

Meshing and Rendering of Patient-Specific Deformation Models with Application to Needle Insertion Simulation

by

Orcun Goksel

B.Sc., Middle East Technical University, 2001

B.Sc., Middle East Technical University, 2002

M.A.Sc., The University of British Columbia, 2004

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

The Faculty of Graduate Studies

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

December, 2009

© Orcun Goksel 2009

Abstract

Tissue deformation is common during many medical interventions. An accurate simulation of these procedures necessitates accounting for tissue displacements by modeling tissue deformation and medical tool interaction. In minimally-invasive procedures, due to lack of visibility, physicians rely on haptic feedback and medical imaging to assess the immediate anatomical configuration and relative medical tool position. These procedures are often difficult to learn and therefore extensive training becomes essential.

Computerized training systems offer an alternative to cadavers and training on patients. To accurately model the tissue deformation, most such systems require a mesh representation of the anatomy. To replicate the medical imaging feedback offered during procedures, a realistic image simulation approach is also needed. In this thesis, a novel energy-based meshing technique taking medical images and producing desirable meshes for the finite element method is introduced. This method employs an image-based discretization energy combined with a geometry-based element quality energy. The former promotes each mesh element to cover similar intensity image regions, while the latter ensures the element suitability for finite element simulation. A method that can mimic realistic B-mode ultrasound images under deformation is also presented in this thesis. This method first maps the pixels of an image from a deformed mesh configuration back to the nominal configuration, and then interpolates them in a B-mode voxel volume reconstructed *a priori*.

Needle insertion is involved in several medical procedures. These percutaneous procedures will benefit significantly from advances in simulating needle-tissue interaction, for which a 3D model is proposed in this thesis. Simulating needle flexibility is achieved fast and accurately using a novel approach employing torsional springs. The needle insertion simulation with haptic feedback is presented for a training scenario for prostate brachytherapy, where simulated ultrasound images coupled with deformation are also displayed. A scheme to generate patient models for this system is also devised using both the conventional meshing techniques in the literature and the proposed variational meshing method.

Table of Contents

Abstract	ii
Table of Contents	iii
List of Tables	vii
List of Figures	viii
List of Abbreviations	x
Acknowledgments	xi
Statement of Co-Authorship	xii
1 Introduction	1
1.1 Motivation	1
1.2 Background	2
1.2.1 Prostate Brachytherapy	2
1.2.2 Tissue Deformation and Needle Interaction Models	4
1.2.3 Model Generation for Medical Simulations	5
1.2.4 Simulating Medical Images	6
1.2.5 Medical Simulators	7
1.3 Thesis Objectives	7
1.4 Chapter Summary	9
References	11
2 Image-Based Variational Meshing	17
2.1 Introduction	17
2.2 Previous Work on Optimal Tessellations	19
2.3 Methods	21
2.3.1 Geometric Energy for Mesh Optimization	21
2.3.2 Element Discretization	23
2.3.3 Objective Function for Image Compliance	23
2.4 Implementation	25
2.4.1 Mesh Initialization	25
2.4.2 Node Updates	26
2.4.3 Connectivity Updates	26
2.4.4 Normalization of Cost Weighting Factor	27

2.4.5	Convergence Measure	27
2.4.6	Element Aspect Ratios	27
2.4.7	Summary of the Algorithm	28
2.5	Results	28
2.6	Discussion	31
2.6.1	Element Thresholding for Segmentation	31
2.6.2	Discretization of Known Elastic Modulus Distribution	32
2.6.3	Connectivity and Node Updates	35
2.6.4	Mesh Sizing	38
2.6.5	Resolution	38
2.6.6	Slivers	39
2.6.7	Mesh Quality Assessment from Aspect Ratios	40
2.6.8	Relationship to deformation and FEM	41
2.7	Conclusions	42
References		43
3	B-Mode Ultrasound Image Simulation in Deformable 3D Medium	46
3.1	Introduction	46
3.2	Previous Work	48
3.3	Methods	49
3.3.1	Inverting Deformation	50
3.3.2	Image Pixels in an FEM Tessellation	51
3.3.3	Fast Equidistant-Point Location on Image Planes in 3D	52
3.3.4	Finding the Pixel Intensity Value	55
3.3.5	Summary of the Proposed Algorithm	56
3.3.6	Computational Analysis	56
3.3.7	Deformation Model	57
3.4	Results	57
3.5	Discussion	62
3.6	Conclusions and Future Work	67
References		68
4	Modeling and Simulation of Flexible Needles	71
4.1	Introduction	71
4.2	Methods	73
4.2.1	Finite Element Method using Tetrahedral Elements	73
4.2.2	Finite Element Method using Nonlinear Beam Elements	74
4.2.3	Angular Springs Model	76
4.2.4	A Common Convergence Criterion for All Models	79
4.3	Experiment	79
4.4	Results	80
4.4.1	Parameter Identification	81
4.4.2	Performance in Modeling Needle Deflection	82
4.5	Discussion	82

4.5.1	Approximating the angular spring constants	87
4.6	Conclusions	88
References		89
5	Haptic Simulator for Prostate Brachytherapy with Simulated Ultrasound	91
5.1	Introduction	91
5.2	Methods	93
5.2.1	Brachytherapy Simulation Components	93
5.2.2	Patient Model Generation	95
5.2.3	Mesh-Ultrasound Registration	96
5.2.4	Pre-computation of Data Structures	96
5.3	Results	97
5.4	Discussion	98
5.4.1	Variational Meshing	98
5.4.2	Brachytherapy Simulation	99
5.4.3	Deformation due to TRUS Probe	100
5.4.4	Parameter Tuning and Validation	102
5.5	Conclusions	102
References		103
6	Conclusions and Future Research	105
6.1	Contributions	105
6.1.1	Modeling Needle Flexibility	105
6.1.2	Rapid Image Slicing Complying Model-Based Deformations	106
6.1.3	Optimal Tessellation of Images	106
6.1.4	Deformation Models Using Elastography	107
6.1.5	Needle Insertion Simulation	107
6.1.6	Patient-Specific Modeling	108
6.2	Future Work	108
References		110
 Appendices		
A	Algebraically Defining the Interior 1-ring Neighbourhood	113
B	Resolving Pixel Discretization Conflicts	114
C	Undeforming Using Barycentric Coordinates	116
D	Euler-Bernoulli Beam Element	117
E	Angle of Deflection for a Bending Moment	119
F	Angle of Deflection for a Torsional (Twisting) Moment	120
G	List of Publications	121

List of Tables

4.1 Tip deflection for various lateral tip forces. 80

List of Figures

1.1	Illustration of the brachytherapy procedure.	3
1.2	The prostate is observed during a needle insertion.	4
1.3	The tissue mesh while the needle is inserted, moved upwards, and retracted. . .	8
2.1	A sliver and the <i>1-ring neighbourhood</i> of a node.	20
2.2	Random 2D/3D mesh initializations and their geometrically optimized configurations.	22
2.3	An initial synthetic phantom with its discretization.	24
2.4	A simple four-element mesh and the combined cost E	24
2.5	The initial phantom mesh in Fig. 2.3 as optimized.	25
2.6	Mesh optimization on a 2D MR image slice of brain ventricles.	29
2.7	Mesh optimization on a 2D CT image slice of the kidney.	30
2.8	A 3D mesh optimization for a synthetic phantom image with a spherical inclusion. .	31
2.9	Mesh optimization for 3D MR image volume of the brain.	32
2.10	Mesh optimization for 3D CT data of the kidney.	33
2.11	Combined cost E during the optimization of some examples.	34
2.12	Initial and optimized meshes of sagittal prostate vibro-elastography images. . .	36
2.13	Demonstration of using a sizing field for variable element sizes throughout the domain.	39
2.14	A sliver and its four face neighbours.	40
2.15	Mesh quality assessment from aspect ratios.	41
3.1	Image slices before and after deformation.	47
3.2	Online and offline steps of the proposed simulation.	49
3.3	Voxel data and image plane in nominal, post-deformation, and undeformed configurations in 2D.	50
3.4	Mesh-based image <i>undeformation</i> illustrated in 2D.	52
3.5	Basic steps of the proposed algorithm.	52
3.6	The intersection of a mesh and the image plane.	54
3.7	Marking mesh elements cross-sections.	55
3.8	Transformations that map a pixel to the reference volume.	55
3.9	The phantom and the image acquisition setup.	58
3.10	Simulated and acquired images with 0, 5, and 10 mm indentations.	59
3.11	Normalized mutual information and sum of squared differences comparisons of simulated and acquired images.	60
3.12	Frame synthesis time with respect to the numbers of pixels and intersected elements.	61

3.13	Real-time ultrasound scanning simulator.	62
3.14	In-vivo data collection.	63
3.15	Simulated ultrasound images of the thigh.	64
4.1	The forces on the needle shaft caused by tissue deformation and needle base manipulation.	72
4.2	Euler-Bernoulli beam element under deformation.	75
4.3	Angles of bending and twisting between two needle segments.	76
4.4	Three joints of our spherical wrist model.	77
4.5	The entire needle in equilibrium state constrained at the base and under nodal forces at its joints.	78
4.6	The deflection of the needle tip and the residual torque at each iteration of a sample bending simulation.	79
4.7	Bent needle shaft.	80
4.8	The tip and the area errors.	81
4.9	Needle discretized by triangular elements.	82
4.10	Distribution of identified Young's moduli and spring constants for each bending experiment using a different load.	83
4.11	The experimental and simulated deformed needles.	84
4.12	The tip and area error comparisons of the three models.	85
4.13	Mean spring constants with respect to needle segment length.	87
5.1	The brachytherapy procedure.	92
5.2	The online simulation components and their data interaction.	94
5.3	Patient-specific model generation procedure.	95
5.4	Surface models from MR images.	96
5.5	The mesh generated from MRI and registered to TRUS.	97
5.6	The graphical interface to the simulation.	98
5.7	Simulation of the seed placement error due to tissue deformation.	99
5.8	Variational meshing result of the prostate labeled volume.	100
5.9	Sample images from a needle insertion simulation.	101
B.1	An illustration of element partial ordering and its corresponding directed graph.	115
C.1	A deformed element and an image pixel.	116
E.1	A short section of a bent cantilever.	119

List of Abbreviations

2D	Two Dimensional
3D	Three Dimensional
CT	Computed Tomography
CVDT	Centroidal Voronoi-Delaunay Tessellation
DVT	Deep-Vein Thrombosis
FEM	Finite Element Method
ICP	Iterative Closest Point
MI	Mutual Information
MR	Magnetic Resonance
MRI	Magnetic Resonance Imaging
NNI	Nearest-Neighbour Interpolation
ODT	Optimal Delaunay Tessellations
SSD	Sum of Squared Differences
TLI	Tri-Linear Interpolation
TRUS	Transrectal Ultrasound

Acknowledgments

I owe particular thanks to my supervisor Prof Tim Salcudean, for his indispensable insight and guidance in research, for his critical feedback that helped hone my work above and beyond, and for his perpetual support in *every* aspect. I would also like to thank the thesis committee members, in particular Prof Robert Rohling, for their ideas and suggestions on improving this thesis.

I offer my enduring gratitude to the faculty, staff, and my fellow students at UBC. I cannot possibly name them all, but here are some of my colleagues that were there with me for a considerable part of my PhD: Julian Guerrero, Reza Zahiri-Azar, Ehsan Dehghan, Hani Eskandari, Xu Wen, Ali Baghani, Diego Prananta, Mengchen Zhu, and Sara Mahdavi. Their discussions with me certainly helped improve this work, and their support and company made my PhD years socially enjoyable as well

I would like to send out very special thanks to my parents for their endless support throughout my *years* of education, although I don't think I can possibly thank them enough for bearing with my infrequent phone calls from several time-zones and many thousands of kilometers away. The last, but definitely not the least, person I would like to thank is Eva, the biggest change in my life during the course of my PhD. I cannot imagine finishing this thesis without her continuing moral support and optimism, and endless understanding and love...

Finally, I would also like to acknowledge the following institutions for their generous financial support through fellowships during the course of my studies: the University of British Columbia, the Canadian IRIS/Precarn Networks of Centres of Excellence, and the National Science and Engineering Council of Canada (NSERC).

Statement of Co-Authorship

This thesis is based on several manuscripts, resulting from collaboration between multiple researchers.

A version of Chapter 2 has been submitted for publication. This paper was co-authored with Prof Tim Salcudean. The author's contribution was developing the idea and numerical implementations, application and evaluation of the methods on image data sets, developing tools for importing and pre-processing such image sets prior to evaluation, and preparation of the manuscript. Prof Salcudean assisted with his insight on relevant research interests, suggestions on fine-tuning numerical approaches, providing and/or contributing to the acquisition of certain image data used as examples of application, and editing the manuscript.

A version of Chapter 3 appeared in the *IEEE Transactions on Medical Imaging* in November 2009. This paper was co-authored with Prof Tim Salcudean. The author's contribution in that paper was developing and implementing the idea; phantom fabrication; designing and conducting phantom and in-vivo experiments; composition of data acquisition instruments and applications; calibration of the experimental setup; meshing the experimental models and simulating their deformation as a tool in validation; development and application of the validation framework and criteria; haptic and graphical interface implementation and optimizing the code for suitable real-time execution; and preparation of the manuscript. Prof Salcudean contributed by identifying this need of real-time image simulation for deformable models in several meetings at earlier research stages as well as providing indispensable feedback and edits at various stages of finalizing the manuscript. Prof Salcudean also provided the experimental tools such as the precision stage and the ultrasound system with an instrumented magnetic position sensor.

A version of Chapter 4 appeared in the journal of *Medical Engineering and Physics* in November 2009. This paper was co-authored with Ehsan Dehghan and Prof Tim Salcudean. The author's contribution was proposing and developing the idea of modeling flexible needles using torsional springs; design and preparation of the experimental setup for collecting bent needle images; acquisition of these images; developing tools for data extraction from such images; formulating the two validation criteria and developing tools for their calculation; design and development of model validation pipeline including the use of optimization methods; implementation of the numerical methods for the angular springs model; and preparation of the manuscript. Mr Dehghan suggested the comparison of the proposed model with the two well-known finite element approaches as a benchmark; developed and implemented these two methods; contributed to the preparation of the two manuscript subsections and the relevant appendix describing these two methods; and helped with the phrasing and editing of certain parts of the manuscript with major contributions being in the Introduction and Discussion sections (4.1 and 4.5). Prof Salcudean assisted with the reformulation of torsional springs using a stable energy-based definition, provided numerous suggestions through the course of concept development, and helped in editing the manuscript.

A version of Chapter 5 has been prepared for submission. This paper is co-authored with Prof Tim Salcudean. The author's contribution in that paper was developing the overall simulation framework, modeling the anatomy for deformation simulation, programming of the haptic and graphical interfaces, integration of various simulation components, developing the tools for offline image/mesh data processing and preparation, and writing the manuscript. Prof Salcudean assisted with ideas of general research directions in application to prostate brachytherapy, provided valuable feedback on the development of underlying needle-tissue interaction model having its foundations from his previous collaborations, provided the haptic interface and the image data acquired using a vibro-elastography setup of his design, and further assisted with editing the manuscript.

Chapter 1

Introduction

1.1 Motivation

Tissue deformation is common during many medical interventions. An accurate simulation of these interventions necessitates accounting for these tissue displacements by modeling tissue deformation and medical tools. Minimally-invasive procedures are appealing due to their reduced complications, however it is often not possible to observe tissue deformation during such procedures due to limited visual feedback. Therefore, in order to assess the immediate anatomical configuration and relative medical tool position, the performing physician has to rely on the haptic¹ feedback via the tool and/or any intra-operative medical imaging, if available. These limitations make such procedures difficult to learn and perform, and training systems that replicate the procedural setup become essential as teaching and rehearsal tools. Such systems are also beneficial in teaching scenarios that are rarely encountered in practice.

The development of the techniques in this thesis was mainly motivated by the *prostate brachytherapy* procedure, which is a treatment method for prostate cancer involving needle insertions. It is a typical minimally-invasive procedure, on which substantial recent research has focused and demonstrated potential for improving the current training and treatment planning methods. The needs and difficulties of this procedure are common to various other minimally-invasive (in particular, percutaneous) procedures, and consequently several techniques presented in this thesis are applicable to different medical procedures as well.

Intra-operative medical imaging modalities are the eyes of a physician during a minimally-invasive procedure such as biopsy. Therefore, a simulation system requires the synthesis of such images as they would appear in an actual procedure. These images are not static, but change during the course of the procedure as the tissue deforms due to tool interactions. A technique to render images matching these deformations is necessary. Furthermore, considering real-time imaging modalities, e.g. ultrasound, such rendering technique also has to perform sufficiently fast to be used in real-time simulation. These aspects of image simulation are addressed in this thesis.

The Finite Element Method (FEM), which involves the discretization of continuum mechanics equations inside *elements* of a *mesh* representation of a given domain, is a common tool for deformation modeling. It is a physically-based model with significant advantages. With the advances in computer technology, fast simulation rates using FEM are readily attainable. Consequently, it is the preferred method in many tissue simulation studies as presented later in Section 1.2.2. A successful simulation relies on a “good” mesh representation of the tissue as well as the equations involved in the FEM solution. A desirable mesh consists of elements that introduce minimal error in the solution and also represent the anatomy and tissue compo-

¹Haptics is the combination of tactile and kinesthetic senses, which refer to the sense of touch and to the sense of movement (from muscles), respectively. In this thesis, *haptics* is used to refer to the kinesthetic sense.

sition sufficiently well in order to successfully approximate its actual behaviour. Conventional generation of such meshes involves two stages: first segmenting the anatomy of interest and next meshing the volume between these segmented surfaces. This often demands the scarcely-available time of the highly-trained clinical personnel. Furthermore, only some parts of the anatomy are segmented using this approach while neglecting the effect of others in deformation. Therefore, a technique that can mesh the visible features in a given imaging modality using elements suitable for further FEM simulation is needed. This need is addressed by a meshing technique introduced in this thesis. For images acquired using elastography², this technique has the essential advantage of readily offering an optimal discretization of the tissue for a good approximation of mechanical behaviour without necessarily identifying/segmenting each individual structure.

Needle insertion in tissue is a special case of minimally-invasive procedures. During percutaneous procedures (e.g., brachytherapy), no direct visual feedback is available, and therefore such procedures are carried out relying on haptic feedback and/or real-time medical imaging. The needle interaction with tissue differs from most other medical tool interactions in several ways: the needle does not manipulate only the organ surface; friction is a significant force during its interaction; and fine needles are flexible unlike many other surgical tools. This requires specialized methods to model and simulate their interaction. Modeling of flexible needles is also studied in this thesis.

A medical training system for simulating the prostate brachytherapy procedure requires several components including: (i) an efficient flexible needle model, (ii) a deformable tissue model, (iii) a needle-tissue interaction model, (iv) a fast image simulation method, (v) haptic feedback, and (vi) generation of deformable patient models. This thesis studies the components (i), (iv), (v), and (vi). Incorporating the other components from our earlier work, the integration and development of a brachytherapy training system is also addressed in this thesis.

1.2 Background

First, prostate brachytherapy is introduced in Section 1.2.1. Previous work on tissue deformation and needle-tissue interaction models is presented in Section 1.2.2. Next, common approaches to mesh construction required by these models are given in Section 1.2.3. Background on simulated medical image generation techniques and in particular simulated ultrasound synthesis is presented in Section 1.2.4. Finally, medical simulators for needle insertion and brachytherapy presented in Section 1.2.5 conclude this background review.

1.2.1 Prostate Brachytherapy

Brachy is a Greek term for “short distance” and brachytherapy is a treatment option involving percutaneous needle insertions to treat tumours and cancer via radiation. High-dose-rate brachytherapy refers to the temporary application of radiation intra-operatively using a powerful source in an inserted catheter, whereas low-dose-rate brachytherapy is the permanent implantation of low energy radioactive pellets (seeds) in and around the tissue region to be

²Elastography is the method of imaging tissue elasticity, such as for the purpose of detecting stiffer cancer tissue, by observing local tissue response (e.g., strain) induced by a mechanical excitation.

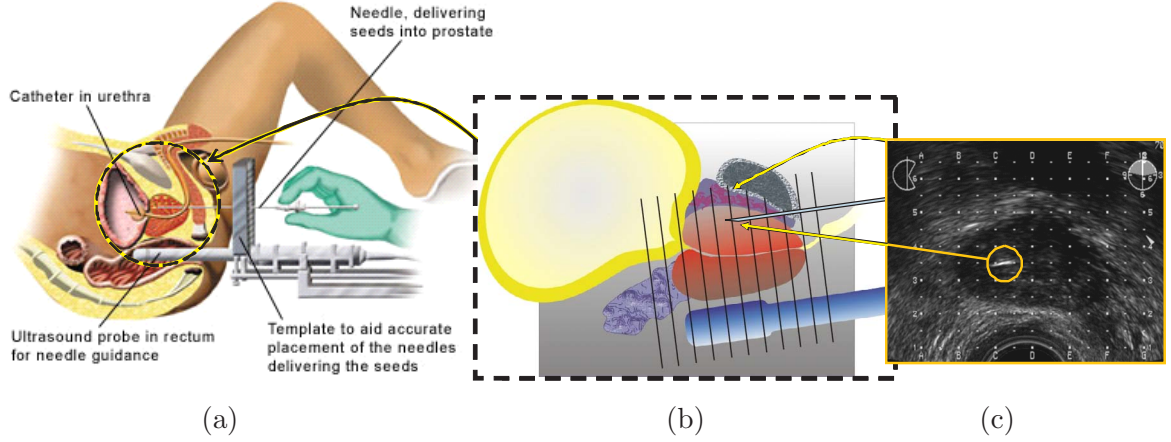


Figure 1.1: Illustration of the brachytherapy procedure: (a) A needle is inserted in the prostate through a template grid. (b) The transrectal ultrasound probe is translated to image at different depths through the pelvic region. (c) The needles are imaged in transverse B-mode slices during seed delivery to ensure accurate placement at the spatial locations as identified pre-operatively at the treatment planning stage.

treated. Some types of cancer to which brachytherapy is applicable include prostate, cervical, breast, and thyroid cancer.

Prostate cancer is the most common cancer among Canadian men with 25,500 new cases and 4,400 deaths estimated in year 2009 alone. One in 6 men will be diagnosed with this disease during their lifetime and its incidence rate is ever increasing due to the aging of the population. Fortunately, over 90% of prostate cancer cases are curable if detected and treated in their early stages.

Low-dose-rate brachytherapy is often the treatment of choice for early-stage locally-confined prostate cancer and, throughout this thesis, *brachytherapy* refers to this particular type of treatment. This treatment option has a high probability of eradicating the cancer while preserving healthy tissue, contingent upon good planning and proper application. During a brachytherapy procedure as illustrated in Fig. 1.1, loaded needles are inserted through the holes of a template grid into the prostate according to a pre-plan. Meanwhile, the physician refers to transrectal ultrasound (TRUS) images to find the needle tip. During a procedure, 80 to 150 seeds are implanted using approximately 25 needles. The brachytherapy needles are 20 cm in length, are quite flexible, and have a beveled tip. See [1] for a detailed description of the prostate brachytherapy procedure.

The prostate is shifted and torqued during needle insertions. In [2], prostate torquing of $\pm 12^\circ$ were recorded on the coronal plane. Figure 1.2 demonstrates its shift on *in-vivo* TRUS data. The given two images were taken at the same transducer depth within one second, before and after a force was applied on the needle which was already in the prostate. Initially, only the bladder is visible (see Fig. 1.2(a)) at the TRUS depth of the prostate *base* (its superior side). When the needle is pushed in by the doctor, the prostate is observed to consequently shift into the plane of view in Fig. 1.2(b).

Despite the low risk of brachytherapy, seed placement errors are still common even by experienced physicians [3]. A sub-optimal application resulting in an undesired dose distribution

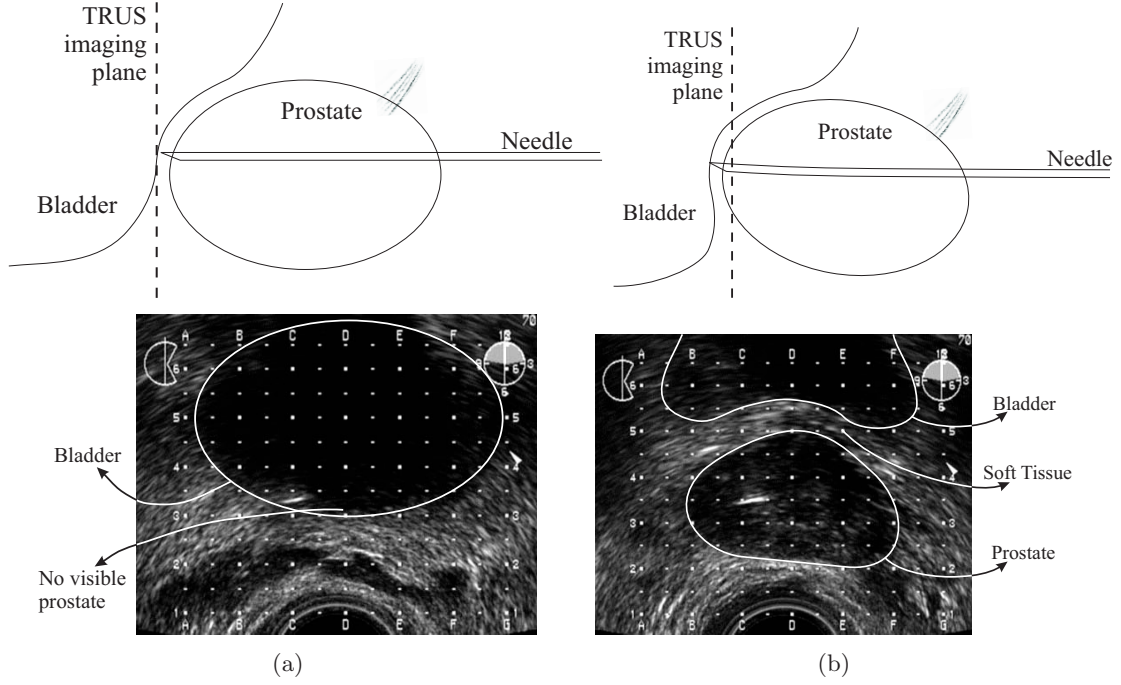


Figure 1.2: The prostate is observed during a needle insertion. Although the TRUS probe is imaging transversally at the same depth, the imaged features are observed to change as the tissue shifts/deforms under interaction forces.

at the target volume not only decreases the effectiveness of the treatment, but it may also lead to subsequent complications such as changes in bowel habits, incontinence, and impotence [4]. Furthermore, errors caused by prostate deformation and motion due to needle insertion may necessitate needle re-insertions, which in turn increase the procedure time and the risk of complications, and cause significant edema. Thus, several studies have aimed at improving the outcome of brachytherapy, either through training/rehearsal systems for medical residents and performing physicians or through planning schemes to increase the targeting accuracy by modeling and correcting for tissue deformation before it occurs. For both, an accurate simulation of the needle-tissue interaction during brachytherapy is required.

1.2.2 Tissue Deformation and Needle Interaction Models

Deformation modeling has been an active topic in computer graphics for the fitting of noisy data and simulation of clothing, facial expressions, and human/animal characters. See [5] for a detailed review. Although mass-spring meshes were successfully implemented in [6, 7], the discretization of continuum mechanics equations using the Finite Element Method (FEM) still has significant advantages in deformation modeling. The *condensation* technique introduced in [8] allows fast surface interactions. Computational advances made real-time rates possible for large meshes using quadratic strain and dynamic FEM [9, 10]. There has also been work on other elasticity formulations, comparison of iterative solution techniques, offline mesh refinement methods, and contact handling algorithms [11, 12]. A review of different models can be found in [13].

The simulation of needle insertion differs from the simulation of other medical tool-tissue interactions in several ways: the needle does not manipulate only the organ surface; friction is a significant force during its interaction; and fine needles are flexible unlike many rigid surgical tools. These three issues were addressed in 2D with a finite-element-based model using the condensation approach, a stick-slip friction model, and local frame rotations with low-rank matrix updates, respectively [14, 15]. This work was extended to 3D [16, 17] and used as part of a needle targeting system [18]. See [13] for a review.

In order to parametrize the interaction of needle and tissue, earlier work focused on the forces on the needle base [19]. Tip cutting and (velocity-dependent) shaft friction force have been also modeled [20, 21]. The tip bevel deflection of *highly-flexible* (also called *steerable*) needles has been studied recently in the context of needle path planning [22, 23].

In this thesis, a linear-elasticity FEM model with 4-node tetrahedral elements is used for tissue deformation modeling. The interaction of the tissue FEM and the needle is achieved using a stick-slip friction model. The cutting and friction parameters required by this model can be identified from controlled insertion experiments, for instance, by using observations from an optical setup [20] or an ultrasound setup [21].

1.2.3 Model Generation for Medical Simulations

Most techniques for deformation simulation, including FEM, require a discretization of the tissue, called the *mesh*. In model generation for medical simulations, conventional approaches treat *segmentation* and *meshing* as two distinct and successive operations. In segmentation, the boundaries of relevant anatomy are identified. Manual segmentation involves contouring these on slices of the volume, which can later be extruded to surface models [24–26]. Alternatively, (semi-)automatic methods find the anatomy requiring certain user input. The domain of interest is next meshed by placing FEM elements inside and around the segmented surfaces. In this approach, the segmented surface becomes a constraint for meshing, preventing any possible of trading off the surface compliance with better mesh quality. Few methods include measures to adjust surface approximation accuracy. However, since the original image is discarded and only the surfaces are retained at this meshing stage, it is not possible to determine at which mesh point and how much of a surface compliance tradeoff is feasible. In addition, the final mesh can only conform to anatomical structures that are initially segmented. This presents the loss of some possibly meaningful data, which existed in the original image, during this process. As a result, a novel meshing technique that takes images as input and generates FEM meshes automatically in a single step is proposed in this thesis.

In mesh generation, various element geometries are possible. Although hexahedral elements are known to produce more accurate results in FEM, they are mostly preferred in structured meshes and tetrahedra are still commonly used due to their flexibility in customizing unstructured meshes to given domain boundaries. Applications of tetrahedral meshes include areas such as solid modeling [27], haptics [28], fracture [29], fluid flow [30], heat transfer [31], surgical simulations [32], and biological tissue modeling such as the brain [33], the knee [11], the liver [34], and the uterus [35].

Most algorithms for tetrahedral mesh generation can be classified in one of the following three groups: (i) the octree technique [36], a recursive division of the model space into smaller elements until a desired resolution is reached for the elements that are cut by the object boundary; (ii) the advancing fronts [37, 38], which progressively builds the tetrahedra

starting from the triangulated surfaces of objects while maintaining an active front that moves inwards/outwards by the formation of new elements; and (iii) methods using the Delaunay criterion [39], which form the largest group of tetrahedral meshing techniques. Since this criterion is just a connectivity definition, not a meshing algorithm itself, additional tetrahedralization methods are required for mesh generation. Since its first use in meshing in [40], numerous Delaunay-based methods have been developed which differ primarily by their decision mechanisms on introducing new points, modifying the current ones, and heuristically fixing quality issues that arise.

Following the meshing approaches above, additional mesh post-processing stages such as smoothing, cleanup, and refinement are often required to improve element quality. Smoothing refers to the slight relocation of the mesh nodes using techniques based on averaging (e.g. Laplacian smoothing), physically-based models (e.g. spring interaction between neighbours) or optimization [41]. Cleanup refers to the step in which element connectivity is optimized using edge/element swaps (*flips*). Refinement is the reduction of the local element sizes by node insertion for the purpose of capturing a local physical phenomenon or for increasing the mesh quality.

Some of the smoothing techniques above were also extended to stand-alone meshing applications [42, 43]. Among these, energy-based methods are of special interest due to their superior results and simpler implementation. They are also easier to analyze mathematically and therefore theoretical guarantees on results can be derived. In this thesis, the state-of-the-art variational meshing technique of Alliez *et al.* [43] is adopted and integrated with an additional image-based energy definition. The optimization of the resulting energy measure aims at finding a mesh with high-quality tetrahedra that also optimally partitions an underlying image.

1.2.4 Simulating Medical Images

Ultrasound is an inexpensive, non-ionizing, real-time medical imaging modality and hence one of the most commonly used examination tools. However, image anisotropy and the existence of various artifacts cause the need for extensive echographer training, which has motivated several computer-based simulation environments [44].

There exist two major approaches for simulating B-mode ultrasound images, *the generative approach* and *the interpolative approach*. The former simulates the ultrasonic wave propagation by using accurate models of the probe, the tissue scatterers, and the wave interaction [45, 46], but involves a very long generation time. The latter approach generates images by interpolating in 3D regular-grid reference volumes *reconstructed* [47, 48] from pre-acquired images. Several studies follow this latter approach [49–55] since it enables data processing with off-the-shelf algorithms. Refer to [44] for a review of ultrasound training simulators.

Since a generative simulation approach with a full-blown wave interaction model is not feasible for real-time applications, some recent work has focused on developing heuristic models that can be computed in real-time [56–58]. Deriving the parameters of such models using CT data was also proposed [59, 60]. Although such pseudo-generative methods for echography simulation are appealing, due to the complex nature of actual wave interactions, it is extremely difficult to generate even common ultrasound phenomena, such as speckle formation, using these methods, let alone realistic images.

As mentioned earlier, tissue deformation is common during medical procedures due to

medical tool interactions. Additionally, an ultrasound examination often involves further deformation caused by the ultrasound probe. Therefore, for a comprehensive realization of a training system involving ultrasound imaging, the deformation must be taken into account in the simulated images. Furthermore, deformation observed in ultrasound images contains essential information in certain applications, such as in the diagnosis of deep-vein thrombosis (DVT) [55]. In addition, the ability to mentally register two-dimensional (2D) image slices within the three-dimensional (3D) anatomy is a non-trivial skill required by any sonographer. Real-time ultrasound simulators also have the potential to accelerate and improve such training. This identified need is addressed in this thesis. A method that generates simulated B-mode images by taking into account tissue deformation is proposed in Chapter 3.

1.2.5 Medical Simulators

Just as a flight simulator is used for training pilots and helps in identifying the complications of current and proposed procedures, a medical simulator is built to recreate the realm of a surgery or examination to be used in training and testing. In this area there exists significant previous work including epidural lumbar puncture [61], breast biopsy [62], interstitial brachytherapy [63], prostate cryotherapy [64], prostate brachytherapy [17, 65], hepatic surgery [66]. There are also current commercialized products for laparoscopy, endoscopy, and intra-vascular access [67–69].

The brachytherapy simulation methods in the literature have addressed the tissue deformation in various ways, such as omitting deformation [70], employing heuristic models for organ surface deformation [71], 2D FEM on a sagittal plane [65], and 3D FEM [17, 18, 72]. See [73] for a detailed review of medical simulations, in particular for urological applications.

The 3D FEM-based simulation of [72] demonstrates insertion of needles into the prostate region for training and planning applications. It uses Lagrange multipliers to couple the deformable models of tissue and the needle. In order to determine the stick-slip states of the needle-tissue interaction to be used in this coupled solution, a *linear complementarity problem* for all possible equation/friction state combinations was solved requiring a significant computational effort in spite of the computation acceleration efforts. Beside the planning use, this simulation was proposed to run at visual refresh rates for the display of anatomical surfaces in training. Unfortunately, it does not offer haptic or ultrasound feedback for training, in spite of these being the only two available real-time intra-operative modality to a physician performing brachytherapy.

The 2D needle interaction simulation methods of [74] were adopted earlier in a 3D framework for prostate brachytherapy [16]. Figure 1.3 illustrates a sample insertion/retraction using this simulation. These techniques are adopted in this thesis in Chapter 5 to extend into a haptic training simulator also having simulated B-mode imaging.

1.3 Thesis Objectives

In this thesis, novel methods addressing various aspects of a medical simulation system are developed. To achieve accurate real-time medical simulations with deformation and to develop a needle insertion training setup, the specific objectives of this thesis are as follows:

1. Developing a meshing technique to automatically generate FEM models from elastography images.

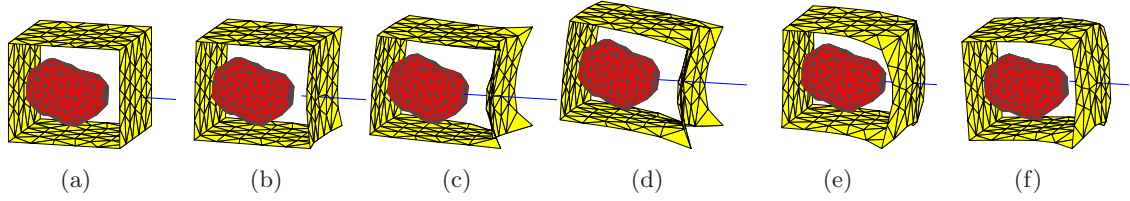


Figure 1.3: The tissue mesh while the needle is inserted (a)-(c), moved upwards (d), and retracted (e)-(f).

2. Providing realistic simulated ultrasound for medical training systems.
3. Simulating the bending of medical needles in a fast and accurate fashion.
4. Development of medical simulation components, such as tissue model, visual simulation, haptic feedback, and user interface, and their integration in a rapid framework that allows for real-time simulation.
5. Designing a patient-specific model generation scheme to acquire and prepare imaging and deformation model data for the use of the targeted simulation framework.
6. By achieving the objectives above, developing methods and designing systems that will enhance medical resident training and treatment planning by providing better computerized alternatives for the current prostate brachytherapy procedure.

In the course of achieving these primary objectives, the following contributions were made:

- A novel use of angular springs in modeling needle bending was introduced and shown to perform fast and accurately. Indeed, this model was shown to provide better performance than common FEM methods in simulating the bending of a brachytherapy needle.
- A fast method of slicing images of volumetric data that is deformed according to a mesh definition. For a given regular-grid volumetric data, realistic-looking medical images can be synthesized in real-time for a tissue deformation simulation that runs in parallel.
- A novel energy-based meshing technique for medical simulations. While it can operate with any imaging modality, its use with elastography imaging was shown to result in tissue deformation models minimizing (elasticity) parameter discretization errors.
- A simulation integration scheme was developed where ultrasound simulation, needle-tissue model, and haptic processes communicate and share data to simulate needle insertion into the prostate.
- A patient model generated from MR images and registered to ultrasound volume for simulation with B-mode feedback was proposed.

1.4 Chapter Summary

The overall goal of the research undertaken in this thesis was to develop algorithms for generating patient-specific tissue deformation models and simulating needle insertion in these models while providing realistic simulated medical imaging feedback. The particular application of prostate brachytherapy is the motivation for this research. The thesis presented here is written in the manuscript-based style, as permitted by the Faculty of Graduate Studies at the University of British Columbia. In the manuscript-based thesis, each chapter represents an individual work that has been published, submitted or prepared for submission to a peer reviewed publication. Each chapter is self-sustained in the sense that it includes an introduction to the work presented in that chapter, the methodology, results and discussion. The references are summarized in the bibliography found at the end of each chapter. The appendices pertaining to each chapter are presented at the end of the thesis.

The components required for a simulation of brachytherapy needle insertion are addressed in different chapters. The simulation of the prostate necessitates a deformable model, the generation of which can be addressed using the variational meshing techniques presented in Chapter 2. The interaction of the needle in such deformable 3D FEM models were already investigated in our earlier work [16, 17]. Chapter 4 presents a novel simulation approach for the bending of brachytherapy needles, caused by forces acting on it during tissue interaction. For a training use of such brachytherapy simulation, the generation of real-time TRUS-mimicking images is also a requirement, which is addressed in Chapter 3. Finally, these different components are integrated into a simulation framework and interface in Chapter 5. Although the development of these techniques target brachytherapy, the individual chapters are presented in a general setting and their integration into a simulation is postponed until Chapter 5. This in accordance with the manuscript-based style and demonstrates the applicability of each chapter's findings in other potential applications. The focus and summary of each chapter is presented below.

In Chapter 2, a novel image-based meshing technique is presented. In medical simulations involving tissue deformation, the FEM is a widely used technique, where the size, shape, and placement of the elements in a model are important factors that affect the interpolation and numerical errors of a solution. Conventional model generation schemes for FEM consist of a segmentation step delineating the anatomy followed by a meshing step generating elements conforming to this segmentation. In this chapter, a single-step model generation technique is proposed based on optimization. Starting from an initial mesh covering the domain of interest, the mesh nodes are adjusted to minimize an objective function which penalizes intra-element intensity variations and poor element geometry for the entire mesh. Trade-offs between mesh geometry quality and intra-element variance are achieved by adjusting the relative weights of the geometric and intensity variation components of the cost function. This meshing approach enables a more accurate rendering of shapes with fewer elements and provides more accurate models for deformation simulation, especially when the image intensities represent a mechanical feature of the tissue such as the elastic modulus. The use of the proposed mesh optimization is demonstrated both in 2D and 3D on synthetic phantoms, MR images of the brain, CT images of the kidney, and vibro-elastography images of the prostate. This automatic mesh generation approach can be used for generating deformable prostate-region models for a brachytherapy training/planning simulation as presented later in Chapter 5.

Chapter 3 presents an algorithm for fast image synthesis inside deformed volumes. Given the node displacements of a mesh and a reference 3D image dataset of a predeformed volume, the method first maps the image pixels that need to be synthesized from the deformed configuration to the nominal predeformed configuration, where the pixel intensities are obtained easily through interpolation in the regular-grid structure of the reference voxel volume. This mapping requires the identification of the mesh element enclosing each pixel for every image frame. To accelerate this point location operation, a fast method of projecting the deformed mesh on image pixels is introduced in this chapter. The method presented was implemented for ultrasound B-mode image simulation of a synthetic tissue phantom. The phantom deformation as a result of ultrasound probe motion was modeled using the finite element method. Experimental images of the phantom under deformation were then compared with the corresponding synthesized images using sum of squared differences and mutual information metrics. Both this quantitative comparison and a qualitative assessment show that realistic images can be synthesized using the proposed technique. An ultrasound examination system was also implemented to demonstrate that real-time image synthesis with the proposed technique can be successfully integrated into a haptic simulation. The methods developed in this chapter are implemented in a haptic brachytherapy training system as presented in Chapter 5.

In Chapter 4, modeling of medical needle flexibility is studied. Needle insertion is performed in many clinical and therapeutic procedures. Tissue displacement and needle bending which result from needle-tissue interaction make accurate targeting difficult. For performing physicians to gain essential needle targeting skills, needle insertion simulators can be used for training. An accurate needle bending model is essential for such simulators. These bending models are also needed for needle path planning. In this chapter, three different models are presented to simulate the deformations of a needle. The first two models use the finite element method and take the geometric non-linearity into account. The third model is a series of rigid bars connected by angular springs. The models were compared to recorded deformations during experiments of applying lateral tip forces on a brachytherapy needle. The model parameters were identified and the simulation results were compared to the experimental data. The results show that the angular spring model, which is computationally the most efficient model, is also the most accurate in modeling the bending of the brachytherapy needle.

Chapter 5 presents a medical simulator for the prostate brachytherapy procedure. Needles are inserted in deformable tissue models using a haptic device while force feedback computed using a needle-tissue interaction model is rendered on a users hand. Transrectal ultrasound images of the region of interest are also displayed in real-time using an interpolation scheme accounting for the mesh-based tissue deformation. Employing a 3D ultrasound volume data reconstructed *a priori*, this simulation method achieves realistic ultrasound feedback coupled with immediate tissue deformation. Models for simulating tissue deformation using the finite element method were obtained by meshing contours on MR images, which are also rigidly registered to ultrasound voxel volumes using the prostate surface. The presented simulation system is both suitable for brachytherapy training using haptic control/feedback and for planning scenarios using the desired base trajectory input methods provided in the interface.

In Chapter 6, the results of the collected works are related to one another and a unified goal of the thesis is discussed. The strengths and weaknesses of the research are then presented, along with future directions for research.

References

- [1] W. M. Butler, “Review of modern prostate brachytherapy,” in *IEEE Engineering in Medicine and Biology Conference (EMBC)*, Chicago, IL, 2000, pp. 748–752.
- [2] V. Lagerburg, M. A. Moerland, J. J. W. Lagendijk, and J. J. Battermann, “Measurement of prostate rotation during insertion of needles for brachytherapy,” *Radiotherapy and Oncology*, vol. 77, pp. 318–323, 2005.
- [3] R. Taschereau, J. Pouliot, J. Roy, and D. Tremblay, “Seed misplacement and stabilizing needles in transperineal permanent prostate implants,” *Radiotherapy and Oncology*, vol. 55, no. 1, pp. 59–63, 2000.
- [4] G. S. Merrick, W. M. Butler, K. E. Wallner, R. W. Galbreath, R. L. Anderson, B. S. Kurko, J. H. Lief, and Z. A. Allen, “Erectile function after prostate brachytherapy,” *Int J of Radiation Oncology Biology Physics*, vol. 62, no. 2, pp. 437–447, 2005.
- [5] S. F. F. Gibson and B. Mirtich, “A survey of deformable modeling in computer graphics,” MERL – Mitsubishi Electric Research Laboratory, Tech. Rep. TR-97-19, 1997.
- [6] M. Downes, M. C. Cavusoglu, W. Gantert, L. W. Way, and F. Tendick, “Virtual environments for training critical skills in laparoscopic surgery,” in *Medicine Meets Virtual Reality (MMVR)*, San Diego, CA, 1998, pp. 316–322.
- [7] D. Terzopoulos and K. Waters, “Physically-based facial modeling, analysis and animation,” *Visualization and Computer Animation*, vol. 1, no. 2, pp. 73–80, 1990.
- [8] M. Bro-Nielsen and S. Cotin, “Real-time volumetric deformable models for surgery simulation using finite elements and condensation,” *Computer Graphics Forum (Proc of Eurographics’96)*, vol. 15, no. 3, pp. 57–66, 1996.
- [9] Y. Zhuang and J. Canny, “Real-time simulation of physically realistic global deformation,” in *SIGGRAPH: Sketches and Applications*, Los Angeles, CA, 1999.
- [10] S. Cotin, H. Delingette, and N. Ayache, “Real-time elastic deformations of soft tissues for surgery simulation,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 1, pp. 62–73, 1999.
- [11] G. Hirota, S. Fisher, and A. State, “An improved finite element contact model for anatomical simulations,” *The Visual Computer*, vol. 19, no. 5, pp. 291–309, 2003.
- [12] H.-W. Nienhuys, “Cutting in deformable objects,” Ph.D. dissertation, Utrecht University, 2003.

- [13] S. Misra, K. T. Ramesh, and A. M. Okamura, “Modeling of tool-tissue interactions for computer-based surgical simulation: A literature review,” *Presence: Teleoperators & Virtual Environments*, vol. 17, no. 5, pp. 463–491, 2008.
- [14] S. P. DiMaio and S. E. Salcudean, “Needle steering and motion planning in soft tissues,” *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 6, pp. 965–974, June 2005.
- [15] —, “Interactive simulation of needle insertion models,” *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 7, pp. 1167–1179, 2005.
- [16] O. Goksel, “Ultrasound image and 3D finite element based tissue deformation simulator for prostate brachytherapy,” Master’s thesis, University of British Columbia, 2004.
- [17] O. Goksel, S. E. Salcudean, and S. P. DiMaio, “3D simulation of needle-tissue interaction with application to prostate brachytherapy,” *Computer Aided Surgery*, vol. 11, no. 2006, pp. 279–288, 2006.
- [18] E. Dehghan and S. E. Salcudean, “Needle insertion parameter optimization for brachytherapy,” *IEEE Transactions on Robotics*, vol. 25, no. 2, pp. 303–315, 2009.
- [19] A. Okamura, C. Simone, and M. O’Leary, “Force modeling for needle insertion into soft tissue,” *IEEE Transactions on Biomedical Engineering*, vol. 51, pp. 1707–1716, 2004.
- [20] S. P. DiMaio and S. E. Salcudean, “Needle insertion modeling and simulation,” *IEEE Transactions on Robotics and Automation*, vol. 19, no. 5, pp. 864–875, 2003.
- [21] E. Dehghan, X. Wen, R. Zahir-Azar, M. Marchal, and S. E. Salcudean, “Needle-tissue interaction modeling using ultrasound-based motion estimation: Phantom study,” *Computer Aided Surgery*, vol. 13, no. 5, pp. 265–280, 2008.
- [22] R. J. I. Webster, J. S. Kim, N. J. Cowan, G. S. Chirikjian, and A. M. Okamura, “Non-holonomic modeling of needle steering,” *Int J of Robotics Research*, vol. 25, pp. 509–525, 2006.
- [23] R. Alterovitz, M. Branicky, and K. Goldberg, “Motion planning under uncertainty for image-guided medical needle steering,” *Int J of Robotics Research*, vol. 27, no. 11-12, pp. 1361–1374, 2008.
- [24] B. Geiger, “Three-dimensional modeling of human organs and its application to diagnosis and surgical planning,” Ph.D. dissertation, École Nationale Supérieure des Mines de Paris, France, 1993.
- [25] G. Treece, R. Prager, A. Gee, and L. Berman, “Surface interpolation from sparse cross sections using region correspondence,” *IEEE Transactions on Medical Imaging*, vol. 19, no. 11, pp. 1106–1114, 2000.
- [26] N. Archip, R. Rohling, V. Dessenne, P. Erard, and L. P. Nolte, “Anatomical structure modeling from medical images,” *Computer Methods and Programs in Biomedicine*, vol. 82, no. 3, pp. 203–215, 2006.

- [27] B. Cutler, J. Dorsey, L. McMillan, M. Müller, and R. Jagnow, “A procedural approach to authoring solid models,” *ACM Transactions on Graphics (SIGGRAPH Proceedings)*, vol. 21, no. 3, pp. 302–311, 2002.
- [28] G. Debunne, M. Desbrun, M.-P. Cani, and A. H. Barr, “Dynamical real-time deformations using space and time adaptive sampling,” *ACM Transactions on Graphics (SIGGRAPH Proceedings)*, vol. 20, pp. 31–36, 2001.
- [29] J. F. O’Brien, A. W. Bargteil, and J. K. Hodgins, “Graphical modeling and animation of ductile fracture,” *ACM Transactions on Graphics (SIGGRAPH Proceedings)*, vol. 21, no. 3, pp. 291–294, 2002.
- [30] K. Dhinsa, C. Bailey, and K. Pericleous, “Investigation into the performance of turbulence models for fluid flow and heat transfer phenomena in electronic applications,” in *Annual IEEE Semiconductor Thermal Measurement and Management (SEMI-THERM)*, 2004, pp. 278–285.
- [31] C. Singh Verma and V. C. V. Rao, “Parallelization of finite volume computations for heat transfer application using unstructured mesh partitioning algorithms,” in *International Conference on High Performance Computing*, 1997.
- [32] F. Ganovelli, P. Cignoni, C. Montani, and R. Scopigno, “Enabling cuts on multiresolution representation,” *The Visual Computer*, vol. 17, no. 5, pp. 274–286, 2001.
- [33] E. Grinspun, P. Krysl, and P. Schröder, “Charms: a simple framework for adaptive simulation,” *ACM Transactions on Graphics (SIGGRAPH Proceedings)*, vol. 21, no. 3, pp. 281–290, 2002.
- [34] X. Wu, M. S. Downes, T. Goktekin, and F. Tendick, “Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes,” in *Computer Graphics Forum*, 2001, pp. 349–358.
- [35] M. Müller and M. Teschner, “Volumetric meshes for real-time medical simulations,” in *Workshop on Bildverarbeitung für die Medizin*, Erlangen, Germany, 2003, pp. 279–283.
- [36] M. A. Yerry and M. S. Shephard, “Automatic three-dimensional mesh generation by the modified octree technique,” *Int J for Numerical Methods in Engineering*, vol. 20, no. 11, pp. 1965–1990, 1984.
- [37] R. Löhner and P. Parikh, “Three-dimensional grid generation by the advancing front method,” *Int J for Numerical Methods in Fluids*, vol. 8, pp. 1135–1149, 1988.
- [38] S. H. Lo, “Volume discretization into tetrahedra - II: 3D triangulation by advancing front approach,” *Computers and Structures*, vol. 39, no. 5, pp. 501–511, 1991.
- [39] B. N. Delaunay, “Sur la Sphere – vide. izvestia akademii nauk sssr,” *VII Seria, Otdelenie Matematicheskii i Estestvennykh Nauk*, vol. 7, pp. 793–800, 1934.
- [40] D. F. Watson, “Computing the delaunay tessellation with application to voronoi polytopes,” *The Computer Journal*, vol. 24, no. 2, pp. 167–172, 1981.

- [41] L. Chen, “Mesh smoothing schemes based on optimal Delaunay triangulations,” in *International Meshing Roundtable*, 2004, pp. 109–120.
- [42] P. O. Persson and G. Strang, “A simple mesh generator in MATLAB,” *SIAM Review*, vol. 46, no. 2, pp. 329–346, 2004.
- [43] P. Alliez, D. Cohen-Steiner, M. Yvinec, and M. Desbrun, “Variational tetrahedral meshing,” *ACM Transactions on Graphics (SIGGRAPH Proceedings)*, vol. 24, no. 3, pp. 617–625, 2005.
- [44] H. Maul, A. Scharf, P. Baier, M. Wüstemann, H. H. Günter, G. Gebauer, and C. Sohn, “Ultrasound simulators: Experience with sonotrainer and comparative review of other training systems,” *Ultrasound Obstet Gynecol*, vol. 24, pp. 581–585, 2004.
- [45] J. C. Bamber and R. J. Dickinson, “Ultrasonic B-scanning: a computer simulation,” *Physics in Medicine and Biology*, vol. 25, no. 3, pp. 463–479, 1980.
- [46] J. A. Jensen, “A model for the propagation and scattering of ultrasound in tissue,” *J Acoust Soc Am*, vol. 89, pp. 182–191, 1991.
- [47] R. N. Rohling, A. H. Gee, and L. Berman, “A comparison of freehand three-dimensional ultrasound reconstruction techniques,” *Medical Image Analysis*, vol. 3, no. 4, pp. 339–359, 1999.
- [48] R. S. José-Estépar, M. Martín-Fernández, P. P. Caballero-Martínez, C. Alberola-López, and J. Ruiz-Alzola, “A theoretical framework to three-dimensional ultrasound reconstruction from irregularly sampled data,” *Ultrasound in Medicine and Biology*, vol. 29, no. 2, pp. 255–269, 2003.
- [49] D. Aiger and D. Cohen-Or, “Real-time ultrasound imaging simulation,” *Real-Time Imaging*, vol. 4, no. 4, pp. 263–274, 1998.
- [50] H.-H. Ehrlicke, “SONOSim3D: a multimedia system for sonography simulation and education with an extensible case database,” *European Journal of Ultrasound*, vol. 7, pp. 225–230, 1998.
- [51] P. Abolmaesumi, K. Hashtrudi-Zaad, D. Thompson, and A. Tahmasebi, “A haptic-based system for medical image examination,” in *IEEE Engineering in Medicine and Biology Conference (EMBC)*, San Francisco, CA, USA, Sep 2004, pp. 1853–1856.
- [52] I. M. Heer, K. Middendorf, S. Müller-Egloff, M. Dugas, and A. Strauss, “Ultrasound training: the virtual patient,” *Ultrasound in Obstetrics and Gynecology*, vol. 24, no. 4, pp. 440–444, 2004.
- [53] M. Weidenbach, F. Wild, K. Scheer, G. Muth, S. Kreutter, G. Grunst, T. Berlage, and P. Schneider, “Computer-based training in two-dimensional echocardiography using an echocardiography simulator,” *Journal of the American Society of Echocardiography*, vol. 18, no. 4, pp. 362–366, 2005.
- [54] Sonofit GmbH. Sonofit ultrasound training system. Darmstadt, Germany. [Online]. Available: <http://www.sonofit.com/>

- [55] D. d'Aulignac, C. Laugier, J. Troccaz, and S. Vieira, "Towards a realistic echographic simulator," *Medical Image Analysis*, vol. 10, pp. 71–81, 2006.
- [56] Y. Zhu, D. Magee, R. Ratnalingam, and D. Kessel, "A virtual ultrasound imaging system for the simulation of ultrasound-guided needle insertion procedures," in *Medical Image Understanding and Analysis*, 2006, pp. 61–65.
- [57] D. Magee, Y. Zhu, R. Ratnalingam, P. Gardner, and D. Kessel, "An augmented reality simulator for ultrasound guided needle placement training," *Med Bio Eng Comput*, vol. 45, pp. 957–967, 2007.
- [58] G. Reis, B. Lappé, S. Köhn, C. Weber, M. Bertram, and H. Hagen, *Visualization in Medicine and Life Sciences*. Springer Berlin Heidelberg, 2008, ch. Towards a Virtual Echocardiographic Tutoring System, pp. 99–119.
- [59] A. Hostettler, C. Forest, A. Forgione, L. Soler, and J. Marescaux, "Real-time ultrasonography simulator based on 3D CT-scan images," in *Medicine Meets Virtual Reality (MMVR)*, 2005, pp. 191–193.
- [60] F. P. Vidal, N. W. John, A. E. Healey, and D. A. Gould, "Simulation of ultrasound guided needle puncture using patient specific data with 3D textures and volume haptics," *Computer Animation and Virtual Worlds*, vol. 19, pp. 111–127, 2008.
- [61] P. J. Gorman, T. M. Krummel, R. Webster, M. Smith, and D. Hutchens, "A prototype haptic lumbar puncture simulator," in *Medicine Meets Virtual Reality (MMVR)*, 2000, pp. 263–274.
- [62] F. S. Azar, D. N. Metaxas, and M. D. Schnall, "Methods for modeling and predicting mechanical deformations of the breast under external perturbations," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2001, pp. 1267–1270.
- [63] S. Miller, C. Jeffrey, J. Bews, and W. Kinsner, "Advances in the virtual reality interstitial brachytherapy system," in *IEEE Canadian Conference on Electrical and Computer Engineering*, Edmonton, Alberta, 1999.
- [64] J. K. Hahn, M. J. Manyak, G. Jin, D. Kim, J. Rewcastle, S. Kim, and R. J. Walsh, "Cryotherapy simulator for localized prostate cancer," in *Medicine Meets Virtual Reality (MMVR)*, 2002.
- [65] R. Alterovitz, K. Goldberg, J. Pouliot, R. Taschereau, and I.-C. Hsu, "Sensorless planning for medical needle insertion procedures," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, NV, 2003.
- [66] G. Picinbono, H. Delingette, and N. Ayache, "Non-linear anisotropic elasticity for real-time surgery simulation," *Graphical Models*, vol. 65, pp. 305–321, 2003.
- [67] Immersion Corp. Medical and surgical simulators. San Jose, CA, USA. [Online]. Available: <http://www.immersion.com>
- [68] Simbionix USA Corp. Procedure rehearsal and training simulation systems. Cleveland, OH, USA. [Online]. Available: <http://www.simbionix.com>

- [69] Mentice AB. Medical simulators for training. Gothenburg, Sweden. [Online]. Available: <http://www.mentice.com>
- [70] A. N. Kimura, J. Camp, R. Robb, and B. Davis, “A prostate brachytherapy training rehearsal system – simulation of deformable needle insertion,” in *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2002, pp. 264–271.
- [71] X. Wang and A. Fenster, “A haptic-enhanced 3D real-time interactive needle insertion simulation for prostate brachytherapy,” in *Proc. SPIE Medical Imaging: Vis., Image-Guided Procedures, and Display*, 2004, pp. 781–789.
- [72] N. Chentanez, R. Alterovitz, D. Ritchie, L. Cho, K. K. Hauser, K. Goldberg, J. R. Shewchuk, and J. F. O’Brien, “Interactive simulation of surgical needle insertion and steering,” *ACM Transactions on Graphics (SIGGRAPH Proceedings)*, vol. 28, no. 3, pp. 1–10, 2009.
- [73] J. Troccaz and *et al.*, “Medical image computing and computer-aided medical interventions applied to soft tissues: work in progress in urology,” *Proceedings of the IEEE*, vol. 94, no. 4, pp. 1665–1677, 2006.
- [74] S. P. DiMaio, “Modelling, simulation and planning of needle motion in soft tissues,” Ph.D. dissertation, Univ. of British Columbia, 2003.

Chapter 2

Image-Based Variational Meshing ³

In order to realize a prostate brachytherapy simulator, a deformable mesh model of the prostate is required. Since this simulation is also intended for treatment planning, or possibly other intra-operative uses, a model generation technique with minimal user input is desired. This further enables the development of the method into a fully-automated technique.

To simulate an accurate deformation using FEM, the mechanical elasticity parameters of a mesh have to be adjusted accordingly. The derivation of such mechanical parameters from in-vivo tissue is now possible with recent advances in the elastography field. Prostate vibro-elastography enables us to acquire the data for the generation of volumetric images of these mechanical parameters. This chapter describes a method for generating tetrahedral prostate meshes from such images. These generated tetrahedra have the advantage of minimally partitioning the input mechanical feature images, so that the error due to this discretization is minimized. This technique is indeed applicable to the meshing of images from different imaging modalities. This chapter describes the general underlying concepts of this variational meshing approach, without particular focus on the prostate. Meshing of a labeled prostate voxel volume using these methods is later demonstrated in Chapter 5.

2.1 Introduction

The finite element method (FEM) is a common technique in medical simulations. Its speed and accuracy depend on the number of nodes/elements used and their shape and placement in the domain. In this chapter, a variational modelling approach is presented to produce high-quality FEM meshes automatically for given tissue domains in both 2D and 3D. The method aligns FEM elements to group similar image intensities, as a way of clustering (segmenting) the domain, while maintaining *good* element aspect ratios for FEM. The use of such a method becomes particularly important when the input image represents a mechanical feature distribution of the tissue such as elastic modulus, because the elasticity parameters of each element is represented with a single value in FEM and the proposed method minimizes an objective function defined by the error between this single-value discretization and the measured modulus distribution within each element. Nevertheless, as presented in this chapter, the method is applicable to the meshing of most medical imaging modalities without the need for an intermediate segmentation.

In the conventional modelling methods for the FEM simulation of tissue deformation, a discrete representation of the anatomy of interest is obtained from an intensity image/volume by

³A version of this chapter has been peer-reviewed and published in the proceedings of the 2009 Medical Image Computing and Computer Assisted Intervention (MICCAI) conference, London, UK. O.Goksel and S.E.Salcudean: “High-Quality Model Generation for Finite Element Simulation of Tissue Deformation,” *Lecture Notes in Computer Science*, 5762:248-256, 2009. A version of this chapter has also been submitted for publication. O.Goksel and S.E.Salcudean, “Image-Based Variational Meshing”.

employing two steps, *segmentation* and *meshing*. Segmentation, which consists of *recognition* and *delineation* of anatomy, has been studied in several medical contexts using numerous different approaches [1]. Although automatic segmentation techniques do exist, recognition is not a computationally well-defined problem and thus is usually achieved with manual intervention leading to semi-automatic implementations. In contrast, delineation, which in many cases can be stated roughly as *grouping similar pixels*, allows for algorithmic approaches. Segmentation overall often requires *a priori* information about both the anatomy and the medical imaging modality, therefore does not have a one-size-fits-all solution.

The result of segmentation is a representation of the organ boundary, which is often in an explicit form such as a surface mesh, although implicit representations are also viable. This anatomical boundary is then supplied to a meshing scheme to tile the space with elements. The final mesh is then used for simulating tissue deformation for procedures such as laparoscopic surgery [2], brain surgery [3], breast biopsy [4], and brachytherapy [5].

It is possible to generate anatomical models using structured meshes such as converting image voxels into hexahedral elements [4]. However, unstructured meshing is a common choice since the same or better surface/volume representations can be achieved using fewer elements [6, 7]. Popular unstructured meshing techniques include octree-based methods, Delaunay approaches, and advancing fronts, most of which were originally developed and tuned for the modelling of mechanical structures. The meshing efforts for medical applications have often focused on generating meshes inside given segmentation contours [8] or 3D surface triangulations [3, 9–11]. Common approaches first tile the bounding box of the anatomy with elements, and then cull the elements outside the anatomy, followed by a final stage of clipping [3, 8, 9] or compressing [10, 11] the elements cutting the anatomical boundaries. Graded meshes with smaller elements closer to the surfaces were employed in applications where the surface mesh resolution is the primary concern [9, 11–13]. Typical implementations of meshing techniques from different conceptual categories were compared by Fedorov *et al.* [14] in the context of deformable registration of brain MR images.

Image segmentation is a special case of the general data *clustering* techniques [18], where not only similarity of intensities but also spatial distribution such as connectivity of voxels (so that they form a single object) is often desired. Such additional constraints are enforced by the inherent functioning of the particular segmentation method such as watershed and active contours [19] and define the overall performance of that method. Indeed, segmentation as part of a meshing process can be seen as an even more-specific clustering problem, where the clustered regions must have concrete geometrical shapes such as tetrahedra. Gevers [20] introduces a 2D method for segmenting photographic images using triangle splitting and merging based on intensity distributions in elements. A similar technique that also considers shape quality measures was recently proposed by Reid *et al.* [21] for segmenting micro-structures in 2D medical images. However, such methods involve several local mesh operations (e.g. annealing, smoothing, fixing in [21]) and their extension to 3D is not trivial. In contrast, our proposed penalty-based mesh evolution scheme to minimal image partitioning problem has multiple advantages.

Most segmentation methods require some sort of a manual intervention not only demanding the scarcely available time of health professionals but also preventing an automatic modelling for FEM. Furthermore, there are certain drawbacks with modeling approaches consisting of two separate steps of segmentation and meshing. For instance, the surface triangulation generated

during the segmentation is often left unchanged using common meshing software since this surface is used as a boundary constraint for meshing. As a result, to guarantee good FEM elements using a limited number of nodes/elements, these intermediate surfaces should either be inherently FEM-friendly delineations or should be *fixed* using additional intermediate steps.

Some methods in the literature generate meshes from given voxel datasets, where each voxel is marked *a priori* as being inside or outside of a structure [8, 15–17]. At different sources, such voxel volumes are named as labeled medical datasets, multi-label partitions, or bit volumes. These can be generated using voxel clustering (segmentation) techniques, such as simple intensity thresholding [12]. Therefore, the above-specified works from the medical image meshing literature, which rely either on *a priori* data or on external query mechanisms for deciding whether a point in space is inside or outside a structure, require a segmented volume as input. By the very design of these methods, an internal surface in the input is used as the boundary on which the mesh faces are fit or refined. Consequently, only the surface information is kept and any other image variation is discarded before starting the meshing process. This prevents the meshing procedure from making any trade-offs between fitting the surface or preserving mesh element qualities for a given number of nodes. Furthermore, any features that are visible in the image, other than the segmented/labeled structures, are also discarded, even when the mesh can also include those features without any further cost.

In contrast to the methods in the literature, that do not address the segmentation within the meshing framework, our method presented in this chapter implements this segmentation intrinsically. It takes raw (unprocessed) intensity images, that have not been segmented or labeled, from a given image modality and generates meshes considering FEM quality measures. At the same time, our method ensures an optimal partitioning (clustering) of voxels into tetrahedra. Essentially, meshing is the discretization of space into elements (geometrical primitives). Our method assigns a cost for each such discretization based on its projection on a given image and chooses one with a minimal cost. Note that such optimally-partitioning tetrahedra can indeed be segmented/labeled into structures following the meshing. These labeled surfaces, which can be determined such as by using thresholding as shown in this chapter, will consist of only internal mesh facets, hence inherently complying with the mesh. Thus, the resulting FEM tessellations can be generated using our method in a single-step variational framework.

2.2 Previous Work on Optimal Tessellations

During finite element modelling of deformation, the two main sources of error are *interpolation errors* of the approximation to the function and its gradient (which is strain for deformation), and *numerical errors* during the solution of the approximation. Numerical errors depend mainly on the conditioning of the finite element matrices involved and also on the numerical precision and type/order of mathematical operations applied in obtaining the solution. Interpolation errors are affected by various characteristics of the mesh and the approximant function, such as the polynomial order of the function approximation. Among these error contributors are the shape and size of mesh elements [22]. Consequently, it has been an active field of research to obtain meshes that will introduce minimal amount of such errors in FEM simulations.

Given a set of vertices, their Delaunay tessellation offers several favorable features compared

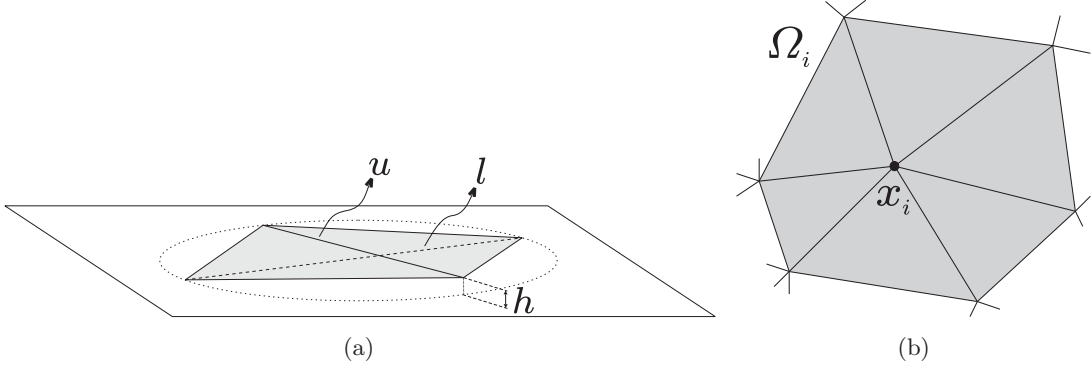


Figure 2.1: (a) A type of ill-shaped flat tetrahedron called *sliver*. (b) *1-ring neighbourhood* of a node.

to other possible tessellations. In particular, when a function f has bounded second derivatives, Delaunay tessellation minimizes the worst interpolation error between f and its piecewise linear approximant defined by this tessellation (mesh). Delaunay tessellation received significant attention in the mesh generation literature.

The major problem with Delaunay refinement approaches to mesh generation in 3D has been the existence of *slivers* in the final meshes. Slivers are a class of degenerate tetrahedra that are not necessarily removed by the Delaunay refinement. The vertices of a sliver are located nearly co-circular with one vertex being slightly out of the plane of the other three vertices as in Fig. 2.1(a). One of the detrimental effects of such tetrahedra on linearly approximating the gradients of a function (e.g. deformation) can be demonstrated by the following example: Consider linearly approximating an underlying function near the center of this circle by first approximating the values at the mid-points of the upper (u) and lower (l) tetrahedron edges. The value at each mid-point is found as the average of the known vertex values at the two (opposite) ends of this edge. If these two mid-point approximations differ, the gradient between them will be arbitrarily large as the sliver degenerates, i.e. $h \rightarrow 0$, as the mid-points are located very close to each other in space. This degeneracy of such tetrahedra are not correctly captured by many commonly-used mesh quality measures such as the edge-radius ratio, which is the ratio between the shortest edge length and the circumsphere radius. Edge ratios, area-volume ratios, and minimum/maximum dihedral angles are among various other quality measures suggested in the literature. Reviews of these measures with their relation to function approximation and FEM were given by Field [23] and Shewchuk [22]. Accordingly, the ratio of inradius to circumradius was presented by Alliez *et al.* [24] as the *fairest* quality measure and comparison in 3D, that punishes all types of poor-geometry elements including slivers, which most other measures fail to identify.

To achieve better-shaped elements, modifications to Delaunay refinement and other local optimization schemes were proposed in the literature [25–27]. However, these methods require significant implementation effort and result in non-convex functionals that are difficult to analyze and derive theoretical guarantees for. More recent approaches to mesh smoothing such as the Centroidal Voronoi-Delaunay Tessellation (CVDt) [28] and Optimal Delaunay Triangulations (ODT) [29, 30] consider minimizing a quadratic energy through updates of vertex

positions and their connectivity. Advantages of these methods are (i) the simplicity of each update and (ii) the implication of eventual convergence due to the monotonically-decreasing energy definitions. Building on these and other previous work, Alliez *et al.* introduces a 3D mesh generation technique producing superior meshes compared to similar methods in the literature [24]. This method is adapted in this work.

For anisotropic meshing, Chen [30] notes that ODT equidistributes the edge lengths of a mesh under a metric related to the Hessian of the approximated function $f(\mathbf{x})$. Alliez *et al.* adjust the edge length distribution in space using a mesh density function, e.g. to have finer elements modeling higher curvature boundary surfaces of a 3D object [24]. However, it is not clear from these and other work, how the information about a feature distribution through the tissue (e.g. distribution of elastic modulus) can be incorporated into such variational meshing techniques, assuming this distribution is known in the continuum *a priori*. Our proposed technique addresses this aspect of meshing.

The methods for image-compliant meshing are introduced in Section 2.3 and a numerical approach for their use is presented in Section 2.4. The results from various imaging modalities are demonstrated in Section 2.5, which is followed by a discussion and conclusions in sections 2.6 and 2.7, respectively.

2.3 Methods

The method involves combining an element geometric quality metric with an image-compliance based metric into an objective function to be minimized. First, the use of the former metric alone is described as it is also employed for initializing the optimization of the combined function. Next, an optimal parameter discretization of the elements of a tessellation is presented. Based on this, an image-compliance measure is defined for an optimal image partitioning and, finally, its combination with the geometric definition above is introduced.

2.3.1 Geometric Energy for Mesh Optimization

Let $g_{\mathcal{T}}(\mathbf{x})$ be a piece-wise linear approximation of a function $f(\mathbf{x})$ on a given tessellation \mathcal{T} . It is known that the integral approximation error $E_G = \|g_{\mathcal{T}} - f\|_{\mathcal{L}^1}$ is minimized by the Delaunay tessellation of the domain $\mathcal{M} \subset \mathbb{R}^N$ when (i) the function f is a paraboloid, i.e. $f(\mathbf{x}) = \|\mathbf{x}\|^2$, and (ii) $g_{\mathcal{T}}$ is its *overlaid* circumscribing piece-wise linear approximant [29]. This was used in iterative optimization schemes in the literature to obtain geometrically high-quality meshes [24, 30] and is also adopted as the *geometrical* mesh quality measure in our method in the form given below:

$$E_G = \sum_{\tau \in \mathcal{T}} \int_{\tau} g_{\tau}(\mathbf{x}) d\mathbf{x} - \int_{\mathcal{M}} f(\mathbf{x}) d\mathbf{x} \quad (2.1)$$

$$= \frac{1}{N+1} \sum_i f(\mathbf{x}_i) |\Omega_i| - \int_{\mathcal{M}} f(\mathbf{x}) d\mathbf{x} \quad (2.2)$$

$$= \frac{1}{N+1} \sum_i \mathbf{x}_i^2 |\Omega_i| - \int_{\mathcal{M}} \mathbf{x}^2 d\mathbf{x} \quad (2.3)$$

where Ω_i is the *1-ring neighbourhood* of node x_i , that is the set of elements having that node as a corner as in Fig. 2.1(b). This neighbourhood will be referred as *1-ring* throughout this

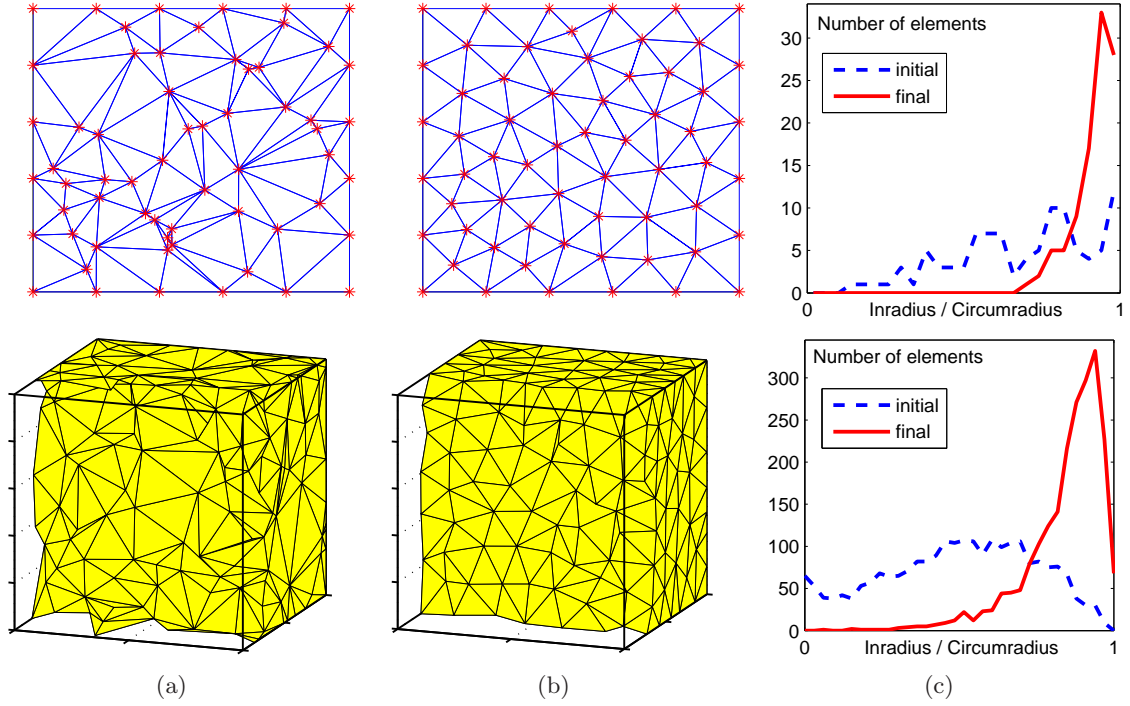


Figure 2.2: (a) Random 2D/3D mesh initializations, (b) their *geometrically* optimized configurations, and (c) the initial/final aspect ratio ρ distributions.

chapter.

Examples of optimizing this geometric energy E_G alone are shown in Fig. 2.2 in 2D and 3D, where the meshes were initialized using a random distribution of vertices as in Fig. 2.2(a). This optimization involves *Lloyd's relaxation*, which is performed as alternate updates: a global Delaunay tessellation of given node locations \mathbf{x}_i and a node relocation so that the above-given cost function is minimized. Figure 2.2(b) shows the optimized meshes after having converged.

During various mesh generation steps and examples in this chapter, inradius-to-circumradius ratio is used to control and observe the quality of elements. This ratio ρ is normalized by the dimension N of the meshing domain [23]:

$$\rho = \frac{N r}{R} \quad (2.4)$$

where r and R are inradius and circumradius, respectively, such that it takes on values in the range $[0, 1]$ and is maximized for a regular simplex. Unlike in 2D where an infinite domain can be meshed with all regular triangles, 3D space unfortunately cannot be tiled by all regular tetrahedra alone. Therefore, 3D meshes are bound to have some sub-optimal elements in practice. Figure 2.2(c) shows the improvement in the inradius-to-circumradius ratio ρ with the optimization procedure. A ρ -histogram compacted towards larger values in general indicates a better mesh having more elements with higher quality.

2.3.2 Element Discretization

In this section, a penalty-function for image discretization is defined based on a mesh element and the image pixels/voxels covered by that element. Since an element uses a discretized value to represent any spatial location within itself, an \mathcal{L}_2 -norm difference of the discretized element value and the voxels represented by that value is used as the metric for the error relating to this discretization. This definition is in a way similar to the common Mumford-Shah functional used in various image segmentation methods to partition pixels based on their similarity [31]. However, in contrast to such methods, where often there is a term constraining the length/curvature of segmenting curve for regularization purpose, the planar faces of mesh elements in our method already constrains the way that image pixels can be partitioned in space.

Let $h(\mathbf{x})$ denote the distribution of a feature in \mathcal{M} , the domain of interest. In the FEM, such a feature is discretized on a mesh so that each mesh element τ_j has a single assigned value, \tilde{h}_j , modeling this feature within that element. Let the discretization error associated with this single value approximation \tilde{h}_j and the values that it represents, $\{h(\mathbf{x}), \mathbf{x} \in \tau_j\}$, be defined as an \mathcal{L}_2 -norm:

$$E_D^j = \int_{\tau_j} |h(\mathbf{x}) - \tilde{h}_j|^2 d\mathbf{x}. \quad (2.5)$$

It is evident that for a constant h within an element, the best \tilde{h}_j is that constant value itself. In general, this error is minimized for \tilde{h}_j being the average mean value of the given distribution. Therefore, for a given element and given (background) feature distribution, a discretized feature value \tilde{h}_j is assigned to that element τ_j as the average of $\{h(\mathbf{x}), \mathbf{x} \in \tau_j\}$. This average mean discretization is further discussed in the context of elastic strain energy formulation for FEM in Section 2.6.2.

2.3.3 Objective Function for Image Compliance

Assigning a single-value equivalent of a known feature distribution (an image) within an element as the average mean value is demonstrated in Fig. 2.3 for a structured mesh overlaid on a synthetic phantom. The element shading in Fig. 2.3(b) denotes the average intensity of the underlying image pixels.

The element discretization error defined in (2.5) is integrated over the mesh to yield the following cost function describing the fitness of a mesh to an image:

$$E_D = \sum_j \int_{\tau_j} |h(\mathbf{x}) - \tilde{h}_j|^2 d\mathbf{x}. \quad (2.6)$$

Note that with the earlier assumption of \tilde{h}_j being the average mean value in the element, (2.6) can be rewritten as:

$$E_D = \sum_j v^j \mathbf{var}(h(\mathbf{x}) : \mathbf{x} \in \tau_j) \quad (2.7)$$

where \mathbf{var} is the second-moment of a distribution around its mean, namely the *variance*. Thus, this definition penalizes elements with larger image intensity variations. However, it does not enforce suitability of element size and shape for the FEM. As a result, in order to derive a

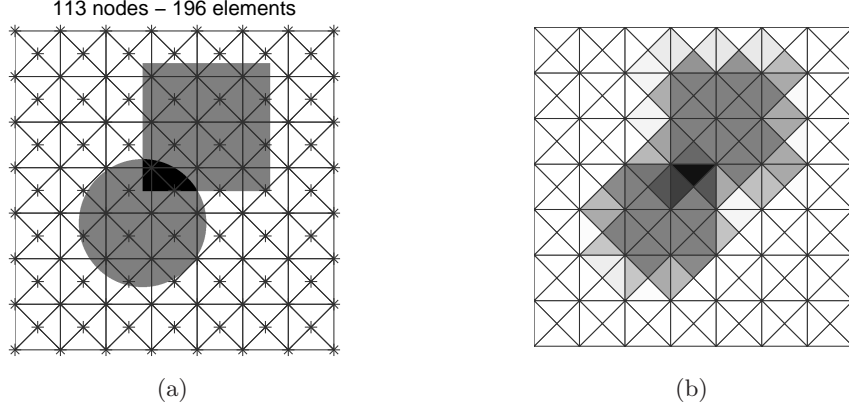


Figure 2.3: An initial synthetic phantom (a) with its discretization (b).

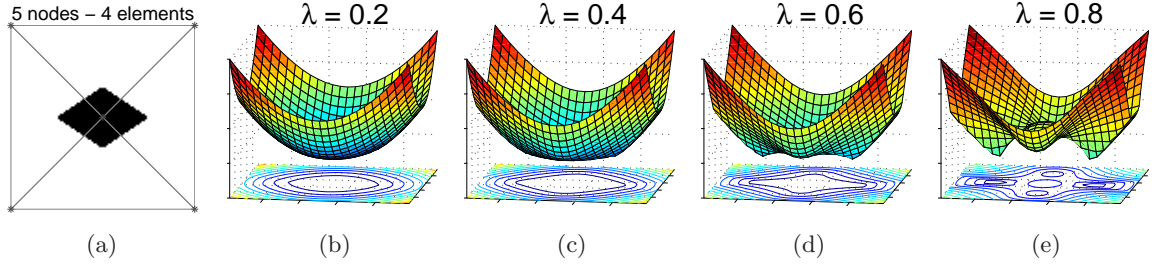


Figure 2.4: A simple four-element mesh (a) and the combined cost E as a function of mid-node position for λ of (b) 20%, (c) 40%, (d) 60%, and (e) 80%.

variational scheme trading off between element geometry and image representation, the two error metrics E_G and E_D above are combined as follows:

$$E = (1 - \lambda)E_G + \lambda E_D \quad (2.8)$$

where $\lambda \in [0, 1)$ is the weighting factor of discretization.

A visualization of this combined error is presented for a simple 2D mesh in Fig. 2.4, where E is plotted as a function of the center node position. For this illustration, a diamond shaped image is used as in Fig. 2.4(a). Then, (2.8) is calculated for the mesh using (2.3) and (2.7), as the position of the center node is changed. For this illustration, the triangulation and the four corner node position defining the boundary are kept fixed. The combined cost E is seen in figures 2.4(b)–(e) as a function of the center node position for different weighting factors λ . Note that with the two competing measures in this example: (i) the best triangle aspect ratios are achieved by minimizing E_G when the mid-node is placed near the center, and (ii) the variance of some triangles reduce to zero minimizing E_D when the mid-node is placed near the corners of the diamond shape. Consequently, for small λ values, E is minimized at the center disregarding the underlying image, whereas the shape corners are captured better by the node as λ increases.

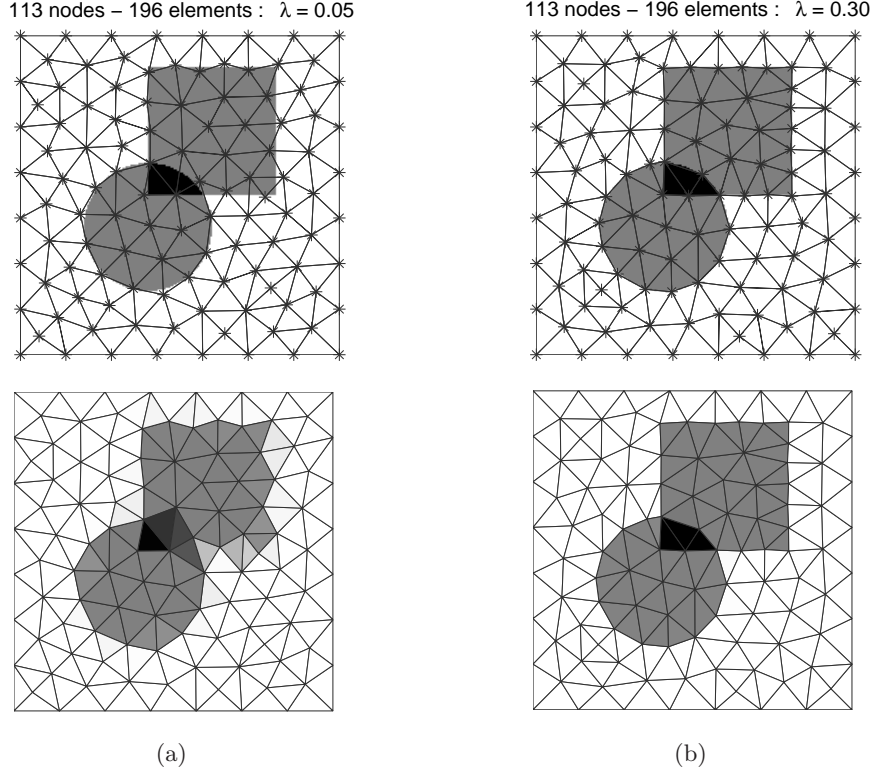


Figure 2.5: The initial phantom mesh in Fig. 2.3 optimized using (a) $\lambda = 0.05$ and (b) $\lambda = 0.30$ are shown as meshes overlaid on the image (top) and the corresponding image approximations (bottom).

2.4 Implementation

The objective E is a function of node location x and mesh connectivity (tessellation) \mathcal{T} . This cost function is minimized in a numerical optimization scheme, where node locations and element connectivity are updated alternately. Figure 2.5 demonstrates two example mesh models generated for the 2D synthetic phantom in Fig. 2.3 after optimization using the presented method with two different weighting constants λ . In order to accommodate for the incorporation of the image-based cost component, the optimization procedure based on Lloyd's relaxation as in other CVDT or ODT optimizations in the variational meshing literature is restructured as described below.

2.4.1 Mesh Initialization

In order to initialize the optimization procedure using the combined cost E , first an optimum geometrical configuration is found by minimizing E_G alone. This is indeed equivalent to taking $\lambda = 0$ in our formulation. This initialization step is primed using nodes that are at uniformly-distributed random locations. The mesh output of this step is referred as the *initial* mesh throughout the rest of this chapter, since it is subsequently used as the starting configuration

for our proposed optimization method.

2.4.2 Node Updates

Consider the optimality condition for a mesh using the combined metric E , with the addition of the image-based component E_D to the geometrical definition. In an *optimal* mesh, the gradient of cost function E with respect to the position of each node within its 1-ring approaches zero. Each node is updated to minimize the cost using this *necessary* optimality condition according to Lloyd's relaxation. To compute E_G and E_D , (2.3) and (2.7) are employed respectively. Finite-differences were then used for the gradient of the combined E in the following two optimization schemes. The first method aims at finding an exact optimum location within each 1-ring at each iteration and the second approximates this location. The benefits of this approximation in terms of computational time and convergence are discussed later in Section 2.6.

Constrained Non-Linear Optimization

The optimal location of \mathbf{x}_i minimizing E is found by taking the 1-ring polygonal/polyhedral region Ω_i seen in Fig. 2.1(b) as the feasible solution region. Then, a non-linear optimization scheme is used where the perimeter of Ω_i is defined as a set of inequality constraints. Ω_i is bounded by a set of lines in 2D (outer edges of neighbouring triangles as in Fig. 2.1(b)) and by a set of planes in 3D (outer triangular faces of neighbouring tetrahedra). Therefore, the feasible region Ω_i can be described as a constraint $A_i \mathbf{x}_i < b_i$ as given in Appendix A, yielding a node location update as:

$$\mathbf{x}'_i = \arg \min_{\mathbf{x}_i \in \Omega_i} E(\mathbf{x}_i) \quad (2.9)$$

$$= \arg \min_{\mathbf{x}_i} E(\mathbf{x}_i) \quad s.t. \quad A_i \mathbf{x}_i < b_i. \quad (2.10)$$

This is implemented in Matlab using *sequential quadratic programming*.

Gradient Descent with Parabola Fit

In this method, each node \mathbf{x}_i is updated (relocated) locally at every iteration using the error E_{Ω_i} in its 1-ring Ω_i , calculated for perturbations in each coordinate axis by a given *step length*. Sub-step length updates are further achieved using a parabola fit in each individual axis. Prior to updating each node, an initial perturbation magnitude (step length) is defined using the distance from this node to its closest element centroid in Ω_i . If a perturbation were to invert a neighbouring tetrahedron, then that immediate step length is reduced using a *golden search method* until a feasible perturbation is reached or the step length becomes too small with respect to the image resolution. Note that the smallest feature that can be detected and modeled is limited naturally by the image pixel/voxel size. This relation to image resolution is further discussed in Section 2.6.

2.4.3 Connectivity Updates

Out of all possible tessellations of given nodes, the Delaunay tessellation offers a mesh that minimizes the geometric measure. Unfortunately, there is no similar result for the combined

objective function. Due to the image-based component, a tessellation that is not optimal for geometry alone can still be optimal for the combined function, for instance, by capturing the image variation better.

In this thesis, a simple connectivity update strategy is adapted, where at each iteration, the Delaunay tessellation of current nodes is considered as a connectivity alternative. Such a tessellation is known to reduce E_G and therefore has potential to reduce the combined cost E as well. Furthermore, there exist well-established fast methods for finding such Delaunay tessellations. The connectivity update is performed as follows: at each iteration following the update of node positions, the mesh is checked to determine whether it satisfies the Delaunay criterion. If this is the case, no further action is taken. Otherwise, first the Delaunay tessellation is found, then the combined error is calculated for this new tessellation and compared to the current E . Accordingly, the tessellation with smaller cost is chosen to proceed for node updates in the following iteration.

2.4.4 Normalization of Cost Weighting Factor

Our design parameter λ sets the trade-off between geometry and image compliance. However, its actual effect on the result depends on many factors such as the domain size, the number of elements, and the intensity variation of the given image. Depending on such parameters, E_G and E_D may have very different scales separated by orders of magnitude. Consequently, in order to have control over the range of practical λ values an additional step of E_G/E_D normalization is employed. In the first iteration where E_G is optimum, a scaling factor between the values of these two error definitions is calculated so that the given λ will indeed be interpreted as the percentage error contribution of discretization rather than an absolute quantity. This scaling factor is then fixed and used for the rest of the iterations. This treatment normalizes λ allowing us to report consistent *effective* values (percentages), which are easier to associate. For instance, a λ of 30% means that the optimization process targets minimizing the cost combined as 30% discretization and 70% geometry components with respect to their initial error contributions.

2.4.5 Convergence Measure

A convergence measure for the optimization is set in this chapter as the root-mean-square nodal position update at iteration t normalized by the mesh bounding-box size b as follows:

$$v = \frac{1}{b} \sqrt{\frac{1}{n} \sum_i^n (\mathbf{x}_i^{t+1} - \mathbf{x}_i^t)^2} \quad (2.11)$$

where n is the number of mesh nodes. A mesh is considered to have converged when $v < 10^{-3}$.

2.4.6 Element Aspect Ratios

In general, a ρ -histogram compacted towards this maximum value indicates a better mesh. However, this qualitative comparison of mesh ρ -distributions is not practical in many cases. In this thesis, the mean value of $(1 - \rho)$ is observed as a quantitative measure for the compactness of these histograms towards the optimal quality value of 1. Theoretically, the worst element

defines the performance, however the overall distribution of the aspect ratios is often also of great interest. As a result, the average, quadratic, and maximum mean values of $(1 - \rho_i)$ for all elements i are observed during optimization steps in this chapter. For instance, the quadratic mean metric is calculated as follows:

$$\sqrt{\frac{1}{n} \sum_i^n (1 - \rho_i)^2} \quad (2.12)$$

The worst aspect ratio $\max\{1 - \rho_i\}$ is also used to identify few cases where an ill-shaped tetrahedron is not removed by the standard meshing procedure. A vertex *insertion/teleportation* to the centroid of such element is employed to recover such cases [24].

2.4.7 Summary of the Algorithm

Below is the overview of the algorithm, showing the concepts introduced above.

```

MeshInitialization()
Normalization()
REPEAT
    MoveNodes()
    UpdateConnectivity()
UNTIL (meshConverged)
PostProcess()
    
```

This block of algorithm starts with an initial mesh, such as a regular grid, or generates a mesh with node positions already optimized for best geometric distribution at the beginning step of *MeshInitialization* (Section 2.4.1). The step *Normalization* refers to finding the scaling between the initial E_G and E_D for a normalized λ (Section 2.4.4). The method *MoveNodes* visits and repositions each node inside its 1-ring, such that the cost of that 1-ring will decrease (Section 2.4.2). This step assumes the connectivity is fixed, while the next step *UpdateConnectivity* considers an alternative tessellation and switches to that one if a lower cost is promised by that (Section 2.4.3). Finally, a new mesh complying with a given image is generated, when the mesh converges according to the given measure in Section 2.4.5. *PostProcess* ensures that the resulting elements have acceptable aspect ratios as later discussed in Section 2.6.6.

2.5 Results

A 2D slice from MR image data of the brain is shown in Fig. 2.6(a). Figures 2.6(b)-(e) present image discretizations by the initial and final meshes for two sample optimizations at different mesh resolutions. The converged mesh of the latter case is seen overlaid on the initial image in Fig. 2.6(b). Similarly, the image and the discretizations from two optimizations for a 2D CT slice of the kidney are presented in Fig. 2.7. The figures referred as initial and final meshes throughout this section denote the configurations before and after the application of our cost minimization technique presented. It is seen in these figures qualitatively that, once a mesh is optimized using the technique presented, a superior image representation is achieved even when a relatively coarse mesh is employed. The reduction in cost for these examples are presented

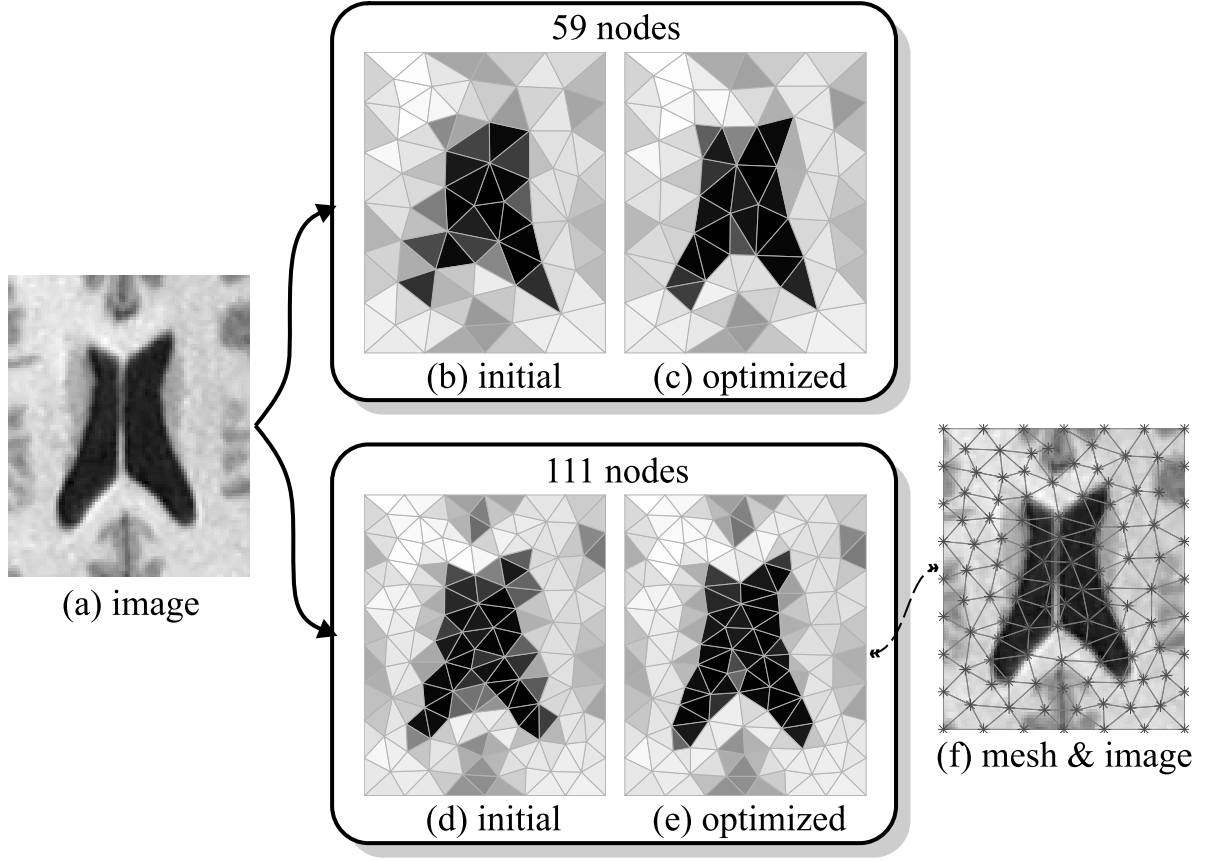


Figure 2.6: Mesh optimization on a 2D MR image slice (a) of brain ventricles. Initial (b) and optimized (c) discretizations using 59 nodes (both having 96 triangles); and initial (d) and optimized (e) discretizations using 111 nodes (both having 192 triangles). The finer optimized mesh is also shown overlaid on the image (f).

numerically later in this section. The minimization of this combined cost indeed quantitatively confirms a better image representation for a given mesh resolution.

The feasibility of our method is next demonstrated in 3D. The initial and final meshes of a synthetic phantom with a spherical inclusion are seen in Fig. 2.8. For the presentation of 3D results, along with cutaway views of the meshes, segmentations of corresponding anatomy of interest using a simple operation of element thresholding are also presented. This thresholding method is detailed in Section 2.6.1.

Slices from a 3D MR image volume of brain ventricles are seen in Fig. 2.9(a) with part of the mesh that was used to initialize the optimization presented in this figure. Initial and final cutaway views along with ventricle meshes thresholded as described above are also shown in Fig. 2.9, where the shading of each element face indicates the discretized \tilde{h}_j value within that element. In Fig. 2.9(h), the final thresholded mesh is seen embedded within the cutaway view. Similarly, Fig. 2.10 presents mesh optimization within a 3D CT volume of the kidney region.

The evolution of the combined cost values E during the optimization of some of the exam-

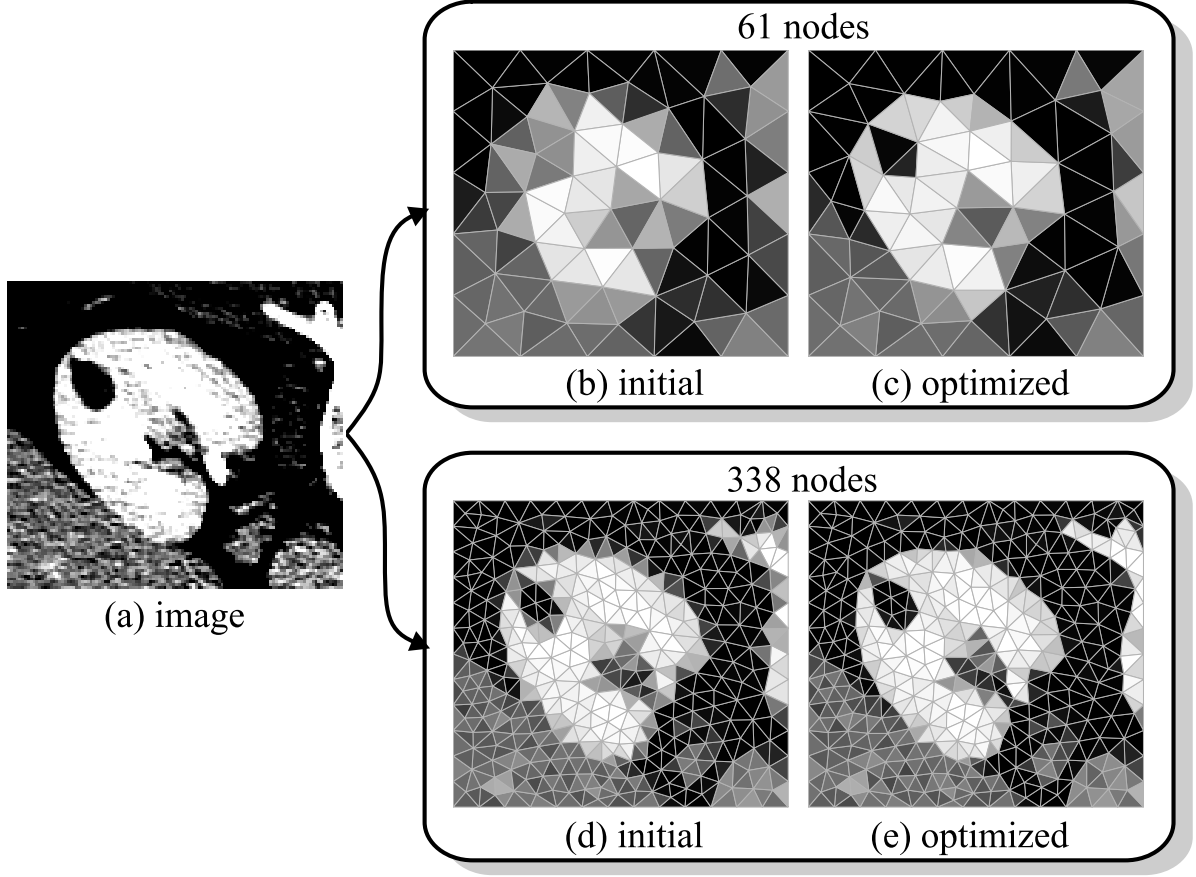


Figure 2.7: Mesh optimization on a 2D CT image slice (a) of the kidney. Initial (b) and optimized (c) discretizations using 61 nodes (both having 100 triangles); and initial (d) and optimized (e) discretizations using 338 nodes (both having 624 triangles).

ples presented in this chapter are plotted in Fig. 2.11. These were normalized to their initial mesh quantities in order to present their change in percentage. The first and last iterations in this figure correspond to the cost at the *initial* and *final* mesh configurations, respectively. These are the meshes optimized using geometry alone and using geometry combined with image compliance errors, respectively, for the medical imaging examples in previous figures. The reduction of cost in all these examples implies that the mesh fits the image *better* when optimized by our method. Note that starting from a geometrically-optimal *initial* configuration ensured by mesh initialization in Section 2.4.1, indicates that it is the image compliance component that is minimizing over iterations. Meanwhile, the geometry component increases slightly until a balance is reached for given weighting constant λ . The image-compliance component is driven from optimal partitioning (segmentation) literature and accordingly is intrinsically indicates the fitting of a discretization to an image.

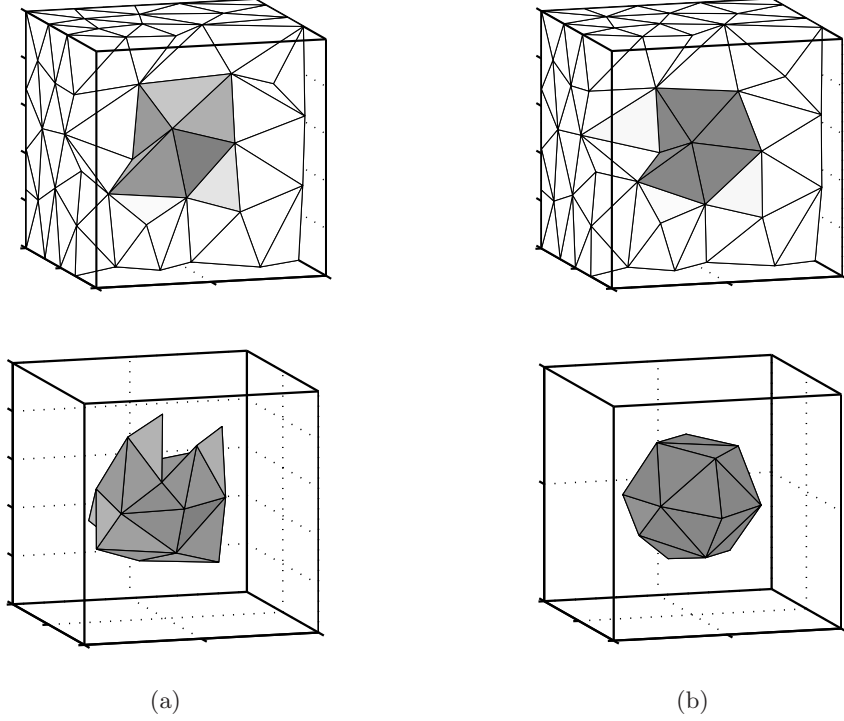


Figure 2.8: A 3D mesh optimization for a synthetic phantom image with a spherical inclusion: (a) initial and (b) optimized meshes. In cutaway views of meshes (top), element shading represents the discretized values of cut elements. The inclusion extracted from each mesh using element thresholding is also seen (bottom).

2.6 Discussion

2.6.1 Element Thresholding for Segmentation

At any optimization iteration including the initial and converged ones, once the average mean \tilde{h} of the image voxels within every element are found, the elements below/above a given threshold can be culled from the visualization. Subsequently, the remaining elements are grouped into bins (*objects of interest*), so that any element can be reached from any other element in the same object by only traversing through neighbouring element faces. Partitioning a discretized mesh into sets of such connected regions presents a way of segmenting this mesh. The largest such connected set is presented in the examples of this chapter as the mesh representation of the anatomy of interest. Note that extracting a surface from these volumetric meshes is not the focus of this thesis and more sophisticated methods can indeed be developed for this. The thresholded mesh figures in this chapter are mainly presented for visualization purposes in order to demonstrate the improvement with respect to the initial meshes.

Recall that our method balances two metrics in the volume, not only focuses on the image compliance of the surface, therefore it is not fair to compare such surfaces extracted from meshes with other surface segmentation methods such as active contours. Nevertheless, it is

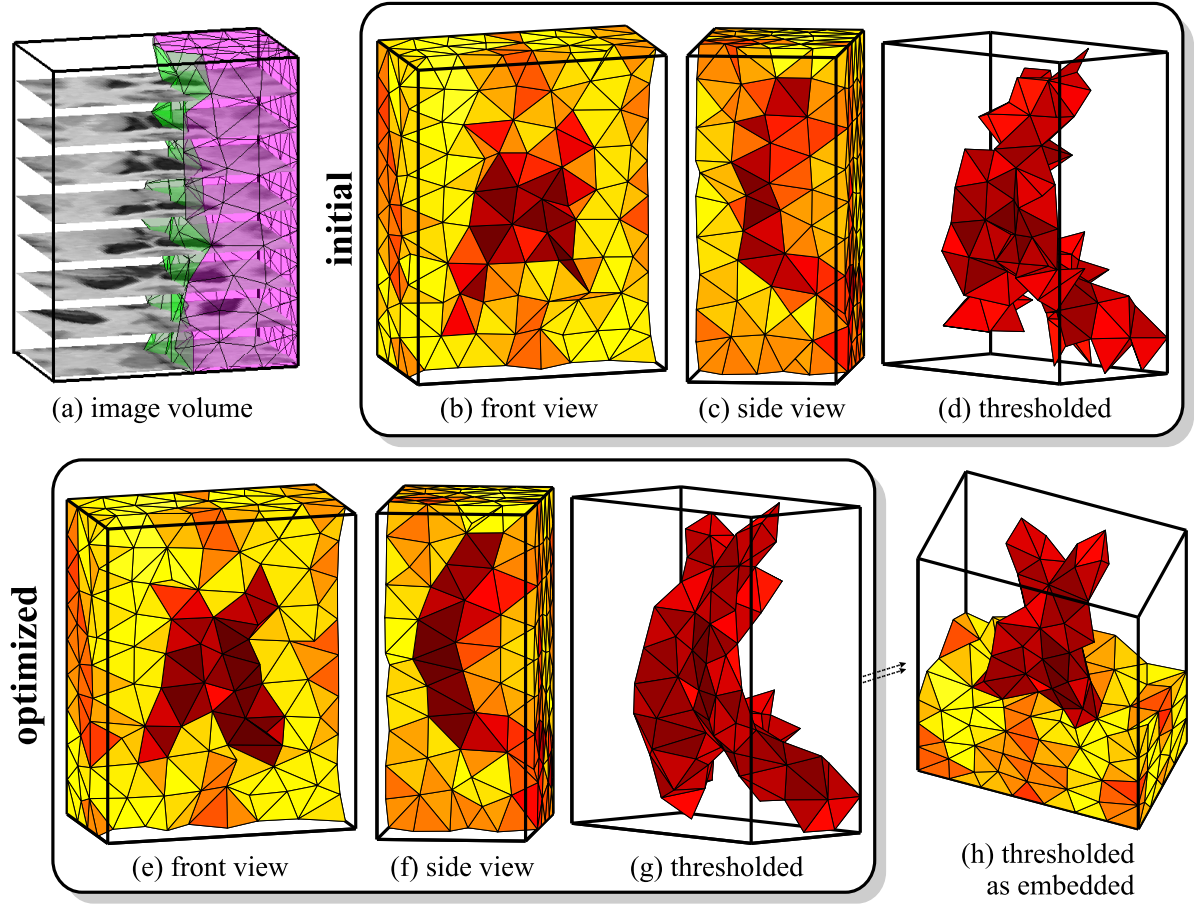


Figure 2.9: Mesh optimization for 3D MR image volume of the brain. Initial mesh with 410 nodes: (a) part of the mesh with some image slices from the voxel volume, (b)-(c) two cutaway views showing discretized element values on faces, and (d) the thresholded elements. A solution after 32 iterations for comparison with the initial mesh: (e)-(f) two cutaway views and (g) the thresholded elements, where (h) presents these elements embedded in a cutaway view.

valuable to show that, once an image is meshed, if a (rough) surface/anatomy segmentation is also needed, the presented thresholding method or others can be used to extract the anatomical structures. Indeed, considering that these meshes are optimal representation of images at a lower-resolution than the original voxel volume, such methods can run faster than conventional voxel-based segmentations also enjoying the (optimal) smoothing effect due to discretization in mesh elements.

2.6.2 Discretization of Known Elastic Modulus Distribution

In this section, the element discretization as the average mean in Section 2.3.2 is shown to be consistent with deformation modeling using FEM. Therefore, using the proposed discretization cost function indeed results in optimal FEM meshes minimizing the error due to parameter

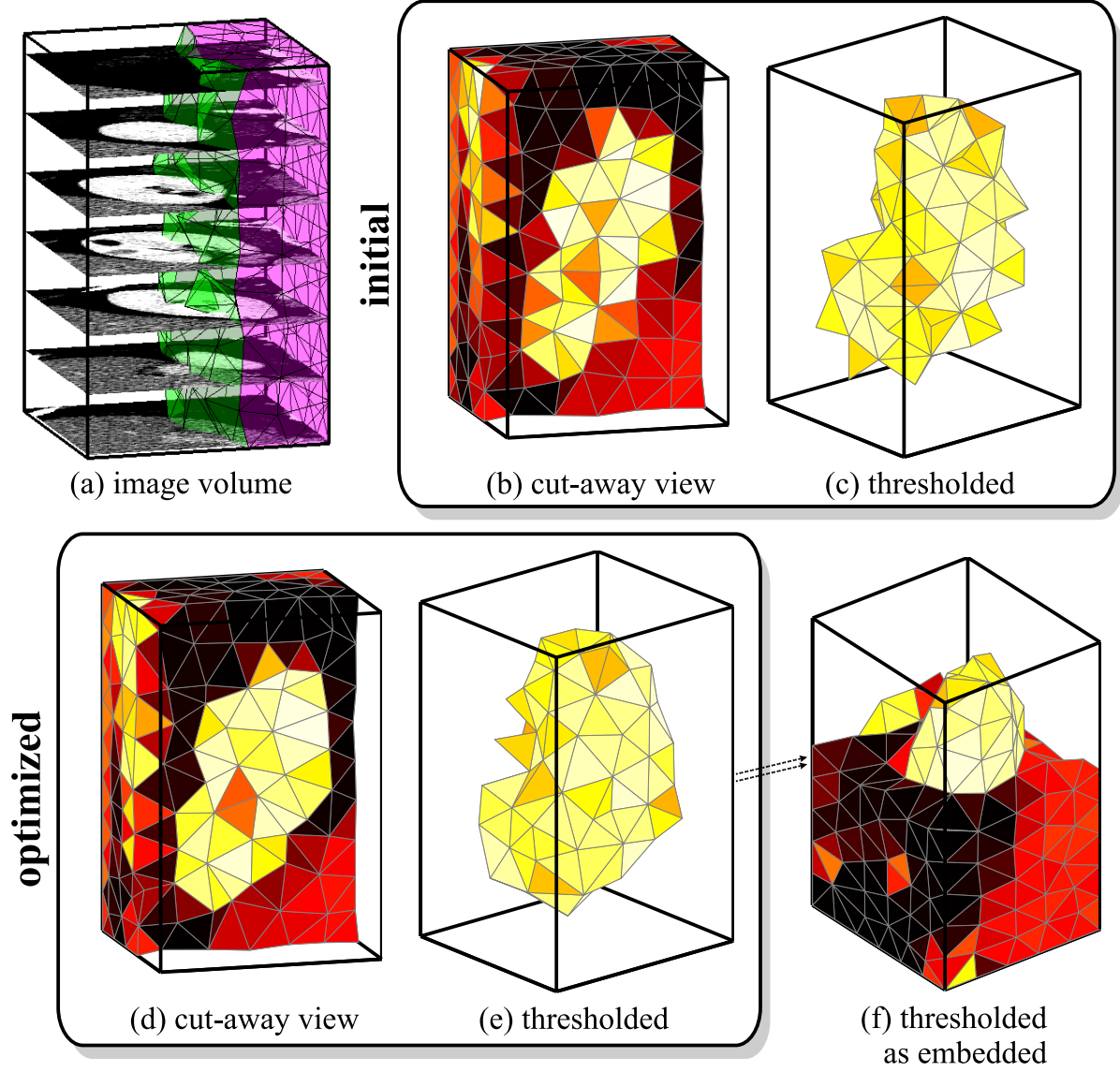


Figure 2.10: Mesh optimization for 3D CT data of the kidney. Initial mesh: (a) with some image slices from the voxel volume, (b) a cutaway view, and (c) the thresholded elements. The solution presented here converged in 25 iterations: (d) a cutaway view and (e) the thresholded elements, where (f) presents these elements embedded in a cutaway view. The meshes have 420 nodes and more than 1600 elements.

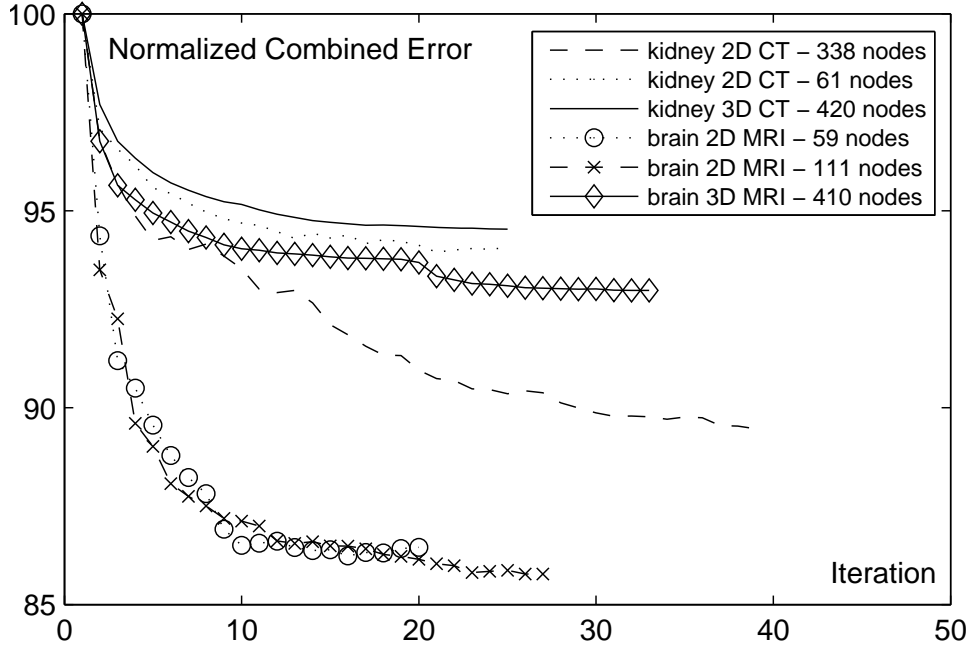


Figure 2.11: Combined cost E during the optimization of the examples in figures 2.6, 2.7, 2.9, and 2.10. In these corresponding figures, the mesh and the resulting image discretizations for the first and the last iterations shown here were depicted as initial and final configurations, respectively.

discretization.

For a linear stress-strain relationship, the elastic strain energy of a 4-node tetrahedral element can be written in terms of the four corner displacements \mathbf{u}^j as:

$$E_{strain}^j(\mathbf{u}^j) = \frac{1}{2} \int_{\tau^j} \mathbf{u}^{jT} B^{jT} C(\mathbf{x}) B^j \mathbf{u}^j d\mathbf{x} \quad (2.13)$$

where $C(\mathbf{x})$ is the element material stiffness matrix and B^j is the constant partial-derivative matrix, which is derived using the integration of barycentric coordinates within the element and hence is fixed for given tetrahedron corner positions [32].

In the conventional derivation of element strain energy, the material stiffness matrix is constant within each element, i.e. $C(\mathbf{x}) = C_j$, since the material properties, Young's modulus and Poisson's ratio, are discretized as constants in each element. Then, (2.13) leads to:

$$E_{strain}^j(\mathbf{u}^j) = \frac{1}{2} \mathbf{u}^{jT} B^{jT} C_j B^j \mathbf{u}^j \int_{\tau^j} d\mathbf{x} \quad (2.14)$$

$$= \frac{1}{2} \mathbf{u}^{jT} B^{jT} C_j B^j \mathbf{u}^j v^j. \quad (2.15)$$

where v^j is the element volume.

C can indeed be written as a linear function of Young's modulus, i.e. $C_j = \tilde{\mathcal{E}}_j C'_j$, where $\tilde{\mathcal{E}}_j$

is the Young's modulus discretization in this element. Consequently, (2.15) yields:

$$E_{strain}^j(\mathbf{u}^j) = \frac{1}{2} \mathbf{u}^{jT} B^{jT} C'_j B^j \mathbf{u}^j \tilde{\mathcal{E}}_j v^j. \quad (2.16)$$

It is a common assumption to take the Poisson's ratio constant in a soft tissue domain. This is acceptable considering the nearly incompressible characteristic of soft tissues. However, the Young's modulus does often change substantially between different tissue structures. Assume that this Young's modulus distribution, $\mathcal{E}(\mathbf{x})$, within the domain \mathcal{M} is known *a priori*. There exist several methods in the literature for the acquisition and derivation of tissue elasticity, see [33] for a review. Consequently, the material stiffness matrix in the element can be formulated as $C(\mathbf{x}) = \mathcal{E}(\mathbf{x}) C'_j$. Substituting this in (2.13) yields:

$$E_{strain}^j(\mathbf{u}^j) = \frac{1}{2} \mathbf{u}^{jT} B^{jT} C'_j B^j \mathbf{u}^j \int_{\tau^j} \mathcal{E}(\mathbf{x}) d\mathbf{x}. \quad (2.17)$$

For the discretization within each element to be optimal, these two energy formulations in (2.16) and (2.17) should be equal, leading to:

$$\tilde{\mathcal{E}}_j v^j = \int_{\tau^j} \mathcal{E}(\mathbf{x}) d\mathbf{x} \quad (2.18)$$

which is satisfied when $\tilde{\mathcal{E}}_j$ is the mean of the distribution within element τ^j .

To demonstrate mesh optimization from mechanical tissue features, the method was applied to prostate elastography images acquired using the *vibro-elastography* technique of Salcudean *et al.* [34]. Elastography is the technique in which tracked localized displacements in response to a mechanical excitation allow for the identification of mechanical tissue properties [33, 35]. For the purpose of this chapter, a 2D sagittal transfer function image of the prostate is meshed. A mesh of the prostate and the pelvic bone using this technique from a labeled voxel volume is presented in Chapter 5 in the context of patient modeling for a prostate brachytherapy training system. The prostate, which is typically stiffer than its surrounding, is seen as the darker region in Fig. 2.12(a) with an initial mesh atop. When converged, this mesh takes the form seen in Fig. 2.12(b).

2.6.3 Connectivity and Node Updates

The approximation accuracy of a mesh depends on the number of nodes/elements involved and, due to the computational cost of FEM, often a limited number of nodes can be accommodated in a simulation. Several meshing studies in the literature define an approximation performance for each node, such as the distance to a given input surface, and continue adding nodes and refining the mesh until a performance bound is met. The nodes are often added in a worst-location-first order. The number of nodes can then be controlled by changing such performance bounds, or stopping the node addition (element refinement) once a given vertex budget for the desired application is reached.

In contrast, our method is initialized using the number of nodes allowed by the vertex budget and, subsequently, the best locations for such nodes are determined during the optimization process. Nevertheless, similarly to the other techniques above, more mesh nodes can also be added at any point in our method, such as at the centroids of elements with the highest error contribution.

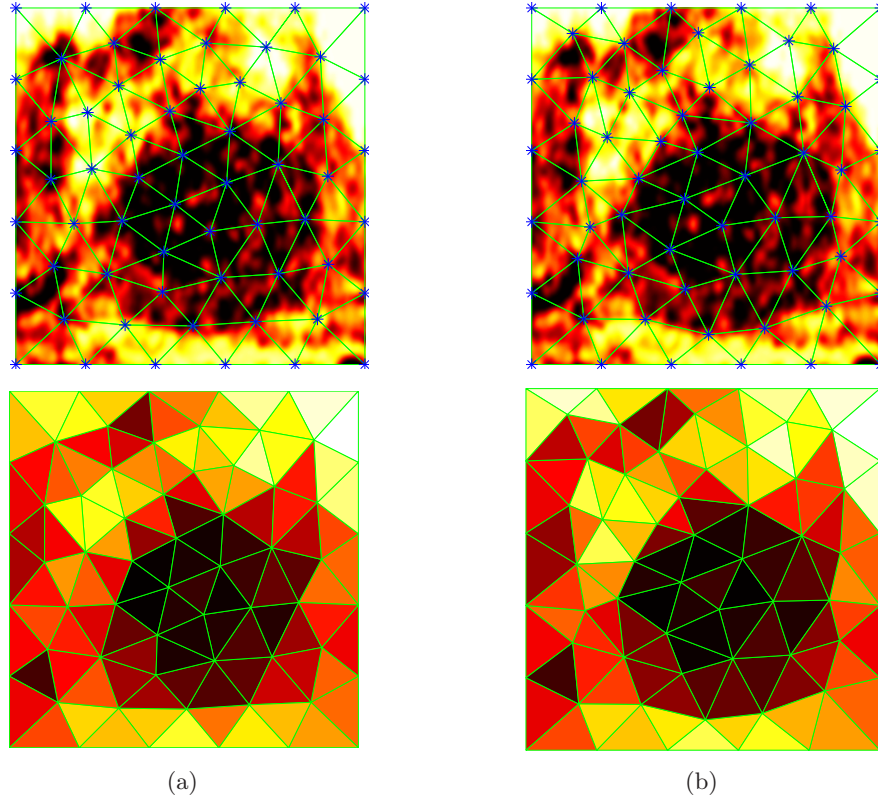


Figure 2.12: Initial (a) and optimized (b) meshes of sagittal prostate vibro-elastography images. The prostate is the darker oval structure in the center.

A mesh is defined not only by its node positions but also by their connectivity and there exist mesh configurations with the same number of nodes such that one may not be reached from the other only by changing node positions. Therefore, while minimizing the given energy definition, different tessellations of given nodes should be considered as well. A full-blown method that searches all possible tessellations of current node positions in order to minimize the combined error E is infeasible in practice. Alternatively, local mesh updates, such as *edge swaps/flips*, can be utilized to find a better connectivity candidate minimizing E . Such heuristic local mesh modifications often require significant implementation effort [24].

In this thesis, at every iteration, the Delaunay tessellation is considered as the single update alternative to the current connectivity. This is an educated guess to update connectivity, since the Delaunay tessellation is known to already minimize the geometric component of the error. Each calculation of the error has a cost in terms of computational time due to numerical integration over image voxels. Therefore, limiting this search for connectivity updates allows our method to perform in a reasonable time. Furthermore, the use of Delaunay tessellation was observed to yield successful final meshes in our examples. Note that, for a particular application, it is also possible to fix the connectivity entirely and only optimize the node positions for a given image.

For the case where the objective function is purely geometric, i.e. $\lambda = 0$, E_G has a simple

algebraic (quadratic) definition as seen in (2.3). This can also be observed for relatively small values of λ as in Fig. 2.4(b). Consequently, the node position \mathbf{x}_i that minimizes E_G alone in the 1-ring neighbourhood Ω_i is the critical point of function E_G . This minimizer point \mathbf{x}'_i , found easily using the derivative of E_G with respect to \mathbf{x}_i , is then used to update the corresponding node location. Alliez *et al.* find the geometrical expression of this critical point to be as follows [24]:

$$\mathbf{x}'_i = \frac{\sum_{\tau_j \in \Omega_i} |\tau_j| c_j}{|\Omega_i|} \quad (2.19)$$

where c_j is the circumcenter of element τ_j and $|\cdot|$ denotes the volume. This formulation suggests that the best node location for an update considering purely E_G is the average of neighbouring circumcenters weighted by corresponding element volumes. This is used during the mesh initialization stage of our method, where only the geometric cost is minimized. Such initial meshes can indeed be precomputed without requiring any image data once the domain size and mesh resolution are known. Alternatively, structured meshes can also be used to initialize the method. Furthermore, it is also possible to start with such a minimizer mesh of one cost component and gradually change the optimization goal by increasing λ until a desired image compliance is reached.

The component E_G can be rapidly computed algebraically using (2.3), where the second term (volume integral) has a closed-form definition for the rectangular/quadrilateral shaped image domains in this chapter. However, the combined E is a function of the image as well and thus is not algebraically defined in a simple form preventing us from using a closed-form update location as for E_G alone in (2.19). Furthermore, contrary to common methods in image processing, it is also difficult to infer the change of E from image-derived values such as the image gradient. This is because such values at a node location do not necessarily characterize the change in E , which is actually caused by the way that the 1-ring element faces sweep through the image voxels, changing voxel variances within those 1-ring neighbours. Due to the lack of a closed-form derivative for E or a direct relation to an image-derived value, we resort to numerical derivation in implementing an optimization scheme minimizing this objective function. During this process, the voxels enclosed by each element are determined using the barycentric coordinates of voxels within the bounding box of this element. This mapping of voxels to their enclosing elements is the computational bottle-neck of our method. This operation can be accelerated using grid-point location approaches such as [36, 37].

Considering the two node update strategies proposed, the former one based on the constrained optimization takes considerable computation time since several recomputations of E locally within Ω_i are required for each node update. Our observations have indicated that finding such *exact* optimum locations locally is not essential within the overall scheme of the optimization. This is because this optimum location also depends drastically on the positions of the immediate neighbouring nodes defining the 1-ring region. If these neighbouring nodes are also updated at the same iteration, this often results in all the nodes being sub-optimal in their updated locations. Alternatively, only the nodes with disjoint 1-rings were optimized at each iteration. Such nodes were picked randomly or chosen from the ones with the highest cost contributions. In most cases, such a treatment still requires a substantial number of iterations to converge. Note that the gradient descent approach presented before requires a constant number of computations per node update. In our examples, that approach yielded superior

meshes in a given time and therefore is used in the results presented. Nevertheless, using the given objective function formulation, it is possible to implement other numerical techniques and optimization strategies such as the *conjugate gradients method*.

In this thesis, the initial nodes in the corners and along the edges of the given rectangular prism shaped domain are kept fixed during the optimization process and the nodes on faces are allowed to move only tangentially. The position of the internal nodes are optimized as described. These constraints can be relaxed and non-prism domains can be accommodated using (outer) domain-boundary complying techniques in [24].

2.6.4 Mesh Sizing

In order to achieve a desired local mesh-size, Alliez *et al.* proposes to modify (2.19) to re-define the element volume as a weighted volume for a background density distribution [24]. Using this simple treatment, they produce smaller elements to model higher curvature surfaces. A similar approach can be used to effectively refine FEM meshes at locations of smaller anatomical features or higher strain regions. For instance, consider the synthetic image in Fig. 2.13(a) representing two circular *anatomical features* of different sizes. A standard optimization presented in this chapter starts from an initial uniform mesh distribution such as in Fig. 2.13(b), which subsequently converges to Fig. 2.13(c). Note that the small feature in the center is not represented well using this uniform mesh. A mesh sizing field (*density function*) shown in Fig. 2.13(d) can be imposed increasing the element resolution in the center as demonstrated in Fig. 2.13(e). An optimization initialized with this specially-crafted nonuniform mesh is observed to successfully model the small feature in the center. Note that although the overall number of nodes/elements are the same in these uniform and nonuniform meshes, the nodes/elements are concentrated where they are actually needed in the latter one.

In order to develop such a varying mesh resolution approach for medical images, the desired sizing function has to be determined such as by extracting the features of interest. Alternatively, such sizing can be integrated into the optimization process using an adaptive method such that the mesh density around elements with high intensity variation is increased.

2.6.5 Resolution

Considering mesh element size, one should note that it cannot be refined arbitrarily for a given image resolution. This is because, as the element sizes decrease, the number of voxels enclosed by each element also decreases. This affects the robustness of the optimization process that relies on the numerical integrations. This can also be seen considering the limit case of having less than one pixel per element. In this chapter, the medical images are resampled (upsampled) as needed in order to provide a minimum required number of voxels on average per element, which was observed to be around 50-200 voxels/tetrahedron (depending on the image modality) in our implementation. However, also note that a high number of voxels slows down the cost computation. In tailoring our method for a given application, one should keep in mind that the computation time will depend on this balance between desired FEM mesh size and desired image representation accuracy. As an alternative to a full-blown numerical integration in cost computation, other schemes such as a variance approximation from multiple element quadrature points can also be employed.

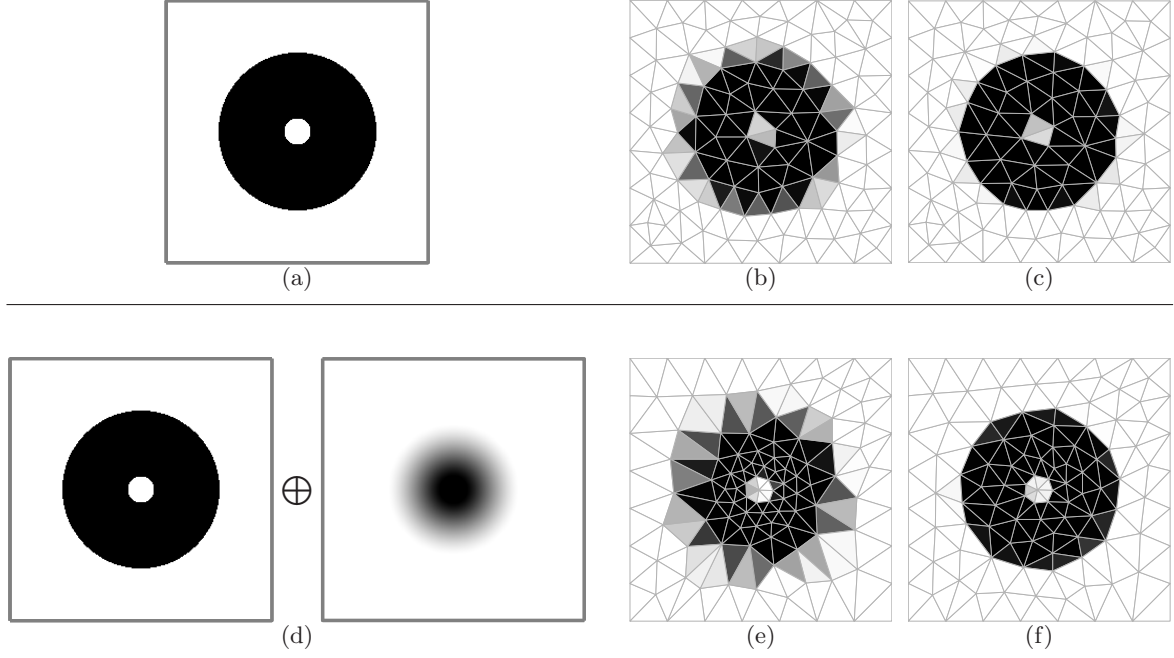


Figure 2.13: Demonstration of using a sizing field for variable element sizes throughout the domain: (a) synthetic image, (b) initial and (c) optimized meshes with uniform element sizing; (d) the image and the element sizing field to be imposed; (e) initial and (f) optimized meshes with the application of this sizing field.

2.6.6 Slivers

In our simulations, the geometric quality of elements, that are initially improved during the geometric initialization step, are kept of high quality while the elements are being aligned to conform to the image. However, in rare cases, a sliver may appear when it substantially minimizes E_D . Consider the case in Fig. 2.14(a) where a sliver is shown with its neighbouring elements. If there is an anatomical surface along the plane of this sliver causing contrast between the upper and lower elements, then this connectivity and node positions can indeed be a local minimizer since other options such as an edge connecting P_1 and P_2 would result in elements with large intensity variance.

If control over the worst element geometry is desired in a particular application, these slivers can be avoided by using a lower value of λ so to emphasize geometrical quality. Alternatively, a node may be added at the centroid of such elements and the following local connectivity update may be performed: All neighbouring elements seen in Fig. 2.14(b) are divided in two using this new node to be a corner. This is also applicable when there are other elements between the two upper/lower face neighbours. This procedure generates faces cutting the equatorial plane of the sliver that are likely desirable to conform to the anatomical surface along that plane. Indeed, E is guaranteed to decrease by this update since (i) addition of a node reduces E_G due to the introduction of a new quadrature point to the linear function approximant and (ii) E_D is also reduced since the face neighbours are all divided in two reducing their variances.

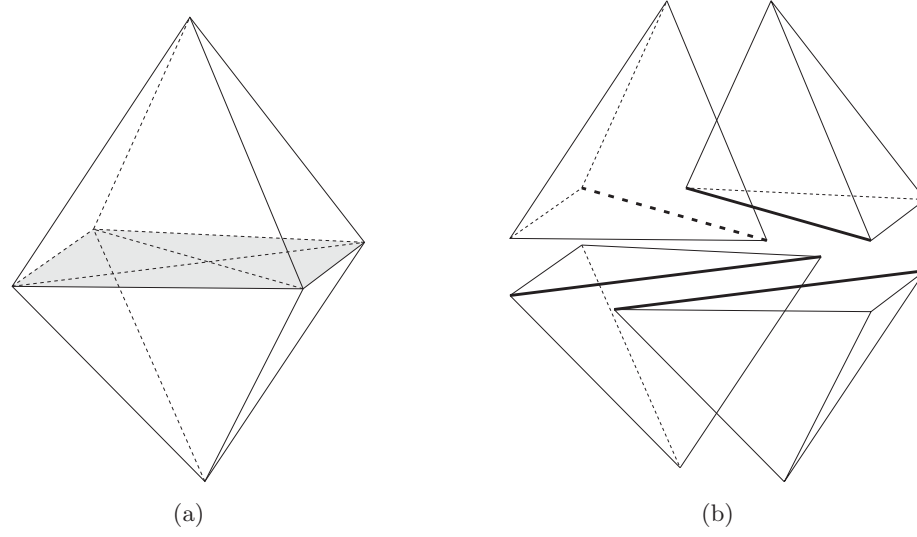


Figure 2.14: A sliver and its four face neighbours (a). The four neighbouring tetrahedra are shown individually (b).

When a strict vertex budget is not a concern, this procedure can be used to remove the few lowest-quality tetrahedra. The application of this sliver removal on the final meshes obtained by the standard techniques presented in Section 2.4 resulted in approximately an additional 1% reduction in the final normalized energy figures reported in Fig. 2.11. It is also possible to use other common sliver *exudation* techniques [26].

2.6.7 Mesh Quality Assessment from Aspect Ratios

In Fig. 2.15(b), the evolution of the aspect ratio error ($1 - \rho$) is seen as the methods of this chapter are applied to the 3D brain MR example in Fig. 2.9. Figure 2.15(a) shows the average, quadratic, and maximum mean values of the element aspect ratio errors that are computed at each major processing step. The purpose of this figure is to show that the aspect ratios of elements, which are initially improved during the geometrical optimization phase, are still kept of high-quality as the overall error is minimized next. The slight increase in average and quadratic means in the combined error optimization phase are due to the elements fitting the given image. Note that this better image fitting, both observed in the example images presented throughout this chapter and also presented quantitatively by the minimized error in Fig. 2.11, does not result a significant degradation of the overall aspect ratios of elements.

The mesh quality can be better seen in the aspect ratio histograms shown in Fig. 2.15(a). These histograms represent three snapshots of the quality distribution over the mesh elements, the evolution of which was presented in Fig. 2.15(a) in a compact form using mean values. These histogram snapshots present mesh quality at critical processing instants: (i) geometrically optimal mesh (*initial mesh*), (ii) optimized mesh after 32 iterations using $\lambda = 25\%$, and (iii) following a final sliver elimination procedure. The time instants (i) and (ii) are represented by the vertical lines in Fig. 2.15(a).

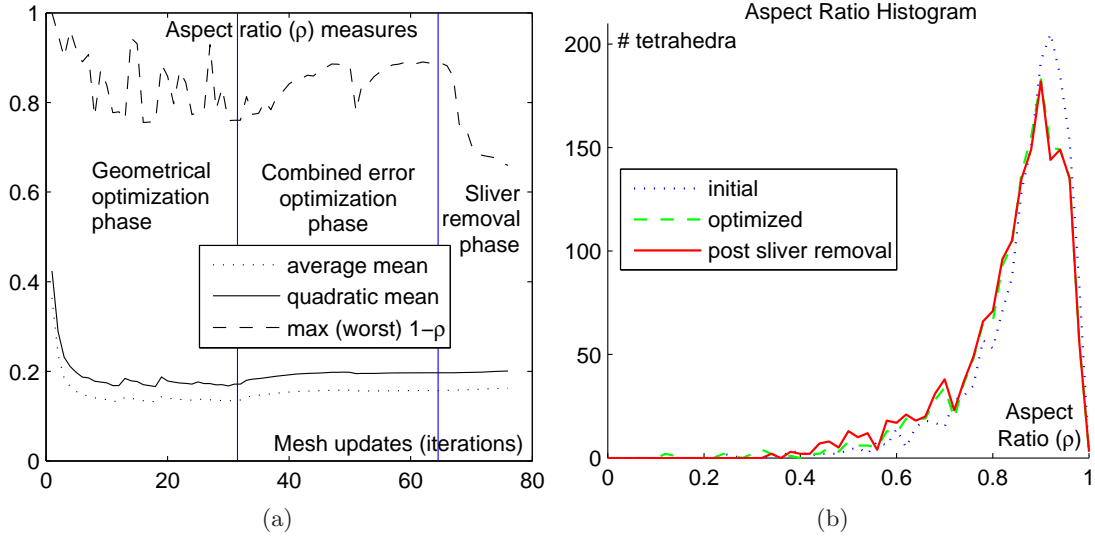


Figure 2.15: Mesh quality assessment from aspect ratios: (a) Average, quadratic, and maximum mean of aspect ratio error distributions, $(1 - \rho)$, are plotted for each iteration. These present a minimal decrease in the aspect ratio as the mesh is optimized to fit the image. In order to illustrate the effect of our optimization, the geometrical and combined error optimization iterations, as well as each sliver removal, are displayed as an *iteration*, although each such step is a significantly different operation in nature. (b) Histograms of element aspect ratios ρ of initial and optimized meshes. These demonstrate minimal mesh quality decline after image fitting. The histogram of elements following a sample case of posterior sliver removal is also presented here, where the elimination of few peaks (low-quality elements) around $\rho = 0.2$ can be observed.

In this example, any sliver elimination was postponed until the end of standard optimization. In this final stage, 10 slivers were removed in this case. In Fig. 2.15(a), for presentation purposes, each sliver removal mesh update is also shown as if it was an optimization iteration. The choice of mean value representation was to better present its change at various stages of our method. Both in the histogram figure and the proposed mean value evolution plot, it is observed that the overall aspect ratio improvement, gained during the geometrical initialization phase, is kept to a great extent during our main optimization phase. The slight quality decline at this stage was expected due to elements fitting the image. In the final sliver removal phase, some lowest quality tetrahedra were eliminated successfully. In practice, sliver elimination can also be done as part of each optimization iteration.

2.6.8 Relationship to deformation and FEM

There exist methods as part of FEM *post-processing* that can refine or modify a mesh based on a computed simulation output such as element strains during deformation. This requires running the simulation first, which in turn necessitates *a priori* knowledge of the boundary conditions. These may not be known prior to meshing, or their location and nature may change substantially from simulation to simulation, e.g. as medical tools interact with the anatomy.

Moreover, although post-process refinement techniques adjust node/element density locally, they do not formulate an intrinsically optimal placement for an element. Unlike in mechanical/civil engineering, larger deformations are involved in medical simulations and simulation accuracy around important anatomical features are often preferred over accuracy at high-strain regions (which are mostly the medical tool contact points). Our proposed method optimizes meshes assuming that there is no prior knowledge of the boundary constraints.

Advances in ultrasound and MR-based elastography offer significant potential for our method, since elastography-recovered tissue values can be assigned to mesh elements for subsequent FEM use with minimal loss of information during discretization. In this chapter, a vibro-elastography image meshing example is provided in Fig. 2.12. There is substantial work in the literature on the identification of mechanical tissue features such as the tissue elastic modulus from such elastography data [33], which is beyond the scope of this thesis.

2.7 Conclusions

In this chapter, a penalty function based on FEM interpolation error is combined with a proposed image-representation cost and the combined objective function is minimized to produce high-quality FEM elements that also discretize a given image in a desirable manner. With the emerging fields of elastography imaging and tissue parameter identification, this method becomes essential for optimal meshes conforming to such parameters. Note that such an optimized discretization can further be used for a fast approximate segmentation since optimized elements represent an image using far fewer degrees-of-freedom than the underlying pixels. The method was presented both in 2D and 3D using synthetic data, MR images of brain, CT images of the kidney, and elastography imaging of the prostate.

References

- [1] M. Sonka and J. M. Fitzpatrick, Eds., *Handbook of medical imaging*. SPIE Press, 2000.
- [2] S. Cotin, H. Delingette, and N. Ayache, “Real-time elastic deformations of soft tissues for surgery simulation,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 1, pp. 62–73, 1999.
- [3] M. Ferrant, A. Nabavi, B. Macq, F. A. Jolesz, R. Kikinis, and S. K. Warfield, “Registration of 3-d intraoperative MR images of the brain using a finite-element biomechanical model,” *IEEE Transactions on Medical Imaging*, vol. 20, no. 12, p. 1384–1397, 2001.
- [4] H. M. Yin, L. Z. Sun, G. Wang, T. Yamada, J. Wang, and M. W. Vannier, “ImageParser: a tool for finite element generation from three-dimensional medical images,” *BioMedical Engineering OnLine*, vol. 3, p. 31, 2004.
- [5] O. Goksel, S. E. Salcudean, and S. P. DiMaio, “3D simulation of needle-tissue interaction with application to prostate brachytherapy,” *Computer Aided Surgery*, vol. 11, no. 2006, pp. 279–288, 2006.
- [6] S. J. Owen, “A survey of unstructured mesh generation technology,” in *International Meshing Roundtable*, vol. 3, 1998.
- [7] R. Löhner, “Automatic unstructured grid generators,” *Finite Elements in Analysis and Design*, vol. 25, no. 1-2, pp. 111–134, 1997.
- [8] N. Archip, R. Rohling, V. Dessenne, P. Erard, and L. P. Nolte, “Anatomical structure modeling from medical images,” *Computer Methods and Programs in Biomedicine*, vol. 82, no. 3, pp. 203–215, 2006.
- [9] W. J. Schroeder, B. Geveci, and M. Malaterre, “Compatible triangulations of spatial decompositions,” in *IEEE Visualization Conference*, Austin, Texas, USA, Oct 2004, pp. 211–217.
- [10] M. Müller and M. Teschner, “Volumetric meshes for real-time medical simulations,” in *Workshop on Bildverarbeitung für die Medizin*, Erlangen, Germany, 2003, pp. 279–283.
- [11] J. Teran, N. Molino, R. Fedkiw, and R. Bridson, “Adaptive physics based tetrahedral mesh generation using level sets,” *Engineering with Computers*, vol. 21, no. 1, pp. 2–18, 2005.
- [12] Y. Zhang, C. Bajaj, and B. Sohn, “3D finite element meshing from imaging data,” *Computer Methods in Applied Mechanics and Engineering*, vol. 194, no. 48-49, pp. 5083–5106, 2005.

- [13] F. Labelle and J. R. Shewchuk, “Isosurface stuffing: fast tetrahedral meshes with good dihedral angles,” *ACM Transactions on Graphics (SIGGRAPH Proceedings)*, vol. 26, no. 3, pp. 57–66, 2007.
- [14] A. Fedorov, N. Chrisochoides, R. Kikinis, and S. K. Warfield, “An evaluation of three approaches to tetrahedral mesh generation for deformable registration of brain MR images,” in *IEEE International Symposium on Biomedical Imaging (ISBI): Nano to Macro*, 2006, pp. 658–661.
- [15] A. Mohamed and C. Davatzikos, “Finite element mesh generation and remeshing from segmented medical images,” in *IEEE International Symposium on Biomedical Imaging (ISBI): Nano to Macro*, 2004, pp. 420–423.
- [16] J.-P. Pons, F. Ségonne, J.-D. Boissonnat, L. Rineau, M. Yvinec, and R. Keriven, “High-quality consistent meshing of multi-label datasets,” in *Information Processing in Medical Imaging (IPMI)*, 2007, pp. 198–210.
- [17] J. Dardenne, S. Valette, N. Siauue, N. Burais, and R. Prost, “Variational tetrahedral mesh generation from discrete volume data,” *The Visual Computer: International Journal of Computer Graphics*, vol. 25, pp. 401–410, 2009.
- [18] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: a review,” *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.
- [19] T. F. Chan and L. A. Vese, “Active contours without edges,” *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.
- [20] T. Gevers, “Adaptive image segmentation by combining photometric invariant region and edge information,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 6, pp. 848–852, 2002.
- [21] A. C. Reid, S. A. Langer, R. C. Lua, V. R. Coffman, S. Haan, and R. E. García, “Image-based finite element mesh construction for material microstructures,” *Computational Materials Science*, vol. 43, no. 4, pp. 989–999, 2008.
- [22] J. R. Shewchuk, “What is a good linear element? interpolation, conditioning, and quality measures,” in *International Meshing Roundtable*, 2002.
- [23] D. A. Field, “Qualitative measures for initial meshes,” *Int J for Numerical Methods in Engineering*, vol. 47, no. 4, pp. 887–906, 2000.
- [24] P. Alliez, D. Cohen-Steiner, M. Yvinec, and M. Desbrun, “Variational tetrahedral meshing,” *ACM Transactions on Graphics (SIGGRAPH Proceedings)*, vol. 24, no. 3, pp. 617–625, 2005.
- [25] L. A. Freitag and C. Ollivier-gooch, “Tetrahedral mesh improvement using swapping and smoothing,” *Int J of Numerical Methods in Engineering*, vol. 40, pp. 3979–4002, 1997.
- [26] S. Cheng, T. K. Dey, H. Edelsbrunner, M. A. Facello, and S. Teng, “Sliver exudation,” in *Annual Symposium on Computational Geometry*, Miami Beach, Florida, United States, 1999, pp. 1–13.

- [27] B. Cutler, J. Dorsey, and L. McMillan, “Simplification and improvement of tetrahedral models for simulation,” in *Proc. of the Eurographics/ACM SIGGRAPH Symp. on Geometry Processing*, Nice, France, 2004, pp. 93–102.
- [28] Q. Du and D. Wang, “Tetrahedral mesh generation and optimization based on centroidal voronoi tessellations,” *Int J for Numerical Methods in Engineering*, vol. 56, pp. 1355–1373, 2003.
- [29] L. Chen and J. Xu, “Optimal Delaunay triangulation,” *Journal of Computational Mathematics*, vol. 22, no. 2, pp. 299–308, 2004.
- [30] L. Chen, “Mesh smoothing schemes based on optimal Delaunay triangulations,” in *International Meshing Roundtable*, 2004, pp. 109–120.
- [31] D. Mumford and J. Shah, “Optimal approximations by piecewise smooth functions and associated variational problems,” *Communications on Pure and Applied Mathematics*, vol. 42, pp. 577–685, 1989.
- [32] O. C. Zienkiewicz and R. L. Taylor, *The Finite Element Method*. Butterworth-Heinemann, 2000.
- [33] J. F. Greenleaf, M. Fatemi, and M. Insana, “Selected Methods for Imaging Elastic Properties of Biological Tissues,” *Annual Review of Biomed. Eng.*, vol. 5, pp. 57–78, 2000.
- [34] S. Salcudean, D. French, S. Bachmann, R. Zahiri-Azar, X. Wen, and W. Morris, “Viscoelasticity modeling of the prostate region using vibro-elastography,” in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2006, pp. 389–396.
- [35] J. Ophir, I. Céspedes, H. Ponnekanti, Y. Yazdi, and X. Li, “Elastography: a method for imaging the elasticity of biological tissues,” *Ultrasonic Imaging*, vol. 13, no. 2, pp. 111–134, 1991.
- [36] A. Fousse, E. Andres, J. Françon, Y. Bertrand, and D. Rodrigues, “Fast point locations with discrete geometry,” in *SPIE Vision Geometry Conference*, Denver, Colorado, USA, Jul 1999, pp. 33–44.
- [37] O. Goksel and S. E. Salcudean, “B-mode ultrasound image simulation in deformable 3-D medium,” *IEEE Transactions on Medical Imaging*, vol. 28, no. 11, pp. 1657–1669, 2009.

Chapter 3

B-Mode Ultrasound Image Simulation in Deformable 3D Medium ⁴

In a prostate brachytherapy procedure, the performing physician uses TRUS imaging both for registering the intra-operative prostate volume to the pre-operative planning images and for following the needle in tissue to ensure accurate placement of seeds at planned locations. Therefore, the simulation of this imaging modality is essential in developing a training simulator. Considering the tissue deforms in brachytherapy during needle insertions, such a simulation must be capable of incorporating deformation and produce images in real-time.

This chapter describes a general methodology of simulating B-mode images. A prostate brachytherapy implementation of this simulation technique is later presented in Chapter 5 for simulating TRUS images of the prostate in the existence of deformations caused by needle interactions.

3.1 Introduction

Ultrasound is a non-invasive and safe medical imaging modality and hence one of the most commonly used examination tools. However, image anisotropy and the existence of various significant artifacts cause the need for extensive echographer training. Current standard education is in the form of supervised examination of real pathologies during clinical practice. Despite its many advantages, this approach involves significant time expenditure of qualified personnel and can only be performed when a supervisor and a patient are available. Furthermore, training on rare pathologies poses a problem. Indeed, students have the chance to learn only 80 % of the important pathologies during one-year of standard education [1]. This need for ultrasound examination training has motivated several computer-based simulation environments [2]. In addition to examination, training of medical procedures that utilize ultrasound imaging, e.g. prostate brachytherapy and breast biopsy, can also significantly benefit from such simulation techniques. The ability to mentally register two-dimensional (2D) image slices within the three-dimensional (3D) anatomy is a non-trivial skill required by any sonographer. Real-time ultrasound simulators have the potential to accelerate and improve such training.

In a typical ultrasound simulation scenario, the user must be presented with an image slicing the target anatomy. In an actual diagnostic or operative procedure, such target anatomy is often deformed under various forces, such as ultrasound probe contact, as illustrated in

⁴A version of this chapter has been published. O.Goksel and S.E.Salcudean, “B-Mode Ultrasound Image Simulation in Deformable 3-D Medium,” *IEEE Transactions on Medical Imaging*, 28(11):1657-1669, Nov 2009. DOI:10.1109/TMI.2009.2016561

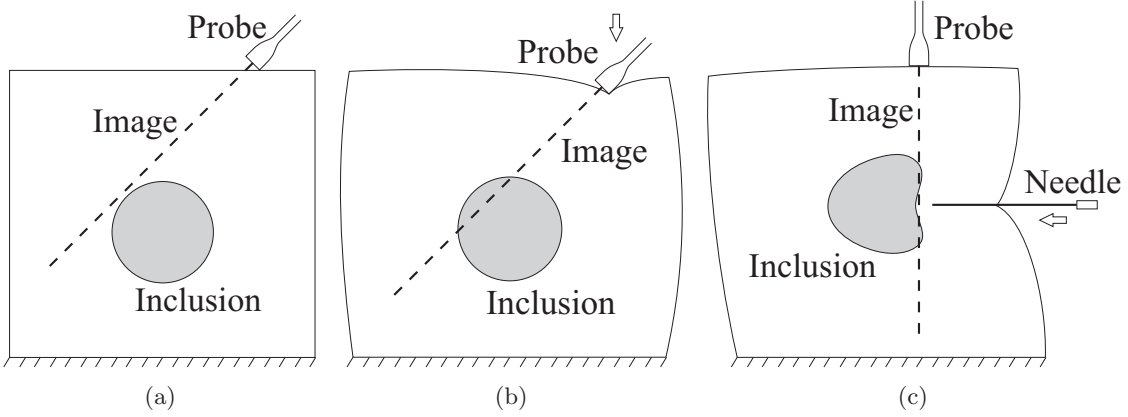


Figure 3.1: Image slices (a) before and (b) after a deformation caused by probe pressure; and (c) illustration of deformation during needle insertion.

Figures 3.1(a) and 3.1(b). Note that an ultrasound probe only compresses the surface of the tissue, whereas there exist other medical tools that further manipulate the tissue internally, e.g. percutaneous needles as in Fig. 3.1(c). A realistic image simulation should take such tissue deformations into account in order to deliver an immediate representation of the anatomy in its current deformed configuration.

Modeling of tissue deformation has been studied extensively in the literature [3–6]. Common techniques such as mass-spring models and the Finite Element Method (FEM) use a discretization (mesh) of the tissue volume and corresponding elasticity parameters to approximate its behaviour under load. In general, these model parameters are abstracted *a priori* and used with given forces in real-time to compute deformation, which is commonly expressed as a set of displacements of the given mesh nodes. To enable a real-time computation of deformation, this discretization often has a significantly coarser structure than the typical resolution of medical imaging modalities. This chapter presents an image generation technique in deformed 3D meshes and addresses the computational challenges for real-time performance. Realistic simulation of ultrasound, which is a real-time imaging modality, is the primary target application of our image generation technique. While the image slicing methodology we propose is described for B-mode ultrasound, it also applies to other modalities such as MR and CT.

The chapter is organized as follows. First, our choice of interpolation procedure, which consists of finding the image pixel intensities by referring their positions back to the nominal pre-deformed configuration, is introduced. Then, its application within meshes that are deformed based on the FEM is outlined in 2D. Next, the pixel location problem that arises in such a scheme and our proposed numerical treatment for this are presented in 3D. In the results section, the proposed technique is demonstrated for real-time ultrasound synthesis in a tissue-mimicking ultrasound phantom and in-vivo thigh data deformed by the ultrasound probe itself. A discussion of the limitations and possible future extensions conclude this chapter.

3.2 Previous Work

An ultrasound simulator necessitates rapid and realistic image rendering of deformed tissue in response to probe or tool manipulation by a trainee. There exist two major approaches for simulating B-mode ultrasound images, *the generative approach* and *the interpolative approach*. The former simulates the ultrasonic wave propagation by using accurate models of the probe, the tissue scatterers, and the wave interaction [7, 8]. Generating a single B-mode frame using this technique takes hours. Thus, this approach is not suitable for real-time applications. Furthermore, in practice it is not possible to extract an exact scatterer model of a complex medium such as the tissue and hence the images generated with this technique typically look artificial. The latter approach generates images by interpolating from pre-acquired images of the volume. While interpolation directly from arbitrarily-oriented B-scans was demonstrated in [9], the construction of a regular-grid reference volume, called *3D ultrasound reconstruction* [10, 11], is commonly the preferred method because it enables data processing with off-the-shelf algorithms. UltraSim [12], which is one of the first commercial ultrasound image simulators, and several others [13–18] follow this latter approach. Refer to [2] for a review on ultrasound training simulators.

Note that anisotropic image artifacts, such as shadowing and reverberation in ultrasound, may not be reproduced correctly by an interpolation scheme due to their direction-dependent characteristics. One attempt to remedy this shortcoming is to acquire real ultrasound images of the volume at several positions/orientations. Subsequently, for a given probe location during simulation, the image that corresponds best with that orientation can be selected from that database and shown to the user. This is not feasible in practice due to the unlimited number of possible probe and/or medical tool configurations during a simulation [1].

Since a generative simulation approach with a full-blown wave interaction model is not feasible for real-time applications, some recent work has focused on developing heuristic models that can be computed in real-time. Some researchers looked at the problem in the context of computer graphics, such as first texture mapping different tissue regions by pre-computed backgrounds, then imposing a Gaussian noise to generate an artificial speckle pattern, and finally applying a depth-dependent radial blurring to simulate a convex probe [19, 20]. Others proposed processing imaginary rays mimicking ultrasound using heuristic interaction functions defined for coarse (pixel level) tissue representations with abstracted parameters, namely attenuation, reflection, and scatterer power [1]. Unfortunately, the adjustment of such parameters was not addressed in this work. Deriving parameters from CT data was also proposed in [21] and in [22], separately, in order to generate ultrasound images by processing CT images. Although such pseudo-generative methods for echography simulation are appealing, due to the substantially complex nature of actual wave interactions, it is extremely difficult to generate even common ultrasound phenomena, such as speckle formation, using these methods, let alone realistic images. As a result, existing simulators that are studied for clinical training scenarios [2, 12, 13, 15–18] are interpolation-based.

In many medical procedures, such as prostate brachytherapy [5], brain surgery, or breast biopsy, significant deformation is caused by medical tools or by the ultrasound probe. In certain applications, such as in the diagnosis of deep-vein thrombosis (DVT), deformation observed in ultrasound images during deliberate probe indentation contains essential diagnosis information. Fast synthesis of ultrasound images in soft tissues under deformation will facilitate the

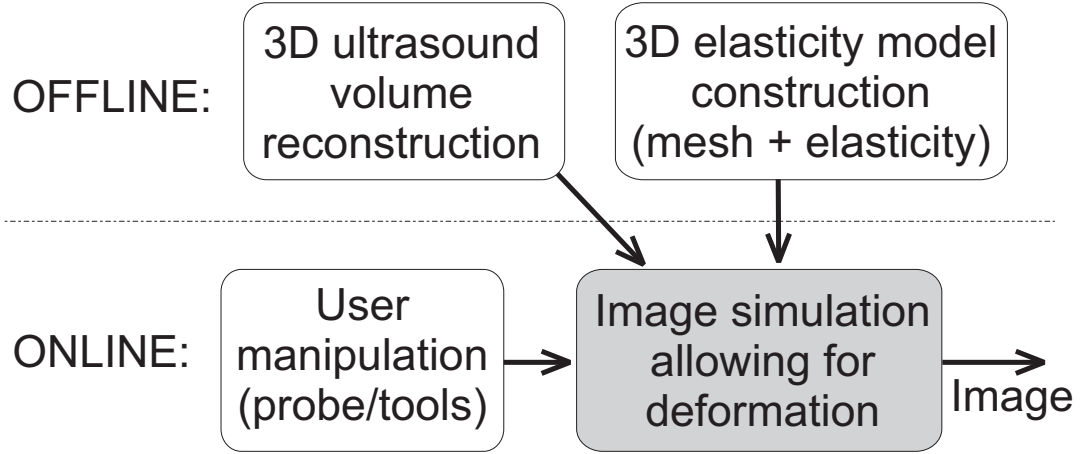


Figure 3.2: Online and offline steps of the proposed interpolation-based simulation.

development of training simulators. With this goal, a DVT diagnosis simulator was proposed in [18]. It simulates the probe pressure by first slicing an image from the 3D ultrasound data set and then applying a 2D elastic deformation to this image using quadtree-splines. This 2D in-plane deformation is pre-computed offline by registering the segmentations of pre-deformed and post-deformed anatomy of a test case [23].

Real-time ultrasound image slicing using physically-valid 3D deformation models has not been addressed in the literature. Our work is motivated by this need. A recent work applies similar techniques to generate ray-traced volume rendering of a deformable liver model [24].

For a deformed-volume image slicing strategy, as illustrated in Fig. 3.2, a reference volume dataset is required. The reference image volume can either be obtained using a 3D ultrasound probe or, alternatively, it can be constructed from individual 2D B-mode slices. This 3D ultrasound reconstruction has been studied extensively in the literature [10, 11, 25, 26]. Given this reference volume and a mesh-based deformation model, the image synthesis component (Fig. 3.2) of an ultrasound simulator is the subject of this chapter, preliminary results of which were presented earlier in [27, 28].

3.3 Methods

Let the spatial voxel locations of an $i \times j \times k$ 3D regular grid be V^0 , where the superscript zero refers to this being the initial (time-zero) configuration of these voxels. Assume that V^0 are the locations of a given reconstructed volume, in other words, the locations at which the intensities (also known as the *gray-values*) $I^0(V^0)$ are known *a priori* (see Fig. 3.3(a)). Note that the illustrations in Fig. 3.3 are given in 2D for the ease of presentation, although they represent 3D concepts.

Let V^0 be transformed to V by the deformation $f(\cdot)$ at a given simulation instance as follows:

$$V = f(V^0) \quad (3.1)$$

as shown in Fig. 3.3(b). Throughout this chapter, these two states V^0 and V are referred as the pre-deformed and the deformed tissue configurations, respectively.

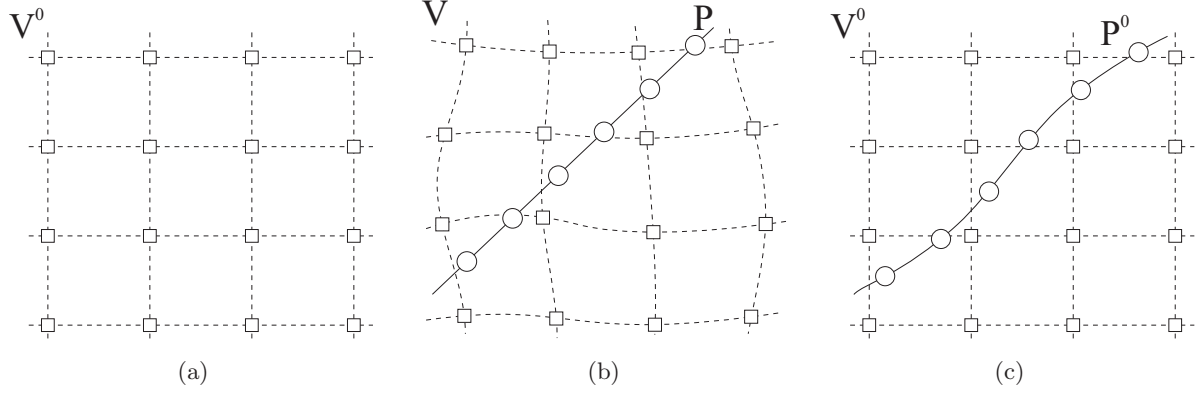


Figure 3.3: Voxel data and image plane in (a) nominal, (b) post-deformation, and (c) undeformed configurations in 2D (circles denote the image pixels and squares denote the volume data voxels).

Consider an image, formed by a set of n planar equidistant pixels P , cutting this deformed volume V . Such an image is shown with circles in Fig. 3.3(b). Synthesizing this image involves finding the immediate intensity values $I(P)$ at these n pixel locations for every image frame to be displayed on the screen. Note that this operation has a lower bound of $\Omega(n)$. Indeed, any algorithm processing an entire image (even just simply displaying it on screen) needs to access all n pixels proving this lower bound.

3.3.1 Inverting Deformation

One approach to the image synthesis above is to first compute the deformed voxel locations V and then to find (interpolate) the pixel intensities $I(P)$ within the known values of $I(V) = I(f(V^0))$. Note that similar deformation computations are employed by common elastic registration techniques [29, 30].

As seen in Fig. 3.3(b), the major disadvantage of the approach above is that the deformed voxels V no longer lie on a regular-grid structure. Consequently, computationally-expensive scattered-data interpolation techniques are needed. Another disadvantage is the need to transform the entire voxel volume from V^0 to $f(V^0)$ for each image frame. This is not practical. Indeed, the interpolation step does not demand the entire volume, since only the voxels *near* the image pixels have an effect on their intensity values $I(P)$. Therefore, it is theoretically possible to compute only the deformation of such nearby voxels—a small subset of V —as required by the particular interpolation technique used. Hence, if this approach is to be used, an effective way of identifying this subset is needed. Determining computational bounds for such a method is difficult, since this subset is not fixed and it changes with both the deformation and the image location.

Due to the above disadvantages, the following approach of first mapping the image pixel locations back to the pre-deformed configuration and then interpolating the regularly-spaced $I^0(V^0)$ at these *undeformed* pixel locations is proposed in this thesis. For an invertible deformation f , the pixel locations P can be mapped to the reference volume as:

$$P^0 = f^{-1}(P) . \quad (3.2)$$

An illustration of such *undeformed* pixels with the reference volume voxels V^0 can be seen in Fig. 3.3(c). Subsequently, the pixel intensities $I^0(P^0)$ at these nominal pixel locations are interpolated from $I^0(V^0)$.

With this method, the required interpolation is on a regular-grid of known values, which enables the use of well-studied simple and fast interpolation techniques. Furthermore, as opposed to the former approach, the inverse deformation needs to be computed only for the image pixels, which yields a fixed number of computations for the synthesis of each frame regardless of the deformation and the image location.

3.3.2 Image Pixels in an FEM Tessellation

Many deformation models employ a mesh to simulate displacements during deformation. One of the most common methods for tissue deformation computation is the FEM, in which tessellations are typically much coarser than medical imaging resolutions due to computational constraints. Mesh-based models simulate deformation in the form of node displacements of the overlaid mesh, so the deformation f is only defined at the nodes. The displacement of other locations within the mesh can be approximated from these node displacements. For example, for tetrahedral meshes, barycentric coordinates can be used for this purpose. Consider the situation in 2D, where an image P slices an object deformed under constraints shown with the arrows in Fig. 3.4(a). For each image pixel P_i ($i \in 1..n$), once its barycentric coordinates with respect to the deformed element e_s enclosing this pixel are found, they point to the corresponding P_i^0 in the pre-deformed configuration (details in Section 3.3.4). Accordingly, for a computed mesh deformation and a given probe position/orientation, our image frame synthesis consists of the following basic steps:

- $\forall i \in \{1..n\}$
- I. Find the element e_s enclosing P_i
 - II. Find the location P_i^0 using its barycentric coordinates and the node displacements of e_s
 - III. Interpolate the given data $I^0(V^0)$ at P_i^0

Figure 3.5 presents a flowchart showing the data flow between these steps and the data interaction with the deformation model.

Steps II and III above are constant-time operations, for which there exist well-studied fast implementations. However, the *point location* problem of step I is computationally demanding and hence the bottleneck of this technique. Note that the enclosing element of an image pixel depends on both mesh deformation and the image position/orientation; hence, it needs to be computed at each time instant t and cannot be decided offline.

In computational geometry, the 2D point location problem has been studied extensively [31], resulting in common techniques such as *slab decomposition* [32], *Kirkpatrick's algorithm* [33], and *trapezoidal maps*. However, only a few of these methods extend to higher dimensions, i.e., point location in 3D spatial subdivisions. Furthermore, most techniques focus on either locating a single point or a set of points scattered over the given domain, whereas the points in our case—the image pixels—are regularly-spaced on a 2D planar surface that is embedded in 3D. Consequently, exploiting this property of our problem to build intermediate data structures for locating all image pixels cumulatively will accelerate the process substantially. Indeed, any

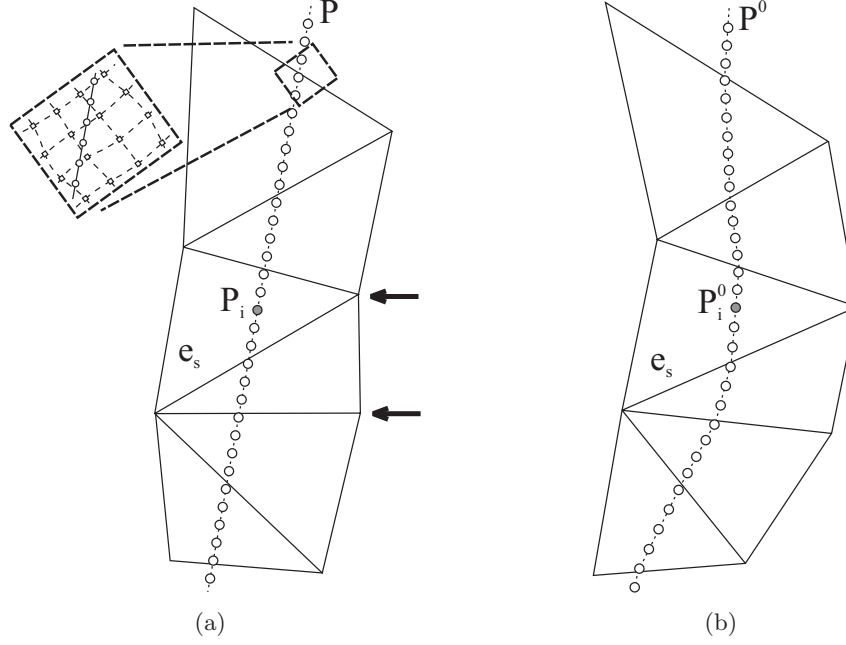


Figure 3.4: Mesh-based image *undeformation* illustrated in 2D: (a) image slice within a mesh that is under force/displacement constraints; and (b) corresponding image pixels mapped to the nominal mesh, where the reference volume is given on a regular-grid structure.

conventional method of locating each point individually would not allow real-time processing of typical medical image resolutions. For instance, the point location routine in the QuickHull package [34] locates the 90K pixels of a typical image presented in our Results section in over 30s. Therefore, the rest of this chapter focuses on exploiting this spatial relationship of image pixels in order to accelerate step I above.

3.3.3 Fast Equidistant-Point Location on Image Planes in 3D

Due to the 3D mesh elements being much larger than the image pixels, numerous neighbouring pixels are enclosed by a single element that is cut by the image. This fact can be exploited to predict the enclosing element of a pixel from its pixel neighbours. Although this yields a significant speed gain, it is still not fully taking advantage of the known grid structure of

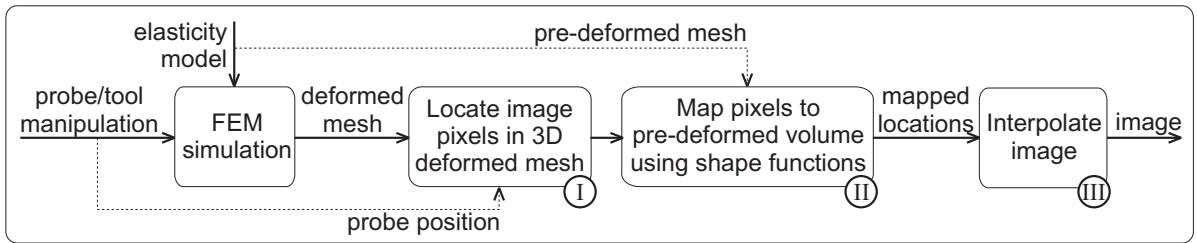


Figure 3.5: Basic steps of the proposed algorithm.

the pixels. Even though most predictions will succeed, each prediction has to be verified by an operation such as *point-in-tetrahedron check*, requiring many additional operations. For failed predictions, finding the enclosing element is again the same non-trivial point location problem above. Therefore, an temporary data structure is proposed for each frame such that, following the construction of this data structure, each and every pixel is located accurately and immediately, i.e., in constant time.

Consider the faces of the mesh elements intersecting the image plane. Note that the mesh-image intersection is the only information needed to locate all image pixel points. Indeed, when moving from a pixel to its neighbour, if no intersection is crossed, the latter pixel still lies in the same element, otherwise, it lies in a different element that can be deduced from the intersection information. Thus, using a scan-line approach, we determine to which mesh element each of the pixels belongs by traversing the image. The traversal occurs along a line parallel to an arbitrary axis, e.g. the axial direction of the ultrasound imaging plane.

The conventional slab decomposition technique for point location, also called the partitioning scan-line algorithm, locates individual (possibly scattered) points in a domain using edge comparisons along a scan-line. In contrast, in our case of an image domain, pixels are positioned at discrete locations, and therefore scan-lines only sweep discrete columns. In this thesis, this discrete pixel structure is utilized to efficiently store the mesh intersection information such that the image intersections with mesh faces are discretized at the pixel locations and are stored for use during line scans. Such a data structure that substantially accelerates the overall image generation procedure is built temporarily prior to every frame being synthesized.

Let us demonstrate a simple 2D case in Fig. 3.6(a). Assume that the pixels just below the mesh intersections, shown with stars, are identified and marked with the corresponding element numbers as depicted. For a downwards traversal (scan) of the image, it is inherent that any pixel encountered on and after a marked pixel is guaranteed to be in that marking element until another marked pixel is encountered. Implementing such a data structure to store the discretized mesh intersections (shaded pixels) requires at most $O(n)$ storage. Although in practice these marked pixels will be a small fraction of the image pixels, allocating an array the size of the image is computationally more efficient and is not demanding on today's computers. So, the discretized mesh intersection information can also be seen as an added property to each pixel, specifying whether it is being intersected by a face and, if so, by which element's face.

Finding the pixels to mark in a 2D case, e.g. the darker circles in Fig. 3.6(a), involves solving for line-line intersections of element edges with the image. Similarly, the intersections of 3D element faces with a planar image can be found in 3D using the deformed-mesh node positions and the image plane equation. However, further processing is required to discretize and mark them on image pixels.

Tetrahedral elements are chosen for presentation in this thesis due to their common use in tessellations. A tetrahedral element intersecting the image is shown in Fig. 3.6(b). Note that a tetrahedron may intersect a plane in one of two possible configurations yielding a triangular or a quadrilateral cross section as illustrated in Fig. 3.6(c). Using the deformed node positions and mesh connectivity, the element edge intersections shown with stars in Fig. 3.6(b) are found easily solving line-plane intersections. Subsequently, the line segments connecting these stars need to be discretized and marked on the corresponding pixels. Recalling our downwards scan-line direction, only the *upper* line segments (shown with darker lines for the instance in Fig. 3.6(b)) need to be marked. This is due to the fact that the top-half of any cross-

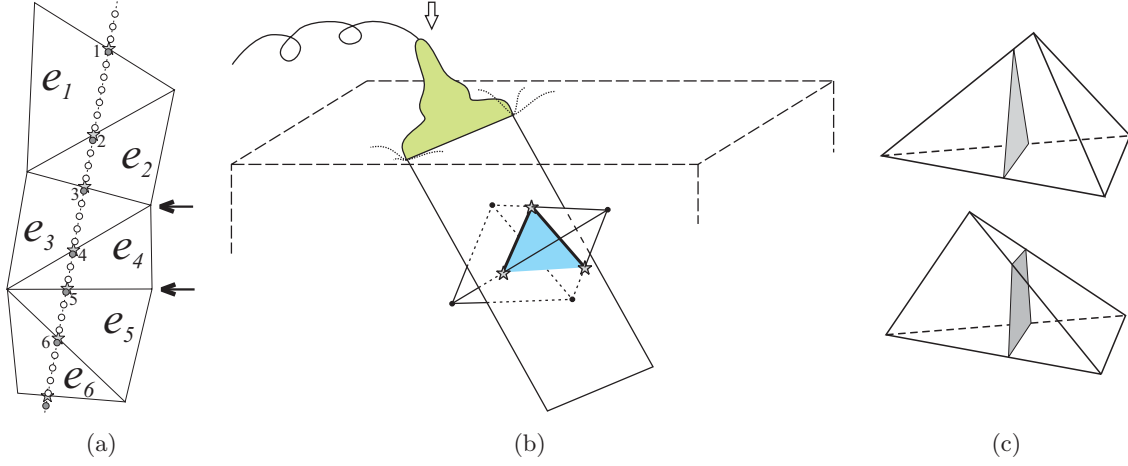


Figure 3.6: The intersection of a mesh and the image plane: (a) Marked image pixels close to projections of element borders (stars) on a 2D mesh; (b) a tetrahedral element intersecting the image plane; and (c) two possible configurations for a planar cross-section of a tetrahedron with either 3 or 4 edges intersected.

section has to be first crossed by the scan-line in order for it to reach the interior of the cross-section. Similarly, once leaving this cross-section, the top portion of the next element below will be crossed indicating that the up-coming pixels do not belong to the previous element anymore. Therefore only marking the edge of elements in the direction of an incoming scan-line is sufficient.

Let us demonstrate this pixel marking for a tetrahedral mesh sliced by an ultrasound image plane in Fig. 3.7. The 3D view in Fig. 3.7(a) shows the element cross-sections. The discretization of these face intersections on the image pixels, which is the above-mentioned intermediate data structure, is seen in Fig. 3.7(b). Part of this structure is illustrated enlarged in Fig. 3.7(c), where the actual face intersections are depicted with dashed-lines and their pixel discretizations with coloured circles. The element numbers marking these pixels are also labeled in the figure.

Bresenham's line drawing algorithm is used to discretize these segments on the grid of image pixels [35]. Note that it is possible for more than one intersection segment to be discretized on the same pixel. Although this is more likely to occur at the corners of the polygons, it can extend to more pixels of a line depending on the relative slopes of neighbouring line segments. For instance, consider the central pixel P shaded in Fig. 3.7(d), which is involved in the discretizations of both the elements e_6 and e_7 . If this pixel were to be marked as belonging to e_6 , then our line scan would mis-locate any pixel in the column below P . Such discretization conflicts, where the same pixel is involved in the discretization of more than one edge intersection, may occur at a substantial number of pixels and each may involve many edges. Therefore, a mechanism is required to ensure that such pixels are marked correctly. In this thesis, we resort to a method of *topologically sorting* all the tetrahedra cross-sections prior to marking them. The details are explained in Appendix B.

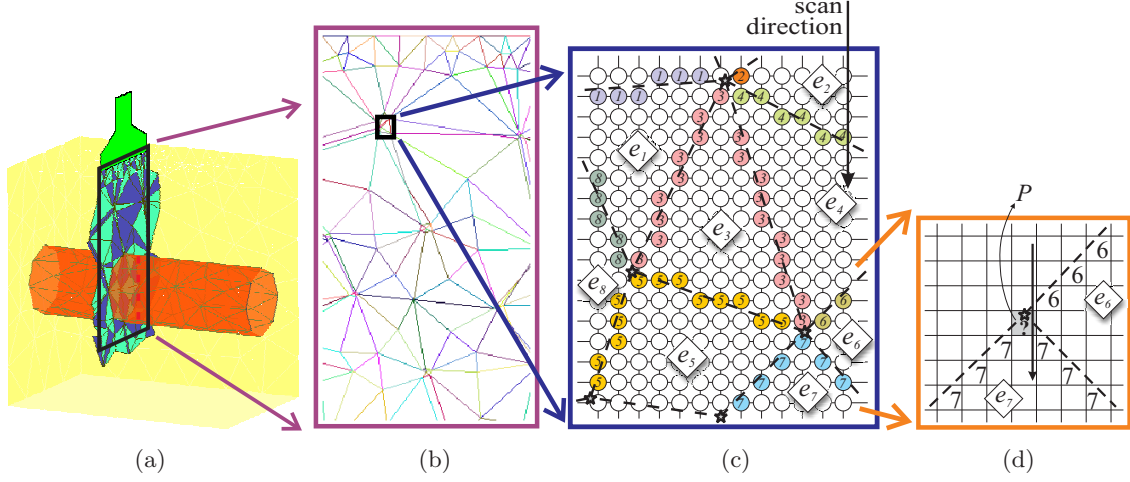


Figure 3.7: Marking mesh elements cross-sections: (a) A mesh and cross-sections of the elements sliced by the image simulated for an ultrasound probe; (b) element boundaries discretized and marked on the image prior to the interpolation; (c) a close-up to this marked image; and (d) a pixel of conflict, that is to be marked according to the scan direction.

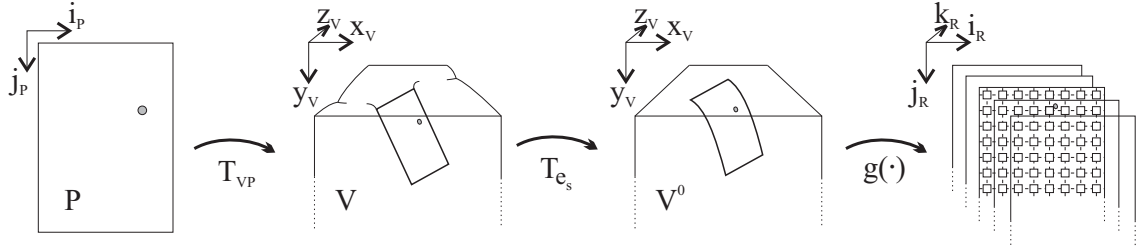


Figure 3.8: Transformations that map a pixel to the reference voxel volume, where an interpolation $g(\cdot)$ finds its intensity value.

3.3.4 Finding the Pixel Intensity Value

Each 3D pixel location is mapped to the reconstructed image voxel volume, where its intensity can subsequently be interpolated. The deformation part of this mapping is addressed using the barycentric coordinates with respect to the enclosing element. The overall mapping from the discrete image plane coordinates to the reconstructed volume coordinates can be expressed as a combination of linear transformations as shown in Fig. 3.8. In this figure, T_{VP} is the image-to-mesh transformation, which is determined by the probe position/orientation and hence is constant for all pixels of the same image frame. T_{e_s} is the deformed-to-nominal mesh transformation defined by the node displacements of the enclosing element e_s due to the deformation. This transformation is constant within an element for the mesh-node displacements of that given instant and it is calculated using barycentric coordinates [36] as described in Appendix C.

Note that T_{VP} changes with probe motion, whereas T_{e_s} with deformation. Consequently,

a cumulative transformation:

$$T_s = T_{e_s} T_{VP} \quad (3.3)$$

can be computed for any given element e_s during the image synthesis of a given probe position/orientation within a known/simulated deformation. Since any pixel within that element is subject to this same transformation, it is calculated only once per intersected element per image frame.

3.3.5 Summary of the Proposed Algorithm

For the synthesis of every frame, first the set of elements that are intersected by the imaging plane is compiled. This set L is composed by traversing the elements in the plane using a 3D mesh element neighbourhood list, which is pre-compiled offline as soon as a 3D mesh is available. Note that the neighbours of an element do not change with deformation. Therefore, given an intersected element, such a list enables us to deduce its neighbours that are also intersected by observing which face of the current element intersects the image plane.

The set L is then sorted topologically and the top-halves of the element projections are discretized and marked on the image in that sorted order using Bresenham's algorithm. Figure 3.7(b) demonstrates an instance of marked image pixels. The intensity of a pixel P_i is found by interpolating $T_s P_i$ in the reference voxel data. This interpolation is denoted by the operator $g(\cdot)$ in Fig. 3.8. For an example of using the nearest-neighbour interpolation (NNI), if we assume that $\text{round}(\cdot)$ function gives the nearest discrete reference-grid location to a point, then $I^0(\text{round}(T_s P_i))$ is the intensity of the pixel sought.

The transformations for all the intersected elements are computed while the set L is compiled. Subsequently, for every edge crossing of the scan-line, a *current transformation* pointer can be switched to point to the transformation matrix of the current enclosing element. Our proposed algorithm for the synthesis of one image frame can be summarized as follows:

1. Compile the set L of intersected elements and their partial-ordering relations.
2. Sort L topologically.
3. Mark the cross-sections of L on the image in the sorted order.
4. Compute the transformation T_s for each element $e_s \in L$
5. For each image pixel P_i
 - 5a. If P_i is marked with element e_s ,
then set the active transformation T to T_s
 - 5b. Find the intensity of P_i by interpolating I^0 at TP_i

3.3.6 Computational Analysis

In the algorithm above, the loop in step 5 processes every image pixel, while the previous steps are used to compile the intermediate data structures to accelerate this step. Consider an n -pixel image intersecting a total of m elements in a 3D mesh and recall that $n \gg m$ for typical medical images and FEM tessellations. Note that the synthesis of an image requires some form of processing of each of its pixels, thus the computation of any synthesis algorithm has a lower bound of $\Omega(n)$. Nevertheless, the individual computational cost for every pixel may render an algorithm infeasible as demonstrated in Section 3.3.2. Indeed, a method that is sub-optimal for our application may demand over 30 s just to locate pixels in mesh elements.

The computational analysis of the methods and the data structures introduced in this chapter are presented below.

Compilation of the intersected element set L starts from an arbitrary initial intersected element, such as one touching the probe. Finding this initial element, which can be done simply by traversing the top surface of the mesh, takes insignificant time. Once one intersected element is found, the traversal of the rest using a pre-compiled neighbourhood list in step **1** takes constant time per element. Computing each transformation in step **4** is also a constant time operation per element. Topological sort takes linear time with respect to the total number of cross-sections and partial relations. Nonetheless, in our case the relations are set by the shared edges between the m cross-sections and thus the number of such edges is bounded from above by a multiple of m . As a result, all the steps **1**, **2**, and **4** compute in $\mathcal{O}(m)$ time.

The computation of step **3**, where the pixels on cross-sections are marked, depends on various implementation choices, such as the specific line drawing algorithm. Nevertheless, it can be approximated by the number of pixels actually marked on the image. This can in turn be regarded as approximately \sqrt{m} rows of elements being marked by Bresenham's algorithm on their upper halves. Considering a row of elements has on the order of \sqrt{n} pixels to be marked, step **3** thus requires $\mathcal{O}(\sqrt{nm})$ time to compute.

Note that the significantly lower computational order of steps **1-to-4** justify the anticipated speed gain during step **5**. In particular, step **5a** of identifying the enclosing element reduces down to a single memory access and, in step **5b**, the transformation of pixels by T_s from discrete 2D image coordinates to the 3D reconstructed volume requires only 12 multiplications per pixel.

3.3.7 Deformation Model

The deformation model employed in our particular implementation is a linear-strain quasi-static finite-element model. For this, first a mesh representation of the region of interest is obtained. Using the Young's modulus and Poisson's ratios of the materials involved, a stiffness matrix \mathbf{K} relating the nodal displacements u of this mesh to the nodal forces f is compiled such that $f = \mathbf{K}u$ [36]. Fundamental boundary constraints (the fixed nodes of the mesh) are next applied on this \mathbf{K} by zeroing its corresponding rows/columns. \mathbf{K} is then inverted and saved for use in the online simulation, where the probe surface is applied as displacement constraints on the nodes that are in contact with it so that the displacements u of all nodes are calculated at each iteration. Modeling such contacts and achieving conforming meshes with contact surfaces is an active field of research with various approaches having been proposed for different deformation models (e.g., local mesh refinement [36], multi-resolution meshes [37], condensation [3], force coupling [6]). In our experiments, it was assumed that the contact locations were known *a priori* so that the mesh was generated ensuring that nodes do exist on those interfaces.

3.4 Results

For our experiments, a $60 \times 90 \times 90$ mm tissue-mimicking gelatin phantom, having a soft cylindrical inclusion of 25 mm in diameter, was constructed. The phantom was meshed using the GiD meshing software [38] yielding 493 nodes and 1921 tetrahedra. The elasticity parameters

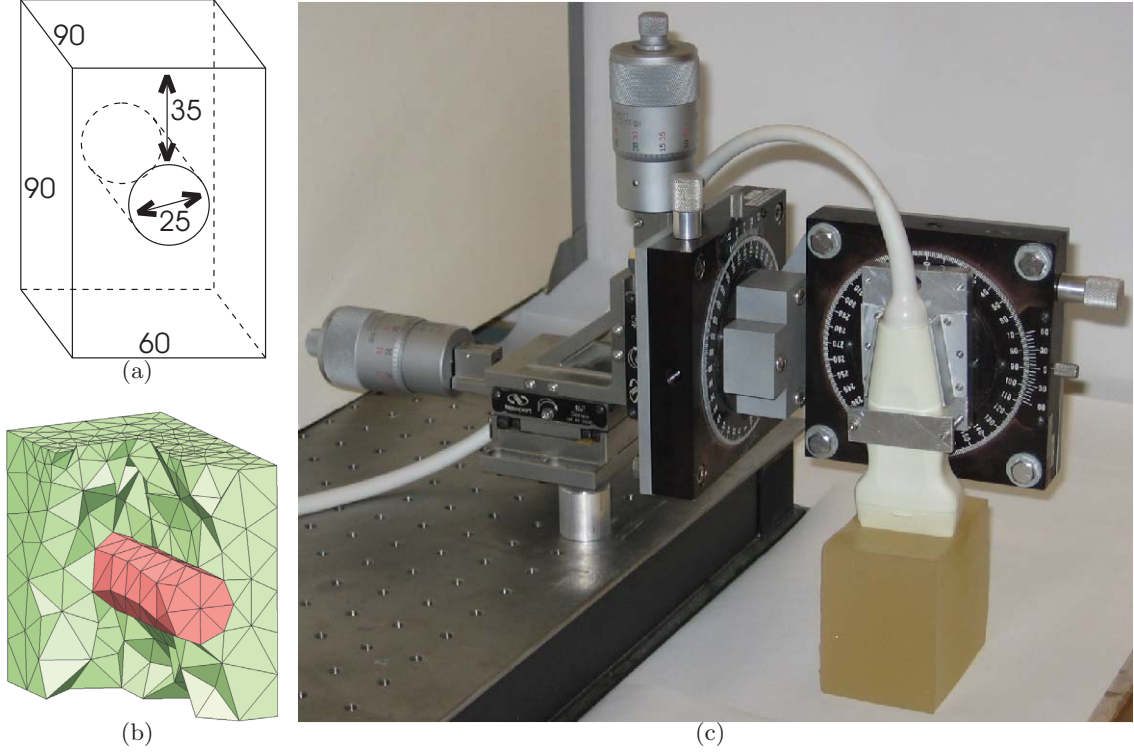


Figure 3.9: (a) The phantom design (in mm), (b) its mesh, and (c) the image acquisition setup.

for the FEM simulation were set to the approximate values known for the gelatin concentrations used. This phantom was imaged using a Sonix RP ultrasound machine from Ultrasonix Medical Corp. with a linear probe mounted on a precision motion stage. Vertical parallel slices with 1 mm separation were acquired with care not to deform the phantom surface. The dimensions and the mesh of our phantom and our imaging setup are seen in Fig. 3.9.

Images, that physically span an area of 37.5×70 mm, were acquired at a resolution of 220×410 from the display pipeline of the ultrasound machine. This is a typical B-mode resolution that this ultrasound machine outputs to the screen for the given probe and default imaging parameters. For our experiment, 75 images were collected at a 1 mm interval while moving the probe in its elevational axis. Accordingly, our reconstructed voxel volume is chosen to be the collection of these parallel slices, which constitutes an average of 8750 voxels per element in our particular phantom mesh.

Deformation was applied by indenting the phantom with the probe. It was simulated using the FEM by fixing the bottom side of the phantom as a fundamental boundary constraint. The probe indentation was applied as a displacement constraint to the mesh nodes coinciding with the probe on the top surface. In the FEM, the Young's moduli were set to 15 kPa and 5 kPa for the background and the circular inclusion, respectively, and a Poisson's ratio of 0.48 was used for both. The ultrasound images were then synthesized using the techniques presented. Some of the acquired and the simulated images are presented in Fig. 3.10.

The images of a simulated probe indentation were compared to the images during an identical physical indentation experiment using their *mutual information* (MI) and sum of

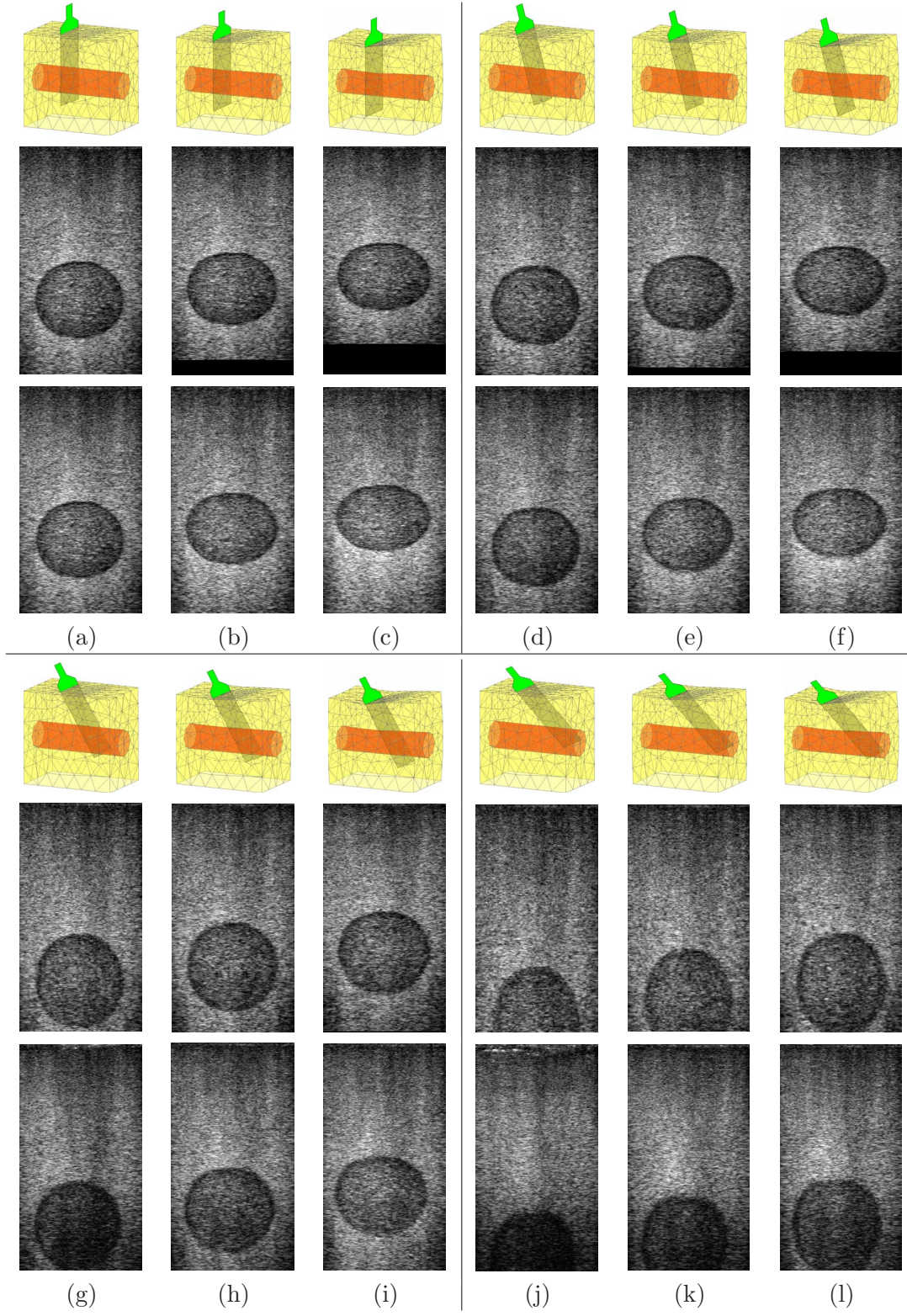


Figure 3.10: Simulated (upper) and acquired (lower) images with 0, 5, and 10 mm indentations for probe tilted at (a-c) 0° ; (d-f) 15° ; (g-i) 30° ; and (j-l) 45° .

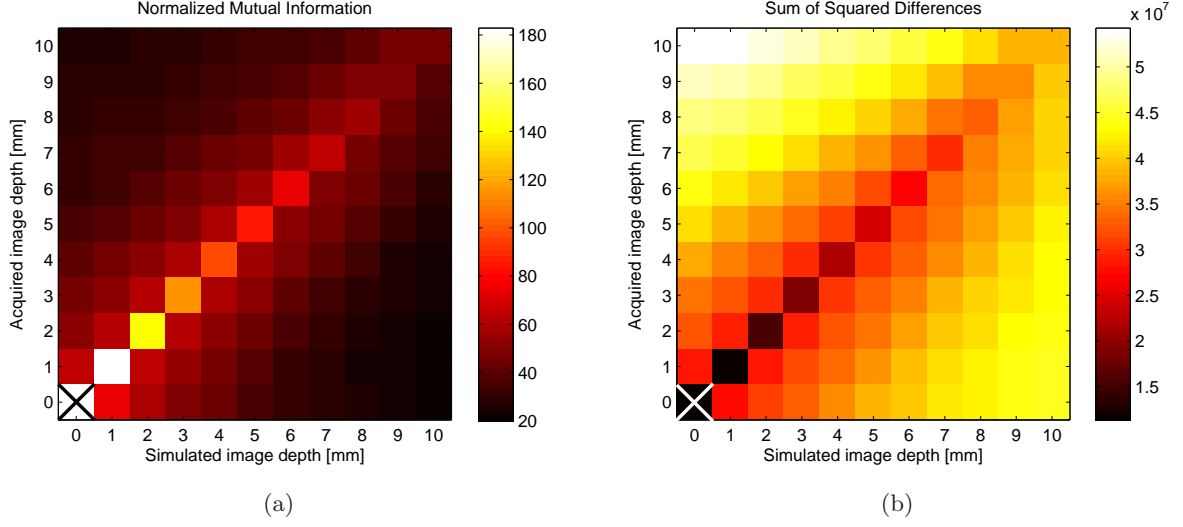


Figure 3.11: Normalized mutual information (a) and sum of squared differences (b) of the images simulated and acquired at 1 mm incremental probe indentations (the values at $[0, 0]$, which match perfectly, are not presented here to preserve the colour map).

squared differences (SSD). 11 images were both simulated and acquired in 1 mm steps up to a 10 mm compression. The MI and SSD between each pair of simulated and acquired images are presented in Fig. 3.11. In this chapter, to present these MI values as a ratio of an absolute measure in our experiments, the values were normalized with the average MI of two images acquired 1 mm apart in the elevational direction. As seen in Fig. 3.11(a), the simulated images and the acquired images for the same indentations have the highest MI and the lowest SSD, as expected. This shows that the simulation can successfully synthesize an image closely resembling a real one. For reference, in our phantom the average MI between an ultrasound image and one that is shifted vertically by one-pixel is 117% of the average MI of two images 1 mm apart (forming the normalization factor) and it is 67% between two acquired images with 1 mm probe indentation, when normalized as defined above.

The method can generate slices of given 220×410 resolution using the nearest-neighbour interpolation (NNI) in less than 13 ms on a 1.87 GHz Pentium computer. Using tri-linear interpolation (TLI), the simulation of a B-mode frame on the same computer takes approximately 25 ms. In order to evaluate the effect of the number of image pixels n and the number of intersected elements m on the computational speed, the frame synthesis time for the NNI was measured using different image parameters. As presented by the computational analysis, a linear dependency on n and a negligible effect of m were expected. First, the image resolution was decreased at 10% decrements of the original resolution while keeping the physical span of the image frame, and then the physical image span was decreased at 10% decrements of the original span while keeping the number of pixels the same. Note that, effectively, the former alters n for a constant m and the latter alters m for a constant n . Consequently, the former decreases the pixel density per element, i.e., the number of pixels falling into each element cross-section whereas the latter increases it. As presented in Fig. 3.12, the number of pixels n was observed to determine the speed in the $\mathcal{O}(n)$ manner as expected, whereas the number

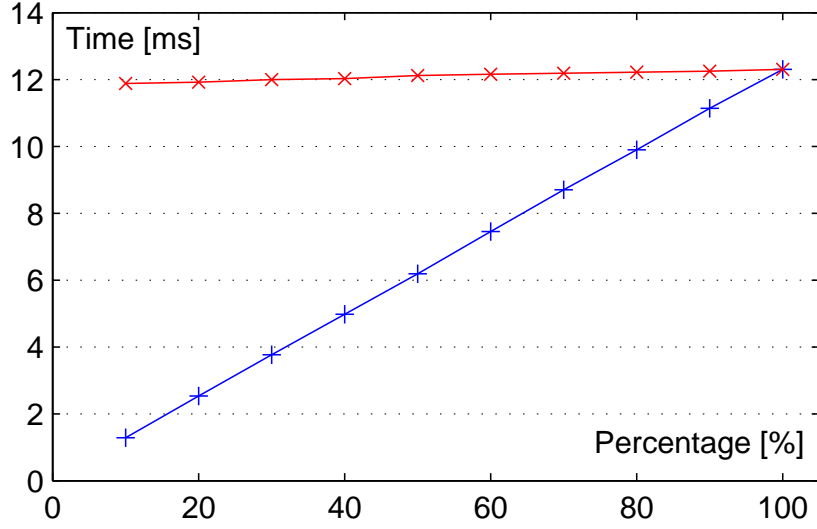


Figure 3.12: Change of frame synthesis time when varying (+) the number of pixels n and (x) the number of intersected elements m expressed in percentages of the full-size phantom images presented.

of intersected elements m exhibited little effect on speed. Note that the slope of this linear dependency on n is defined by the hidden cost of processing each pixel (step 5). Our approach of introducing additional steps and data structures in order to reduce this hidden cost to a few multiplications is the key to accelerating such a method.

The technique above has been implemented for interactive ultrasound visualization with simulated deformation as seen in Fig. 3.13. In this system, a SensAble PhantomTM Premium device mimicking the probe is manipulated by the user while a visual interface displays the tissue mesh and the probe in 3D. At the same time, the ultrasound images are also synthesized by our algorithm and displayed at real-time visual rates (over 40 Hz even when additional computation time for FEM simulation and haptic feedback are added on top of the 13 ms/frame image synthesis time). A simple feedback force normal to the nominal tissue surface and dependent on the current indentation depth was applied to user's hand. The probe indentation was modeled as displacement constraints on the closest surface tissue nodes and the mesh deformation was computed using a pre-computed inverse stiffness matrix.

For an in-vivo assessment of this simulation method, B-mode images of the thigh of a volunteer were also simulated. The 3D volume was reconstructed from B-mode images acquired by the same Sonix RP ultrasound machine. The spatial positions of these images were recorded using a magnetic position sensor, Ascension MiniBIRD, attached to a linear ultrasound probe so that a free-hand scan can be conducted. The anterior upper thigh of a volunteer was then scanned as seen in Fig. 3.14(a) to a depth of 45 mm by translating the probe orthogonal to its imaging plane similarly to the phantom experiment. The spatial arrangement of the collected images can be seen in Fig. 3.14(b). The volume represented by these images is then resampled on a regular grid of 0.1 mm spacing using the Stradwin software [26].

Considering the thigh tissue locally, the femur is the fixed displacement boundary condition, that plays a major role in the way the thigh deforms. The femur anterior surface was segmented

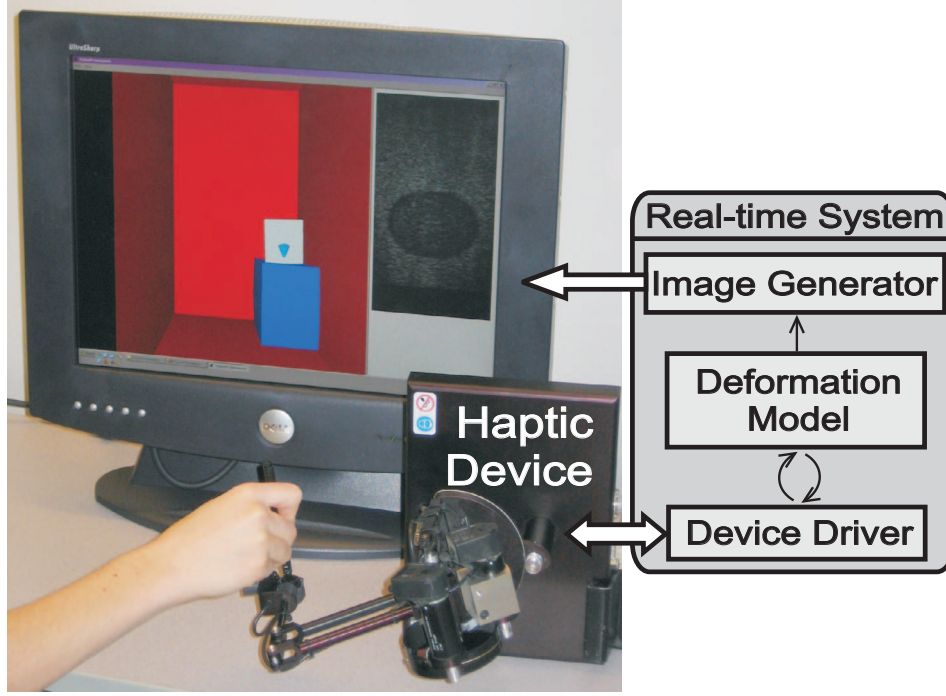


Figure 3.13: Real-time ultrasound scanning simulator.

from a deeper ultrasound scan of the same thigh region. An FEM mesh of 3805 elements and 997 nodes was generated using the GiD software and displacement boundary conditions were defined by spatially fixing the nodes on the femur surface. For the purpose of this thesis, all soft tissue was given a fixed Young's modulus of 15 kPa and a Poisson's ratio was set to 0.48 in this in-vivo example. Ultrasound images of two different probe orientations were simulated using tri-linear interpolation. Simulated images at different indentation depths are seen in Fig 3.15, where the anterior thigh anatomy is observed to deform under probe pressure.

3.5 Discussion

In the literature, there have been *in-plane image deformation* strategies for image registration, deformation correction for volume reconstruction [10, 25], and a training simulation for DVT [18, 23]. However, these 2D approaches cannot simulate out-of-plane deformations. Even if the deformation is driven by motion only in the imaged plane, the boundary conditions may couple image plane motion with motion orthogonal to the imaged plane, and such methods simply cannot account for motion orthogonal to the imaging plane. Indeed, consider the case of brachytherapy, where the prostate is imaged with the transversal crystal of the endo-rectal probe while the needle may push the prostate through the imaging plane (see Fig. 3.1(c)). For such deformation induced orthogonally to the imaging plane, 2D approaches simply do not apply and our method becomes essential.

Our choice of phantom geometry aims at presenting out-of-plane deformation. During the indentation experiments with the probe tilted, not only do the phantom structures compress,

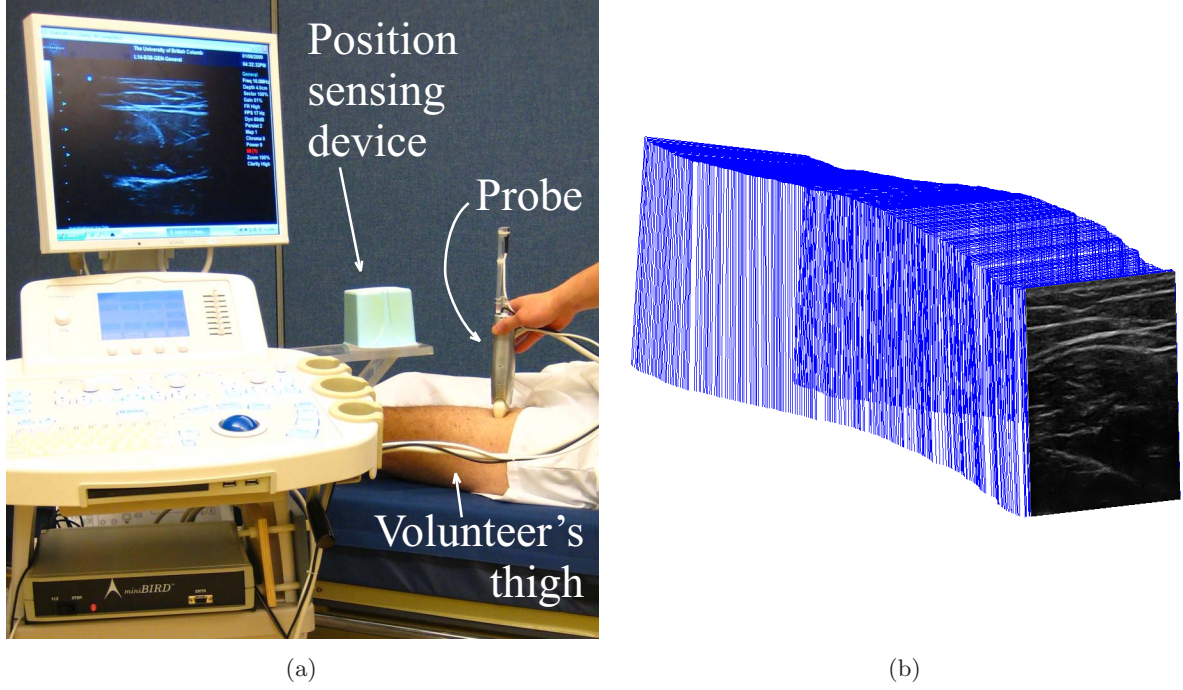


Figure 3.14: (a) The setup for in-vivo data collection and (b) the spatial arrangement of the collected ultrasound images.

but also the image plane moves through the volume in its elevational axis.

Our simulation method is not limited to linear imaging geometry, but can also accommodate other probe geometries such as sector probes. Indeed, regardless of the transducer crystal geometry, the conventional format for displaying, processing, and storing B-mode images is Cartesian-based in compliance with the common screen hardware and image processing algorithms. Consequently, our method simulates the pixels on a regular-grid regardless of the original data acquisition format, similarly to other interpolation-based implementations in the literature. The simulation of a sector image, for instance, can be accommodated in the current technique by simulating all pixels internal to the bounding box of the non-rectangular image footprint and then masking off the pixels lying outside the actual sector image region.

Locating the points of a 3D grid was studied in [39] to resample a 3D mesh to use in the finite difference method for simulating seismic wave propagation. However, note that we are interested in a 2D plane that intersects the mesh in arbitrary orientation. Furthermore, our case involves deformed meshes and registering an image plane manifold between different mesh configurations.

A deformation simulator must have two components: one that predicts how tissue deforms based on a physically valid model and one that can display the result at a high enough frame rate to match the simulation context, i.e., be “real-time”. Considering the former, many medical simulations employ the FEM to produce the mesh deformation in response to tissue forces or changes in the tissue constraints. Coupled with this FEM simulation, the user of a medical simulator should be able to examine the simulated tissue in a manner that is accurate and real-time. The techniques presented in this chapter serve to this latter need.

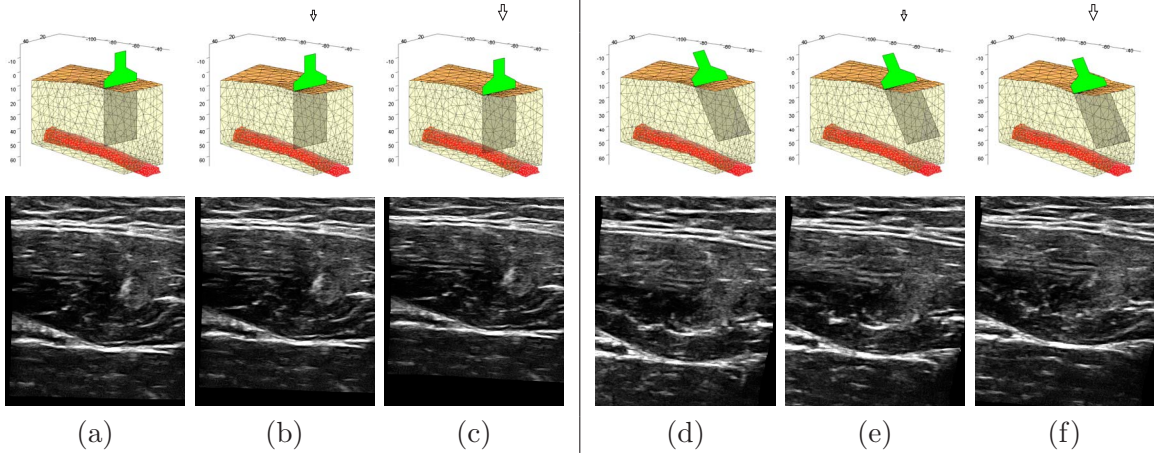


Figure 3.15: Simulated ultrasound images of the thigh with 0, 2, and 5 mm indentations (shown with arrows) for probe tilted at (a-c) 0° and (d-f) 30° .

It is important to realize that our presented method is independent of the FEM simulation and does not introduce any additional approximations and errors beyond those intrinsic to the FEM simulation, i.e., there is no trade-off between image synthesis acceleration and image quality. The image synthesis method uses the same mesh and the same interpolation functions as the FEM simulation, and the re-sampling is performed over a grid of the same resolution as the original medical image data while keeping the highest possible image resolution given the reference image data set. Considering the B-mode image synthesis in particular, the only assumption employed is the isotropy for the ultrasound interaction model. In addition, note that the method introduced for image pixel location gives an exact solution, not an approximation. Such a method is superior to the previous techniques in locating regularly-spaced points at the expense of a small additional storage and its initialization time.

Our method aims at the development of a real-time realistic image synthesis technique, which can be integrated with training systems that involve tissue deformation. In this context, the simulated images were found to be adequate for medical simulators by physicians following a preliminary visual inspection. This will be assessed further by expert sonographers in specific clinical procedures, such as prostate brachytherapy, in the future. An extension of the quantitative evaluation method presented for the phantom images to the collected in-vivo leg scan is not trivial. In phantoms, the boundary constraints and material properties can be adjusted precisely, so that an assessment evaluates strictly the image simulation performance and does not involve any errors associated with the identification of such parameters. Although these parameters can be approximated at some level for actual anatomy, it is difficult to produce exact FEM models and therefore to decouple the effects of the deformable model from the image simulation method in in-vivo data.

As explained in Section 3.3.6 and justified by the results in Fig. 3.12, the frame synthesis time of our simulation depends mainly on the number of pixels, but is also affected by the number of elements intersecting the image to a small degree. Nevertheless, note that it is independent both from the entire physical span of the FEM mesh and from the total number of nodes/elements in it. It is indeed further independent from the entire physical span of the reconstructed image data set, assuming it fits in the computer memory. Consequently, much

larger tissue representations can be handled successfully. Also note that the extent of the reference image volume does not have to match the extent of the FEM mesh. For instance, in our experiment the bottom part of the phantom was not imaged for the reference volume due to the depth setting of the probe and hence the black regions appear at the bottom of the simulated images when indentation is applied. However, the deformation simulation was considering the entire phantom volume for an accurate simulation.

Observing the relatively small effect of the number of intersected elements on synthesis time in Fig. 3.12, it is seen that more elements on the plane will not be detrimental to the simulation. Consequently, techniques such as local mesh refinement will not impede the speed, since $n \gg m$ is still valid. Indeed, one can analyze the data on such a figure to estimate maximum image resolution for a given mesh simulated on a particular machine.

In general, 3D ultrasound reconstruction involves resampling the acquired images in order to generate volume data on a grid-structure. In our phantom image collection, ultrasound slices were parallel and relatively dense creating a regular grid, therefore no further processing was needed for reconstruction. For the in-vivo data, the Stradwin software was used to resample the data in a grid structure. In this thesis, the nearest-neighbour and tri-linear interpolations were employed in the image synthesis step. Other interpolation schemes are also applicable depending on the computational limitations for achieving a required frame-rate in a particular implementation.

Note that, for significantly large deformations, the acquired and the simulated images may not match exactly using the linear-strain approximation due to rotationally-variant linear elements. Nevertheless, there exist numerical treatments in the literature to achieve rotational invariance [40]. Non-linear elasticity with dynamic models has also been proposed in the literature [4]. All these models provide mesh node displacements at given time instants, which can be used by our technique to synthesize images as in the given implementation. Moreover, even other deformation models that provide node displacements, such as finite differences or spring networks, can be used since the barycentric transformation proposed does not assume, nor is bound to, any property of the FEM.

We assume that linear interpolation is used within the FEM mesh, therefore 4-node tetrahedra were employed in our simulation. The interpolation accuracy is maintained by reducing the size of the mesh elements as necessary. The presented techniques still apply to deformations computed by higher-geometry elements, such as 10-node tetrahedra, by using only the corner nodes for image synthesis. Considering element geometries other than tetrahedra, e.g. hexagons, or meshes with mixed elements used for FEM deformation, note that the image plane cross-sections of such convex polyhedra are also convex. Thus, these cross-sections can still be found and marked using the techniques presented. However, an alternative undeforming transformation will be needed instead of using the barycentric coordinates.

Our phantom has the same cross-section longitudinally, therefore two 1 mm-apart images look (and should look) the same to the human eye. Although their speckle pattern may differ, for an observer they both carry the same information about the location, size, and other features of the inclusion. The MI of this minimal and practically in-differentiable speckle pattern change is taken as the base for normalization.

Note that, once the image is marked, the processing of each and every scan-line is independent of another. This allows for completely disjoint synthesis of each scan-line, which is a significant advantage for parallel processing of the computationally intensive step 5. For

instance, on the emerging multi-core CPU architectures, scan-lines can be distributed among CPU cores. Moreover, the simple computation scheme of step 5 is highly aligned with the SIMD (single instruction, multiple data) execution model for parallelization on recent GPU computation/programming technologies, such as CUDA [41].

The approach presented in this chapter is an enabling technique for many medical simulations and in particular for any ultrasound simulator that will allow for tissue deformation. Fast image simulation is also essential for certain deformable registration techniques, where image slices of a deformed volume are compared to a set of reference images while the deformation constraints are being optimized iteratively. Studying accelerated image synthesis is of significant value especially in such registration schemes, where the generation of sliced images through mesh-based deformations is one of the computationally intensive steps. Faster synthesis is also crucial for the processing of high resolution images. A faster algorithm allows for the generation of possibly more than one slice at a time instant (such as to render the volume in 3D) and also facilitates the presentation of more than one modality to a trainee during a training simulation (such as additional deformed MR and/or CT to better understand the anatomy scanned).

This thesis treats the problem of image simulation with deformation using an interpolative approach due to real-time processing limitations and parametrization difficulties of generative models. An ultrasound image is indeed a product of very complex wave interactions in the tissue. In an interpolative approach, instead of attempting to model this complex wave behaviour, image features (and similarly artifacts) are reproduced and placed in simulated images at the 3D location where they were originally acquired. Similarly, our technique assumes that the B-mode image of a deformed tissue region is similar to the properly deformed version of a B-mode image taken prior to deformation. However, the visibility of an actual image feature (e.g. a sharp tissue interface) or an artifact in B-mode imaging may depend on the direction of the incidence of the ultrasound beam, whereas the visibility, with an interpolative approach, depends on the direction of the original data collection regardless of the simulation-time probe orientation. Therefore, interpolative B-mode techniques may not be optimal or applicable for certain tasks. Nevertheless, they are still of great value in the medical field and are employed in various B-mode applications such as volume reslicing in commercial 3D ultrasound machines. Our approach equips the user with a fast and powerful tool for a range of applications from training simulators to fast deformable registration, after carefully considering the interpolative nature of the simulation method and its related limitations for that particular application. Note that, as opposed to ultrasound, for other imaging modalities such as MR, the assumption above may be satisfied.

Our method can be extended for applications and anatomy in which artifacts are prominent by acquiring multiple reconstructions of the same volume where the images are collected at various probe incidence angles. Subsequently, for each frame simulation, the reconstructed volume acquired by the closest probe orientation to the simulated probe will be used. This will effectively maximize the likeness and direction of artifacts in simulation. Note that such a modification introduces additional memory storage cost due to multiple reconstructed volumes; nevertheless, it does not increase the overall computational complexity of the approach.

3.6 Conclusions and Future Work

In this chapter, a technique for synthesizing planar images in deformed meshes of tissue models was presented. The method uses 3D image data of pre-deformed tissue. A pixel enumeration technique originating from common scan-line algorithms was adopted to enable fast identification of mesh elements enclosing each image pixel during deformations given by mesh node displacements. This allows the generation of medical images of considerable size at frame rates that are suitable for real-time applications. This technique was implemented to simulate B-mode images of a deformable phantom and anterior thigh of a volunteer. Synthesized images of probe indentations were then compared to the corresponding images acquired by physically deforming the phantom. A real-time ultrasound simulator implemented on a haptic device was also demonstrated. The results show that the proposed method produces realistic-looking B-mode images. The methods presented are easily adaptable to other imaging modalities and deformation models.

References

- [1] G. Reis, B. Lappé, S. Köhn, C. Weber, M. Bertram, and H. Hagen, *Visualization in Medicine and Life Sciences*. Springer Berlin Heidelberg, 2008, ch. Towards a Virtual Echocardiographic Tutoring System, pp. 99–119.
- [2] H. Maul, A. Scharf, P. Baier, M. Wüstemann, H. H. Günter, G. Gebauer, and C. Sohn, “Ultrasound simulators: Experience with sonotrainer and comparative review of other training systems,” *Ultrasound Obstet Gynecol*, vol. 24, pp. 581–585, 2004.
- [3] M. Bro-Nielsen, “Finite element modeling in medical VR,” *Journal of the IEEE*, vol. 86, no. 3, pp. 490–503, 1998.
- [4] G. Picinbono, H. Delingette, and N. Ayache, “Non-linear anisotropic elasticity for real-time surgery simulation,” *Graphical Models*, vol. 65, pp. 305–321, 2003.
- [5] O. Goksel, “Ultrasound image and 3D finite element based tissue deformation simulator for prostate brachytherapy,” Master’s thesis, University of British Columbia, 2004.
- [6] J. Berkley, G. Turkiyyah, D. Berg, M. A. Ganter, and S. Weghorst, “Real-time finite element modeling for surgery simulation: An application to virtual suturing,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 10, no. 3, pp. 314–325, 2004.
- [7] J. C. Bamber and R. J. Dickinson, “Ultrasonic B-scanning: a computer simulation,” *Physics in Medicine and Biology*, vol. 25, no. 3, pp. 463–479, 1980.
- [8] J. A. Jensen, “A model for the propagation and scattering of ultrasound in tissue,” *J Acoust Soc Am*, vol. 89, pp. 182–191, 1991.
- [9] R. W. Prager, A. Gee, and L. Berman, “Stradx: Real-time acquisition and visualisation of freehand 3D ultrasound,” *Medical Image Analysis*, vol. 3, no. 2, pp. 129–140, 1999.
- [10] R. N. Rohling, A. H. Gee, and L. Berman, “A comparison of freehand three-dimensional ultrasound reconstruction techniques,” *Medical Image Analysis*, vol. 3, no. 4, pp. 339–359, 1999.
- [11] R. S. José-Estépar, M. Martín-Fernández, P. P. Caballero-Martínez, C. Alberola-López, and J. Ruiz-Alzola, “A theoretical framework to three-dimensional ultrasound reconstruction from irregularly sampled data,” *Ultrasound in Medicine and Biology*, vol. 29, no. 2, pp. 255–269, 2003.
- [12] D. Aiger and D. Cohen-Or, “Real-time ultrasound imaging simulation,” *Real-Time Imaging*, vol. 4, no. 4, pp. 263–274, 1998.

- [13] H.-H. Ehrlicke, “SONOSim3D: a multimedia system for sonography simulation and education with an extensible case database,” *European Journal of Ultrasound*, vol. 7, pp. 225–230, 1998.
- [14] P. Abolmaesumi, K. Hashtrudi-Zaad, D. Thompson, and A. Tahmasebi, “A haptic-based system for medical image examination,” in *IEEE Engineering in Medicine and Biology Conference (EMBC)*, San Francisco, CA, USA, Sep 2004, pp. 1853–1856.
- [15] I. M. Heer, K. Middendorf, S. Müller-Egloff, M. Dugas, and A. Strauss, “Ultrasound training: the virtual patient,” *Ultrasound in Obstetrics and Gynecology*, vol. 24, no. 4, pp. 440–444, 2004.
- [16] M. Weidenbach, F. Wild, K. Scheer, G. Muth, S. Kreutter, G. Grunst, T. Berlage, and P. Schneider, “Computer-based training in two-dimensional echocardiography using an echocardiography simulator,” *Journal of the American Society of Echocardiography*, vol. 18, no. 4, pp. 362–366, 2005.
- [17] Sonofit GmbH. Sonofit ultrasound training system. Darmstadt, Germany. [Online]. Available: <http://www.sonofit.com/>
- [18] D. d’Aulignac, C. Laugier, J. Troccaz, and S. Vieira, “Towards a realistic echographic simulator,” *Medical Image Analysis*, vol. 10, pp. 71–81, 2006.
- [19] Y. Zhu, D. Magee, R. Ratnalingam, and D. Kessel, “A virtual ultrasound imaging system for the simulation of ultrasound-guided needle insertion procedures,” in *Medical Image Understanding and Analysis*, 2006, pp. 61–65.
- [20] D. Magee, Y. Zhu, R. Ratnalingam, P. Gardner, and D. Kessel, “An augmented reality simulator for ultrasound guided needle placement training,” *Med Bio Eng Comput*, vol. 45, pp. 957–967, 2007.
- [21] A. Hostettler, C. Forest, A. Forgione, L. Soler, and J. Marescaux, “Real-time ultrasonography simulator based on 3D CT-scan images,” in *Medicine Meets Virtual Reality (MMVR)*, 2005, pp. 191–193.
- [22] F. P. Vidal, N. W. John, A. E. Healey, and D. A. Gould, “Simulation of ultrasound guided needle puncture using patient specific data with 3D textures and volume haptics,” *Computer Animation and Virtual Worlds*, vol. 19, pp. 111–127, 2008.
- [23] D. Henry, J. Troccaz, J. L. Bosson, and O. Pichot, “Ultrasound imaging simulation: Application to the diagnosis of deep venous thromboses of lower limbs,” in *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 1998, pp. 1032–1040.
- [24] M. Nesme, “Milieu mécanique déformable multi résolution pour la simulation interactive,” Ph.D. dissertation, Université Joseph Fourier, June 2008.
- [25] M. R. Burcher, L. Han, and J. A. Noble, “Deformation correction in ultrasound images using contact force measurements,” in *IEEE Workshop Math Methods Bio Img Anal*, Kauai, HI, USA, 2001, pp. 63–70.

- [26] A. Gee, R. Prager, G. Treece, C. Cash, and L. Berman, “Processing and visualizing three-dimensional ultrasound data,” *British Journal of Radiology*, vol. 77, pp. S186–S193, 2004.
- [27] O. Goksel and S. E. Salcudean, “Real-time synthesis of image slices in deformed tissue from nominal volume images,” in *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, Brisbane, Australia, Oct 2007, pp. 401–408.
- [28] —, “Fast B-mode ultrasound image simulation of deformed tissue,” in *IEEE Engineering in Medicine and Biology Conference (EMBC)*, Lyon, France, Aug 2007, pp. 2159–2162.
- [29] J. F. Krücker, G. L. LeCarpentier, J. B. Fowlkes, and P. L. Carson, “Rapid elastic image registration for 3-d ultrasound,” *IEEE Transactions on Medical Imaging*, vol. 21, no. 11, pp. 1384–1394, 2002.
- [30] D. Zikic, W. Wein, A. Khamene, D.-A. Clevert, and N. Navab, “Fast deformable registration of 3d-ultrasound data using a variational approach,” in *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, Copenhagen, Denmark, 2006, pp. 915–923.
- [31] J. E. Goodman and J. O’Rourke, *Handbook of Discrete and Computational Geometry*, 2nd ed. Chapman & Hall/CRC, 2004.
- [32] D. Dobkin and R. J. Lipton, “Multidimensional searching problems,” *SIAM Journal on Computing*, vol. 5, no. 2, pp. 181–186, 1976.
- [33] D. G. Kirkpatrick, “Optimal search in planar subdivisions,” *SIAM Journal on Computing*, vol. 12, pp. 28–35, 1983.
- [34] C. B. Barber, D. P. Dobkin, and H. T. Huhdanpaa, “The Quickhull algorithm for convex hulls,” *ACM Trans Math Softw*, vol. 22, no. 4, pp. 469–483, 1996.
- [35] J. E. Bresenham, “Algorithm for computer control of a digital plotter,” *IBM Systems Journal*, vol. 4, no. 1, pp. 25–30, 1965.
- [36] O. C. Zienkiewicz and R. L. Taylor, *The Finite Element Method*. Butterworth-Heinemann, 2000.
- [37] G. Debunne, M. Desbrun, M.-P. Cani, and A. H. Barr, “Dynamical real-time deformations using space and time adaptive sampling,” *ACM Transactions on Graphics (SIGGRAPH Proceedings)*, vol. 20, pp. 31–36, 2001.
- [38] R. Löhner, “Progress in grid generation via the advancing front technique,” *Engineering with Computers*, vol. 12, pp. 186–210, 1996.
- [39] A. Fousse, E. Andres, J. Françon, Y. Bertrand, and D. Rodrigues, “Fast point locations with discrete geometry,” in *SPIE Vision Geometry Conference*, Denver, Colorado, USA, Jul 1999, pp. 33–44.
- [40] M. Müller, J. Dorsey, L. McMillan, R. Jagnow, and B. Cutler, “Stable real-time deformations,” in *SIGGRAPH/Eurographics Symp on Computer Animation*, 2002, pp. 49–54.
- [41] T. R. Halfhill, “Microprocessor report: Parallel processing with CUDA,” Reed Electronics Group: In-Stat, Tech. Rep., 2008.

Chapter 4

Modeling and Simulation of Flexible Needles ⁵

4.1 Introduction

Percutaneous insertion of long and flexible needles into soft tissue is involved in many clinical and therapeutic procedures such as biopsy and prostate brachytherapy. As a result of needle-tissue interaction, the needle base movements and, for needles with a beveled tip, the asymmetric cutting force on the tip, the tissue deforms and the needle bends (see Fig. 4.1). The targeting procedure is complicated by the bending of the needle shaft, target displacement due to tissue deformation, and insufficient visual feedback from medical imaging modalities. Accurate needle insertion requires significant skill and training of the performing physician. Modeling, simulation, and path planning of needle insertion are emerging fields of research aimed at providing the physicians with training devices and accurate pre-surgery plans.

Needle insertion simulators usually include a soft tissue model, a flexible needle model and a needle-tissue interaction model. The simulation of needle bending and tissue deformation together in one combined model is generally not feasible for two reasons: (i) a fast solution for a large tissue mesh requires exploiting simplifications in deformation equations, which are not suitable for estimating the needle behavior and (ii) the interaction surface of the needle and tissue changes during insertion, which many techniques like the Finite Element Method (FEM) cannot inherently accommodate. Therefore, generally two separate models for the tissue and the needle are employed [1–4] with a third needle-tissue contact model governing their interaction.

Deformable tissue models have been studied extensively in simulating tissue deformation during surgery and needle insertion [1–5]. Models based on the FEM are the most common techniques employed for tissue deformation simulation during needle insertion [1, 2, 4, 6]. Mass-spring models [7] have also been used for this purpose. The interaction of needles with such deformable tissue models has also been studied widely [2, 5, 8, 9].

The aim of this chapter, the preliminary results of which were presented in [10], is to compare different models for simulating the bending of flexible needles. In general, medical needles can be categorized into three major groups: rigid needles, highly flexible needles, and (moderately) flexible needles. Rigid needles keep a straight posture regardless of the forces applied on them during insertion. Due to their simplicity, rigid needle models have been used when the needle physical properties and the insertion procedure lead to negligible bending [1, 2, 6]. On the opposite end of the flexibility range are the *highly* flexible needles.

⁵A version of this chapter has been published. O.Goksel, E.Deaghan, and S.E.Salcudean, “Modeling and Simulation of Flexible Needles,” *Medical Engineering and Physics*, 31(9):1069-1078, Nov 2009. DOI:10.1016/j.medengphy.2009.07.007

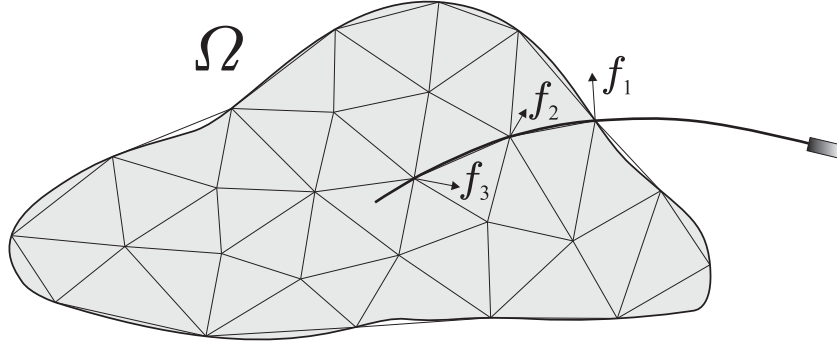


Figure 4.1: The forces on the needle shaft caused by tissue deformation and needle base manipulation.

These needles are assumed to bend in the direction of their tip bevel with a constant curvature, without applying considerable force to the tissue in the lateral direction. These needles were modeled as a non-holonomic system [3] and were used for needle insertion simulations and planning in [11–13].

Some needles, such as brachytherapy needles, cannot be categorized as either rigid or highly flexible. They are not rigid, since their deflection during procedures is significant. They are not highly flexible either, since a considerable lateral force is necessary to bend them. Several groups have modeled this type of needles. DiMaio and Salcudean simulated the needle as an elastic material using FEM with geometric non-linearity and 3-node triangular elements and validated this method in phantom studies [14]. This method was later extended to 3D using 4-node tetrahedral elements [4]. Glozman and Shoham used linear 2D beam elements to simulate the needle bending for needle steering [7]. Linear beam theory was also used to introduce a needle steering model with online parameter estimator [15], to estimate the needle tip deflection during insertion due to tip bevel [16], and to identify the shaft force profile due to the bevel [17]. Models based on linear beam theory are relatively simple and fast. However, they are not rotationally invariant and cannot preserve the needle length during large deformations/deflections.

Many needles, such as brachytherapy needles, consist of a stylet sliding inside a hollow cannula. Physical modeling of this combination without any simplifications is indeed very complicated as it requires separate models for the stylet and the cannula and an interaction model to simulate their interface. In this thesis, flexible needles are approximated as solid bars and, accordingly, models applicable to solid bars are examined to simulate their flexion.

Three different models were used to simulate needle bending. The first two models are based on the FEM and were chosen due to their frequent use in the literature, while the third model is an angular springs model. The first model uses 4-node tetrahedral elements, where nonlinear geometry is accommodated to simulate large deformations. The second model also accommodates nonlinear geometry and uses Euler-Bernoulli nonlinear beam elements. In this work, nonlinear beam elements were preferred over more common linear ones in the literature due to their superiority in modeling large deformations. The third model is novel and utilizes angular springs for the quasi-static simulation of needle bending. In the literature, angular springs have been used to model cantilever-like structures [18] such as beams in mechanical

engineering [19] and hair deformation in computer graphics [20]. They have been also incorporated in 3D mass-spring models to simulate large volume deformations [21]. In this thesis, this type of a spring finite-difference model is implemented for medical needles, for which its performance is compared with two other common types of physically-based models using FEM.

The Young's modulus is the parameter that describes needle bending in the first two models. Similarly, the third model is identified by its spring constant. The parameters defining each model are identified for a brachytherapy needle through experiments, where several lateral forces were applied to the needle tip and the shaft deflection was recorded. The models were then studied for their accuracy in simulating the actual needle deflection observed during experiments.

In this thesis, the needle bending models are devised in 3D. Note that, for the single force applied at the needle tip during the experiments, the needle deformation is entirely planar. Therefore, the parameter identification and the model validations were performed in 2D. Nevertheless, the same identified parameters also describe a needle in 3D, since the shafts of most medical needles are built with axial symmetry resulting in symmetric deflection for the same force rotated around their long axis.

The following section derives the models in 3D. Next, the experimental method to validate the models is described in Section 4.3. The results and a discussion follow in sections 4.4 and 4.5, respectively. Finally, conclusions are presented in the last section.

4.2 Methods

4.2.1 Finite Element Method using Tetrahedral Elements

The Finite Element method is a powerful tool for approximating a solution to the continuum mechanics equations. In this method, an entire body Ω is divided into several discrete elements Ω_e . Then, the constitutive equations are approximated over each element and combined to give an approximation to the global solution. Various types of elements can be chosen depending on the nature of the problem. 4-node tetrahedral (TET4) or 3-node triangular (TRI3) elements are the simplest elements to use in 3D or 2D deformation analysis, respectively.

Considering a deformable body, the 3D coordinates of a point in the undeformed configuration can be written as $[x_1 \ x_2 \ x_3]$ in the reference frame. The coordinates of that point in the deformed configuration can then be expressed using the displacements $[u_1 \ u_2 \ u_3]$ added onto the undeformed coordinates. Using this definition, the Green strain and the second Piola-Kirchhoff stress tensors are given as follows:

$$\epsilon_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} + \sum_{k=1}^3 \frac{\partial u_k}{\partial x_i} \frac{\partial u_k}{\partial x_j} \right), \quad (4.1)$$

$$\sigma_{ij} = \sum_{k=1}^3 \sum_{l=1}^3 \mathbf{C}_{ijkl} \epsilon_{kl}, \quad (4.2)$$

where $i, j \in \{1, 2, 3\}$ and \mathbf{C}_{ijkl} is the material moduli tensor for linear elastic materials which is a function of Young's modulus and Poisson's ratio.

When the deformation is small, the second order terms in (4.1) can be neglected leading to a linear relation between strain and displacement. This linear relation leads to a set of

linear algebraic equations with a constant coefficient matrix. Such a system can be solved rapidly using off-line computation and condensation [2, 22]. However, the assumption of small deformation is not valid for needle bending, where the deformation is large with respect to the needle diameter [14]. Thus, nonlinear geometry as in (4.1) must be included in the model. This leads to a nonlinear relation between the nodal forces and displacements:

$$\underline{f} = \hat{\mathbf{K}}(\underline{u}) \underline{u} \quad (4.3)$$

where \underline{u} and \underline{f} are the vectors of nodal displacements and forces, respectively, and $\hat{\mathbf{K}}$ is the stiffness matrix, the elements of which can be represented as functions of nodal displacements \underline{u} . This nonlinear equation can accommodate the axial displacements during lateral deformation and, therefore, preserves the needle length. However, the use of condensation is not possible in the nonlinear case and this leads to a time-consuming solution.

The Newton-Raphson method is an effective iterative technique for solving the nonlinear algebraic equations in (4.3) [23]. In this method, a tangent stiffness matrix \mathbf{K}_T is computed at each iteration and used to find a descent direction for the solution using the iteration scheme below:

$$\mathbf{K}_T \Delta \underline{u}^t = \underline{f} - \hat{\mathbf{K}}(\underline{u}^t) \underline{u}^t \quad (4.4)$$

$$\underline{u}^{t+1} = \underline{u}^t + \Delta \underline{u}^t \quad (4.5)$$

to update the nodal displacements at each iteration t .

In our simulations, the Newton-Raphson method converged to a solution for (4.3) in a few iterations. Unfortunately, computing the tangent stiffness matrix and solving (4.4) at each iteration are both computationally intensive operations.

4.2.2 Finite Element Method using Nonlinear Beam Elements

Beam elements were designed and introduced in the FEM literature in order to model the deformation of bars, rods, and other beam-like structures [24]. These elements are expected to be better suited for needle deformations than triangular/tetrahedral elements. Thus, a beam-element model was also implemented to compare its performance in modeling needles. Euler-Bernoulli beam element is a common element formulation to model thin bars and therefore is used as a needle model in this chapter.

Euler-Bernoulli beam theory suggests that each plane perpendicular to the beam axis prior to any deformation remains as a plane that is still perpendicular to the beam axis after the deformation. Under this condition the displacements of each material point can be written as a function of the displacement along the neutral axis of the beam as follows:

$$u(x, y, z) = u_0(x) - y \frac{\partial v_0(x)}{\partial x} - z \frac{\partial w_0(x)}{\partial x}, \quad (4.6)$$

$$v(x, y, z) = v_0(x) - z \alpha_0(x), \quad (4.7)$$

$$w(x, y, z) = w_0(x) + y \alpha_0(x), \quad (4.8)$$

where u_0 , v_0 , and w_0 are the neutral axis displacements along the x , y , and z axes, respectively, and α_0 is the twist around the beam axis. It is assumed that the beam axis is lying along the x axis as shown in Fig. 4.2. Please note that, in order to conform to the beam-element literature,

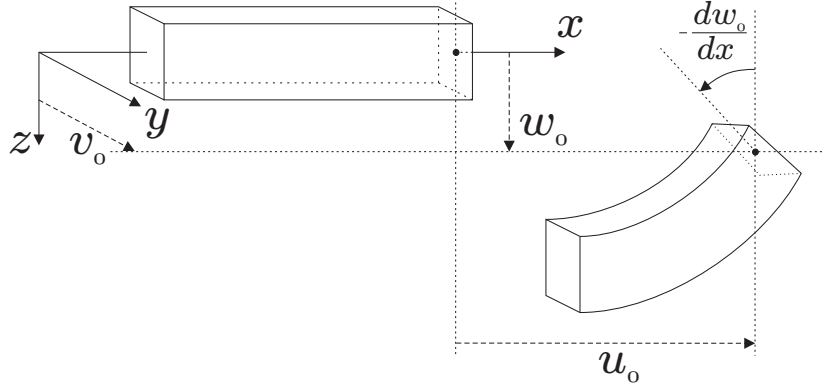


Figure 4.2: Euler-Bernoulli beam element under deformation.

the node positions $[x_1 \ x_2 \ x_3]$ and the node displacements $[u_1 \ u_2 \ u_3]$ in the previous subsection are referred here as $[x \ y \ z]$ and $[u \ v \ w]$, respectively.

Assuming that the beam twist is small while the lateral deformations are moderately large, strains in the beam element can be approximated as [25]:

$$\epsilon_{xx} \approx \frac{du_0}{dx} - y \frac{d^2 v_0}{dx^2} - z \frac{d^2 w_0}{dx^2} + \frac{1}{2} \left(\left(\frac{dv_0}{dx} \right)^2 + \left(\frac{dw_0}{dx} \right)^2 \right), \quad (4.9)$$

$$\epsilon_{xy} \approx -\frac{1}{2} z \frac{d\alpha_0}{dx}, \quad (4.10)$$

$$\epsilon_{xz} \approx \frac{1}{2} y \frac{d\alpha_0}{dx}. \quad (4.11)$$

In the beam element, the displacements of the material nodes on the beam axis are interpolated from the nodal variables of the two end-points (nodes) of the beam. The nodal variables at node p are the axial displacement u_p , the twist α_p , the transverse displacements v_p and w_p (along the y and z axes, respectively), and their derivatives $\phi_p = -dv/dx$ and $\psi_p = -dw/dx$. Euler-Bernoulli beam theory employs cubic interpolation functions for lateral displacements, and linear interpolation functions for axial displacements and twist angle. Therefore, within each element between nodes p and $p+1$, the variables are interpolated as follows:

$$u_0(x) = N_1(x)u_p + N_2(x)u_{p+1} \quad (4.12)$$

$$\alpha_0(x) = N_1(x)\alpha_p + N_2(x)\alpha_{p+1} \quad (4.13)$$

$$v_0(x) = M_1(x)v_p + M_2(x)\phi_p + M_3(x)v_{p+1} + M_4(x)\phi_{p+1} \quad (4.14)$$

$$w_0(x) = M_1(x)w_p + M_2(x)\psi_p + M_3(x)w_{p+1} + M_4(x)\psi_{p+1} \quad (4.15)$$

where $N_i(x)$ and $M_i(x)$ denote the linear and cubic interpolation functions, respectively.

An element equilibrium equation relating loads on the nodes to the nodal variables is written as:

$$\mathbf{K}(\underline{u}) \underline{u} = \underline{f}, \quad (4.16)$$

where \underline{u} is the vector of nodal variables described above and \underline{f} is the vector of corresponding nodal forces and torques. Please see Appendix D for detailed information on the calculation

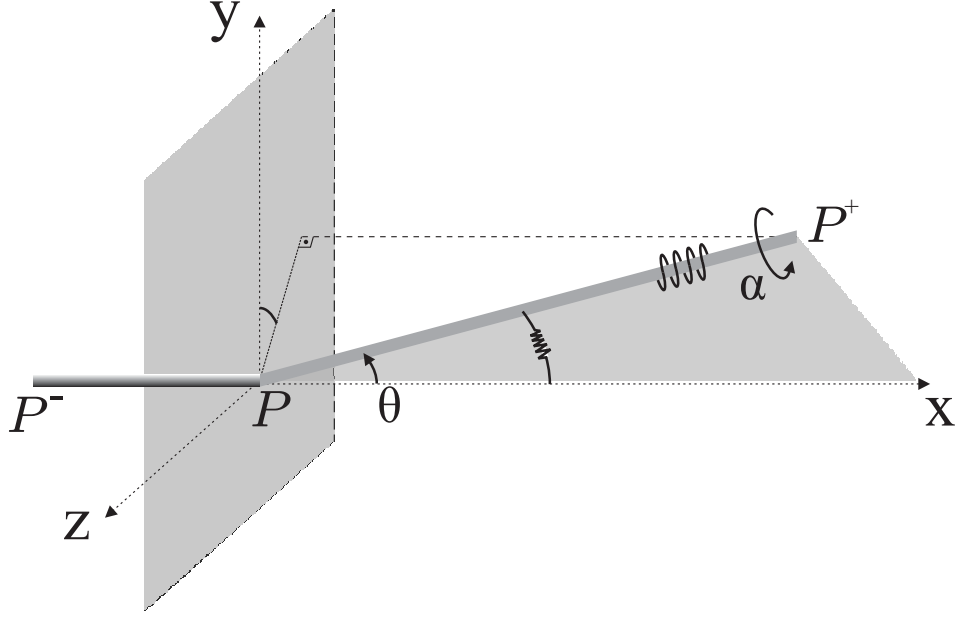


Figure 4.3: Angles of bending and twisting between two needle segments.

of the stiffness matrix and its integration. To solve (4.16), Picard's method [24] is used. This method employs the following iterative steps:

$$\mathbf{K}(\underline{u}^t) \underline{u}^{t+1} = \underline{f} \quad (4.17)$$

where t is the iteration number.

4.2.3 Angular Springs Model

In this section, the needle is modeled as a discrete structure composed of n connected rigid rods. A joint P between two consecutive segments of such a model is shown in Fig. 4.3, where the segment PP^+ is bent and twisted relative to the segment before it, P^-P , under external loads. The magnitude of bending and twisting can be described by two angles: the bending (zenith) angle θ from the x axis to the segment PP^+ , and the twisting angle α about the segment PP^+ due to torsion around its axis.

Naturally, a deformed (bent/twisted) needle structure applies internal reaction (“straightening”) torques/forces to *un-bend* and *un-twist* itself. These are modelled as torques applied to link PP^+ at joint P to restore its rest configuration. The effect of these torques can be visualized as two rotational springs—one un-bending and the other un-twisting the segment—shown in Fig. 4.3. It is evident that the magnitudes of such restoring torques are functions of the deviations (angles θ and α) from the rest configuration. Let us define the magnitudes of these reaction torques, τ_b for bending and τ_t for twisting, as functions of the angles as follows:

$$\tau_b = h_b(\theta) , \quad (4.18)$$

$$\tau_t = h_t(\alpha) . \quad (4.19)$$

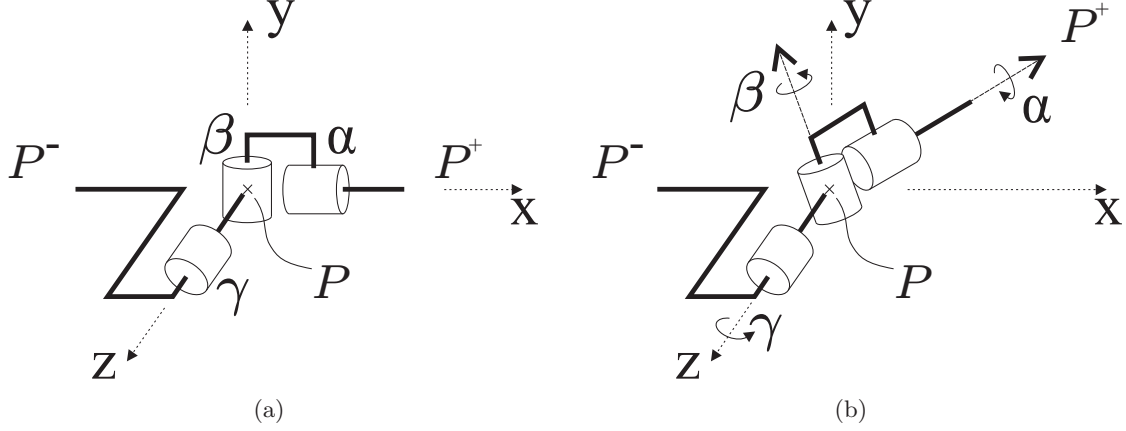


Figure 4.4: Three joints of our spherical wrist model in (a) initial and (b) bent configurations.

Although the actual characteristics of h_b and h_t may be complex and possibly nonlinear, for the elastic range of deformations, a linear relationship is expected. Indeed, when an infinitesimal section of a cantilever shaft is analyzed for its bending (in Appendix E) and its twisting (in Appendix F), linear angle-torque dependencies are observed for both of them. These insights motivate us to devise our model composed of linearly-acting springs as follows:

$$\tau_b = k_b \theta , \quad (4.20)$$

$$\tau_t = k_t \alpha . \quad (4.21)$$

Then, the energy stored in these springs can be written as:

$$V(\theta, \alpha) = \frac{1}{2} k_b \theta^2 + \frac{1}{2} k_t \alpha^2 . \quad (4.22)$$

As a model of the connections between each pair of segments, a spherical wrist —the rest configuration of which is shown in Fig. 4.4(a)— is chosen in this thesis. These three wrist joints, seen in a bent/twisted configuration in Fig. 4.4(b), define the orientation of the segment to its right with respect to the segment to its left. Note that, whereas the third joint α defines the twisting angle independently, the bending angle θ is a combination of the joint angles γ and β . It can be geometrically shown that:

$$\cos \theta = \cos \gamma \cos \beta . \quad (4.23)$$

In order to derive compliance models for the joints γ and β , let us rewrite the energy as a function of these joint variables, e.g. $V(\gamma, \beta, \alpha)$. For sufficiently small angles, $\theta \approx \sin \theta$. This greatly simplifies the following derivation and the solution of final system of equations. Thus, replacing $\theta^2 = \sin^2 \theta$ and (4.23) in (4.22):

$$V(\theta, \alpha) = \frac{1}{2} k_b (1 - \cos^2 \theta) + \frac{1}{2} k_t \alpha^2 , \quad (4.24)$$

$$V(\gamma, \beta, \alpha) = \frac{1}{2} k_b (1 - \cos^2 \gamma \cos^2 \beta) + \frac{1}{2} k_t \alpha^2 . \quad (4.25)$$

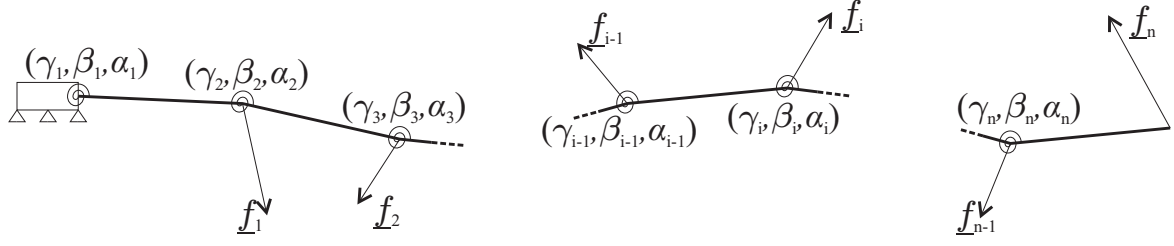


Figure 4.5: The entire needle in equilibrium state constrained at the base and under nodal forces at its joints.

Considering the invariance of the Langrangian at equilibrium, the partial derivatives of joint angles are equal to joint torques, which are the gradients of the potential with respect to the joint angle coordinates, as follows:

$$\tau_\gamma = \frac{\partial V}{\partial \gamma} = k_b \sin \gamma \cos \gamma \cos^2 \beta , \quad (4.26)$$

$$\tau_\beta = \frac{\partial V}{\partial \beta} = k_b \sin \beta \cos \beta \cos^2 \gamma , \quad (4.27)$$

$$\tau_\alpha = \frac{\partial V}{\partial \alpha} = k_t \alpha . \quad (4.28)$$

The small angle approximation above also applies to γ and β , i.e. $\sin \gamma \approx \gamma$, $\sin \beta \approx \beta$, $\cos \gamma \approx 1$, and $\cos \beta \approx 1$, thus yielding:

$$\tau_\gamma \approx k_b \gamma , \quad (4.29)$$

$$\tau_\beta \approx k_b \beta . \quad (4.30)$$

Consider a 3D bent needle posture in equilibrium with force loads \underline{f}_i applied at its joints as in Fig. 4.5. Using our spherical wrist model, a Jacobian matrix \mathbf{J} can be written for this $3n$ -joint robotic arm [26], where the joint torques $\underline{\tau} = [\tau_1' \dots \tau_n']'$ relate to forces \underline{f} as follows:

$$\underline{\tau} = \mathbf{J}' \underline{f} . \quad (4.31)$$

From (4.28–4.30), the $3n$ vector of joint torques $\underline{\tau}$ can be written as a combination of $3n$ joint angles $\underline{\Phi} = [\gamma_1 \beta_1 \alpha_1 \dots \gamma_n \beta_n \alpha_n]'$ and the corresponding joint stiffnesses as follows:

$$\underline{\tau} = \mathbf{K}_s \underline{\Phi} \quad (4.32)$$

where \mathbf{K}_s is a diagonal matrix with diagonal elements $\{k_{b,1} k_{b,1} k_{t,1} \dots k_{b,n} k_{b,n} k_{t,n}\}$ being the joint stiffnesses .

Note that since the Jacobian \mathbf{J} is configuration dependent, (4.31) is indeed a nonlinear system of equations as follows:

$$\mathbf{K}_s \underline{\Phi} = \mathbf{J}'(\underline{\Phi}) \underline{f} . \quad (4.33)$$

In this thesis, this nonlinear system is solved iteratively using Picard's method:

$$\underline{\Phi}^{t+1} = \mathbf{K}_s^{-1} \mathbf{J}'(\underline{\Phi}^t) \underline{f} \quad (4.34)$$

which consists of two phases: (i) given the needle configuration, the Jacobian is composed, and (ii) the torques found by $\mathbf{J}' \underline{f}$ are divided by corresponding joint stiffnesses to find the joint angles and hence the needle posture at the next iteration.

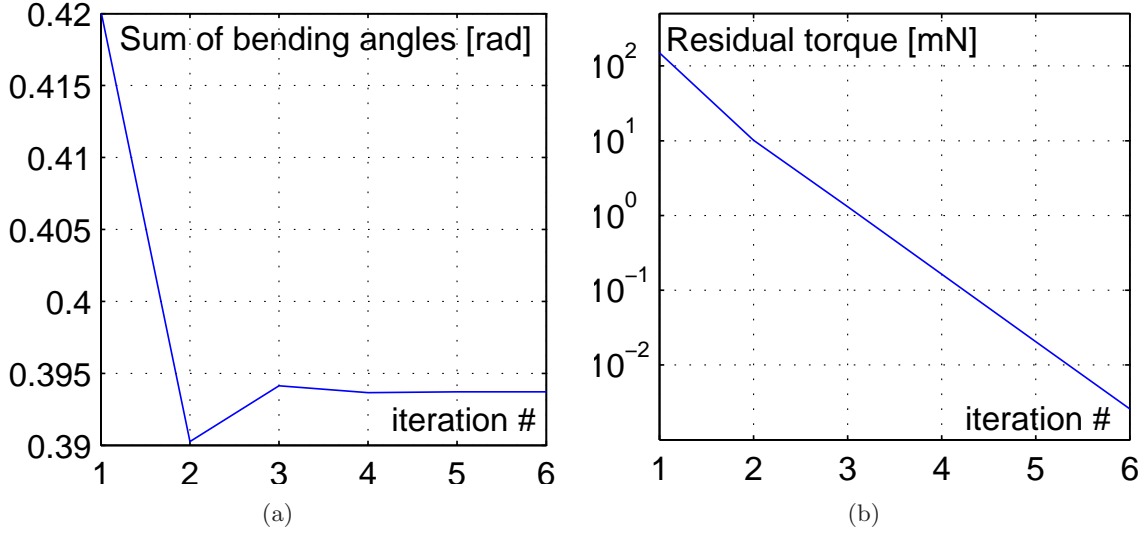


Figure 4.6: For a sample bending simulation, (a) shows the sum of joint angles γ , which is the deflection of the needle tip, and (b) shows the residual torque at each iteration.

4.2.4 A Common Convergence Criterion for All Models

In order to have a fair comparison between the three needle models, a common convergence criterion was devised to be used in the simulations as follows:

$$Error = \frac{\|\underline{u}^t - \underline{u}^{t-1}\|}{\|\underline{u}^t\|} < \lambda \quad (4.35)$$

where \underline{u}^t is the vector of nodal displacements at the iteration t and λ is the tolerance. In this thesis, a tolerance of 10^{-4} is used.

When the convergence criterion in (4.35) is met, the residual of error for all the models were checked to ensure convergence. For the proposed angular-springs model in (4.33), that is the torque residual $\|\mathbf{J}'(\Phi^t)\underline{f} - \mathbf{K}_s\Phi^t\|$. For all the simulations in this chapter, the terminal residuals of the three models were sufficiently small indicating model convergence to equilibrium.

Figure 4.6 shows the convergence during a sample planar deflection with a tip force applied in $-y$ direction. The sum of angles γ is plotted in Fig. 4.6(a). Note that for this planar bending, this sum is equal to the deflection angle of the needle tip with respect to the needle base frame lying along x axis. The torque residual during this simulation is also plotted in Fig. 4.6(b).

4.3 Experiment

In order to show the feasibility of the models and compare their accuracy, the following experiments were conducted using an 18 gauge 20 cm Bard[®]BrachyStar[®] needle (C. R. Bard, Inc., Covington, GA), that is used in prostate brachytherapy seed-implant procedures. In these experiments, the needle was bent under several known forces and its bent shaft form is recorded for evaluating our model simulations.

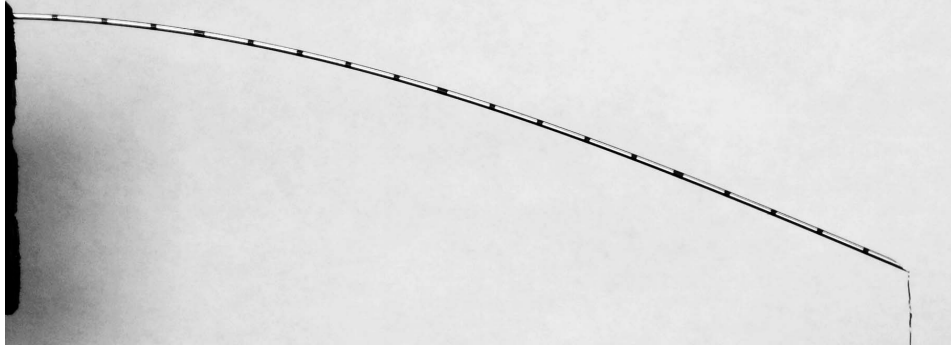


Figure 4.7: The needle shaft as it is clamped at its base and a vertical force is applied at its tip.

During the experiments, the needle was clamped at its base while its shaft lay horizontal to the ground. Due to the clamping mechanism, the effective bending shaft-length of the needle was reduced to 18.7 cm. Subsequently, several vertical forces were applied to its tip using combinations of scaling weights. The weights were placed on a stage hanging on a tiny hook bent at the tip in order to achieve a perpendicular force to the needle base at all times. The stage setup for hanging weighs 1.3 g and the weights applied were 5, 10, 15, 20, 25, 30, and 40 g.

For each force applied, the needle was imaged as in Fig. 4.7 over a white background using a digital camera, the shutter of which was controlled via computer. Ordinary brachytherapy needles are marked with black stripes at every 1 cm. These markers were located on each image to determine the bent configuration of the needle. Needle-tip lateral deflections from the nominal axis, measured using this procedure, are shown in Table 4.1.

Table 4.1: Tip deflection for various lateral tip forces.

Force [mN]	63	113	163	213	263	313	413
Deflection [mm]	8.3	14.5	20.9	27.0	33.0	39.0	50.2

To minimize the effect of lens warping on the data, the needle was centered close to the image center and a far camera focal distance was used. By imaging straight lines, this effect was found to be negligible in the given setup. In a calibration step using a ruler, the image resolution on the needle plane was measured to be better than 9 pixels/mm.

The observations of shaft bending in these experiments were next used to (i) identify the model parameters for the given brachytherapy needle, and (ii) evaluate the accuracy of each model by comparing the simulated bent needles to the experimental data.

4.4 Results

In the experiments, the needle deformations were all planar. Therefore, the model equations were reduced to their 2D equivalents. The equations for a 2D nonlinear beam element are obtained by simply removing the rows and columns of the stiffness matrix in (4.16) that correspond to v , ϕ and α in the vector of nodal variables (reducing \mathbf{K} to its parts \mathbf{K}^{11} , \mathbf{K}^{13} ,

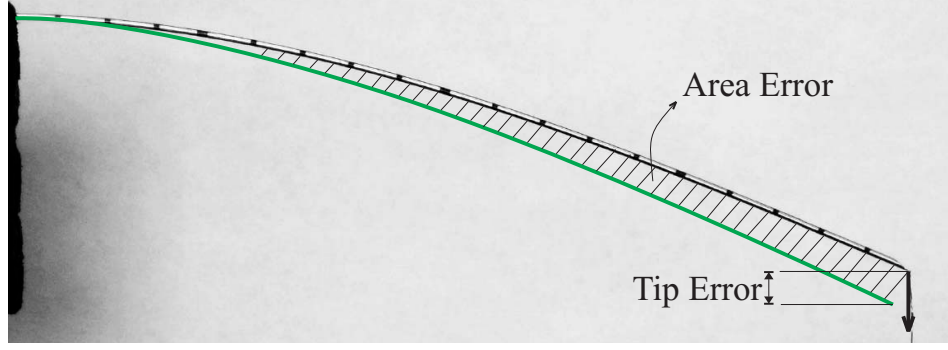


Figure 4.8: The definitions of the tip error and the area error between the simulated model and the observed needle postures.

\mathbf{K}^{31} , and \mathbf{K}^{33} in Appendix D). The angular spring model can be used in 2D simply by neglecting the torsion. The FEM with tetrahedral elements reduces to triangular elements in 2D. A 2D plane-stress analysis was then performed by also taking the out-of-plane thickness of the elements into account. The Poisson's ratio was taken to be 0.3, which is the Poisson's ratio of steel. Therefore, the model parameters to be identified are the Young's modulus E in the two FEM-based models and the bending spring constant k_b in the angular spring model. Equal-length segments were chosen so that the needle bending is described by a single parameter in each model.

4.4.1 Parameter Identification

For parameter identification, two independent goals were specified: (i) a best fit to the experimental tip position, and (ii) a best fit of the entire needle shaft. Accordingly, two cost functions relating an actual bending to a simulated one were defined as seen in Fig. 4.8: (i) a lateral tip position error, that is, the vertical difference between the simulated and the measured tip position, and (ii) the area lying between the simulated and the observed needle shafts. These goals, namely the *tip fit* and the *area fit*, were formulated as two separate optimization problems. Subsequently, the model parameters were found for each of them, independently, using the Nelder-Mead simplex search method [27].

In the first phase the model parameters were identified for different numbers of elements using each pair of force-displacement data. The effect of the number of elements on the identified parameters and the accuracy of the simulation were studied using models with 10 and 20 segments. In the triangular FEM model, meshes with 11×2 and 21×2 nodes were employed corresponding to 10 and 20 segments, respectively. The 11×2 triangular mesh employed is seen in Fig. 4.9, where the shaft thickness was set to that of a brachytherapy needle.

Figure 4.10 shows the range of the identified Young's moduli and the identified angular spring constants over the range of applied forces for different numbers of segments. In this figure, the identified parameters for both optimization goals are presented as box-plots displaying the distribution of parameters for varying tip loads.

Figure 4.10(a) presents the identified bending spring constants for 10, 20, and 50 segments. Similarly, Figure 4.10(b) presents the identified Young's moduli for 10 and 20 triangular elements. As seen in the plots, Young's modulus for triangular elements and spring constants for

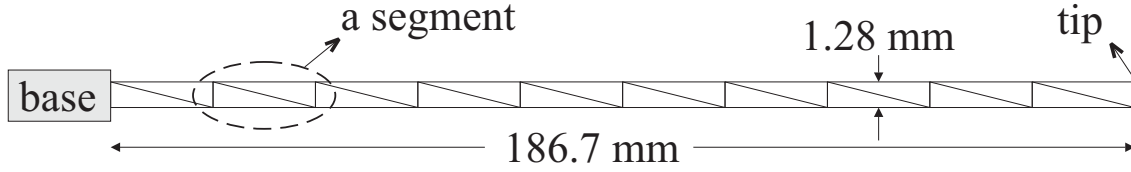


Figure 4.9: Needle discretized by triangular elements on a 11×2 mesh.

angular springs are highly dependent on the number of segments. This is due to the nature of these models where adding segments increases the shaft flexibility since the deflections of each segment add up. As opposed to these two models above, the beam model is not sensitive to the number of segments. Consequently, the Young's moduli identified for both 10 and 20 segments are the same and therefore they are presented using a single plot in Fig. 4.10(c).

Note that, for a given number of segments and optimization goal (e.g., the tip fit), the variation of the identified parameters with the changing force is relatively small in all three models. This slight variation is observed to be mainly due to experimental errors such as the resolution of the imaging system and the localization of the black stripes on the shaft.

4.4.2 Performance in Modeling Needle Deflection

During a needle insertion simulation, the forces are not known a priori. Thus, the goal of a needle deformation model is to successfully predict needle deflections for a wide range of possible loads using only a single set of fixed parameters. In this chapter, we use a single identified parameter for each model. These parameters, identified in the previous section, vary slightly with the applied force. Consequently, the effect of using a single fixed parameter for all applied loads is studied next. In total, six parameters have been identified for three different models with 10 and 20 segments each.

As seen in Fig. 4.10, the parameters identified using the tip fit and the area fit are close, thus either of these error measures could be used to determine a fixed parameter for each model. The mean of the parameter values identified using the tip fit method was chosen as the fixed parameter in simulating each model for evaluation.

Each fixed parameter was then used to simulate the needle deflection for all the seven different tip loads and these simulated deflections were compared with the experimental results. Figure 4.11 shows these simulated deflections for the three models with 20 segments along with the experimentally observed needle configurations. The tip and area errors for 10 and 20 segments using fixed parameters are presented in Fig. 4.12.

4.5 Discussion

Thanks to the fast computation of the presented models, and in particular, the angular springs model, they can easily be integrated into real-time medical training simulations systems. Due to their high accuracy, they can also be used in simulations for needle steering and path planning.

Computationally, the beam element model is more efficient than the triangular/tetrahedral element model. Note that a beam element and a triangular element both have 6×6 stiffness

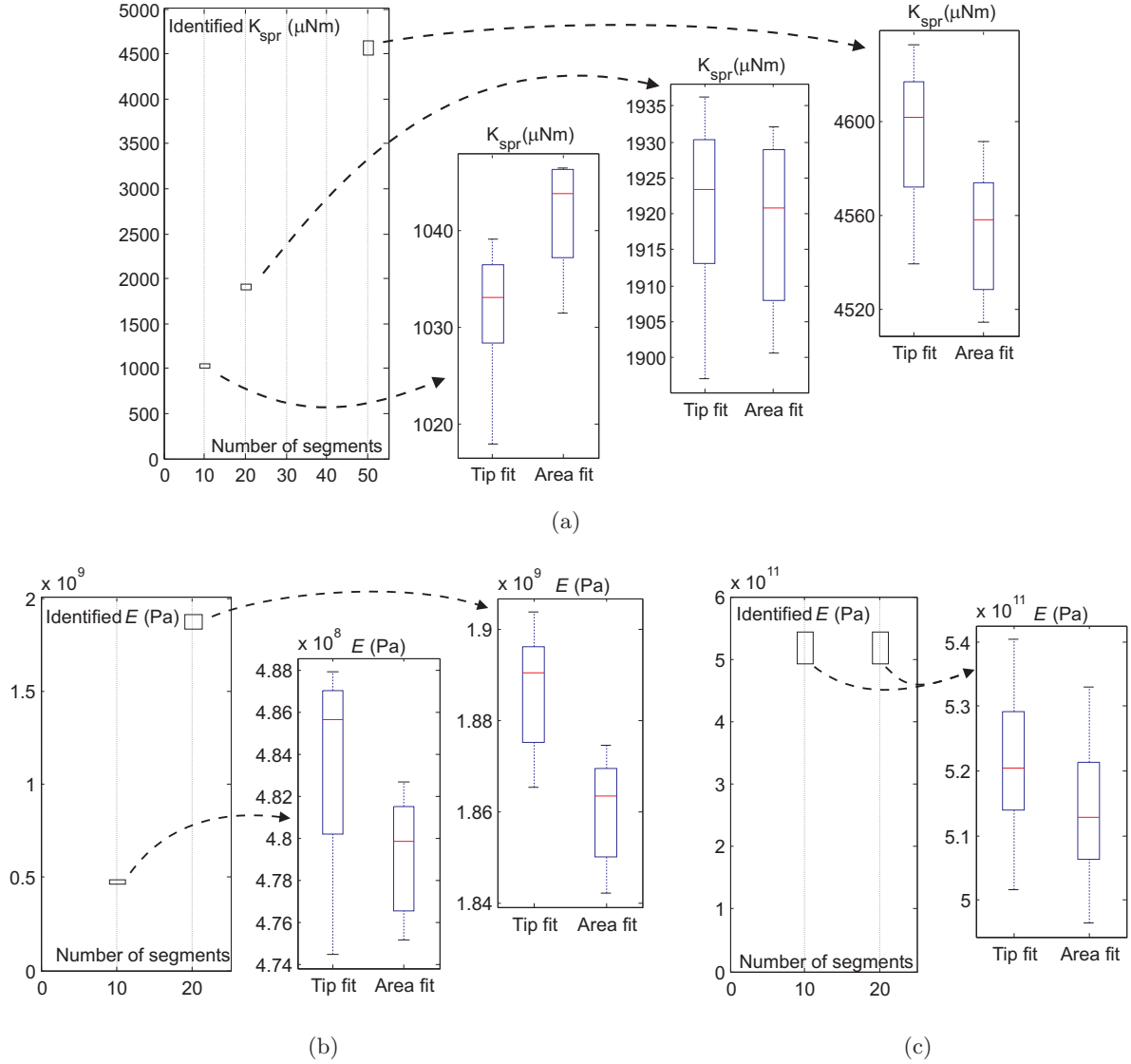
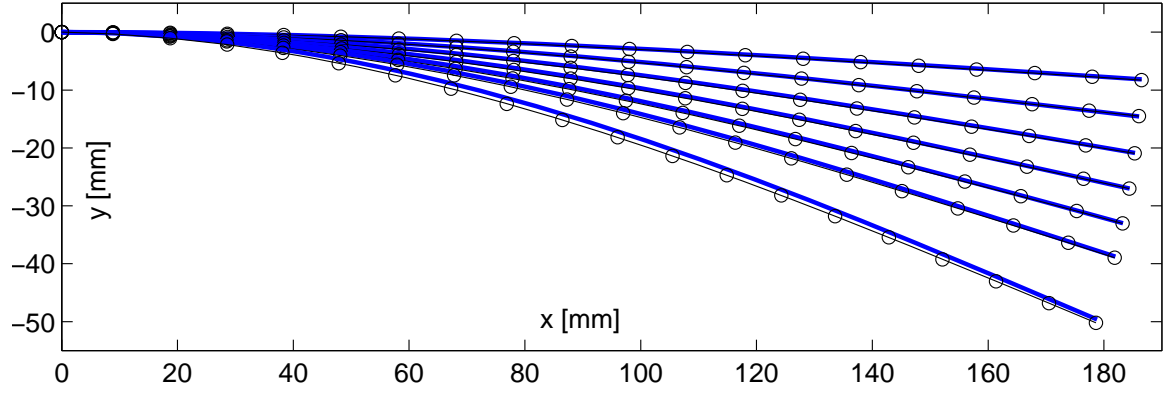
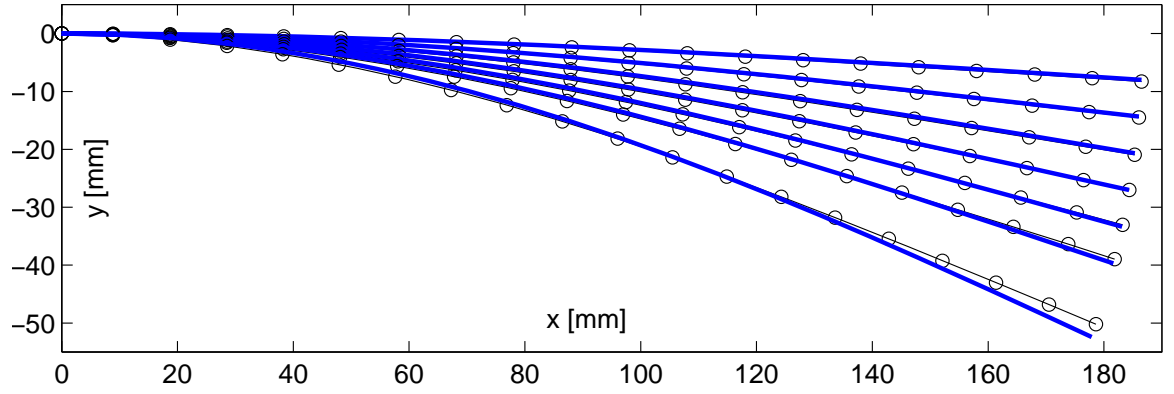


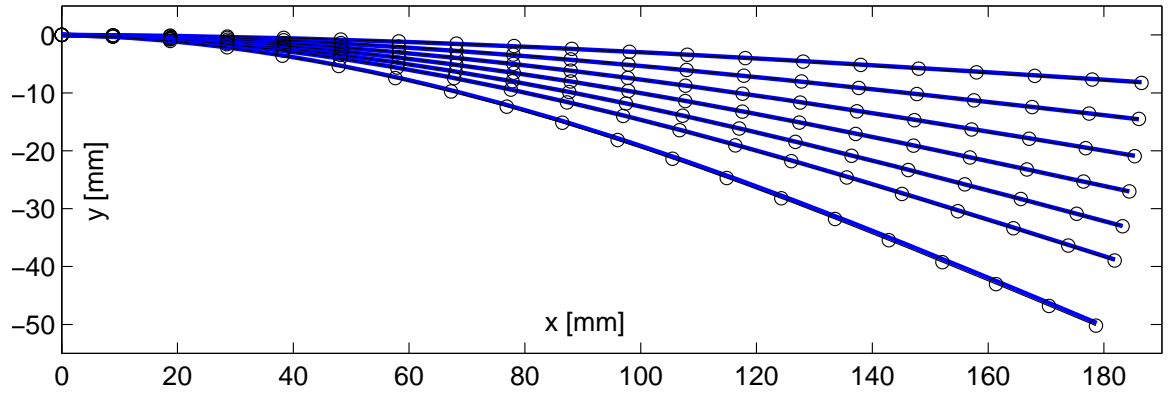
Figure 4.10: Distribution of identified Young's moduli and spring constants by minimizing tip and area errors for each bending experiment using a different load. Boxes extend from the lower quartile to the upper quartile value with the median marked. Whiskers show the full extent of data for models (a) 10/20/50 segment spring, (b) 10/20 segment triangular, and (c) 10/20 segment beam (having equal values). The extents of values are also shown in separate figures for each model to facilitate comparing them within full range of values.



(a) The triangular element model with a 21×2 mesh.



(b) The nonlinear beam model with 20 elements.



(c) The angular springs model with 20 segments.

Figure 4.11: The experimental deformed needle posture (circles) and the simulated needle (thicker solid lines) using the mean value of identified parameters.

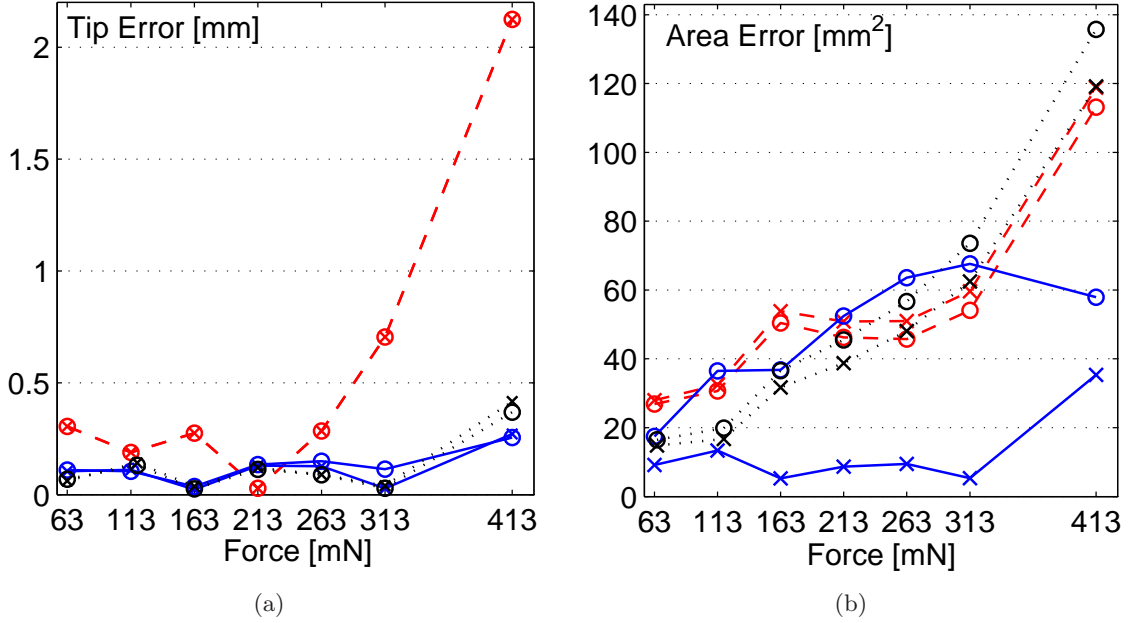


Figure 4.12: The tip (a) and area (b) error comparison of the three models (dotted-lines for the triangular FEM, dashed-lines for the beam FEM, and solid-lines for the spring model) simulated using the mean value of identified parameters with 10 and 20 segments (marked with circles and crosses, respectively).

matrices in 2D and, similarly in 3D, a tetrahedral element and a 3D beam element both have 12×12 stiffness matrices. However, modeling a needle using triangles/tetrahedra requires more elements than the beam element approach. For instance, if the needle is modeled in 2D using a mesh of $2n$ nodes (resulting in $n - 1$ segments), the tangent stiffness matrix will be $4n \times 4n$. In contrast, an equivalent beam model with $n - 1$ elements will result in a $3n \times 3n$ stiffness matrix.

Extension to 3D escalates this computational discrepancy, since a significantly larger number of elements are required for tetrahedral FEM to model the same number of segments in 3D, whereas the beam model and, similarly, the angular springs model use the same number of nodes and elements both in 2D and 3D. Furthermore, achieving an accurate and symmetric 3D model requires many tetrahedral elements [4], whereas the beam and angular springs models are inherently symmetric. Also note that, assuming the needle twist to be negligible facilitates the beam-element and the angular-springs models since the dimensions of the beam-element stiffness matrix can be reduced as shown in Appendix D and, in the angular springs case, skipping the computations involving the twist angle accelerates the convergence of the system in (4.33) by decreasing the number of unknowns. However, the tetrahedral model does not benefit from this simplification.

In our simulations, the beam-element and the angular-springs models often converged in fewer iterations than the triangular element model. Not only is the angular springs model robust requiring at most 6 iterations in our simulations, but its each iteration is also computed very fast since it consists of forward vector-algebraic equations with few trigonometric

functions, that can be implemented using look-up tables, if necessary. Indeed, during a coupled tissue-needle simulation, the forces on the needle change smoothly. Therefore, an initial needle-posture guess from the previous time instant is often close to its final solution and thus generally only a single iteration is required to find the new equilibrium state.

There are alternative methods to Picard's method that was used to solve (4.34) for the angular springs model. In particular, a physically-based method based on potential energy could be developed using (4.25) as a starting model. The equilibrium configuration of the system would satisfy (4.33), with guaranteed convergence.

The identified Young's modulus did not change with the number of elements in the beam element model. Therefore, the beam element model can be used in an adaptive deformation simulation scheme. In this scheme, the simulation starts with a minimum number of elements and, as the needle perforates the tissue and interacts with more tissue, the number of elements is increased (as done in [7]).

Note that the given angular springs model can be simplified to neglect the twist along the shaft by ignoring/zeroing the angles α_i . This is a valid assumption for many needles since the twist is often negligible due to the high shear modulus of the needle shaft and/or small twisting torques acting on it.

The angular springs model as presented deals with force loads only. Nevertheless, torque loads can easily be integrated in the formulation by extending the Jacobian definition in (4.31) as follows:

$$\underline{\tau} = \mathbf{J}' \begin{bmatrix} \underline{f} \\ \underline{g} \end{bmatrix} \quad (4.36)$$

where \mathbf{J}_i is now a $6 \times 3n$ Jacobian matrix and \underline{g} is the vector of torque loads at joints.

Although in our formulation of the angular springs model the force loads were considered only at nodes/joints, in practice, forces at any location on the model can be accommodated, since their corresponding moment arms can be calculated regardless of them being applied at a joint or not.

In our experiments, a single tip force was applied on the needle. However, soft tissue applies a distribution of force along the needle shaft during a needle insertion. In order to employ the presented discrete models, these forces need to be integrated on the discrete needle node locations first. Such integration is common in the FEM literature. Then, these multiple discretized forces can be applied on the discrete needle model. Both FEM models and angular springs are known to perform stably with multiple loads applied [20, 23]. Overall, the coupled needle-tissue system is solved in an iterative scheme, which therefore takes the insertion history into account. Such a coupled system was presented in [4].

The experiments were conducted with the needle stylet inside the needle cannula. During brachytherapy procedures, there are often at most 5-6 seeds loaded in each needle at 1 cm intervals. Consequently, the stylet, which is inserted in the cannula up to the last loaded seed, covers at least 14-15 cm of the 20 cm cannula. Therefore, in our simulator presented later in Chapter 5, the identified angular spring parameter with the stylet is used for the entire needle. Alternatively, the model parameters can be identified through experiments for a needle with and without the stylet. Subsequently, for a simulation where the stylet is partially inside the cannula, the corresponding parameter can be used for the sections of the needle with and without stylet.

In this thesis, equal length segments were used in the angular springs model that led to

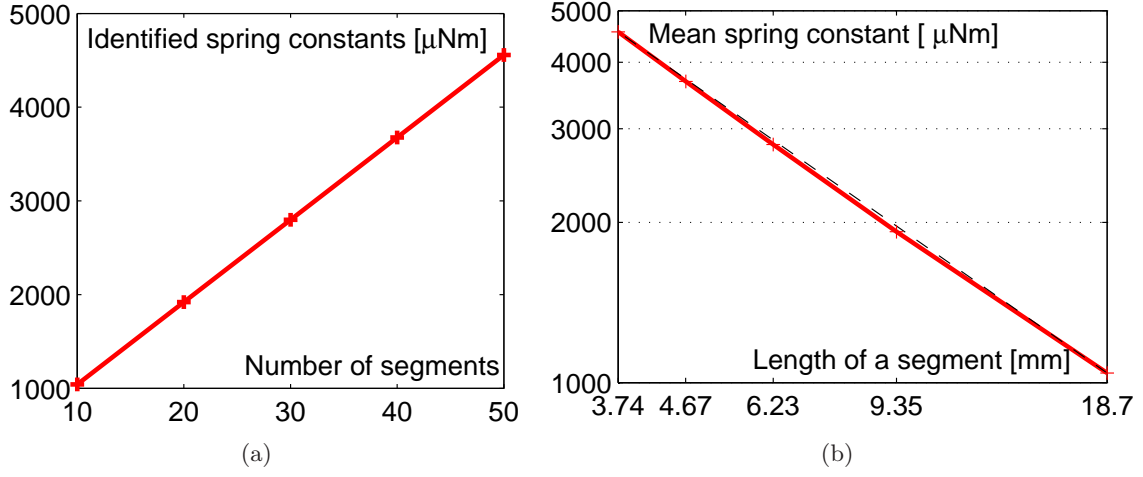


Figure 4.13: (a) Mean angular spring constants identified using 10, 20, 30, 40, and 50 segments and (b) a log-log plot of mean spring constants as a function of segment length (dashed-line is a line fitted to the first and the last values).

equal spring constants for all the joints. It should be noted that both bending and twisting spring constants of a segment are defined for a specific segment length and they are both inversely proportional to this length as formulated in appendices E and F. This relation between segment length and the bending spring constant was further studied (Section 4.5.1) to devise a spring-constant approximation scheme for any given length. It was confirmed that the bending spring constant is inversely proportional to the segment length.

4.5.1 Approximating the angular spring constants

It is not always possible to determine the best segment length to use in a needle simulation at the time the bending model is developed. For a specific simulation, a finer or coarser model than the one for which parameters were initially identified may be required. For each segment length, running simulations to identify individual parameters may not be feasible. Furthermore, segment lengths may need to be adjusted online during a simulation. Therefore, identifying a relation between the segment length and the spring parameter is beneficial. For this, the angular springs model, being the most accurate in modeling needle bending and the fastest to simulate, is studied next.

The angular spring constants identified using 10, 20, 30, 40, and 50 segments are plotted in Fig. 4.13(a). In this figure, only the mean value of constants identified through different loading experiments are shown since the other values lie within 1% range as also observed in Fig. 4.10(c-e) for the case of 10, 20, and 50, respectively. It is seen that the mean identified spring constant has a linear relation with the number of segments, which is inversely proportional to the segment length. This relation between the spring constant and the segment length is indeed expected from (E.3).

The change of this mean spring constant with segment length is presented in Fig. 4.13(b) in a log-log scale. Recall that the effective bending needle shaft is 186.7 mm. As seen in this figure, the angular spring parameter for a given segment length can be interpolated easily from

the known values. For instance, if the parameters for 20, 30, or 40 segments are identified using a line fit (dashed line in Fig. 4.13(b)) to the values of 10 and 50 segments, the interpolated values fall within 2.5% of the spring parameters that were actually identified through individual simulations.

4.6 Conclusions

Three different models were presented to simulate the deformations of a needle. The first two models use tetrahedral and beam elements, while the third model uses rigid bars connected through angular springs. All the models can preserve the needle length during moderately large deformations. The efficacy of the models in simulations of needle bending was evaluated through experiments during which several lateral forces were applied to a brachytherapy needle and the resulting deformations were recorded. The model parameters —Young’s modulus for the FEM-based models and spring constants for the angular springs model— were identified to fit each model to these experimental data. The lateral tip error and the area error were independently minimized to find the parameters defining each model. Later, one single parameter value was extracted for each model and the needle deformations simulated using these values were compared to the experimental deformations. A scheme for interpolating spring constants from values identified through simulations is also presented.

The angular springs model demonstrated the highest accuracy compared to the other two models. Furthermore, this method is computationally the most efficient and it is the simplest to implement. It was shown that the Young’s modulus in the tetrahedral/triangular model is significantly dependent on the number of elements. However, the same parameter is independent of the segment length in the beam element. This property can be used in adaptive simulation to increase the speed.

The models are capable of simulating the needle twist. However, twist is often negligible in medical procedures and it is not validated through our experimental setup.

References

- [1] R. Alterovitz, J. Pouliot, R. Taschereau, I. C. Hsu, and K. Goldberg, “Needle insertion and radioactive seed implantation in human tissue: Simulation and sensitivity analysis,” in *IEEE Int Conf on Robotics and Automation (ICRA)*, vol. 2, 2003, pp. 1793–1799.
- [2] S. P. DiMaio and S. E. Salcudean, “Needle insertion modeling and simulation,” *IEEE Transactions on Robotics and Automation*, vol. 19, no. 5, pp. 864–875, 2003.
- [3] R. J. I. Webster, J. S. Kim, N. J. Cowan, G. S. Chirikjian, and A. M. Okamura, “Non-holonomic modeling of needle steering,” *Int J of Robotics Research*, vol. 25, pp. 509–525, 2006.
- [4] O. Goksel, S. E. Salcudean, and S. P. DiMaio, “3D simulation of needle-tissue interaction with application to prostate brachytherapy,” *Computer Aided Surgery*, vol. 11, no. 2006, pp. 279–288, 2006.
- [5] E. Dehghan, X. Wen, R. Zahiri-Azar, M. Marchal, and S. E. Salcudean, “Needle-tissue interaction modeling using ultrasound-based motion estimation: Phantom study,” *Computer Aided Surgery*, vol. 13, no. 5, pp. 265–280, 2008.
- [6] E. Dehghan and S. E. Salcudean, “Needle insertion parameter optimization for brachytherapy,” *IEEE Transactions on Robotics*, vol. 25, no. 2, pp. 303–315, 2009.
- [7] D. Glozman and M. Shoham, “Image-guided robotic flexible needle steering,” *IEEE Transactions on Robotics*, vol. 23, pp. 459–466, 2007.
- [8] A. Okamura, C. Simone, and M. O’Leary, “Force modeling for needle insertion into soft tissue,” *IEEE Transactions on Biomedical Engineering*, vol. 51, pp. 1707–1716, 2004.
- [9] J. R. Crouch, C. M. Schneider, J. Wainer, and A. M. Okamura, “A velocity-dependent model for needle insertion in soft tissue,” in *Medical Image Computing and Computer Assisted Intervention*, 2005, pp. 624–632.
- [10] E. Dehghan, O. Goksel, and S. E. Salcudean, “A comparison of needle bending models,” in *Medical Image Computing and Computer Assisted Intervention*, 2006, pp. 305–312.
- [11] R. Alterovitz, M. Branicky, and K. Goldberg, “Motion planning under uncertainty for image-guided medical needle steering,” *Int J of Robotics Research*, vol. 27, no. 11-12, pp. 1361–1374, 2008.
- [12] R. Alterovitz, A. Lim, K. Goldberg, G. S. Chirikjian, and A. M. Okamura, “Steering flexible needles under Markov motion uncertainty,” in *Int Conf on Intelligent Robots and Systems*, 2005, pp. 120–125.

- [13] W. Park, J. S. Kim, Y. Zhou, N. J. Cowan, A. M. Okamura, and G. S. Chirikjian, "Diffusion-based motion planning for a nonholonomic flexible needle model," in *IEEE Int Conf on Robotics and Automation (ICRA)*, 2005, pp. 4600–4605.
- [14] S. P. DiMaio, "Modelling, simulation and planning of needle motion in soft tissues," Ph.D. dissertation, Univ. of British Columbia, 2003.
- [15] K. G. Yan, T. Podder, D. Xiao, Y. Yu, T.-I. Liu, C. Cheng, and W. S. Ng, "An improved needle steering model with online parameter estimator," *Int J of Computer Assisted Radiology and Surgery*, vol. 1, pp. 205–212, 2006.
- [16] N. Abolhassani, R. V. Patel, and A. Farzam, "Minimization of needle deflection in robot-assisted percutaneous therapy," *Medical Robotics and Computer Assisted Surgery*, vol. 3, no. 2, pp. 140–148, 2007.
- [17] H. Kataoka, T. Washio, M. Audette, and K. Mizuhara, "A model for relation between needle deflection, force, and thickness on needle penetration," in *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2001, pp. 966–974.
- [18] A. H. Slocum, *Precision Machine Design*. SME, 1992.
- [19] X. Huang, T. X. Yu, G. Lu, and H. Lippmann, "Large deflection of elastoplastic beams with prescribed moving and rotating ends," *Proceedings of the Institution of Mechanical Engineers, Part C: J of Mech Eng Sci*, vol. 217, no. 9, pp. 1001–1013, 2003.
- [20] K. Anjyo, Y. Usami, and T. Kurihara, "Simple method for extracting the natural beauty of hair," *ACM Computer Graphics*, vol. 26, no. 2, pp. 111–120, 1992.
- [21] E. Anshelevich, S. Owens, F. Lamiraux, and L. E. Kavraki, "Deformable volumes in path planning applications," in *IEEE Int Conf on Robotics and Automation (ICRA)*, 2000, pp. 2290–2295.
- [22] M. Bro-Nielsen and S. Cotin, "Real-time volumetric deformable models for surgery simulation using finite elements and condensation," *Computer Graphics Forum (Proc of Eurographics'96)*, vol. 15, no. 3, pp. 57–66, 1996.
- [23] O. C. Zienkiewicz and R. L. Taylor, *The Finite Element Method*. Butterworth-Heinemann, 2000.
- [24] J. N. Reddy, *An Introduction to Nonlinear Finite Element Analysis*. Oxford University Press, 2004.
- [25] I. Sharf, "Geometrically non-linear beam element for dynamics simulation of multibody systems," *Int J of Numerical Methods in Engineering*, vol. 39, pp. 763–786, 1996.
- [26] L. Sciavicco and B. Siciliano, *Modeling and Control of Robot Manipulators*, 2nd ed. Springer, 2000.
- [27] J. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the nelder-mead simplex method in low dimensions," *SIAM Journal of Optimization*, vol. 9, no. 1, pp. 112–147, 1998.

Chapter 5

Haptic Simulator for Prostate Brachytherapy with Simulated Ultrasound ⁶

5.1 Introduction

Low-dose-rate *brachytherapy* is the permanent implantation of radioactive seeds in tissue and it is the treatment of choice for early-stage locally-confined prostate cancer. In brachytherapy, the seeds are delivered using long, flexible needles inserted through a *template grid* according to a pre-plan as seen in Fig. 5.1(a). Similar to most minimally-invasive procedures, brachytherapy lacks direct visual feedback and thus the seeds are implanted under medical image guidance. A transrectal ultrasound (TRUS) probe mounted on a translation stage is used by the physician to image the anatomy and the needles. Figure 5.1(b) shows a photography of the probe and the stage. During the procedure, a knob on the side of this stage is used to translate it to image the anatomy at different depths. A sample ultrasound image from the procedure can be seen in Fig. 5.1(c).

Due to the lack of direct visual feedback, the performing physician has to rely on the TRUS images and the kinesthetic feedback on the needle in order to assess the needle path (and hence the positions of the seeds to be implanted) with the respect to the current anatomy. The prostate, the pathology, the seeds, and the needle are not always clearly visible in TRUS images. Furthermore, the procedure must be completed in a relatively short period of time to minimize edema and anesthesia side-effects.

Errors in seed placement resulting in an undesired dose distribution at the target volume decrease the effectiveness of the treatment and may lead to complications such as incontinence and impotence [1]. Furthermore, errors caused by prostate deformation and motion due to needle insertion may necessitate needle re-insertions, which in turn increase the procedure time and the risk of complications, also causing significant edema. For developing these required skills, the conventional training of a medical resident involves observing expert physicians perform the procedure and using mannequins, animals, or cadavers to practice, which either do not offer very realistic experiences or may involve ethical issues. In this chapter, an alternative method of training via a computational brachytherapy simulator system is proposed to aid medical residents in gaining hands-on experience on various aspects of this procedure, such as the relation of ultrasound images and the anatomy, and the effects of tissue deformation. Such

⁶A version of this chapter has been peer-reviewed and accepted for publication in the proceedings of the 2010 International Symposium in Biomedical Imaging (ISBMS). O. Goksel and S.E. Salcudean, “Haptic Simulator for Prostate Brachytherapy with Simulated Ultrasound”. A version of this chapter will also be submitted for publication.

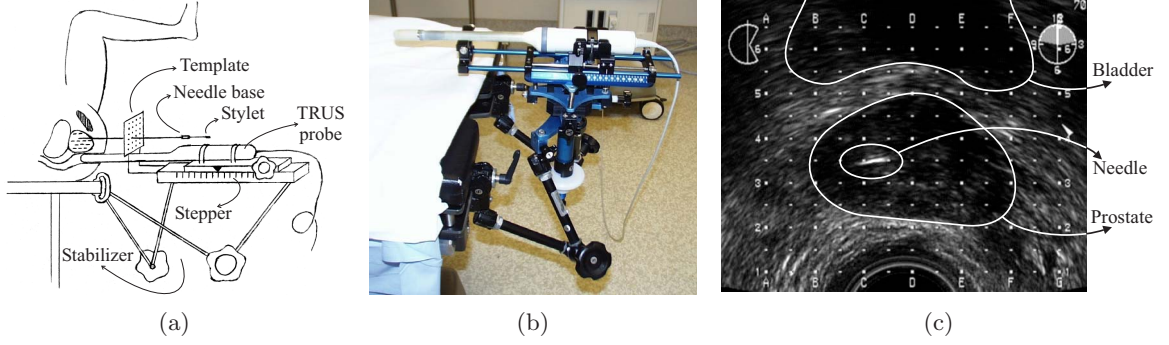


Figure 5.1: (a) The brachytherapy procedure setup, (b) a picture of the probe and its stage, (c) and an intra-operative ultrasound image.

simulation of this procedure is also important in accurate planning of the seed locations, where the anticipated (*a priori* simulated) tissue deformation of needle insertions can be compensated for in the planning.

During needle insertion in the operating room, physicians have to rely on TRUS images along with haptic feedback on the needle. Therefore, a training or rehearsal simulator requires realistic rendering of both images and haptics, while also simulating tissue deformation. Furthermore, for a rehearsal or planning application, an accurate patient-specific model is essential since the tissue mesh and surrounding boundary constraints play a significant role on the outcome of the procedure [2].

The simulation presented in this chapter runs on a computer system and mimics the images and the deformation of tissue in response to the needle being inserted. The user holds and moves a physical (haptic) device in order to insert/retract the needle, while kinesthetic feedback computed in the simulation is applied on the user's hand by the haptic device. At the same time, simulated TRUS images are presented to the user on the screen.

This chapter addresses the integration of the following components of the proposed simulator: a needle model, a tissue model with deformation, needle-tissue interaction, image simulation, haptic feedback, and generation of patient models. The flexible needle model presented in Chapter 4 is employed here. The tissue deformation and needle interaction were adopted from our earlier work [3]. The image simulation method is adapted from Chapter 3 with the addition of needle appearance simulation proposed in [4]. Haptic feedback and a patient model generation framework are introduced in this chapter, while also addressing the real-time system integration and user interface aspects. The application of the variational meshing technique in Chapter 2 is also discussed and illustrated for prostate model generation in this chapter.

Ultrasound is a real-time, cost-effective, non-ionizing, and hence widely-used medical imaging technique. For simulation of this real-time modality, several *generative* methods have been proposed. However, even the most sophisticated generative methods do not capture the realism and complexity of actual B-mode images. Chapter 3 presents a method of employing actual B-mode volumes to synthesize deformation-coupled images using an *interpolative* scheme in the mesh [5]. This method, which was also validated through phantom studies, is adopted and integrated into the brachytherapy simulation in this chapter.

Several studies on modeling tool interaction with deformable tissue exist in the literature.

See [6] for a review. A subcategory of these methods motivated by training, especially laparoscopic procedure simulators, target graphical rendering of simulated camera images as visual feedback. However, for brachytherapy, a real-time simulation of TRUS images within deformed tissue volume is required, where our method becomes essential. Brachytherapy has received significant attention from the research community targeting tissue deformation and needle interaction simulations [3, 7]. However, haptic and ultrasound feedback, which are both essential in a training setup, have not been provided in any previous work.

5.2 Methods

5.2.1 Brachytherapy Simulation Components

The components described below are illustrated in Fig. 5.2. At the heart of the simulation lies the needle-tissue interaction model, which runs in a separate thread at high-priority. Tissue deformation is modeled using linear-strain FEM and is coupled to the needle shaft using a stick-slip friction model [3, 8, 9]. In this approach, the needle shaft poses a displacement constraint for the tissue nodes laterally so that these nodes can slide along the shaft, but cannot leave it. Friction forces that are typically caused on the needle shaft surface during such sliding are also applied on these tissue nodes [8]. These displacement and force constraints are then solved using a condensed version of the tissue stiffness matrix computed using FEM [10]. As the needle advances in the tissue, this coupled needle-tissue model is updated quasi-statically and new tissue nodes are added on the needle shaft, if required [3]. More details on this 3D needle-tissue interaction model can be found in [9] and [3].

A low-level haptic process running at 1 KHz reads the haptic device position to control the needle base location and provides force feedback computed in the simulation to the user's hand. This process also handles other low-level control tasks such as creating a lateral constraint on the needle to simulate the brachytherapy template grid through which the needles pass during the actual procedure. In this thesis, a low cost haptic device solution, Novint Falcon, was used to control the needle base location. This device allows for position input and force feedback in three translational axes within a workspace of $10 \times 10 \times 10 \text{ cm}^3$ and provides a software development platform in Windows.

The flexibility of brachytherapy needles is modeled similarly to the bending of a cylindrical rod. This model is discretized as a set of linear rod segments connected via torsional springs. These springs force the segments to align, which was shown to model shaft flexibility accurately in Chapter 4 [11]. As a result of the constraints that the needle imposes on tissue as described above, the tissue generates reaction forces (computed using the FEM). Subsequently, at a simulation time instant, using the needle posture and the forces exerted on the needle shaft, first the torques at each segment joint and then the corresponding spring deviations (hence a new needle posture) are found, therefore yielding a new bent needle posture at the end of the simulation iteration.

For imaging feedback, the interpolative ultrasound simulation technique described in Chapter 3 was used to display real-time B-mode TRUS images to the user [5]. These images are simulated using pre-operative voxel volume data that is deformed to match the location of the nodes computed by the FEM. The same tissue mesh used in needle interaction model above is used for the image deformation simulation as well. Consequently, at each image simulation

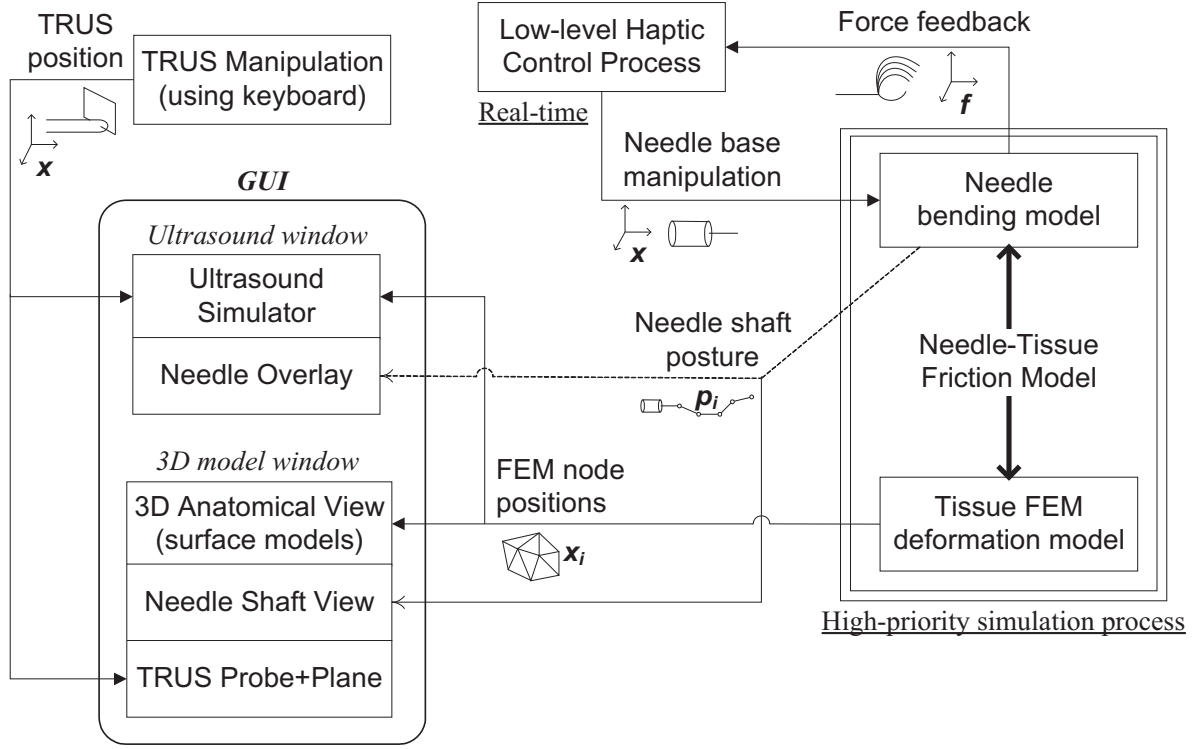


Figure 5.2: The online simulation components and their data interaction.

iteration, pixels of an image plane emanating from the TRUS probe into the deformed mesh are simulated. As described in Chapter 3, the plane intersection with the deformed mesh is marked on the pixels to find encapsulating mesh elements and, once the pixel locations are transformed to the nominal mesh frame, their intensities are interpolated using tri-linear interpolation. This image simulation task is run in a lower-priority thread at visual refresh rates. In this thesis, the translation of the virtual TRUS probe is controlled using the keyboard.

B-mode needle appearance is also synthesized using the needle-image simulation method of [4] on the presented ultrasound images. This technique employs a large set of needle images collected *a priori* in water. At simulation time, for a given relative arrangement of the probe with respect to a needle, the closest set of such *a priori* images are determined. Subsequently, an ultrasound needle appearance is interpolated using a combination of nearest-neighbour and tri-linear techniques in a tensor interpolation scheme. This simulated needle image is then overlaid/blended with the simulated ultrasound images at the location and orientation that it should appear in the image. The current image dataset includes needle tip and shaft images for transversal TRUS ultrasound probe at several depths and orientations, and can be extended to the B-mode simulation of brachytherapy seeds in the future [4].

For the overlay of needle images on the ultrasound images, the addition of pixel intensities was proposed in [4]. Although B-mode formation is indeed a result of very complex ultrasound interactions, such an intensity blending method was further shown to perform satisfactorily. Prior to blending, the needle images also need to be rotated and positioned over the location where they should appear [4]. In this thesis, the hardware acceleration of the graphics card is

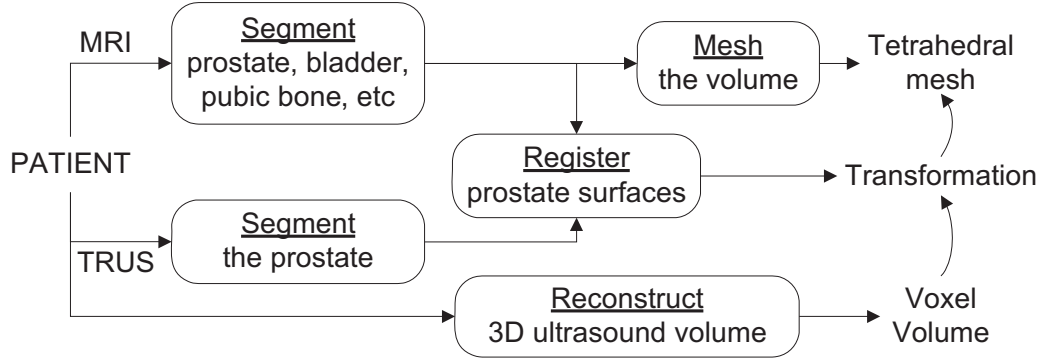


Figure 5.3: Patient-specific model generation procedure.

utilized for this task using OpenGL. At every display iteration, the simulated needle image is texture-mapped on a surface, which is then positioned over the desired region of the simulated B-mode image. The transparency of the needle texture is then used to obtain a desired blending effect, such as intensity addition. This approach achieves a very fast rotation and blending on the graphics hardware.

The proposed brachytherapy simulator displays a 3D view of the anatomical structures as well. In this display, the anatomical surfaces are rendered during the simulation so that their displacement in 3D can be observed relative to the needle and the TRUS probe. In medical training, the comprehension and association of observed ultrasound images within a 3D anatomical volume is one of the challenges frequently mentioned by radiation oncologists practicing the procedure. This 3D view aims to facilitate the learning of the 3D anatomical layout with respect to the TRUS probe and the needle. In certain other training scenarios such as evaluating a trainee, this view can simply be disabled using the interface.

5.2.2 Patient Model Generation

Each patient model is generated using two sets of imaging data: (i) pre-operative TRUS images and (ii) pre- or post-operative MRI of the prostate region. The TRUS images are used to reconstruct a 3D B-mode voxel volume for the on-the-fly ultrasound simulation. The MR volume is used to generate a mesh representation of the relevant anatomy. These two modalities are registered using the prostate surface. An outline of the model generation steps described above is seen in Fig. 5.3.

For 3D mesh generation for a given anatomy, anatomical surfaces extracted from contours segmented on image slices are used in a Delaunay-based meshing strategy, as described below in this section. The feasibility of using the energy-based meshing technique from Chapter 2 is studied later in Section 5.4.1. The use of prostate vibro-elastography imaging in this latter approach will be the ultimate modeling strategy for the brachytherapy simulator. However, the patient data used in this work, which contains TRUS B-mode voxel volume scans, does not include volumetric elastography images. Therefore, the MRI volumes and their segmentations, which were readily available to us from these same patients, are primarily employed in this chapter. In these MRI volumes, the pelvic bone and the tissue near the skin, which are typically not covered by a TRUS scan, are also seen in detail.

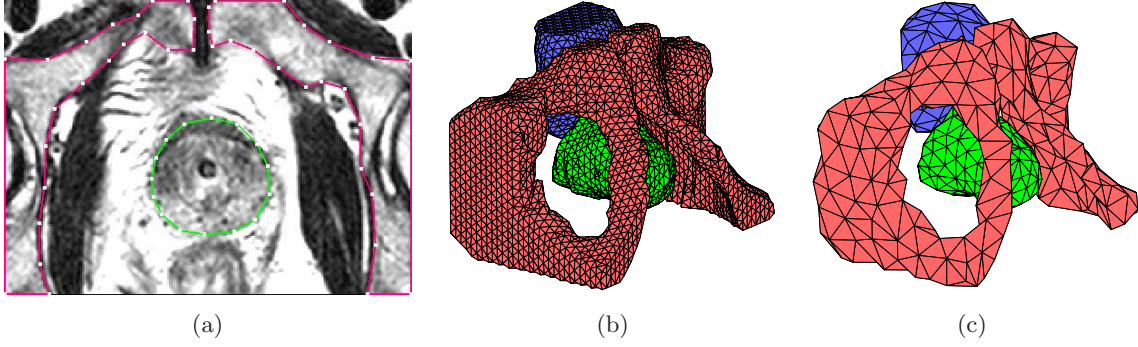


Figure 5.4: (a) A segmented MR slice, (b) the fine surface meshes generated from segmented contours, and (c) coarser surface models after farthest point sampling.

First, the prostate, the pelvic bone, and the bladder are segmented in MR images as seen in Fig. 5.4(a) using the Stradwin software, where the anatomy surfaces seen in Fig. 5.4(b) are interpolated from these cross-sections [12]. The tissue inside and around these surfaces are then meshed using the TetGen volume meshing tool [13]. Since most meshing applications taking explicit surface representation do not modify the given surface vertices, depending on the desired mesh resolution, these anatomical surface models are resampled at a lower resolution using *fast-marching geodesic farthest-point sampling* [14] prior to meshing. This reduced model as seen in Fig. 5.4(c) also accelerates the meshing process.

5.2.3 Mesh-Ultrasound Registration

Next, the prostate is segmented on TRUS images. Accordingly, the mesh generated above and seen in figures 5.5(a)–(b) is registered rigidly to the TRUS image volume using the prostate surface models from both imaging modalities. A variation of *iterative closest point* (ICP) method is used to match the point clouds of the prostate models. The registered surfaces are seen in Fig. 5.5(c). For the FEM deformation simulation, the pelvic bone is assumed as a fixed boundary constraint in the offline computation of an inverse stiffness matrix. This matrix and other required mesh vertex/connectivity information are saved *a priori* to be employed in real-time by the needle-interaction model. The ultrasound simulation also expects certain mesh-related data structures in order to accelerate its online processing [5]. Also, the needle-image simulation necessitates a set of needle images acquired *a priori* [4]. These are also prepared in advance and saved for online use.

5.2.4 Pre-computation of Data Structures

Certain data structures of the proposed system do not change during a simulation and/or their initial values require lengthy computations. Such parameters are computed in an offline step and saved to be later used in real-time simulation. For instance, the mesh faces corresponding to anatomical surfaces in the *3D anatomical view* are identified prior to the simulation. This allows for adjusting the parameters of the stick-slip interaction model on such surfaces to simulate different tissue characteristics, such as an increased tip-cutting threshold simulating the prostate capsule, that is typically harder to penetrate than the surrounding tissue. Such

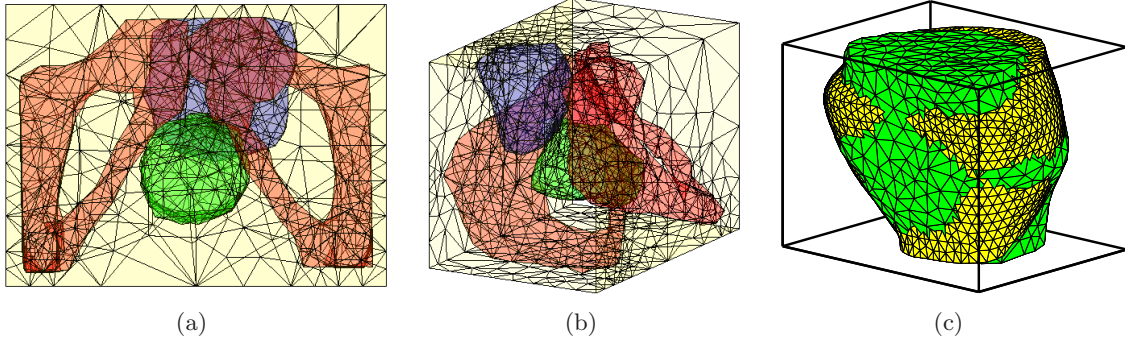


Figure 5.5: (a)-(b) The volume mesh generated from segmentation of MRI and (c) the rigidly registered prostate surface models from MRI and TRUS.

friction and cutting parameters of the interaction model are currently set using empirical values. A tissue stiffness matrix that is to be used in FEM is also computed offline from the mesh. Young's modulus and Poisson's ratio of different tissue layers are assigned from approximate values in the tissue biomechanics literature.

5.3 Results

Haptic interaction is currently implemented using a Novint Falcon device, that has three degrees-of-freedom for both position control and force feedback. Simulating the template grid that constraints the lateral motion of the needle, the device is fixed in those two axes using a PID controller once the insertion point on skin is selected in the ultrasound interface. Subsequently, the device controls the needle base in the insertion axis and provides feedback in the same axis during the simulation. In practice, the needle bending is relevant only beyond the *template grid*. This can be simulated by simply neglecting/zeroing the bending angles of the deformable needle model segments at and before the template location.

The corresponding components of the system run at real-time haptic and visual rates. Refresh rates of the needle interaction and ultrasound simulations can be set in the interface at run-time with typical values being 250 Hz and 10 Hz, respectively. The ultrasound image pixels that are mapped back from the immediate deformed mesh to the nominal time-zero mesh at each iteration are interpolated using tri-linear interpolation in that nominal image dataset. The image simulation takes a large portion of the processing time, therefore the interface including this ultrasound view is isolated in a separate thread running at a lower priority than the insertion model. For the simulation scenarios in which *remeshing* [3] is desired during haptic control, a pre-insertion is performed in the background once the insertion template location is chosen. This ensures that the nodes are created/moved on the needle path and a subsequent insertion with haptic feedback does not require any time-steps possibly exceeding this rate. The graphical interface to this simulation is seen in Fig. 5.6.

This simulation can also be used as part of a treatment planning system. For a given needle-base trajectory, which is read from a file or is commanded in the interface, the simulation performs the insertion and finds the resulting needle configuration in the tissue. This is presented in Fig. 5.7 to simulate a conventional brachytherapy needle insertion, where the

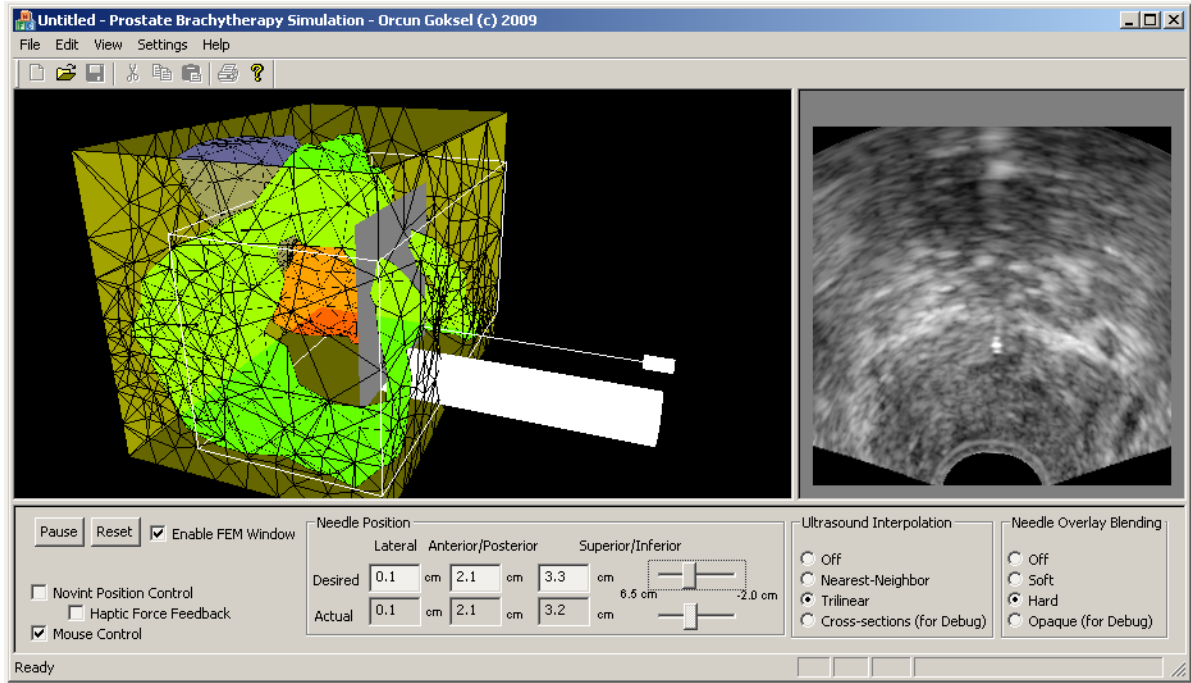


Figure 5.6: The graphical interface to the simulation displaying the anatomical surface models on the left and the simulated ultrasound window on the right.

needle tip is targeted at the *circled* 3D pre-plan seed location. In conventional brachytherapy, tissue deformation is not taken into account. Therefore, when the needle tip arrives at this 3D position in the nominal frame as shown in Fig. 5.7(a), the target is considered reached and a seed is implanted in the tissue. This seed shown with *plus* coincides with the nominal targeted location (*circle*) in this deformed configuration. However, it is actually subject to a change of position, once the needle is retracted and the tissue returns to its nominal configuration. The mismatch between the seed and its original target is seen in Fig. 5.7(c).

5.4 Discussion

5.4.1 Variational Meshing

This section demonstrates the use of the variational meshing approach presented earlier in Chapter 2 for the model generation for the brachytherapy simulator. This method will allow a user-input-free generation of patient-specific prostate models. These can be generated intra-operatively, enabling on-the-fly treatment planning/correction. The meshing technique from segmented surfaces in Section 5.2.2 requires user input and significant segmentation time, therefore is not applicable in such a scenario. Furthermore, it does not offer a guaranteed way of limiting the maximum number of vertices in the mesh. This vertex budget is enforced by the FEM simulation speed and assures the completion of a simulation in a reasonable time on a particular hardware. In our variational meshing approach, such a vertex budget can be given as an input, while a trade-off between capturing intensity changes or achieving higher-quality

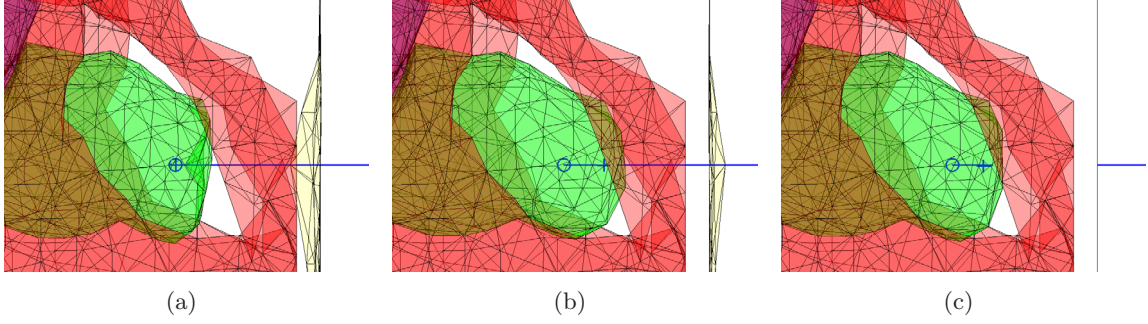


Figure 5.7: Simulation of the seed placement error due to tissue deformation. The circle denotes the pre-plan target, whereas the plus sign marks the actual implanted seed location (a) at the time of insertion, (b) during needle retraction, and (c) after the needle leaves the tissue.

elements can be set.

At the time of this simulator development, the patient data having other image modalities required by our simulation does not include volumetric elastography images. Therefore, our variational meshing method is demonstrated in this brachytherapy simulation context by employing a *labeled volume* as an intermediate image representation, similarly to various other medical meshing studies in the literature [15–18]. This labeled volume is a regular grid of voxels, where each voxel is assigned a value for being inside/outside a segmented anatomy. In our case, the surfaces segmented from MRI as described in Section 5.2.2 are voxelized for this purpose, similarly to the examples of [18].

The labeled volume is then fed to the image-based variational meshing approach, which replaces the tessellation step (“Mesh”) in the model generation diagram in Fig. 5.3. The resulting tessellation is seen in Fig. 5.8. In this figure, only the elements forming the prostate and the pelvic bone are shown using the *element thresholding* method described in Chapter 2. The isosurface of the labeled volume is also overlaid in these figures to show the agreement between the mesh and the image. This mesh can next be registered to TRUS similarly to the result of the TetGen meshing method above.

The meshing approach described in Section 5.2.2 has been chronologically developed first and used accordingly in our simulator prototype. Note that prostate vibro-elastography is work that was carried out in parallel to that in this thesis. Fine-tuning the parameters of that system to obtain reliable tissue elasticity is still in progress. Once such elastography images are available to us, the introduced variational meshing method will be applied without a labeled intermediate volume, as done in Chapter 2.

5.4.2 Brachytherapy Simulation

The needle targeting example presented above shows the capability of the system for predicting a seed implant location. The ultrasound images of such seeds can also be simulated and blended with images similarly to the needle image simulation. Furthermore, these predicted implant locations can be used in a needle path planning system using iterative optimization techniques as presented by [19].

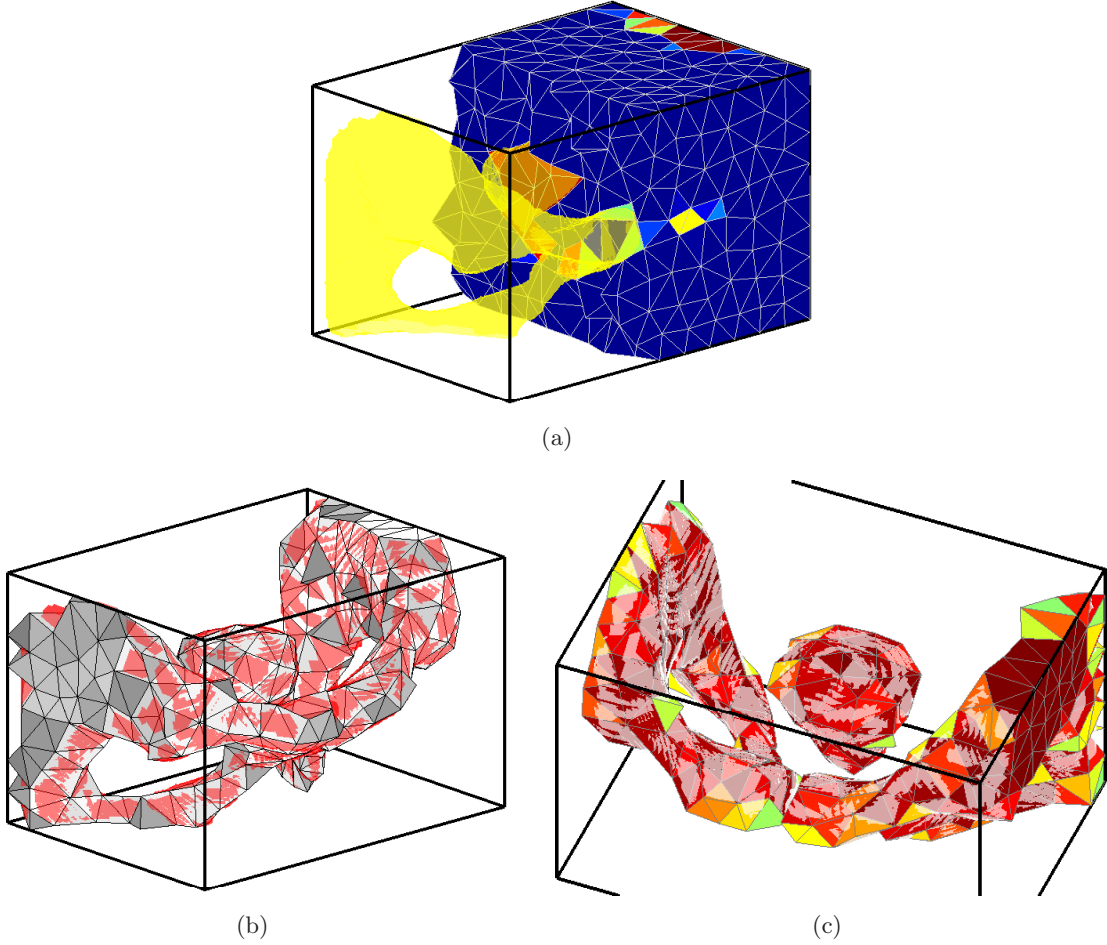


Figure 5.8: Variational meshing result of the prostate labeled volume with the labeled volume isosurface overlaid. A good agreement between the labeled volume boundary and our mesh is apparent from the isosurface closely following our thresholded mesh surface.

In the preliminary testing of the training system, the ultrasound simulation for brachytherapy with deformation was found to be very realistic-looking. The accuracy of this approach was already studied in phantoms in Chapter 3. Furthermore, the needle image simulation work of Zhu [4], that was incorporated in this brachytherapy simulation, generated realistic needle images in ultrasound. These findings were confirmed by a brachytherapy expert testing the system. Sample images of the system during the insertion of a needle can be seen in Fig. 5.9.

5.4.3 Deformation due to TRUS Probe

In the current simulation, the interaction between the probe and the tissue is not modeled. Indeed, the probe is moved minimally in transversal axis during needle insertions and any probe-tissue friction during the axial translation of the probe can be neglected considering the lubrication of the probe encapsulating balloon. Nonetheless, it is possible to model the probe-tissue interaction both in lateral and axial axes also using a stick-slip model similar to

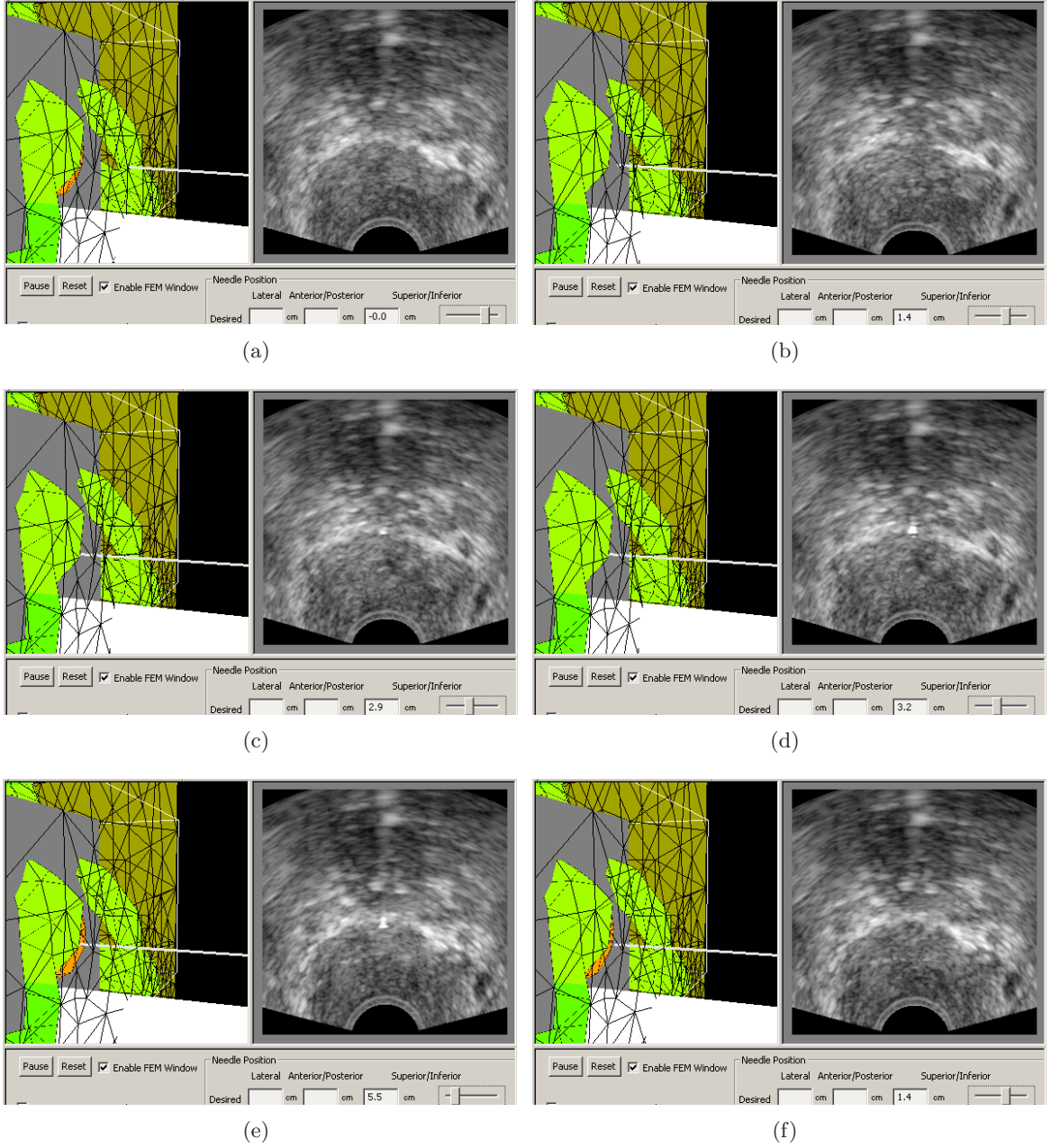


Figure 5.9: Sample images from the simulation of a brachytherapy needle insertion. The needle is kept rigid and the TRUS position (imaging depth) is not changed between these images. The ultrasound image is observed to change, reflecting the deformation of the tissue, also showing the needle when it intersects the imaging plane, during the insertion (a)-(d) and the retraction (e)-(f) of the needle. The ultrasound simulation and its realism are better appreciated with the animated images during the simulation, since the speckle pattern can then be observed to change during needle motion.

needle-tissue interaction.

The pre-operative MR and the intra-operative TRUS volumes are currently registered rigidly. This makes the assumption that the deformation of the prostate during the volume registration at the beginning of the brachytherapy procedure is minimal. In the future, these models and consequently the corresponding TRUS voxel volume can be registered using well-known deformable registration techniques. Either intensity-based techniques or surface-based methods, given that the pelvic bone is also (partially) segmented in TRUS images, can be followed. Note that, for an FEM-based registration approach, the mesh generated using elastography readily provides a good deformable model having its elements assigned with their relative elasticity values. Alternatively, the deformable model can be generated from the TRUS imaging, such as using elastography techniques, eliminating the need for registration.

5.4.4 Parameter Tuning and Validation

There exist methods in the literature for finding both the biomechanical properties of the prostate region [20] and the parameters of needle interaction [21,22]. Acquiring such simulation parameters non-invasively is still a challenging research problem. Swelling of the prostate was not incorporated in our simulation, but it may be modeled with the addition of a pressure component on the prostate model increasing with time or the number of needles inserted.

The presented ultrasound simulation technique is also of value for evaluating the accuracy of deformation simulation by comparing deformed ultrasound images to the ones collected intra-operatively with the needle inserted. Validation of the simulation is to be addressed in future work. Such simulated ultrasound images in comparison with actual intra-operative ones can further be used for optimizing tissue deformation and/or needle insertion parameters.

5.5 Conclusions

In this chapter, a novel needle insertion simulation with simulated ultrasound feedback is presented for the prostate brachytherapy procedure. This is the first real-time deformable tissue interaction model in the literature that also provides coupled realistic deformable ultrasound feedback. The visual realism of the simulated ultrasound is ensured intrinsically by the use of actual TRUS images as input. The needle in the developed simulation is controlled using one of various methods including a haptic device, allowing for utilization in training, rehearsal, or planning applications. A patient-specific model generation procedure for this simulation is also presented in this chapter. Parameter identification and the validation of the presented system will be studied next.

References

- [1] G. S. Merrick, W. M. Butler, K. E. Wallner, R. W. Galbreath, R. L. Anderson, B. S. Kurko, J. H. Lief, and Z. A. Allen, “Erectile function after prostate brachytherapy,” *Int J of Radiation Oncology Biology Physics*, vol. 62, no. 2, pp. 437–447, 2005.
- [2] S. Misra, K. J. Macura, K. T. Ramesh, and A. M. Okamura, “The importance of organ geometry and boundary constraints for planning of medical interventions,” *Medical Engineering and Physics*, vol. 31, no. 2, pp. 195–206, 2009.
- [3] O. Goksel, S. E. Salcudean, and S. P. DiMaio, “3D simulation of needle-tissue interaction with application to prostate brachytherapy,” *Computer Aided Surgery*, vol. 11, no. 2006, pp. 279–288, 2006.
- [4] M. Zhu, “Real-time B-mode ultrasound image simulation and artifacts modelling of needles and brachytherapy seeds,” Master’s thesis, University of British Columbia, 2009.
- [5] O. Goksel and S. E. Salcudean, “B-mode ultrasound image simulation in deformable 3-D medium,” *IEEE Transactions on Medical Imaging*, vol. 28, no. 11, pp. 1657–1669, 2009.
- [6] S. Misra, K. T. Ramesh, and A. M. Okamura, “Modeling of tool-tissue interactions for computer-based surgical simulation: A literature review,” *Presence: Teleoperators & Virtual Environments*, vol. 17, no. 5, pp. 463–491, 2008.
- [7] N. Chentanez, R. Alterovitz, D. Ritchie, L. Cho, K. K. Hauser, K. Goldberg, J. R. Shewchuk, and J. F. O’Brien, “Interactive simulation of surgical needle insertion and steering,” *ACM Transactions on Graphics (SIGGRAPH Proceedings)*, vol. 28, no. 3, pp. 1–10, 2009.
- [8] S. P. DiMaio and S. E. Salcudean, “Interactive simulation of needle insertion models,” *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 7, pp. 1167–1179, 2005.
- [9] O. Goksel, “Ultrasound image and 3D finite element based tissue deformation simulator for prostate brachytherapy,” Master’s thesis, University of British Columbia, 2004.
- [10] M. Bro-Nielsen and S. Cotin, “Real-time volumetric deformable models for surgery simulation using finite elements and condensation,” *Computer Graphics Forum (Proc of Eurographics’96)*, vol. 15, no. 3, pp. 57–66, 1996.
- [11] O. Goksel, E. Dehghan, and S. E. Salcudean, “Modeling and simulation of flexible needles,” *Medical Engineering and Physics*, vol. 31, no. 9, pp. 1069–1078, 2009.
- [12] G. Treece, R. Prager, A. Gee, and L. Berman, “Surface interpolation from sparse cross sections using region correspondence,” *IEEE Transactions on Medical Imaging*, vol. 19, no. 11, pp. 1106–1114, 2000.

- [13] H. Si, “TetGen, a quality tetrahedral mesh generator and three-dimensional delaunay triangulator, v1.3 user’s manual,” WIAS, Berlin, Germany, Tech. Rep. 9, 2004.
- [14] G. Peyré and L. D. Cohen, *Advances in Computational Vision and Medical Image Processing: Methods and Applications*. Springer, 2008, ch. Geodesic Methods for Shape and Surface Processing, pp. 29–56.
- [15] A. Mohamed and C. Davatzikos, “Finite element mesh generation and remeshing from segmented medical images,” in *IEEE International Symposium on Biomedical Imaging (ISBI): Nano to Macro*, 2004, pp. 420–423.
- [16] N. Archip, R. Rohling, V. Dessenne, P. Erard, and L. P. Nolte, “Anatomical structure modeling from medical images,” *Computer Methods and Programs in Biomedicine*, vol. 82, no. 3, pp. 203–215, 2006.
- [17] J.-P. Pons, F. Ségonne, J.-D. Boissonnat, L. Rineau, M. Yvinec, and R. Keriven, “High-quality consistent meshing of multi-label datasets,” in *Information Processing in Medical Imaging (IPMI)*, 2007, pp. 198–210.
- [18] J. Dardenne, S. Valette, N. Siauve, N. Burais, and R. Prost, “Variational tetrahedral mesh generation from discrete volume data,” *The Visual Computer: International Journal of Computer Graphics*, vol. 25, pp. 401–410, 2009.
- [19] E. Dehghan and S. E. Salcudean, “Needle insertion parameter optimization for brachytherapy,” *IEEE Transactions on Robotics*, vol. 25, no. 2, pp. 303–315, 2009.
- [20] S. Salcudean, D. French, S. Bachmann, R. Zahiri-Azar, X. Wen, and W. Morris, “Viscoelasticity modeling of the prostate region using vibro-elastography,” in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2006, pp. 389–396.
- [21] J. Hing, A. Brooks, and J. Desai, “Reality-based estimation of needle and soft-tissue interaction for accurate haptic feedback in prostate brachytherapy simulation,” *Robotics Research, STAR 28*, vol. 28, pp. 34–48, 2007, sTAR 28.
- [22] A. Okamura, C. Simone, and M. O’Leary, “Force modeling for needle insertion into soft tissue,” *IEEE Transactions on Biomedical Engineering*, vol. 51, pp. 1707–1716, 2004.

Chapter 6

Conclusions and Future Research

In chapters 2, 3 and 4, novel algorithms were proposed contributing to different aspects of medical simulations. These proposed methodologies were tested on simulated data, tissue-mimicking phantoms, and/or actual patient imaging data. In Chapter 5 a simulation framework for prostate brachytherapy was presented. At this point, the feasibility of a training simulation for needle-insertion as was proposed in Chapter 1 is demonstrated. In this chapter, the materials in the previous chapters are analyzed in light of current research in the field and the contributions are delineated. The future work and directions are discussed at the end.

6.1 Contributions

6.1.1 Modeling Needle Flexibility

Elastic rods have been studied in various contexts in the literature such as hair modeling in computer graphics [1] and modeling catheters and guidewires for medical procedures [2]. Based on the Kirchhoff theory of elastic rods [3], a sophisticated dynamic elastic model for various types of discrete rods for computer graphics applications was recently presented in [4]. In Chapter 4 of this thesis, we present a novel use of a discrete elastic rod for needle bending. Using this method, bending and twisting of medical needles are modeled as a collection of rod segments connected by angular-springs. An iterative solution scheme to determine the needle posture for given forces is also presented.

The efficacy of the model was evaluated through experiments, where the bending of an actual prostate brachytherapy needle was observed for several different lateral tip forces. As a benchmark for our approach, it was compared with two common FEM models for the same needle. For the evaluation of the methods, two error measures were proposed to define a good fit of a model to experimental data. These errors were minimized to identify the model parameters, which are Young's modulus for the FEM-based models and spring constant for the angular springs model. The needle deformations simulated using the identified parameters were then compared to the experimental deformations for different experiments.

The angular springs model, which is computationally the most efficient and is the simplest to implement, also demonstrated the highest accuracy in modeling actual needle flexibility compared to the other two models. The models are capable of simulating the needle twist. The twist is often negligible for standard needles in medical procedures such as brachytherapy and it was not validated through our experimental setup. However, recent work introduces the use of *highly-flexible* (also called *steerable* needles) for improved steer-ability in targeted needle insertion applications [5, 6]. These needles are steered using their tip bevel, the angle

of which is controlled by twisting the needle base. In order to model this controlled steering for such very flexible needles, a model of needle shaft torsion around its axis is essential.

6.1.2 Rapid Image Slicing Complying Model-Based Deformations

In Chapter 3, a technique for synthesizing planar images in deformed meshes of tissue models was introduced. The method uses the 3D image data of the anatomical region of interest. A pixel enumeration technique was developed for the well-known *point location* problem to enable fast identification of mesh elements enclosing each image pixel during deformations given by mesh node displacements. This leads to the generation of medical images of considerable size at frame rates that are suitable for real-time applications.

This technique was implemented to simulate B-mode images of a tissue-mimicking phantom and the anterior thigh in-vivo. Synthesized phantom images of known probe indentations were also used to evaluate the method by comparing these with corresponding images acquired by physically deforming the phantom. They exhibit substantial similarity demonstrating the feasibility of the method in simulating B-mode images. A real-time ultrasound simulator implemented on a haptic device was also presented. Both qualitative and quantitative results show that the proposed method produces realistic-looking B-mode images.

The methods presented are easily adaptable to other imaging modalities and deformation models. Fast image simulation is essential for certain deformable registration techniques as well. The presented pixel enumeration technique for element identification, which lies at the heart of the rapid simulation method, is readily highly-parallelizable for implementations on multi-core CPU and GPU architectures. Furthermore, the speed of the technique depends only on the number of pixels in the simulated images, but not on the overall size of the FEM mesh and/or the 3D reconstructed voxel volume. This computational advantage was presented on an example in Chapter 3.

This method makes the assumption that the ultrasound images and artifacts will not change significantly with deformation. This is not necessarily true in every clinical scenario, nevertheless can be safely employed in many applications such as training and deformable registration [7–10]. Considering applications and anatomy in which artifacts are prominent, multiple reconstructions of the same volume can be employed where the images are collected at various probe incidence angles. Subsequently, for each frame simulation, the reconstructed volume acquired by the closest probe orientation to the simulated probe is used. This will effectively maximize the likeness and direction of artifacts in simulation.

6.1.3 Optimal Tessellation of Images

In order to partition images into FEM elements in an optimal fashion, a novel energy definition was proposed in Chapter 2 based on FEM interpolation and image-partitioning errors [11–13]. This combined error is minimized using a proposed optimization scheme producing high-quality image-compliant FEM elements. The method was demonstrated in 2D and 3D on numerical data, MR images of the brain, and CT images of the kidney. Such mesh models of the anatomy are required by medical simulations and hence their generation is an active field of research [14–18].

Such an optimized discretization represents an image using far fewer degrees-of-freedom than the underlying image pixels/voxels. Therefore, it is possible to also extract an approxi-

mate segmentation of the structures of interest from the meshes optimized using this method. This was demonstrated with a simple segmentation technique selecting connected elements from a thresholded subset of all tetrahedra.

Our proposed method does not require any *a priori* knowledge of the boundary constraints, which are often fully defined only at the time of the online simulation but not during the model generation. Local mesh element sizes thought the domain can also be adjusted using the inherent formulation of the technique as was demonstrated on a numerical example. This allows for having finer mesh resolution near certain anatomical features, if desired. A sliver exudation method that is compatible with the proposed energy definition was also presented in Chapter 2.

6.1.4 Deformation Models Using Elastography

With the emerging fields of elastography imaging and tissue parameter identification, the optimal image partitioning method proposed in Chapter 2 becomes essential for generating optimal deformation models from those identified parameters. In a deformation model based on FEM, such mechanical features of the tissue are discretized as a constant in each mesh element. The error due to this discretization is readily minimized using the presented method. This technique was presented on an elastography image of the prostate, where the tissue elastic modulus was approximated to be inversely-proportional to the imaged strain [19]. Indeed, this and other mechanical parameters of a tissue region can be extracted using several methods presented in the elastography literature [20–23].

6.1.5 Needle Insertion Simulation

Computerized simulation of needle insertion was studied in applications for medical training [18, 24], treatment-planning [25], and needle path planning/steering [5, 6]. Based on the pioneering work of [26], the basic concepts of a 3D insertion simulation were developed in [24]. With the integration of the B-mode simulation techniques and the needle flexibility model presented in Chapters 4 and 3, respectively, a complete simulation framework was developed in Chapter 5. The integration of the sub-systems with a haptic device, a user interface, and with each other was also addressed in that chapter. The presented technique can simulate realistic ultrasound images coupled with a physically-based deformation model. This simulation addresses several desired aspects of typical needle insertion applications, such as simulating haptics and imaging, and acquiring patient models, making it a first in its field.

This work is novel and has the potential for an immediate and significant impact on the practice of brachytherapy. There is great interest and enthusiasm in such a training system for prostate brachytherapy from our medical collaborators and hence it has strong potential to serve in a clinical training setting. While extremely important to brachytherapy, there are many other procedures, from biopsy to drug delivery, that require accurate targeting, and hence a fast simulation. Techniques developed for the simulation of needle insertions are also applicable to haptic surgical simulators and planners in general. Planning and correction of needle insertion paths require the prediction such path [25]. The proposed simulator is also of use in such applications.

For an accurate simulation of the tissue deformation during needle insertion, mechanical properties of the tissue have to identified first. Recent advances in elastography techniques

allow for such parameter acquisition [20–23]. Other parameters required by needle-tissue interaction model are the shaft friction and tip cutting forces for each tissue type/interface. There exist methods proposed to identify such parameters in the literature [27, 28]. Nevertheless, no such method has yet been applied and recovered the brachytherapy needle interaction parameters in-vivo. Therefore, approximate values were used in the current simulation and must be addressed for a clinical use.

6.1.6 Patient-Specific Modeling

In this thesis, the models for the finite element simulation of tissue deformation during brachytherapy procedures were obtained from pre-operative MR images. These models are subsequently registered to the ultrasound voxel volumes using the prostate surface in both models. Currently, a rigid registration approach is followed. Nevertheless, given the pelvic bone (and possibly other anatomical structures) delineated manually or automatically in ultrasound images, elastic registration techniques can also be employed.

It is also possible to generate FEM models using elastography imaging. The use of such images in an automatic meshing technique and the application of this technique given a voxel volume were studied separately in Chapters 2 and 5, respectively. The author believes that currently available and recently emerging techniques in elastography [21] combined with the proposed mesh generation method in Chapter 2 is the future of fully-automatic physically-based patient modeling.

6.2 Future Work

The algorithms presented in this thesis yield novel methods for the simulation of minimally-invasive procedures, in particular needle insertion. A prostate brachytherapy simulator was the application platform for these methods and hence the direct outcome of this thesis. This training platform is important for presenting the value of simulated ultrasound in creating a realistic simulation environment. The proposed brachytherapy system was designed as a tool for clinical training, patient-specific rehearsal, and treatment planning. Nevertheless, the methods developed are applicable to several other procedural setups, such as biopsy (e.g. in prostate, breast, liver, brain), lumbar puncture, and epidural needle insertions. The methodology and ideas in this thesis can be easily packaged in a marketable product for clinical use.

The presented ultrasound simulation alone can also be used in other applications, such as in fast deformable registration. This simulation technique applies similarly to fast slicing of images from other modalities to be used for registration, training, or any other purpose. The proposed meshing technique introduces a novel approach, which has the potential to change the traditional “first segment, *then* mesh” view of model generation from medical images. The proposed technique is anticipated to promote further research in similar energy-based formulations for meshing from images, which will lead to optimal and automatic meshing for simulations.

A number of interesting and relevant areas of future work can be pursued from here. Some of these are as follows:

- Identification of the needle interaction model parameters (friction/cutting) for individual tissue types/patients, for which tracking using ultrasound as in [28] may be used.

- Comparison and validation of haptic force-feedback in the simulator with in-vivo needle insertions. For this, (i) an instrumented needle can be used intra-operatively to measure forces, or (ii) a clinical user study can be performed with experienced physicians using our simulator.
- Modeling and simulation of the deflection caused by tip bevel. This could extend to needles that are purposely steered in tissue.
- A clinical simulator system to be used in training, with template overlay in B-mode, etc, and including procedural details such as choosing needles. Possibly, the deformation caused by the probe motion can also be simulated.
- Validation of the ultrasound simulation for prostate imaging. In-vivo ultrasound images can be compared with the images generated by the simulator. Although this may include possible errors in modeling the tissue and needle interaction, such a study has not been done before and would quantify the simulator realism.
- Validation of the predicted seed locations using our simulator against the seed locations identified from a medical imaging modality. The seeds may be located using B-mode ultrasound, fluoroscopy, or emerging elastography techniques. While locating all implanted seeds in brachytherapy is an active field of research, localizing a few seeds can be sufficient for validation.
- Designing a completely automatic patient modeling system with the possible integration of elastography and variational meshing. Given that such models are obtained rapidly, the proposed insertion simulation can be used intra-operatively for on-the-fly treatment planning/correction.
- Experiments with targeting and treatment-planning using the proposed system on phantoms and patients.

References

- [1] K. Ward, F. Bertails, T.-Y. Kim, S. R. Marschner, M.-P. Cani, and M. C. Lin, “A survey on hair modeling: Styling, simulation, and rendering,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 2, pp. 213–234, 2007.
- [2] J. Lenoir, S. Cotin, C. Duriez, and P. Neumann, “Interactive physically-based simulation of catheter and guidewire,” *Computer & Graphics*, vol. 30, no. 3, pp. 417–423, 2006.
- [3] J. Langer and D. A. Singer, “Lagrangian aspects of the kirchhoff elastic rod,” *SIAM Review*, vol. 38, no. 4, pp. 605–618, 1996.
- [4] M. Bergou, M. Wardetzky, S. Robinson, B. Audoly, and E. Grinspun, “Discrete elastic rods,” *ACM Transactions on Graphics (SIGGRAPH Proceedings)*, vol. 27, no. 3, pp. 63:1–12, 2008.
- [5] R. J. I. Webster, J. S. Kim, N. J. Cowan, G. S. Chirikjian, and A. M. Okamura, “Non-holonomic modeling of needle steering,” *Int J of Robotics Research*, vol. 25, pp. 509–525, 2006.
- [6] R. Alterovitz, M. Branicky, and K. Goldberg, “Motion planning under uncertainty for image-guided medical needle steering,” *Int J of Robotics Research*, vol. 27, no. 11-12, pp. 1361–1374, 2008.
- [7] J. F. Krücker, G. L. LeCarpentier, J. B. Fowlkes, and P. L. Carson, “Rapid elastic image registration for 3-d ultrasound,” *IEEE Transactions on Medical Imaging*, vol. 21, no. 11, pp. 1384–1394, 2002.
- [8] D. Zikic, W. Wein, A. Khamene, D.-A. Clevert, and N. Navab, “Fast deformable registration of 3d-ultrasound data using a variational approach,” in *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, Copenhagen, Denmark, 2006, pp. 915–923.
- [9] H. Maul, A. Scharf, P. Baier, M. Wüstemann, H. H. Günter, G. Gebauer, and C. Sohn, “Ultrasound simulators: Experience with sonotrainer and comparative review of other training systems,” *Ultrasound Obstet Gynecol*, vol. 24, pp. 581–585, 2004.
- [10] D. d’Aulignac, C. Laugier, J. Troccaz, and S. Vieira, “Towards a realistic echographic simulator,” *Medical Image Analysis*, vol. 10, pp. 71–81, 2006.
- [11] J. R. Shewchuk, “What is a good linear element? interpolation, conditioning, and quality measures,” in *International Meshing Roundtable*, 2002.
- [12] P. Alliez, D. Cohen-Steiner, M. Yvinec, and M. Desbrun, “Variational tetrahedral meshing,” *ACM Transactions on Graphics (SIGGRAPH Proceedings)*, vol. 24, no. 3, pp. 617–625, 2005.

- [13] D. Mumford and J. Shah, “Optimal approximations by piecewise smooth functions and associated variational problems,” *Communications on Pure and Applied Mathematics*, vol. 42, pp. 577–685, 1989.
- [14] M. Müller and M. Teschner, “Volumetric meshes for real-time medical simulations,” in *Workshop on Bildverarbeitung für die Medizin*, Erlangen, Germany, 2003, pp. 279–283.
- [15] G. Picinbono, H. Delingette, and N. Ayache, “Non-linear anisotropic elasticity for real-time surgery simulation,” *Graphical Models*, vol. 65, pp. 305–321, 2003.
- [16] S. Cotin, H. Delingette, and N. Ayache, “Real-time elastic deformations of soft tissues for surgery simulation,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 1, pp. 62–73, 1999.
- [17] N. Archip, R. Rohling, V. Dessenne, P. Erard, and L. P. Nolte, “Anatomical structure modeling from medical images,” *Computer Methods and Programs in Biomedicine*, vol. 82, no. 3, pp. 203–215, 2006.
- [18] N. Chentanez, R. Alterovitz, D. Ritchie, L. Cho, K. K. Hauser, K. Goldberg, J. R. Shewchuk, and J. F. O’Brien, “Interactive simulation of surgical needle insertion and steering,” *ACM Transactions on Graphics (SIGGRAPH Proceedings)*, vol. 28, no. 3, pp. 1–10, 2009.
- [19] J. Ophir, I. Céspedes, H. Ponnekanti, Y. Yazdi, and X. Li, “Elastography: a method for imaging the elasticity of biological tissues,” *Ultrasonic Imaging*, vol. 13, no. 2, pp. 111–134, 1991.
- [20] A. Samani, J. Bishop, and D. B. Plewes, “A constrained modulus reconstruction technique for breast cancer assessment,” *IEEE Transactions on Medical Imaging*, vol. 20, no. 9, pp. 877–885, 2001.
- [21] J. F. Greenleaf, M. Fatemi, and M. Insana, “Selected Methods for Imaging Elastic Properties of Biological Tissues,” *Annual Review of Biomed. Eng.*, vol. 5, pp. 57–78, 2000.
- [22] S. Salcudean, D. French, S. Bachmann, R. Zahiri-Azar, X. Wen, and W. Morris, “Viscoelasticity modeling of the prostate region using vibro-elastography,” in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2006, pp. 389–396.
- [23] H. Eskandari, S. E. Salcudean, R. Rohling, and J. Ohayon, “Viscoelastic characterization of soft tissue from dynamic finite element models,” *Physics in Medicine and Biology*, vol. 53, no. 22, pp. 6569–6590, 2008.
- [24] O. Goksel, S. E. Salcudean, and S. P. DiMaio, “3D simulation of needle-tissue interaction with application to prostate brachytherapy,” *Computer Aided Surgery*, vol. 11, no. 2006, pp. 279–288, 2006.
- [25] E. Dehghan and S. E. Salcudean, “Needle insertion parameter optimization for brachytherapy,” *IEEE Transactions on Robotics*, vol. 25, no. 2, pp. 303–315, 2009.
- [26] S. P. DiMaio and S. E. Salcudean, “Needle insertion modeling and simulation,” *IEEE Transactions on Robotics and Automation*, vol. 19, no. 5, pp. 864–875, 2003.

- [27] A. Okamura, C. Simone, and M. O’Leary, “Force modeling for needle insertion into soft tissue,” *IEEE Transactions on Biomedical Engineering*, vol. 51, pp. 1707–1716, 2004.
- [28] E. Dehghan, X. Wen, R. Zahiri-Azar, M. Marchal, and S. E. Salcudean, “Needle-tissue interaction modeling using ultrasound-based motion estimation: Phantom study,” *Computer Aided Surgery*, vol. 13, no. 5, pp. 265–280, 2008.

Appendix A

Algebraically Defining the Interior 1-ring Neighbourhood

For a node $\mathbf{x} = \{x, y, z\}^T$ with k neighbouring tetrahedra in its 1-ring, a system of inequalities can be written as the polyhedron constraint as follows:

$$A^{k \times 3} \mathbf{x}^{3 \times 1} < b^{k \times 1} \quad (\text{A.1})$$

omitting the subscript i since only a single node is explained here. Each inequality on each row refers to a neighbouring element and can be defined for the node to be on the *interior* side of that outer face. Let us define one such row, where the other three corners of the element are known positions \mathbf{x}_1 , \mathbf{x}_2 , and \mathbf{x}_3 . Assuming that these element nodes are ordered right-handed, \mathbf{x} can be guaranteed to be on the interior side by checking the following constraint:

$$\begin{vmatrix} x & x_1 & x_2 & x_3 \\ y & y_1 & y_2 & y_3 \\ z & z_1 & z_2 & z_3 \\ 1 & 1 & 1 & 1 \end{vmatrix} > 0 \quad (\text{A.2})$$

where $|\cdot|$ is the determinant of a matrix. Note that this indeed bounds the volume of that element to be positive for that given node order. To yield the structure in (A.1), this determinant can be rewritten as follows:

$$-x \begin{vmatrix} y_1 & y_2 & y_3 \\ z_1 & z_2 & z_3 \\ 1 & 1 & 1 \end{vmatrix} - y \begin{vmatrix} x_1 & x_2 & x_3 \\ z_1 & z_2 & z_3 \\ 1 & 1 & 1 \end{vmatrix} - z \begin{vmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ 1 & 1 & 1 \end{vmatrix} < \begin{vmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ z_1 & z_2 & z_3 \end{vmatrix} \quad (\text{A.3})$$

$$A^{1 \times 3} \mathbf{x}^{3 \times 1} < b^{1 \times 1} \quad (\text{A.4})$$

Appendix B

Resolving Pixel Discretization Conflicts

The correct element for marking a pixel in conflict depends on the scan-line direction. Normally, a pixel on an edge intersection must be labeled by the element which a line scan passing through that pixel is just entering. For instance, in Fig. 3.7(d), with respect to our chosen scan-line direction, the pixels immediately below pixel P belong to e_7 , thus P should be labeled as 7.

To resolve such discretization conflicts, we first sort the cross-sections prior to marking them. Discretizing the cross-section edges on image pixels given a suitable order enables each cross-section label to over-write a previous one by leaving the desired label on the pixel in the end. To formulate such a sorting so that the last value marking a pixel is the correct one, consider a pair of neighbouring cross-sections. A scan-line passing through both of these will visit first one and then the other, since they are convex polygons. Furthermore, the order will be the same for every scan-line traversing both cross-sections. Note that, for a scan-line to correctly locate the pixels below a corner (or, similarly, an edge) shared by these cross-sections, those shared pixels need to be marked by the last-traversed cross-section. In Fig. 3.7(d), e_7 is the latter-traversed element cross-section. Therefore, e_7 's edges have to be discretized after e_6 's, consequently, label 7 over-writes label 6.

This traversal precedence criterion is indeed a *partial ordering* relation in-between all the pairs of neighbouring element cross-sections in the image plane. Let ' $<$ ' denote this relation such that $e_i < e_j$ indicates that e_j is traversed later than e_i by any given scan-line. Due to the convex cross-section geometry, this relation is transitive such that:

$$(e_i < e_j) \text{ and } (e_j < e_k) \implies (e_i < e_k) \quad (\text{B.1})$$

Therefore, all cross-sections can be represented by a *directed acyclic graph*⁷, where the intersected elements are the nodes and their traversal precedence orders are the directed graph edges connecting the pairs of neighbouring elements. Such a graph can be sorted linearly into a global (total) order that satisfies every given partial order relation. This procedure is called *topological sorting*⁷. Note that there may exist more than one such global ordering, any of which can be used for our purposes. For example, the illustration of cross-sections in Fig. B.1(a) has a graph as shown in Fig. B.1(b), and one possible ordering of this graph is as follows:

$$e_{10} < e_2 < e_9 < e_1 < e_8 < e_4 < e_3 < e_5 < e_6 < e_7 < e_{11} \quad (\text{B.2})$$

⁷T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. MIT Press & McGraw-Hill, 2001.

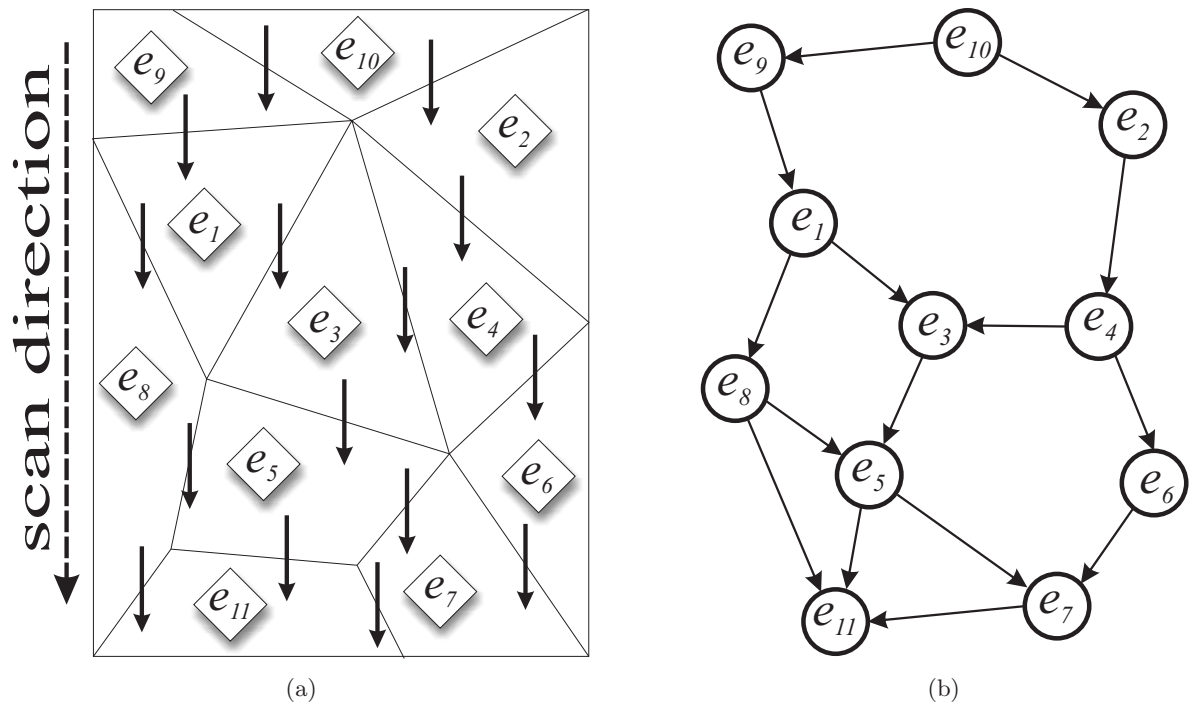


Figure B.1: (a) An illustration of element partial ordering and (b) its corresponding directed graph.

Appendix C

Undeforming Using Barycentric Coordinates

The 3D position of a point $\mathbf{x} = [x \ y \ z]^T$ inside a tetrahedral element e , seen in Fig. C.1(a), can be expressed as follows:

$$\mathbf{x} = \sum_{k=1}^4 r_k \mathbf{c}_k \quad (\text{C.1})$$

where r_k are the barycentric coordinates with respect to the element corners $\mathbf{c}_k = [x_k \ y_k \ z_k]^T$, which are the deformed node positions. This equation can be rewritten for normalized coordinates in the following matrix form:

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix} \quad (\text{C.2})$$

$$\mathbf{x} = C \mathbf{r} \quad (\text{C.3})$$

so that the barycentric coordinates can be found as $\mathbf{r} = C^{-1} \mathbf{x}$.

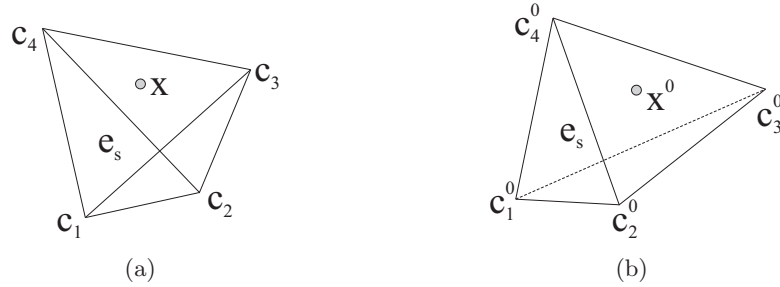


Figure C.1: (a) A deformed element at a time instant t and an image pixel \mathbf{X} lying within; and (b) the nominal pre-deformed geometry of this same element with the corresponding undeformed pixel location.

The point \mathbf{x}^0 seen in Fig. C.1(b) corresponding to the same barycentric coordinates in the nominal pre-deformed configuration of the same element can also be written similarly as $\mathbf{x}^0 = C^0 \mathbf{r}$ for the nominal node positions of \mathbf{c}_k . As a result, this corresponding nominal location is found as:

$$\mathbf{x}^0 = C^0 C^{-1} \mathbf{x} \quad (\text{C.4})$$

$$\mathbf{x}^0 = T_{e_s} \mathbf{x} . \quad (\text{C.5})$$

Appendix D

Euler-Bernoulli Beam Element

The strain energy in a beam can be written as:

$$U = \frac{1}{2} \int_V [E\epsilon_{xx}^2 + 4G(\epsilon_{xy}^2 + \epsilon_{xz}^2)] dV, \quad (\text{D.1})$$

where E and G are the Young's and shear moduli, respectively. Substituting (4.9)–(4.11) in (D.1), one can write the strain energy as⁸:

$$\begin{aligned} U = & \frac{GJ}{2} \int \left(\frac{\partial \alpha_0}{\partial x} \right)^2 dx + \frac{EA}{2} \int \left(\frac{\partial u_0}{\partial x} \right)^2 dx \\ & + \frac{EI_z}{2} \int \left(\frac{\partial^2 v_0}{\partial x^2} \right)^2 dx + \frac{EI_y}{2} \int \left(\frac{\partial^2 w_0}{\partial x^2} \right)^2 dx \\ & + \frac{EA}{2} \int \left[\left(\frac{\partial u_0}{\partial x} \right) \left(\frac{\partial v_0}{\partial x} \right)^2 + \left(\frac{\partial u_0}{\partial x} \right) \left(\frac{\partial w_0}{\partial x} \right)^2 \right] dx \\ & + \frac{EA}{8} \int \left[\left(\frac{\partial v_0}{\partial x} \right)^4 + \left(\frac{\partial w_0}{\partial x} \right)^4 + 2 \left(\frac{\partial v_0}{\partial x} \right)^2 \left(\frac{\partial w_0}{\partial x} \right)^2 \right] dx \end{aligned} \quad (\text{D.2})$$

where A is the cross-sectional area, and $J = I_y + I_z$ where I_y and I_z are the second moments of inertia about the y and z axes, respectively. By substituting (4.12)–(4.15) in (D.2), the beam strain energy can be written as a function of nodal variables. Using the virtual work principle, the relation between the nodal variables and nodal forces/torques can be derived by symbolically differentiating the strain energy with respect to the nodal variables. Based on the structure of the equations for a beam element (interpolation functions and energy equation), this can be written in the form of (4.16) which is detailed as follows:

$$\underline{f} = \begin{bmatrix} \mathbf{K}_{(2 \times 2)}^{11} & \mathbf{K}_{(2 \times 2)}^{12} & \mathbf{K}_{(2 \times 4)}^{13} & \mathbf{K}_{(2 \times 4)}^{14} \\ \mathbf{K}_{(2 \times 2)}^{21} & \mathbf{K}_{(2 \times 2)}^{22} & \mathbf{K}_{(2 \times 4)}^{23} & \mathbf{K}_{(2 \times 4)}^{24} \\ \mathbf{K}_{(4 \times 2)}^{31} & \mathbf{K}_{(4 \times 2)}^{32} & \mathbf{K}_{(4 \times 4)}^{33} & \mathbf{K}_{(4 \times 4)}^{34} \\ \mathbf{K}_{(4 \times 2)}^{41} & \mathbf{K}_{(4 \times 2)}^{42} & \mathbf{K}_{(4 \times 4)}^{43} & \mathbf{K}_{(4 \times 4)}^{44} \end{bmatrix} \underline{u} \quad (\text{D.3})$$

$$\underline{u} = [u_p \ u_{p+1} \ \alpha_p \ \alpha_{p+1} \ v_p \ \phi_p \ v_{p+1} \ \phi_{p+1} \ w_p \ \psi_p \ w_{p+1} \ \psi_{p+1}]' \quad (\text{D.4})$$

$$\underline{f} = [f_p^x \ f_{p+1}^x \ \tau_p^x \ \tau_{p+1}^x \ f_p^y \ \tau_p^y \ f_{p+1}^y \ \tau_{p+1}^y \ f_p^z \ \tau_p^z \ f_{p+1}^z \ \tau_{p+1}^z]' \quad (\text{D.5})$$

⁸I. Sharf, “Geometrically non-linear beam element for dynamics simulation of multibody systems,” *Int J of Numerical Methods in Engineering*, vol. 39, pp. 763–786, 1996.

where \underline{u} is the vector of nodal variables; f^x , f^y , and f^z are the nodal forces; and τ^x , τ^y and τ^z are the nodal torques. The blocks of the stiffness matrix \mathbf{K} can be calculated as follows⁹:

$$\mathbf{K}^{11}(i, j) = EA \int_L \dot{N}_i \dot{N}_j dx, \quad (\text{D.6})$$

$$\mathbf{K}^{22}(i, j) = GJ \int_L \dot{N}_i \dot{N}_j dx, \quad \mathbf{K}^{2t} = \mathbf{K}^{t2} = \mathbf{0} \quad \forall t \neq 2, \quad (\text{D.7})$$

$$\mathbf{K}^{31}(i, j) = 2 \mathbf{K}^{13}(i, j) = EA \int_L \dot{v}_0 \dot{N}_i \dot{M}_j dx, \quad (\text{D.8})$$

$$\mathbf{K}^{33}(i, j) = EI_z \int_L \ddot{M}_i \ddot{M}_j dx + \frac{1}{2} EA \int_L \dot{v}_0^2 \dot{M}_i \dot{M}_j dx, \quad (\text{D.9})$$

$$\mathbf{K}^{34}(i, j) = \mathbf{K}^{43}(j, i) = \frac{1}{2} EA \int_L \dot{v}_0 \dot{w}_0 \dot{M}_i \dot{M}_j dx, \quad (\text{D.10})$$

where L is the element length; $(\dot{}) = \frac{\partial}{\partial x}$ and $(\ddot{}) = \frac{\partial^2}{\partial x^2}$. Blocks \mathbf{K}_{14} , \mathbf{K}_{41} , and \mathbf{K}_{44} are computed similarly to (D.8) and (D.9) by substituting w_0 for v_0 and I_y for I_z .

During the bending of a cantilever beam, such as the needle, no stretching occurs on the beam's neutral axis. Therefore, the strain component ϵ_{xx} in (4.9) should be equal to zero on the beam axis. However, this constraint cannot be met since v_0 and w_0 are interpolated by cubic functions while a linear interpolation function is employed for u_0 . As a result, *membrane locking* occurs resulting in an over-stiff element⁹. One solution for the membrane locking problem is to use one-point Gauss quadrature or reduced integration for the calculation of the sub-matrices of \mathbf{K} that include \dot{v}_0 or \dot{w}_0 in (D.8)-(D.10)⁹. Other parts can still be integrated using two-point Gauss quadrature.

As seen in (D.7), nodal twists and axial torques (the second row/column blocks of the \mathbf{K} matrix) form a separate linear set of equations that is independent of the other variables, i.e. the lateral forces and torques. Therefore, if the twist angles or the twisting torques are negligible, these rows/columns responsible for twist can be removed to decrease the dimension of the stiffness matrix.

⁹J. N. Reddy, *An Introduction to Nonlinear Finite Element Analysis*. Oxford University Press, 2004.

Appendix E

Angle of Deflection for a Bending Moment

Consider a short section of a beam bent under a constant uniform bending moment τ_b . Figure E.1 shows its neutral axis of length L , that is located along a radius of curvature ρ . Let c be the distance of the neutral axis from the outer fibre. From the similarity of the arcs, we obtain the following:

$$\frac{L + \delta}{L} = \frac{\rho + c}{\rho} \Rightarrow \frac{\delta}{L} = \frac{c}{\rho} \quad (\text{E.1})$$

The fibre strain $\epsilon(x)$ of a fibre at a radial distance x to the neutral fibre is given as:

$$\epsilon(x) = \frac{\tau_b x}{EI} \quad (\text{E.2})$$

where E is the Young's modulus of the material and I is the moment of inertia of the section. Substituting (E.1) in the definition of the outer fibre strain $\epsilon(c) = \delta/L$, which is the amount of elongation divided by the initial length, gives $\epsilon(c) = c/\rho$. Substituting this in (E.2) for a fibre distance of c results in:

$$\epsilon(c) = \frac{\tau_b c}{EI} = \frac{c}{\rho} \Rightarrow \frac{\tau_b}{EI} = \frac{1}{\rho}. \quad (\text{E.3})$$

An arc of angle ω at radius ρ has a length $L = \omega\rho$. Thus, (E.3) can be rewritten as follows:

$$\frac{\tau_b}{EI} = \frac{\omega}{L} \Rightarrow \omega = \frac{\tau_b L}{EI}. \quad (\text{E.4})$$

Note that the bending angle ω is equal to the angle of curvature θ . Thus, the linear relation between the bending moment and the angle of deflection can be written as $\theta = \tau_b/k_b$ where $k_b = EI/L$.

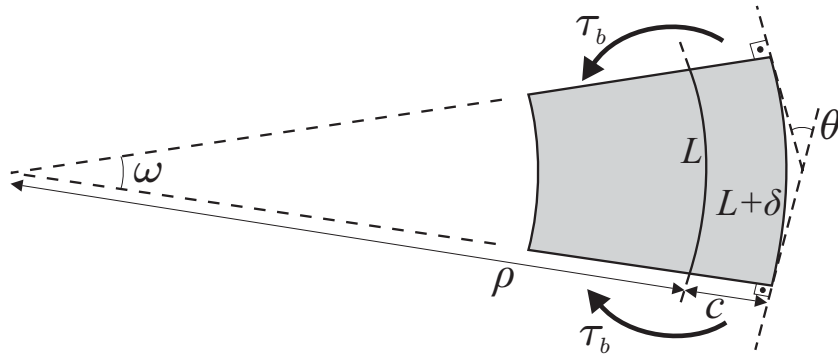


Figure E.1: A short section of a bent cantilever.

Appendix F

Angle of Deflection for a Torsional (Twisting) Moment

The angular deflection α of a shaft under torsional moment M can be expressed as:

$$\alpha = \frac{584 L \tau_t}{G D^4} = \frac{\tau_t}{k_\alpha} \quad (\text{F.1})$$

where L is the length of the twisting section, G is the shear modulus of the material, and D is the shaft radius.

Appendix G

List of Publications

Journal Articles:

- **O. Goksel**, E. Dehghan, and S.E. Salcudean: “Modeling and Simulation of Flexible Needles,” *Medical Engineering and Physics*. 31(9):1069-1078, 2009.
- **O. Goksel**, and S.E. Salcudean: “B-Mode Ultrasound Image Simulation in Deformable 3-D Medium,” *IEEE Transactions on Medical Imaging*. 28(11):1657-1669, 2009.
- **O. Goksel**, S.E. Salcudean, and S.P. DiMaio: “3D Simulation of Needle-Tissue Interaction with Application to Prostate Brachytherapy,” *Computer Aided Surgery*. 11(6):279-288, 2006.
- D.G. French, J. Morris, M. Keyes, **O. Goksel**, and S.E. Salcudean: “Computing Intraoperative Dosimetry for Prostate Brachytherapy Using TRUS and Fluoroscopy,” *Academic Radiology*. 12:1262-1272, 2005.

Selected Refereed Conference Publications:

- **O. Goksel** and S.E. Salcudean: “High-Quality Model Generation for Finite Element Simulation of Tissue Deformation,” *Proceedings of Medical Image Computing and Computer Assisted Intervention (MICCAI)*:248-256, London, UK, Sep 2009.
- S. Mahdavi, **O. Goksel**, and S.E. Salcudean: “3D Segmentation of the Prostate in Ultrasound Images,” *MICCAI*:960-967, London, UK, Sep 2009.
- **O. Goksel** and S.E. Salcudean: “Automatic Prostate Segmentation From Transrectal Ultrasound Elastography Images Using Geometric Active Contours,” *International Conf on Ultrasonic Meas and Imaging Tissue Elasticity*:34, Vlissingen, Netherlands, Sep 2009.
- R. Zahiri-Azar, **O. Goksel**, T.S. Yao, E. Dehghan, J. Yan, and S.E. Salcudean: “Multi-Dimensional Sub-Sample Motion Estimation: Initial Results,” *Proceedings of IEEE Ultrasonics Symposium*, Rome, Italy, Sep 2009.
- **O. Goksel** and S.E. Salcudean: “Real-time Synthesis of Image Slices in Deformed Tissue from Nominal Volume Images,” *MICCAI*:401-408, Brisbane, Australia, Oct 2007.
- **O. Goksel**, R.-Z. Azar, and S.E. Salcudean: “Simulation of Ultrasound Radio-Frequency Signals in Deformed Tissue for Validation of 2D Motion Estimation with Sub-Sample Accuracy,” *IEEE Engineering in Medicine and Biology Conf*:87-90, Lyon, France, Aug 2007.
- **O. Goksel**, S.P. DiMaio, S.E. Salcudean, R. Rohling, and J. Morris: “3D Needle-Tissue Interaction Simulation for Prostate Brachytherapy,” *MICCAI*:827-834, Palm Springs, CA, USA, Oct 2005.