Ultrasound-guided robotic assistance for prostate brachytherapy

by

Jay T. Carriere

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Biomedical Engineering

Department of Electrical and Computer Engineering

University of Alberta

Abstract

Percutaneous procedures, those involving the insertion of needles into tissue, encompass a wide array of clinical applications and are used in both therapeutic and diagnostic modalities. In either of these modalities, the needle must be steered accurately towards a target location to maximize the therapeutic or diagnostic efficacy. Specific procedures, including biopsy, drug delivery, or radiotherapy delivery, require the use of long flexible needles to reach a target location deep inside the body. During insertion, these long flexible needles will bend, or deflect, away from the desired target location or target insertion axes. Surgical robotic systems can be used to reduce or eliminate this deflection during insertion, thus increasing needle placement accuracy. Of particular clinical and research interest are systems which assist, rather than replace, the clinician to capitalize on the clinician's intuition, training, and expertise.

These assistive surgical systems, commonly referred to as surgeon-in-the-loop systems, can be applied to percutaneous procedures to control some aspect of the needle insertion while allowing the clinician to maintain control over other elements of the insertion. The surgeon-in-the-loop systems focused on in this work take the form of needle steering systems, whereby a steering device compensates for any needle deflection while the clinician manually inserts the needle to the desired target depth.

The primary target therapy in this work is prostate brachytherapy, a procedure where long flexible needles are loaded with radioactive seeds, with the needles then being inserted into the patient, to permanently deposit the seeds into the prostate in order to treat cancerous tissue. In current clinical practice, clinicians rely on live ultrasound images to determine the accuracy with which the needles are being placed during insertion. This work aims to use technologies which are already incorporated into clinical practice to minimize the cost and complexity of future clinical integration. The first portion of this thesis thus focuses on needle-tip tracking in ultrasound images. The first work proposes a needle-shape predictive framework for the static case where the needle is fully inserted and thus supported by tissue. The framework uses a mechanical model and incorporates shape information captured in a single ultrasound image near the base of the needle, in order to predict the shape of the entire needle. The second work presents a real-time needle tip-path predictor, incorporating a particle filter and the well known kinematic bicycle model so that needle-tip path prediction can be iteratively updated as each ultrasound image is processed.

The second half of this thesis contains the work related to surgeon-in-the-loop needle steering systems. Two control frameworks are developed which both model needle motion using a reduced order 3D kinematic bicycle model. The first controller implements a switching style regulator using ultrasound image deflection feedback and is proven to asymptotically, in the sense of Lyapunov, stabilize the needle deflection to zero. This switching controller uses direct ultrasound image feedback to measure the deflection of the needle and modulates the needle bevel angle, thus steering the needle, to bring the needle tip back to the axis of insertion. The second controller is based on an event triggered control framework which incorporates needle-tip path prediction to determine the future needle tip deflection. The event triggering controller then uses the predicted needle path to optimize the location of various event points, points of particular needle insertion depth, where the control output is changed to steer the needle.

The last portion of the thesis contains an image processing algorithm to autonomously detect the contour and location of the prostate within 2D B-mode ultrasound images. This prostate contouring, or segmentation, algorithm can be used to determine the location of the prostate prior to needle insertion to correct for any change in prostate location between the original seed deposition plan and the observed prostate during the insertion procedure.

Preface

In the following, all content of this thesis published in peer-reviewed conferences and journals is listed. The contents of Chapter 3 were presented at *2015 IEEE International Conference on Robotics and Automation* [1]. The contents of Chapter 4 were presented at *2016 IEEE International Conference on Advanced Intelligent Mechatronics* [2]. The contents of Chapter 5 were published in [3] and the contents of Chapter 6 were published in [4]. The analysis, implementation, and experimental evaluation of the control systems along with the needle shape prediction and image processing algorithms contained within this thesis is my own work.

The quasi-static spring-beam model in Chapter 3 was the work of Carlos Rossa. The reduced-order kinematic bicycle model, within Chapters 5 and 6, is the work of Moshen Khadem [5]. The hand-held needle steering device was developed and fabricated by Carlos Rossa with the device published originally in [6].

The research project, of which this thesis is a part, received ethics approval from the Alberta Cancer Research Ethics Committee, *Quantifying Needle Deflection and Tissue Deformation in Transperineal Interstitial Permanent Prostate Brachytherapy: A Basis for Development of Next-Generation Implant Technique* No. 25837, 04 Nov 2011.

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada under grant CHRP 446520, the Canadian Institutes of Health Research (CIHR) under grant CPG 127768, and by the Alberta Innovates - Health Solutions (AIHS) under grant CRIO 201201232.

To my parents, Thom and Lenora, my sister, Rachel,
for their love and support during this journey
and to my friends for the laughs and encouragement
along the way.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In percutaneous procedures, a needle is inserted into tissue and is guided by a clinician towards a desired target location. These procedures have a wide range of clinical applications, including biopsy, drug delivery, and brachytherapy. In the case of biopsy, the goal is to gather tissue samples for diagnostic analysis, and for brachytherapy, the objective is to deposit radioactive seeds for the treatment of cancerous tissues. The diagnostic and therapeutic efficacy of both depends upon the accuracy with which the needle tip is positioned with respect to the target point. The ability of a clinician to accurately position the needle tip is impacted in various therapeutic and diagnostic modalities when, due to anatomical constraints, long flexible needles are used. These long flexible needles will bend during insertion due to needle/tissue interactions resulting in suboptimal needle tip placement.

The work covered in this thesis looks at measuring this needle deflection and developing needle steering methodologies implemented in a robotic surgical assistant that reduce deflection during insertion. The first portion of the thesis covers work related to needle tracking and needle shape prediction based on mechanical and kinematic models. The second portion presents two controllers which are designed to steer the needle during insertion based on image feedback. The controllers are implemented in a surgeon-in-the-loop modality where the robot assistant steers the needle while a clinician is manually inserting it towards the target. The final work presented in the thesis uses image processing to locate the target location in images to reduce any error caused

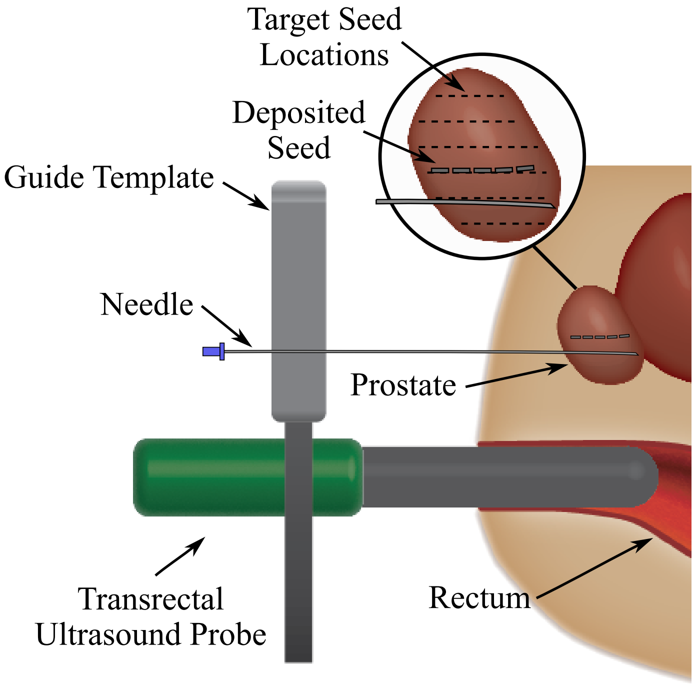


Figure 1.1: Standard clinical brachytherapy setup with Transrectal Ultrasound (TRUS) probe, needle, and guide template.

by patient placement differences between the planning phase of the procedure and the intraoperative phase.

1.1 Prostate Brachytherapy

The therapeutic procedure focused on in this thesis is prostate brachytherapy; a procedure whereby long flexible needles, preloaded with radioactive seeds, are inserted through the perineum into the prostate to treat prostate cancer. This procedure is an attractive treatment option used in early-stage locally-confined prostate cancers [8, 9]. The therapeutic benefit and side-effect profile of this surgery depend on the accuracy with which the radioactive seeds are placed within the prostate with respect to planned dosimetric target locations [10][11][9]. During insertion, however, each needle will bend away from its target location, requiring continuous observation and correction of the needle path.

The first stage in the procedure [12] is to develop an intraoperative plan which defines the target seed locations based on radiation dosing requirements. To begin planning, a patient's prostate is scanned in a series of axial slices, spaced 5 mm apart, which show a cross-section of the prostate. From this pre-operative (”pre-scan”) data, a map of the patient's prostate is generated. In each of the pre-scan slices, the patient's prostate is manually segmented by a clinician and then a dosimetry plan is created. This dosimetry plan maps out the needle target locations and the number of required needle insertions for successful treatment of the cancerous tissue.

With the dosimetry plan created from the pre-scan data, the second stage of the clinical procedure consists of the needle insertion. In standard brachytherapy setup, see Fig. 1.1, clinicians will use a guide template to place needles in line with the target, where each is needle inserted through guide holes in the template. A transrectal ultrasound (TRUS) probe is used to observe the needle placement in tissue. During insertion, only a single image plane at a particular insertion depth is used to verify needle placement, and the clinician must manually step the ultrasound probe forward and backward to change the depth of the plane being imaged. Thus, as the needle is sent off course from its target during insertion, the surgeon is unable to see or predict the needle's total deflection. If the needle has deflected substantially away from the target location after it is fully inserted the needle is removed and reinserted.

Even with the ultrasound images and the clinician manually steering the needle, experienced clinicians are only able to place seeds with an average accuracy of 5 mm [13]. This 5 mm error is significant given the 50 mm diameter of the average prostate. Increasing the efficacy of this procedure is of growing practical importance; with a projected prostate cancer diagnosis for one in eight Canadian men in their lifetime [14].

1.2 Dissertation Outline and Contributions

This thesis introduces new methods for needle shape estimation from ultrasound images, surgeon-in-the-loop needle steering systems informed by the needle shape estimation, and prostate segmentation for target planning and localization. These methods will be used to develop surgical assistance systems to aid a clinician during the pre-operative planning and intra-operative needle insertion stages of a prostate brachytherapy procedure. One such system to be presented in this thesis is a surgeon-in-the-loop needle steering assistant. This surgeon-in-the-loop system incorporates a hand-held needle steering device, originally presented in [6], which rotates the needle at its base to control needle deflection as the surgeon manually inserts the needle; see Fig. 1.2. This thesis will also present a fully autonomous prostate segmentation system which finds the 3D contour of the prostate, captured in TRUS image slices, to assist a clinician during the surgical planning of the procedure. The prostate segmentation algorithm can be utilized when creating the dosimetric plan for a prostate brachytherapy procedure based on these pre-planning TRUS image slices.

As an overview of the material to be covered in this thesis, a brief literature review will be presented in Chapter 2 covering needle tracking in ultrasound images, needle steering and control work, and prostate segmentation in ultrasound images. Following this background material will be original work on needle segmentation in ultrasound images, both in the axial plane and the sagittal plane. Chapter 3 will propose a method, incorporating material presented in [1], to estimate the entire shape of a long flexible needle, suitable

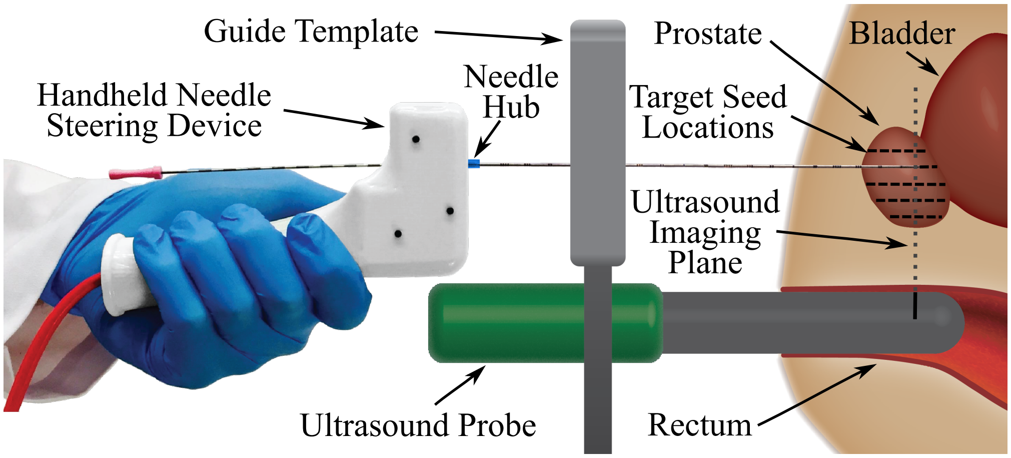


Figure 1.2: Standard clinical brachytherapy setup with the proposed addition of the hand held needle steering device.

for a needle insertion assistant robot. This method bases its prediction on a small segment of a needle, imaged via ultrasound in the sagittal plane, after needle insertion. An algorithm will be developed to segment the imaged portion of the needle from the ultrasound images. The segmented needle will then be used to estimate the parameters of a needle-tissue interaction model that will predict the entire needle shape, with a proof of concept study done in a plastisol tissue phantom.

In Chapter 4 a needle tip-path estimator using real-time TRUS axial image feedback will be demonstrated, based on material presented in [2]. For needle tip path estimation, an algorithm for tracking the position of the needle tip in a series of 2D axial TRUS image slices will be developed. The tracked needle position will then be input into a particle filter to estimate and update the parameters of a kinematic bicycle model in a robust manner to predict the shape of the entire needle after it is fully inserted. The performance of the needle tip-path estimator will be evaluated in both ex-vivo beef tissue phantoms and in-vivo patient images.

Building on this needle tip tracking and needle shape prediction work the following three chapters will cover the the major contributions within this dissertation. The major contributions of this thesis are as follows:

- **A 3D needle steering control algorithm using live TRUS image feedback with a Lyapunov based proof of needle deflection minimization:** A 3D needle steering controller incorporating live TRUS image feedback, originally published in [3], will be developed in Chapter 5 using the axial needle tracking algorithm. A continuous-time control law is built using a depth-dependent 3D reduced order bicycle model, which is a nonholonomic model of needle tip motion. An advantage of this proposed controller over others presented in the literature is that it is proven to asymptotically stabilize the needle, in the sense of Lyapunov, to zero deflection requiring only real-time needle-tip deflection feedback, which can be found from needle-tip tracking in TRUS images. The control law is shown to remain effective in practice even when the magnitude of the needle rotation velocity is limited for practical or patient safety reasons. A system incorporating a hand-held needle steering device will be used to test the controller, such that it will be shown to work in a surgeon-in-the-loop setting, where the needle is inserted manually and the steering controller will minimize deflection. Acceptable needle deflection control performance of the needle steering system will be validated using insertions in three distinct tissue phantoms, with an average needle tip deflection of 0.54 (± 0.27) mm and a maximum needle tip deflection of 1.04 mm at the target insertion depth of 120 mm. Both the average and maximum needle tip deflections are better than the average deflection of 1.22 mm reported in a survey of 26 other 2D and 3D needle steering algorithms in the literature [7].

- **A predictive 3D needle steering controller based on an event-trigger framework to minimize needle deflection and limit rotational needle steering velocities:** In Chapter 6 an improved controller for the surgeon-in-the-loop system, originally published in [4], will be presented. The controller is based on an event-triggered control system where the needle steering output is changed a particular insertion depths of the needle (the event points). This controller will incorporate online parameter estimation of the reduced-order kinematic bicycle model using needle-tip tracking in ultrasound images. The magnitude and location of the steering changes, the event points, will be optimized during insertion based on the current estimate of parameters of the bicycle model. For efficient parameter estimation and control, an analytic solution to the reduced-order bicycle model will be derived. This analytic solution to the bicycle model offers a large fundamental improvement in planning speed and parameter estimation over methods previously used in the literature. In addition, the predictive event-triggered control law offers an improvement over the continuous-time control law, and many controllers presented in the literature, in that the needle deflection performance is optimized while obeying user set limits on the maximum needle rotation velocity (and number of total rotations); which may need to be limited for practical reasons or to minimize tissue damage during insertion. The controller will also be validated using the hand-held device within three distinct tissue phantoms. The experimental results will show the system performing with an average needle tip deflection 0.47 (± 0.21) mm with a maximum needle tip deflection of 0.90 mm at the insertion target depth of 120 mm, which is a slight improvement over the performance of the previous continuous-time control law.

- **A fully-autonomous prostate segmentation algorithm which finds the 3D contour of the prostate based on a set of TRUS images:** The goal of the prostate segmentation algorithm is to find the 3D prostate contour in a set of B-mode TRUS images taken before needle insertion in clinical patients. The contouring algorithm uses a novel superpixel algorithm to extract relevant TRUS image texture and edge information which is combined with statistical prostate shape model information, through a graph-cut based optimization procedure, to segment out the 3D prostate contour. This contour will be compared to the pre-planning contour manually created by the clinician for the dosimetric planning. One advantage of our proposed prostate segmentation system, with respect to many systems presented in the literature, is that the prostate segmentation algorithm is fully-autonomous, requiring no user input during the segmentation procedure. The algorithm will be evaluated on 9 TRUS image sets and a mean absolute difference of 2.52 (±1.66) mm, an average maximum absolute difference of 7.19 (±1.22) mm, and an average Jaccard index accuracy of 0.79 (± 0.07) will be shown. The difference between the reference clinician segmented contours and the algorithmic prostate contours is of similar magnitude to that reported in the literature between various clinicians when manually segmenting the same dataset [15, 16].

The final chapter, Chapter 8, will summarize the material and results contained in this thesis and future research and goals will be discussed.

Chapter 2

Literature Review

2.1 Needle Tracking in Ultrasound Images

In prostate brachytherapy, the images from a transrectal ultrasound probe (TRUS) are used by the clinician to monitor the needle insertion. Standard clinical TRUS probes provide 2D B-mode images in one of two anatomical planes, either the axial plane or the sagittal plane (see Fig. 2.1). Ultrasound image based needle segmentation and tracking have been approached in both 2D axial and sagittal image slices, taken along the length of the needle as it is inserted, as well as in 3D volumes. This thesis will focus on tracking the needle position in 2D images and 3D image volumes. These 3D volumes are created from a series of 2D image slices that are mechanically scanned over a period of time, as due to the small size of the TRUS probe used in prostate brachytherapy, real-time 3D volume ultrasound probes are not yet commercially available.

Sagittal images capture the needle lengthwise along the axis of insertion, see Fig. 2.2(b). In this plane, needle segmentation has been done using a linear Hough transform [17] or principal components analysis (PCA) in conjunction with thresholding and morphological operations [18]. A curved formulation of the 2D Hough transform has also been proposed [19] and has been augmented in the literature by [20, 21] by using a Gabor filter to enhance needle contrast. Sagittal image needle segmentation has the disadvantage that a substantial portion of the needle needs to be captured in the image plane in order for segmentation to work.

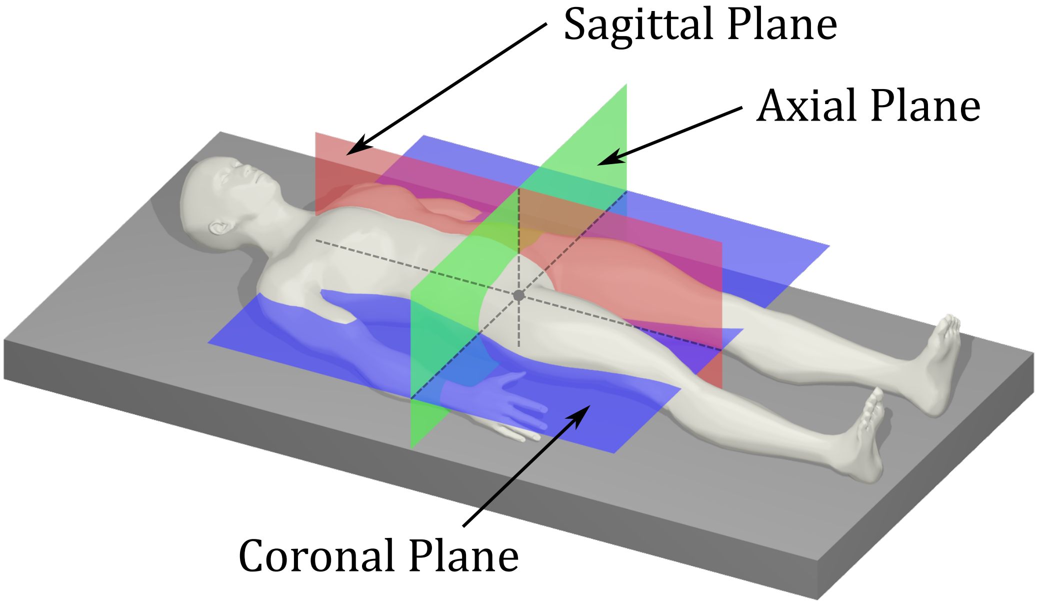
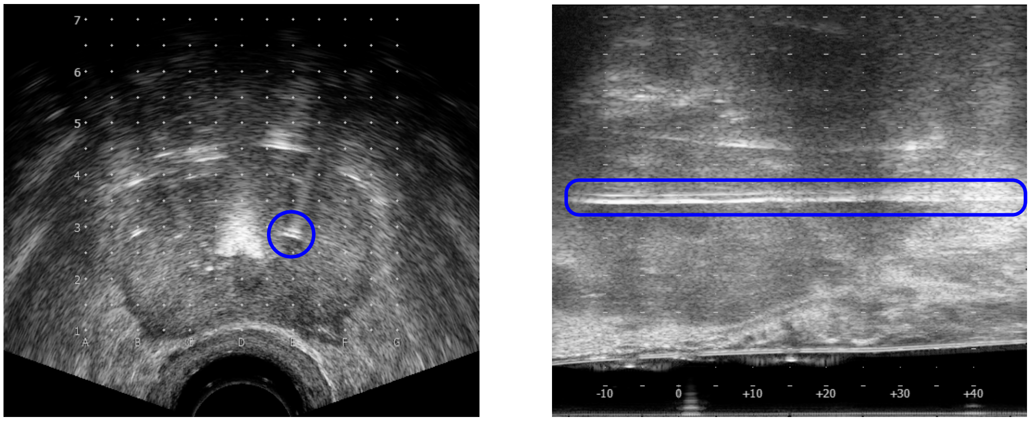


Figure 2.1: Diagram indicating axial, sagittal, and coronal anatomical planes. In this work the patient-attached coordinate system is defined with the x,y,z axes being normal to the sagittal, coronal, and axial planes respectively.



(a) Ultrasound image of needle in axial plane.

(b) Ultrasound image of needle in sagittal plane.

Figure 2.2: Ultrasound images in the two planes used for prostate brachytherapy.

In the axial image plane, only a small cross-section of the needle is observed, with a number of imaging artifacts and low contrast between background tissue and the needle [22, 23, 24] making segmentation in axial images more challenging. In contrast to the sagittal imaging plane, the curvature of the needle cannot be observed from a single axial image. Therefore to get an accurate measurement of needle shape, during or after insertion, the needle cross-section must be segmented or tracked in multiple axial image slices. A commonly used approach for needle shape measurement in the literature is to attach an ultrasound probe to a manually controlled linear stage or robotic linear actuator system so that the probe can be moved to scan the tissue after the needle has been inserted. The image slices captured by such a system can be analyzed as a 3D volume in an offline manner. If a small portion of the needle is to be tracked, or if the needle is sufficiently rigid so as not to bend during insertion, line segmenting algorithms such as 3D projection based methods [25, 26] or the 3D Hough transform [27, 28] may be used to segment the needle in the 3D volume.

In the presence of significant background noise, enhancing the needle appearance in the axial image or filtering of the 3D image volume may be required. Non-traditional TRUS imaging methods may be used to enhance needle appearance, for instance, the work of [29] used vibration at the base of the needle to allow for needle segmentation using Doppler ultrasound. When using conventional imaging methods, the 3D Gabor filter can be used to reduce noise in the 3D volume [30]. Statistical filtering methods have also been used to segment a straight needle from a 3D volume with a linear-least squared based filter being presented in [31] and a phase-grouping approach outlined in [32]. The work of [33] used a RANSAC filtering to fit an arbitrary 3D polynomial to segment out a curved needle shape from the 3D volume.

While requiring additional hardware to perform autonomous needle scanning, axial imaging has the advantage of being able to capture the 3D shape of the needle without requiring any assumption about the in-plane nature of the needle. If the ultrasound probe is positioned using a robotic or computer controlled linear actuator, the probe can be moved in synchrony with the needle tip as the needle is inserted. Such systems are controlled so that the tip of the needle is contained within each axial image slice. In general US image artifacts hamper needle segmentation; the comet-tail artifact which appears around a needle in axial images [23, 24], however, can be used for segmentation of the needle tip in single axial images [34] or to aid in tracking the needle tip between adjacent image slices [35]. Several real-time methods for needle tracking using the comet-tail artifact have been proposed in the literature. With the comet-tail artifact presenting as a bright line in the US image the Hough transform may be used for needle detection [36, 37] or Kalman filtering may be used to track the comet tail between images [38, 39]. By using a robotic probe holder with both translational and rotational degrees of freedom the angle at which the US probe images the needle tip can be controlled to improve the appearance and contrast of the comet-tail artifact [40].

2.2 Needle Modelling and Control

For general percutaneous procedures, a number of different models for estimating needle deflection have been proposed in the literature. A thorough review of the various models and control methodologies presented in the literature is given in a survey by [7]. In general, the needle deflection models can be categorized as either being kinematic or mechanical in nature.

The most commonly used kinematic model in the literature is the bicycle model, which was developed by [41] and [42] to describe the motion of a beveled-tip needle as it is inserted into tissue. This model states that the 3D trajectory of a beveled needle-tip, during insertion into tissue, is analogous to the trajectory of a bicycle with its front wheel locked at a single angle. This “locked wheel” models the needle trajectory as following a 3D path with a constant radius of curvature (i.e., moving along the edge of a 3D circle) with the direction of curvature related to the angle of the needle-tip bevel. Research in [41] developed this model for use as a 3D path planning model for needle insertion. This model was used by [42] for needle deflection modeling in the control of a needle directed toward a target in a single 2D plane.

The bicycle model is typically solved in state-space where the 2D or 3D position and orientation of the needle tip are the states of the system. The radius of curvature of the needle tip's trajectory approximates needle-tissue interaction forces and is, therefore, a tissue-dependent parameter to be known a priori [43] or estimated. The work of [44] showed the parameters of a linearized version of the 2D bicycle model can be estimated by a Kalman filter for control and path-planning. When needle-tip deflection is not controlled, [45] show that an Unscented Kalman Filter can be used for 3D state and parameter estimation.

State estimation of the bicycle model is often coupled with a controller. Using modern control techniques, the researchers in [46] developed a linearizing transform for a 2D version of the bicycle model and implemented it in an observer-controller combination. Sliding mode control has also proven very effective at the combination of state-estimation and control applied to the bicycle model with [47] implementing 2D and [48, 49] implementing 3D sliding mode controllers. Trajectory control using the bicycle model in 3D has also been achieved by using a 2D in-plane needle deflection models [36, 37, 50], where the orientation of the (2D) plane of deflection is related to the orientation of the needle-tip bevel and is assumed, or controlled, to be constant over small insertion lengths. Once the initial 2D plane of needle deflection is estimated from ultrasound images, the 3D trajectory control is implemented by rotating the base of the needle, thus changing the orientation of the 2D deflection plane, to compensate for 3D target motion and account for 3D obstacle motion inside tissue.

Mechanical models incorporate physical forces and bending moments caused by needle/tissue interaction and control actions to estimate needle deflection. Static mechanical models have been used in the literature to describe needle deflection when the needle is fully inserted, with pseudo-static or dynamic mechanical models used for needle deflection calculation during insertion. One class of mechanical needle model used for prediction and control considers the needle to be an Euler–Bernoulli beam and was originally presented by [51] where a constant bending moment along the entire needle due to tissue forces was assumed. This class of model has expanded to incorporate needle diameter, tissue stiffness, needle-tip cutting forces, and friction as factors affecting the needle's overall deflection [52, 53, 54, 55, 56, 57, 58, 59, 60]. Dynamic needle-tissue interaction forces, such as mechanical work to insert the needle [61], needle-tissue interaction force profiles [61, 62], and complex finite element tissue deformation models [63] have also been incorporated into this model.

Needle deflection control using Euler-Bernoulli beam based models typically requires that tissue parameters and needle-tissue interaction are known before insertion. This parameter identification and tissue characterization is a non-trivial task, especially with regard to in-vivo human tissue, and typically requires trial insertions observed in US images to determine these parameters [52, 61, 62, 63]. When using Euler-Bernoulli beam models for online control of needle deflection, models which contain approximations of the needle-tissue interaction forces and simplifications of the beam deflection models are often used. The work of [64] showed that point-force loads to approximating needle-tissue interaction parameters which were estimated from needle deflection in ultrasound images. For controlling needle deflection, the effect of tissue on the needle can be represented by spring elements supporting an Euler–Bernoulli beam with (rigid) linear [65] or (flexible) nonlinear beam elements [58, 6]. The work of [66] developed a helical spring-beam linearization Euler-Bernoulli beam model for a needle in free space. The spring-beam model considers the needle to consist of a series of massless rods interconnected by radial springs. This model was extended by [67] to incorporate needle tip cutting forces as the needle is inserted into tissue.

2.3 Prostate Segmentation

There are several artifacts and limitations inherent in ultrasound images which much be compensated for in computer-aided tissue segmentation [22]. In the specific case of prostate segmentation the well-known shadowing and edge enhancement artifacts, which obscure the edge of the prostate in US images, has motivated the use of active-shape, active-contour, texture-based, and other methods for segmentation. Regardless of segmentation modality, algorithmic prostate segmentation can be grouped into two general methods, semi-autonomous and fully-autonomous systems. Semi-autonomous systems require some user input for each patient in order to perform prostate segmentation. Fully-autonomous systems, in contrast, require no such user input. The field of prostate segmentation in multiple imaging modalities has been well explored in the literature with [68] and [69] providing comprehensive surveys. A more recent survey of focusing on automated diagnosis using algorithmic prostate contouring in ultrasound images was given by [70].

In active contour model segmentation, a simple contour model is deformed through some forcing function such that the contour will come to rest on the edge of the prostate. Active contour segmentation is typically semi-autonomous, requiring user input for the initial contour placement, and can be particularly sensitive to areas of low contrast or shadowing/enhancement in US images. One of the first active contour segmentation systems was presented by [71], wherein an initial user-defined 2D contour model (later extended to 3D by [72, 73]) is fit to the edge of the prostate, with the user editing the segmented contour in areas affected by US artifacts. To compensate for incomplete edge information algorithmically, [74] used an active contour model which incorporates a Kalman filter to probabilistically detect the edge of the prostate around user input seed points. To avoid the instability caused by the forcing functions in the active contour model, [75] instead uses non-linear warping of the axial image to an ellipse based contour model. The work of [76] represented the contour of the prostate with 2D superellipses. This concept was futher developed in [77] information from a full series of these 2D axial superellipses is refined into a single 3D ellipsoid contour.

To enhance the robustness of contour-based segmentation algorithms in the presence of shadowing/enhancement and low organ-to-background contrast, a statistics-based approach known as active shape modeling has been employed. In active shape segmentation, a large number of manually segmented contours (or feature points) are input into a system to derive the probability that the contour will take a particular shape (or has particular feature point locations)[78]. The work of [79] presented a 2D prostate contouring algorithm using weak membrane that is deformed to segmented the prostate based by edge information, enhanced by an edge-preserving filter, in TRUS images. A dyadic wavelet transform was used by [80] to filter input TRUS images before a spline based discrete dynamic contour is deformed to segment the prostate. Using the statistical inference from a series of prostate contours, [81] created a point distribution model and [82] used a spherical harmonics model, which were fit to a series of user input points and then updated based on the likelihood that nearby prostate edges in the US image correspond to the statistical model.

By incorporating both contour and image information from manually segmented contours, [83, 84] used non-rigid registration to warp an input US image to all of the manually segmented US images in a database. A series of warped manually-segmented images which strongly matched the input image is then used to calculate the contour statistics, with the mean of this statistic being used as the contour for segmentation.

To provide a basis of comparison for the performance of the prostate segmentation algorithm presented in Chapter 7, a selection of prostate segmentation articles closely related to the work presented in this thesis is given in Table 2.1. This table summarizes various aspects of the referenced prostate segmentation works, for instance indicating if the prostate segmentation algorithm returns 2D prostate contours or 3D prostate volumes and if the algorithm uses an active contour model or active shape model, or both, for segmentation. User input for the referenced segmentation methods is also shown, such as initial point selection and requiring a clinician to manually correct the prostate contours produced by the algorithms after image processing. Table 2.1 incorporates information presented in [68] (Table III) and [69] (Table 5) along with the information presented in the cited papers within the table. The various accuracy and similarity metrics for evaluating the performance of referenced prostate contouring algorithms is given in Table 2.2 which are based on[69] (Tables 2 and 3).

Table 2.1: Summary of Related Ultrasound Prostate Segmentation Literature

Author and Citation	Year	2D	3D	Initial Point Selection	Manual Correction	Active Contour Model	Active Shape Model	Contour Accuracy Values (mm)		Volume Accuracy Values (mm)		Volume Similarity Value	
[79] Pathak et al.	2000	x		x		x		MAD	1.50				
								MaxD	4.00				
[71] Ladak et al.	2000	x		x	x	x		MAD	$\approx 0.63 \pm 0.25$				
								MaxD	$\approx 2.5 \pm 0.98$				
[80] Chiu et al.	2004	x		x	x	x		MAD	$\approx 0.58 \pm 0.18$				
								MaxD	$\approx 2.14 \pm 0.91$				
[76] Gong et al.	2004	x		x		x		MAD	1.36 ± 0.58				
								MaxD	3.42 ± 1.52				
[73] Hu et al.	2003		x	x	x	x		MAD	1.19 ± 0.14			VD	0.072 ± 0.034
								MaxD	7.01 ± 1.04				
[81] Hodge et al.	2006		x	x	x		x	MAD	1.09 ± 0.49	AVD	3.28 ± 3.16	VD	0.0022 ± 0.0458
								MaxD	7.27 ± 2.32				
[82] Tutar et al.	2006		x	x		x	x	MAD	1.26 ± 0.41			JI	0.835 ± 0.042
								MaxD	4.06 ± 1.25				
[77] Mahdavi et al.	2011		x	x		x				AVD	2.43 ± 6.08	VE	0.0663 ± 0.0090

Table 2.2: Evaluation Metrics for Prostate Segmentation

Metric Type	Parameters	Metric	Equation
Contour	A distance d_j between corresponding points on the algorithmically segmented contour and a ground truth contour	Mean absolute distance (MAD)	$\text{MAD} = \frac{1}{N}\sum_{j=1}^{N} \lvert d_j \rvert$
		Maximum distance (MaxD)	$\text{MaxD} = \max \lvert d_j \rvert$
Volume	Algorithm contour volume (ACV) and ground truth contour volume (GTV)	Volume Difference(VD)	$\text{VD} = \frac{(\text{GTV}\cup\text{ACV})-(\text{GTV}\cap\text{GTV})}{2\times\text{GTV}}$
		Average Volume Difference (AVD)	$\text{AVD} = \frac{\text{GTV}-\text{ACV}}{\text{GTV}}$
		Jaccard Index (JI)	$\text{JI} = \frac{\text{GTV}\cap\text{ACV}}{\text{GTV}\cup\text{ACV}}$
		Volume Error (VE)	$\text{VE} = \frac{\text{GTV}+\text{ACV}-2(\text{GTV}\cap\text{ACV})}{\text{GTV}+\text{ACV}}$

Chapter 3

Needle Shape Prediction from Partial Observation in Sagittal US Images

The first step in predicting the needle shape in our method is to acquire ultrasound images of a portion of the brachytherapy needle inserted into tissue. Subsequently, we introduce a novel technique for segmenting bent needle images by combining edge detection and thresholding techniques with RANdom SAmple Consensus (RANSAC) filtering[85]. The segmentation algorithm outputs a second order polynomial that fits to the portion of the needle seen in the ultrasound image. An affine transformation is used to convert the polynomial into the real-world frame so that it can be used by the needle-tissue interaction model, which predicts the shape of the entire length of the needle. To validate this prediction, we compare the shape output from the model with the shape of the needle as seen in the camera images, which show the entire needle; the same technique as used to segment the needle shape in ultrasound is also used for the camera image. The overall flowchart of the algorithm is given in Fig. 3.1.

This chapter is organized as follows. In Section 3.1, we introduce the algorithm to segment a needle in both ultrasound and camera images; this segmentation results in two second order polynomial equations that follow the curvature of the needle. Section 3.2 deals with the conversion of the needle shapes found from the camera and ultrasound images into physical

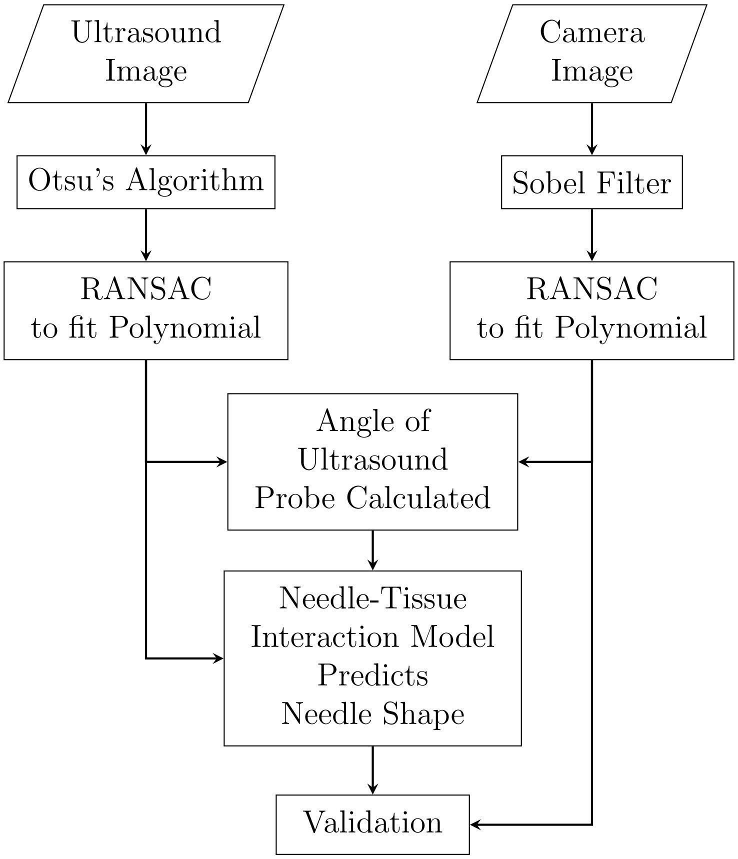


Figure 3.1: Needle Prediction and Validation Algorithm

coordinates, and thus coregistration into a common frame. The polynomial equations and transform matrices calculated by the coregistration are used in Section 3.3 as inputs to our needle-tissue interaction based model. This needle-tissue interaction model is then used to predict the entire needle shape from the observed portion of the needle seen in the ultrasound image. Finally, the results of the needle shape prediction are compared to the needle shape as found in the camera image, which represents the ground truth.

3.1 Camera and Ultrasound Image Processing

In this section we present a technique that is used to find a curved needle in an ultrasound image, giving an observation of a small portion of the needle, which will be used to inform the total needle shape prediction. This technique for segmenting a curved needle in an ultrasound is then extended to a camera image. The needle segmentation algorithm results in a second order polynomial equation that follows the curvature of the needle in its respective frame of reference, i.e. either the camera frame or the ultrasound image frame.

The experimental setup consists of a tissue phantom imaged from overhead by a camera and from the side by an ultrasound scanner. The images show a needle that is inserted into the tissue and deflects during insertion. Fig. 3.2 shows a schematic representation of the experimental set up in order to define the different coordinate frames that arise from combining imaging modalities. It is important to note that points in both the camera and ultrasound images, are throughout the chapter, defined in their respective pixel domains. In the convention here, superscripts and subscripts on the left side of a symbol refer to the frame of reference for the coordinate system. The coordinate frame for the ultrasound image is denoted by U; a point ${}^{U}P$ in the ultrasound image will have coordinates (${}^{U}x$, ${}^{U}y$) defined with respect to the lower left hand side of the ultrasound image, as shown in Fig. 3.2. The coordinate frame for the camera image is denoted by C; a point ${}^{C}P$ in the camera image will have coordinates (${}^{C}x$, ${}^{C}y$), defined with respect to the upper left hand side of the camera image, as indicated in Fig. 3.2. A third frame is defined not in

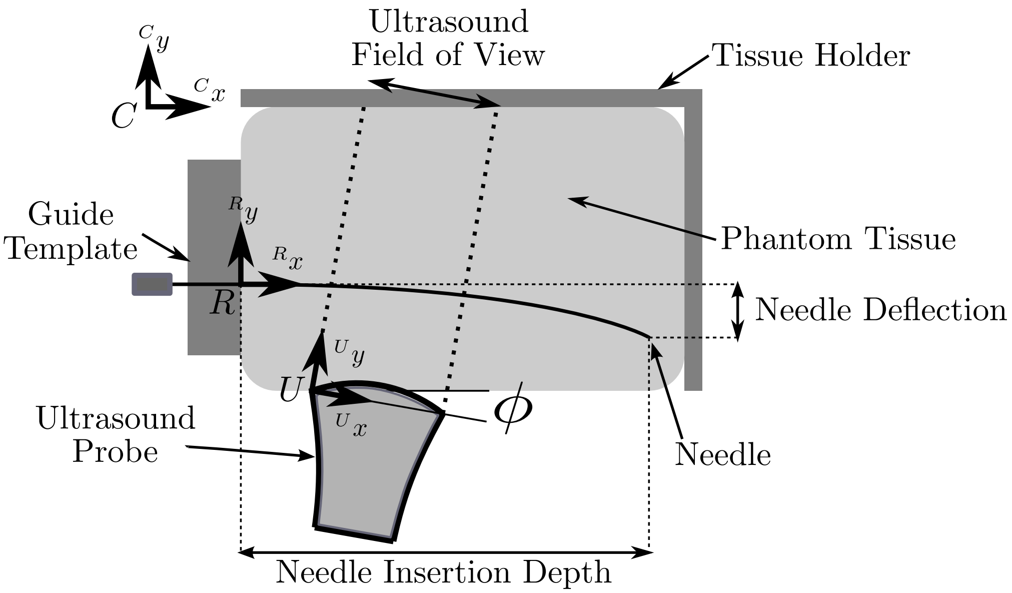


Figure 3.2: Brachytherapy procedure simulation setup including ultrasound probe, phantom tissue, needle template, and needle. The setup is represented from the field of view of the camera.

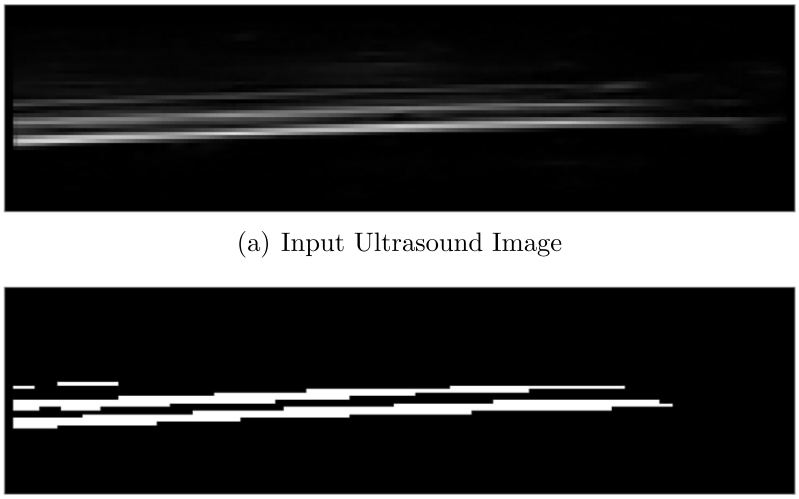
a pixel domain but in physical coordinates and referred to as the real-world frame, denoted by R. A point ^{R}P in the real-world frame consists of (^{R}x, ^{R}y) measured in meters with respect to an origin located where the needle exits the template; see Fig. 3.2.

In Section II-A and Section II-B, we will introduce methods to process the ultrasound and camera images, respectively. Section II-C describes how the points from the image processing are segmented, i.e. converted into polynomial curves.

3.1.1 Ultrasound Image Processing

Needle segmentation in 2D ultrasound images has previously been done via Gabor Filtering [20] and the Hough Transform for straight needles [17] or curved needles [19]. Additionally, needles have been segmented in 3D ultrasound images using projection methods [86, 25] or RANSAC [87]. In this chapter, we combine RANSAC filtering with Otsu's Algorithm [88] and Sobel edge detection [89].

Ultrasound images along the length of the needle are taken near the needle's



(a) Input Ultrasound Image

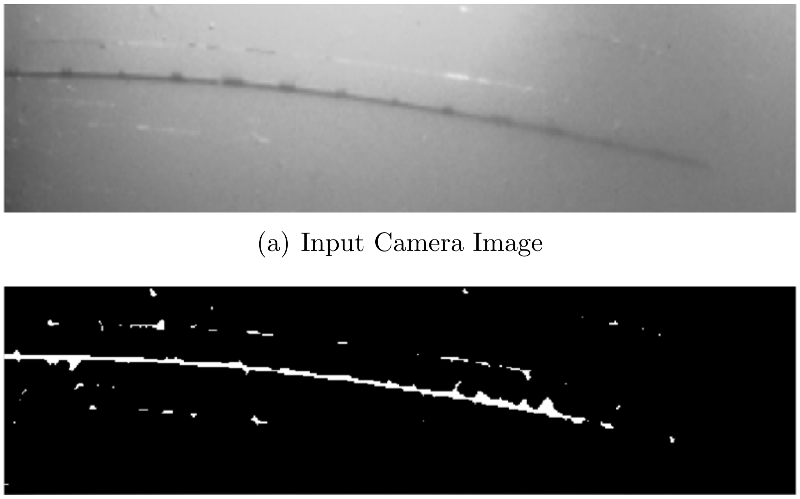
(b) Ultrasound Image after Image Processing

Figure 3.3: Image Processing for Ultrasound

entry point into tissue, as this represents the most challenging scenario for prediction. The probe used to capture the images is a linear transducer and as such shows a limited field of view, in this case a linear 40 mm wide image that does not widen with depth. These images are captured at arbitrary distances from 2 to 66 mm away from the needle's entry point into the tissue. When the needle is imaged under ultrasound while in tissue, the needle is quite distinct from the background. The ultrasound image of the needle is shown in Fig. 3.3(a). The routine for segmenting the needle in an ultrasound image involves thresholding using Otsu's algorithm [88]. This converts the grayscale ultrasound image into a segmented binary (black and white) image, as shown in Fig. 3.3(b). The white values in the binary image are converted into a list of points ${}^{U}P$, which is the input for the RANSAC algorithm. Image processing is implemented in Matlab using the functions from the image processing toolbox.

3.1.2 Camera Image Processing

In contrast to the ultrasound image, the needle in the camera image is not readily segmentable based on a grayscale threshold. As seen in Fig. 3.4(a), the camera image suffers from variations in lighting intensity and a global threshold that can segment the needle in all portions of the image cannot be



(a) Input Camera Image

(b) Camera Image after Sobel Filtering

Figure 3.4: Image Processing for Camera Image

found. Therefore, segmenting the needle from the camera image is performed using the Sobel edge detection filter. The Sobel edge detector is a 3x3 gradient filter that approximates the 2D derivative of the image with respect to image intensity [89]. From this, a threshold is applied to the filtered image (i.e., the 2D derivative) and only those edges exceeding the threshold are accepted, thus creating a binary image. The output of the camera image processing is shown in Fig. 3.4(b). After the Sobel filter is applied to the image, the white values are converted to a list of points ${}^{C}P$ to be input to perform RANSAC filtering on the image (Section 3.1.3).

3.1.3 Processed Image Conversion to Polynomial via RANSAC

Given that the image processing routines output binary images, the next step in the algorithm is to convert a list of segmented points from those images into polynomial curves. Typically, a method such as a linear least squares can be used to fit data to a polynomial, however the output of a simple linear least squares fit will consider all points to be inliers in the data set and as such can be skewed by outliers or noise. Instead, RANSAC or RANdom SAmple Consensus [85] is the algorithm that we will use to fit polynomials to the noisy datasets as RANASC rejects outliers from a dataset.

The input to the RANSAC algorithm is a collection of points from either the camera ${}^{C}P$ or ultrasound ${}^{U}P$, in their respective frames, that represent possible points along the needle. We want to convert the points from ${}^{C}P$ and ${}^{U}P$ into second order polynomials

$$ {}^{C}y = \alpha_2({}^{C}x)^2 + \alpha_1({}^{C}x) + \alpha_0 \tag{3.1} $$

$$ {}^{U}y = \beta_2({}^{U}x)^2 + \beta_1({}^{U}x) + \beta_0 \tag{3.2} $$

where the α and β terms are the coefficients of the desired polynomials that best describe the needle.

The RANSAC algorithm, illustrated in Fig. 3.5, takes 3 random points from the sets ${}^{C}P$ or ${}^{U}P$ and uses linear least squares to calculate the polynomial coefficients in (3.1) and (3.2) respectively. Outliers are defined as points that are further than a given distance from the curves (3.1) and (3.2). The ratio of inliers to outliers is then calculated and if this ratio is greater than a threshold the algorithm calculates the coefficients for polynomials (3.1) and (3.2) considering the entire set of inlier points. If the ratio is less than the threshold, then three different points from ${}^{C}P$ or ${}^{U}P$ are chosen and the algorithm restarts. For our implementation we use maximum distance values of 3 pixels for the camera image and 7 pixels for ultrasound image. For the inlier to outlier ratio we use 70% and 80% for the camera image and ultrasound image respectively. These pixel distance values were found empirically by reviewing the results of several trials; in the future we intend to automate the process of finding these values. After the RANSAC routine has successfully been run on the two sets of points ${}^{C}P$ and ${}^{U}P$, the resulting polynomials follow the curve of the needle in either image. The results after RANSAC polynomial curve fitting are shown in Fig. 3.6.

3.2 Camera and US image coregistration

In order to have both polynomials (3.1) and (3.2) referred to the same frame, or coregistered, three transformation matrices are needed for both camera image and ultrasound image conversion. For the transformation matrix from the

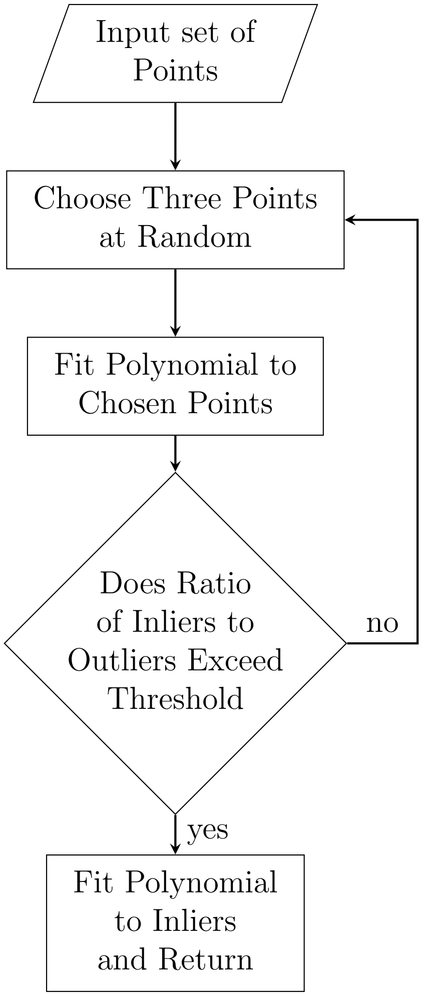
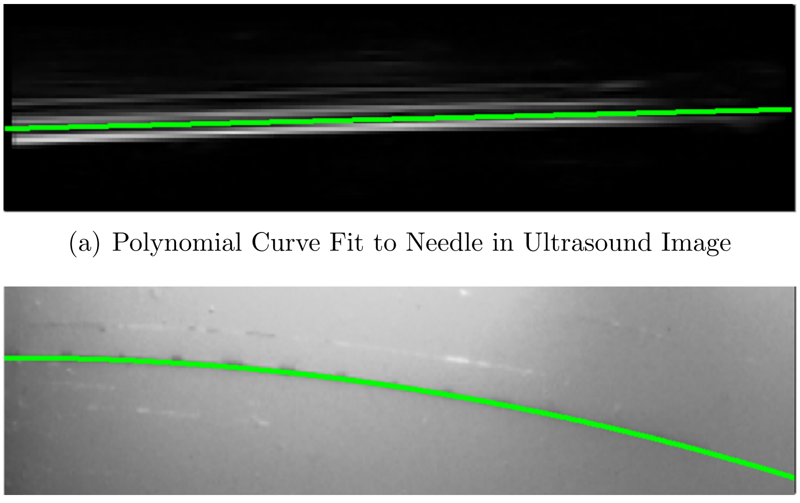

Input set of Points
Choose Three Points at Random
Fit Polynomial to Chosen Points
Does Ratio of Inliers to Outliers Exceed Threshold
no
yes
Fit Polynomial to Inliers and Return

Figure 3.5: RANSAC Algorithm



(a) Polynomial Curve Fit to Needle in Ultrasound Image

(b) Polynomial Curve Fit to Needle in Camera Image

Figure 3.6: Polynomial Curves Following Segmented Needle Shape

ultrasound image's pixel domain to the real-world frame, we assume that a simple affine transformation matrix, ${}^{R}_{U}T$, will suffice as the linear transducer on the ultrasound machine results in a rectangular image that does not widen with depth. Likewise for the camera, since it is located above the center of the setup and is adjusted so that its focal plane coincides with that of the experiment we assume another affine transform matrix, denoted by ${}^{R}_{C}T$, will suffice. The general forms for the affine transformation matrices to be used in this chapter are given by

$$ {}^{R}_{U}T = {}^{R}_{U}A \cdot {}^{R}_{U}B \cdot {}^{R}_{U}C \tag{3.3} $$

$$ {}^{R}_{C}T = {}^{R}_{C}A \cdot {}^{R}_{C}B \cdot {}^{R}_{C}C \tag{3.4} $$

where A is a translation matrix from the origin of one frame to another, B is a rotation matrix that corresponds to the angles between the frames, and C is a scaling matrix that converts from the units of one frame to another.

3.2.1 Ultrasound Image Transformation Matrix

The matrix ${}^{R}_{U}A$ that translates the origin of the ultrasound frame to the origin of the real-world frame is of the form

$$ {}^{R}_{U}A = \begin{bmatrix} 1 & 0 & {}^{R}_{U}t_x \\ 0 & 1 & {}^{R}_{U}t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.5} $$

where ${}^{R}_{U}t_x$ and ${}^{R}_{U}t_y$ are the x and y offset, in meters, of the leftmost point of the transducer element from the base of the needle, see Fig. 3.2. Since the ultrasound probe is deliberately rotated from experiment to experiment, we require a matrix ${}^{R}_{U}B$ given as

$$ {}^{R}_{U}B = \begin{bmatrix} \cos\phi & \sin\phi & 0 \\ -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.6} $$

where ϕ is defined as the angle of the ultrasound probe relative to the closest edge of the tissue, as shown in Fig. 3.2. Finally, conversion from the ultrasound image's pixel domain to real-world coordinates is done by the scaling matrix ${}^{R}_{U}C$ given by

$$ {}^{R}_{U}C = \begin{bmatrix} {}^{R}_{U}S_x & 0 & 0 \\ 0 & {}^{R}_{U}S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.7} $$

where ${}^{R}_{U}S_x$ is the pixel domain width scaling factor and ${}^{R}_{U}S_y$ is the height scaling factor. The total transformation matrix is given by

$$ {}^{R}_{U}T = {}^{R}_{U}A \cdot {}^{R}_{U}B \cdot {}^{R}_{U}C = \begin{bmatrix} {}^{R}_{U}S_x \cos\phi & -{}^{R}_{U}S_x \sin\phi & {}^{R}_{U}t_x \\ {}^{R}_{U}S_y \sin\phi & {}^{R}_{U}S_y \cos\phi & {}^{R}_{U}t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.8} $$

which describes the affine transform between the ultrasound frame and the real-world frame, with the components of ${}^{R}_{U}T$ coming from the translation, scaling, and rotation values in matrices ${}^{R}_{C}A$, ${}^{R}_{C}B$, and ${}^{R}_{C}C$ respectively.

The elements of the matrices ${}^{R}_{U}A$ and ${}^{R}_{U}C$ can be found by using the coordinates of landmarks common to the two coordinate frames. ${}^{R}_{U}B$, on the other hand, requires the value of ϕ. This value, while relating to a physical quantity, is difficult to measure with enough accuracy for our purposes. We will discuss how we find the value of ϕ in Section 3.2.3 as this requires having the matrix ${}^{R}_{C}T$.

3.2.2 Camera Image Transformation Matrix

The matrix ${}^{R}_{C}A$ that translates the origin of the camera frame to the origin of the real-world frame is given by

$$ {}^{R}_{C}A = \begin{bmatrix} 1 & 0 & {}^{R}_{C}t_x \\ 0 & 1 & {}^{R}_{C}t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.9} $$

where ${}^{R}_{C}t_x$ and ${}^{R}_{C}t_y$ are the x and y offset, in meters, of the upper leftmost point of the camera image to the base of the needle, see Fig. 3.2. Since the camera frame is purposely aligned to the real-world frame, this implies

$$ {}^{R}_{C}B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.10} $$

Finally, conversion from the camera image's pixel domain to meters is done by the scaling matrix ${}^{R}_{C}C$ which is

$$
{}^{R}_{C}C = \begin{bmatrix} {}^{R}_{C}S_0 & 0 & 0 \\ 0 & {}^{R}_{C}S_0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.11}
$$

where ${}^{R}_{C}S_0$ is the pixel domain scaling factor for both the x and y axes, since the experiment is centered in the camera field of view. As with the ultrasound matrices, the camera transformation matrices ${}^{R}_{C}A$ and ${}^{R}_{C}C$ can be found by converting measured landmark points between the two coordinate frames. Again, the total transformation matrix can be found through

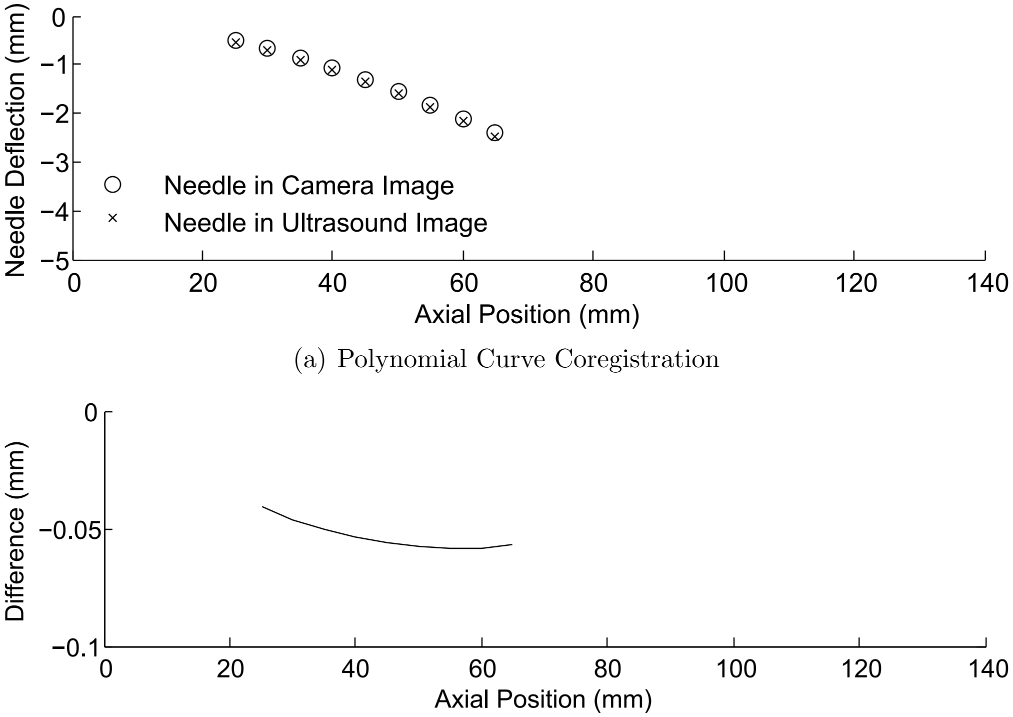
$$
{}^{R}_{C}T = {}^{R}_{C}A \cdot {}^{R}_{C}B \cdot {}^{R}_{C}C = \begin{bmatrix} {}^{R}_{C}S_x \cos\phi & -{}^{R}_{C}S_x \sin\phi & {}^{R}_{C}t_x \\ {}^{R}_{C}S_y \sin\phi & {}^{R}_{C}S_y \cos\phi & {}^{R}_{C}t_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.12}
$$

with the elements of ${}^{R}_{C}T$ being derived from matrices ${}^{R}_{C}A$, ${}^{R}_{C}B$, and ${}^{R}_{C}C$.

3.2.3 Solving for the Ultrasound Image Rotation

In order to use ${}^{R}_{U}B$, the rotation matrix for conversion between the ultrasound image's coordinate frame and the real-world coordinate frame, the value of ϕ is required. For our research group's future goal of a robotic needle insertion assistant robot, or alternatively in a traditional clinical setting, ϕ can be directly and accurately measured via position senors. For this experiment, we do not have access to such sensors and instead will be solving for the angle ϕ using an iterative technique. This angle is found by comparing the results of applying the ultrasound transformation matrix ${}^{R}_{U}T$ to the polynomial ${}^{U}P$ (found in Section 3.1.3), using an initial value of ϕ, to the results of the transformation matrix ${}^{R}_{C}T$ applied to the polynomial ${}^{C}P$ (found in Section 3.1.3). The value of ϕ is iterated in steps of 0.1° and the value that results in the best fit between both curves is chosen. Fig. 3.7 shows the results after the matrices ${}^{R}_{U}T$ and ${}^{R}_{C}T$ have been found[1]. The two matrices allow for the portion

[1] After coregistration, the scaling elements of the transform matricies ${}^{R}_{U}T$ and ${}^{R}_{C}T$ were found to be ${}^{R}_{U}S_x = 0.21 \cdot 10^{-3}$, ${}^{R}_{U}S_y = 0.17 \cdot 10^{-3}$, ${}^{R}_{C}S_x = 0.25 \cdot 10^{-3}$, and ${}^{R}_{C}S_y = 0.25 \cdot 10^{-3}$. The values ${}^{R}_{U}S_x$ and ${}^{R}_{U}S_y$ are the x and y pixel-to-millimeter ratios for the ultrasound image and ${}^{R}_{C}S_x$ and ${}^{R}_{C}S_y$ are the pixel-to-millimeter ratios for the camera image.



(a) Polynomial Curve Coregistration

(b) Coregistration Error

Figure 3.7: Polynomial curve corregistration and error for a single prediction trial.

of the needle seen in the ultrasound image to be converted to the real-world coordinate frame and used as an input for the needle-tissue interaction model.

3.3 Needle-Tissue Interaction Model

In order to predict the curvature of the needle based on observing only a portion of the needle we will use a physical model that incorporates characteristics of both the needle and the phantom tissue. This model is an extension of [66] and is fully presented in [67]. The model is a non-linear needle deflection estimator using only the portion of the needle imaged by ultrasound and converted into the real-world coordinate frame, needle insertion length, and the needle mechanical properties as its inputs. The total inserted length of the needle L is discretized into n non-deformable weightless bars of length ℓ such that $\ell = L/n$, as shown in Fig. 3.8. Let us define θ_i as the relative angle between a bar segment of the needle $i-1$ and the subsequent bar segment i.

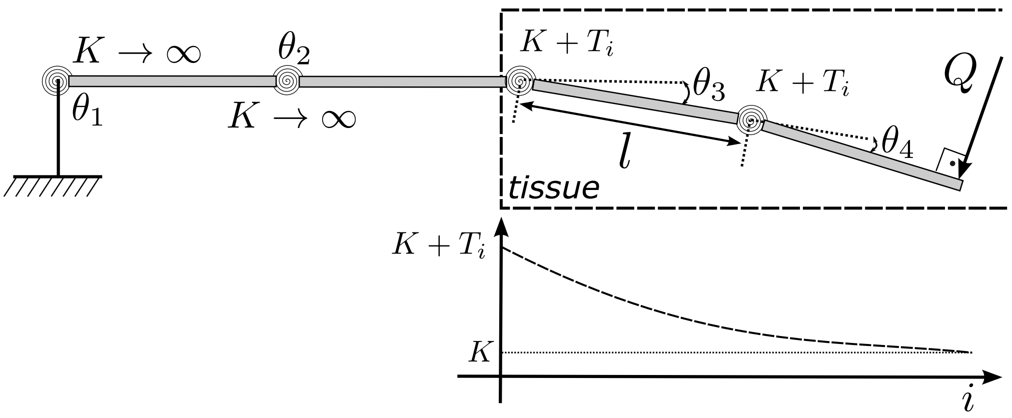


Figure 3.8: Needle-tissue interaction lumped model. The length of the needle is represented by discrete weightless bars ℓ

Two neighboring bar segments $i-1$ and i are considered to be connected by helical springs, all of which have the stiffness K, given by

$$K = \frac{EI}{L}n \tag{3.13}$$

where E and I are the needle Young's modulus of elasticity and its second moment of inertia. As the needle is inserted into the tissue, a force Q is applied to the tip of the needle perpendicular to last segment, such that the needle bends. As the needle bends during insertion it increasingly compresses the tissue located under the needle, this in turn causes the tissue to apply a reaction force back onto the needle. The actions of these forces are modeled as helical springs of stiffness T_i, which is added to the original spring stiffness of K. Together, these two stiffnesses resist the deflection of the needle. This results in the helical springs inside the tissue having a stiffness of $K + T_i$; conversely for the springs joining the needle segments outside of the tissue $T_i = 0$. The stiffness T_i is modeled as

$$T_i = k_0[\ell(n-i+1)]^{k_1} \tag{3.14}$$

where k_0 and k_1 are variables that depend on tissue parameters. Together k_0, k_1, and Q represent the model parameters that will be found interactively using a pattern search algorithm implemented in Matlab.

To obtain the needle deflection we consider the torque τ_i generated on each joint i. The torque τ_i is a result of the effect of the needle tip force Q, and its influence at each joint is $\tau_i = QG_i$, where $1 \leq i \leq n$ and G_i is a Jacobian transformation. The components G_i of the Jacobian matrix relating the force Q to the relative angles τ_i as $\tau_i = G_i F_t$ are

$$G_i = \ell \left(1 + \sum_{p=i}^{n} \cos\theta_p - \cos\theta_i \right) \cos\left(\sum_{p=i}^{n} \theta_p - \theta_i \right) \tag{3.15}$$

Thus, the relative angular displacement for each of the segments can be calculated as $\theta_i = \tau_i / K_i$. Finally, we can calculate the Cartesian position of each helical spring in the transverse (v_i) and axial (u_i) directions (with respect to the unbent needle) as

$$v_i = v_{i-1} + \ell \sin\left(\sum_{p=1}^{i} \theta_p \right) \tag{3.16}$$

$$u_i = u_{i-1} + \ell \cos\left(\sum_{p=1}^{i} \theta_p \right) \tag{3.17}$$

Solving for the needle's total deflection requires knowledge of the needle tip force Q and the tissue reaction force as modeled by k_0 and k_1, from (3.14). Since these equations cannot be solved analytically, we need to use an iterative method to find a solution. In order to do that the polynomial coefficients of the needle from the ultrasound image are used to generate a series of points in the real-world frame, originally from ${}^{C}P$, that represent a portion of the needle. The points in the real-world frame will be known as ${}^{R}P_U$ having coordinates (${}^{R}x_U$, ${}^{R}y_U$) in the axial and transverse direction and are found by taking points from the polynomial (3.2) and transforming them through ${}^{R}_{U}T$ such that a new polynomial is created that describes the deflection seen by the ultrasound probe in the real frame, given by

$${}^{R}y_U = \gamma_2({}^{R}x_U)^2 + \gamma_1({}^{R}x_U) + \gamma_0 \tag{3.18}$$

where the parameters γ_2, γ_1, γ_0 are fit to the transformed points by linear least squares, as after RANSAC this polynomial is now noise free.

The values of Q, k_0, and k_1 are found through minimizing the following multivariate cost function using a pattern search algorithm

$$C(Q, k_0, k_1) = \sum_{i=j_1}^{j_2} \left| v_i(Q, k_0, k_1) - {}^{R}y_{Ui} \right| \qquad (3.19)$$

where ${}^{R}y_{Ui}$ are discrete sample points taken from (3.18), such that ${}^{R}x_{Ui} = u_i$ for every element i that is in the portion of the needle simulation corresponding to the same portion of the needle imaged under ultrasound; thus $j_1 \leq i \leq j_2$, with j_1 and j_2 being the boundaries on the left and right side of the needle portion respectively. $v_i(Q, k_0, k_1)$ is the simulated needle deflection at the corresponding axial position u_i for a given set of Q, k_0, and k_1. The values of Q, k_0, and k_1 are iteratively changed to minimize the cost function and thus find the model parameters. With these parameters found, the equations of the system can be used to predict the total needle shape.

3.4 Experimental Results

The setup for the experiment to simulate a prostate brachytherapy procedure is shown in Fig. 3.9. An 18-gauge by 200 mm prostate seeding needle (Eckert & Zielger BEBIG GmbH, Berlin, Germany) is inserted through a brachytherapy template (Model D0240018BK, C. R. Bard, Inc., Covington, USA) into a tissue phantom. The tissue phantom is a plastisol gel, consisting of a one to four ratio of liquid plastic to plastic softener from M-F Manufacturing Co, Fort Worth, USA. The physical characteristics of the plastisol gel mimic human tissue with the added benefit that the phantom tissue transmits both visible light and ultrasound waves, allowing simultaneous capturing of images in both modalities. A portion of the needle is imaged by a 4DL14-5/38 Linear 4D transducer (different from the transrectal ultrasound probe shown in Fig. 1.1) which is connected to an SonixTouch ultrasound machine (Ultrasonix, Richmond, Canada). For this experiment, we only use the 2D functionality of the ultrasound probe, as very few clinics have access to a 3D ultrasound probe. The entire needle is imaged from above by a XCD-SX90CR video camera (Sony Corporation of America, New York, USA). The prostate seeding

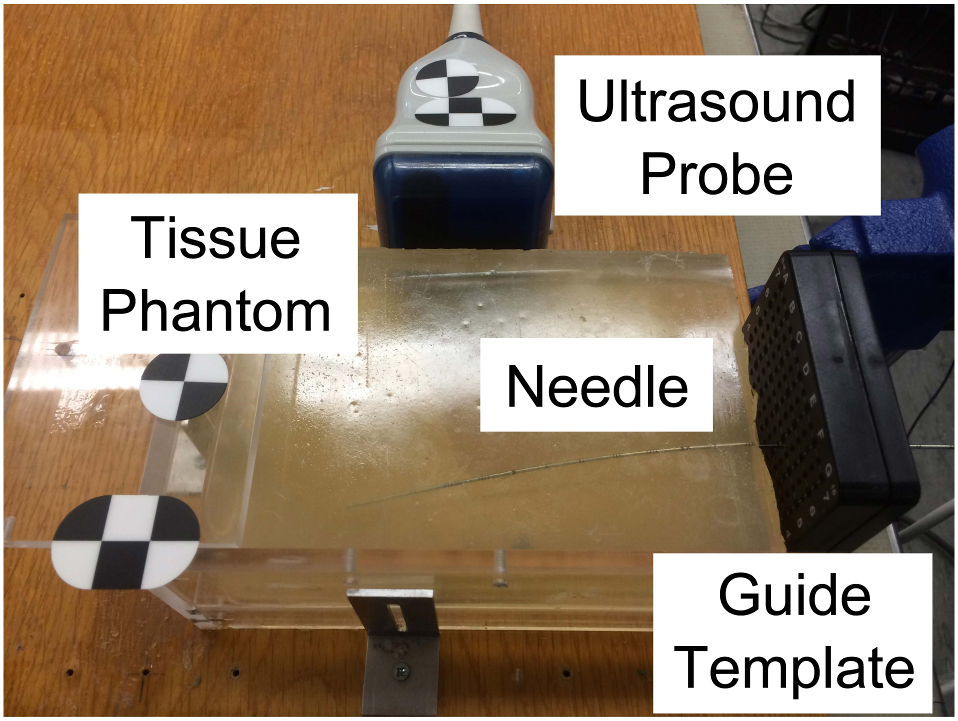


Figure 3.9: Brachytherapy procedure simulation setup, as seen in the camera frame, including ultrasound probe, phantom tissue, needle template, needle, and markers.

needle is inserted such that it deflects in the plane imaged simultaneously by the camera and the ultrasound probe. Due to the process by which ultrasound and camera images are captured, the ultrasound probe does not interfere with the camera's field of view even while they both image the same plane.

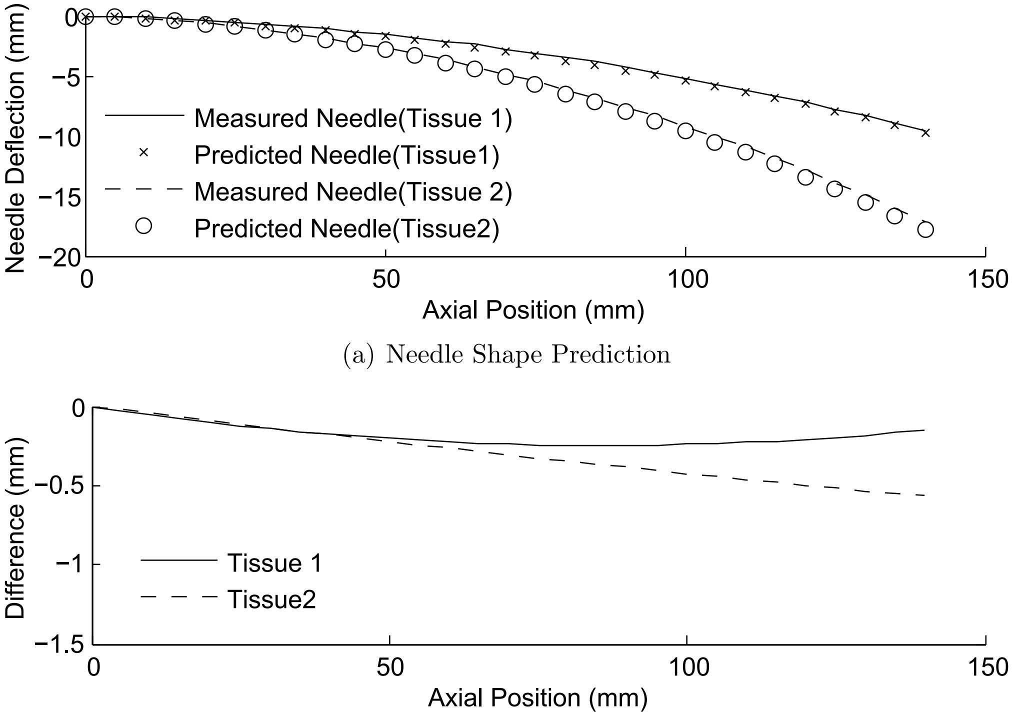
Two different tissue phantoms are used, so called Tissue 1 and Tissue 2, where Tissue 2 is considerably stiffer than Tissue 1. Tissue 1 was created using a four to one ratio of liquid plastic to softener while Tissue 2 was created using a five to one ratio. With the needle inserted into the tissue phantom, at an insertion depth of 140 mm, 5 images from the camera and ultrasound are taken simultaneously. Between images. the orientation of the ultrasound probe ϕ is changed. The images are input into the aforementioned image processing algorithms (Section 3.1) in order to calculate the needle shape polynomials. The ultrasound polynomial equation that contains the partial observation of the needle is subsequently inserted into the needle-tissue interaction model in order to predict the entire needle shape. The predicted needle shape is

Table 3.1: Plastisol A Results

Exp #	Probe Distance from Guide	Probe Orientation	Measured Deflection	Model-Prediction Deflection	Absolute Deflection Error
1	40 mm	2.1°	9.6 mm	9.7 mm	0.1 mm
2	17 mm	1.4°	10.1 mm	11.4 mm	1.3 mm
3	28 mm	4.1°	10.0 mm	12.2 mm	2.2 mm
4	66 mm	1.5°	9.0 mm	13.4 mm	4.4 mm
5	66 mm	1.7°	8.7 mm	10.2 mm	1.5 mm
Avg			9.5 mm	11.4 mm	1.9 mm

Table 3.2: Plastisol B Results

Exp #	Probe Distance from Guide	Probe Orientation	Measured Deflection	Model-Prediction Deflection	Absolute Deflection Error
1	14 mm	1.8°	17.6 mm	12.5 mm	5.1 mm
2	19 mm	4.5°	17.3 mm	18.9 mm	1.6 mm
3	36 mm	0.3°	17.5 mm	23.0 mm	5.6 mm
4	15 mm	-0.8°	17.3 mm	17.8 mm	0.5mm
5	2 mm	11.9°	17.2 mm	21.0 mm	3.8 mm
Avg			17.4 mm	18.5 mm	3.32 mm



(a) Needle Shape Prediction

(b) Needle Shape Prediction Error

Figure 3.10: Sample Needle Shape Prediction with Errors

compared with the measured needle shape as found by the camera in Fig. 3.10. Table 3.1 and Table 3.2 summarize the needle shape prediction results for each of the 5 trials in both of the tissue phantoms, through comparing needle tip deflection. Of note in these results, is that neither the offset location of the ultrasound probe nor the probe orientation affects the needle shape prediction results. When one compares the average measured deflection to the average deflection error both tissues have a prediction error of about 17%, validating that the model is tissue independent.

3.5 Conclusion

In this chapter we have shown the feasibility of predicting the shape of an entire needle when only a portion of it has been imaged via ultrasound. Our method is based on combining image processing, coregistration of different image sources, and a needle-tissue interaction model. The routines described for image processing allow for a needle shape to be found in either ultrasound or camera imaging modalities. We have developed a new method for ultrasound image segmentation that combines thresholding, using Otsu's algorithm, and RANSAC. This combination allows for the ultrasound imaged needle segment to be succinctly described as a polynomial. In addition we have extended this technique to be able to segment a needle inserted into semi-transparent tissue imaged by a camera.

As a general tool to help validate a needle-tissue interaction model, we have devised a method for coregistering ultrasound and camera images using affine transformations. To the best of the authors' knowledge, such a technique has not been presented in the literature. It has the advantage that it is both mathematically straightforward and computationally efficient. Potential applications of our method to medical robotics include needle pose estimation and simultaneous camera and ultrasound visual servoing.

The needle-tissue interaction model outlined here is used to predict the entire shape of the needle based on partial observation. This model only requires simple mechanical characteristics of the needle and is independent of tissue characteristics and needle-tissue force interactions. Furthermore, the input to this model can be provided by either an ultrasound or camera image. The algorithm as outlined by this chapter has been validated in phantom tissue. The experimental results have shown that the needle shape can be estimated with an average accuracy of 2.6 mm, and an maximum accuracy of under 5.6 mm. These preliminary results show the soundness of the underlying partial prediction concept, however further work to reduce experimental errors and increase the robustness of the frame coregistration would need to be done before this work can be translated into a clinical environment.

Chapter 4

Needle Tip-path Prediction from TRUS Axial Image Slices

Building on the needle segmentation and model fitting algorithm presented in the last chapter, the primary goal of this chapter is to combine a kinematic model of the brachytherapy needle with an image processing routine that tracks the needle tip during insertion to build a robust needle shape estimation system. Our proposed algorithm predicts the shape of the entire needle based on the observation of only a portion of the needle with the assumption that the needle shape can be described in a single 2D plane. Using the needle shape information a clinician will then be able to determine if the needle tip position, and subsequently the deposited seed position, will be placed at a satisfactory location. This allows corrective action to be taken without withdrawing and fully reinserting the needle and thereby reducing tissue damage. Additionally, by predicting the needle shape and tip position, the TRUS ultrasound probe can remain stationary after a pre-defined stopping point, thus minimizing the effect of TRUS probe movement on needle shape and final tip displacement. In the future, this needle tracking and estimation routine will be used to inform a needle steering control system.

For tracking and prediction the position of the needle cross-section is segmented out of each of the 2D axial slices, taken normal to the insertion direction of the of the needle. The tracked needle cross section and the distance between consecutive axial slices is incorporated by a particle filter to update the parameters of a non-holonomic kinematic bicycle model for each

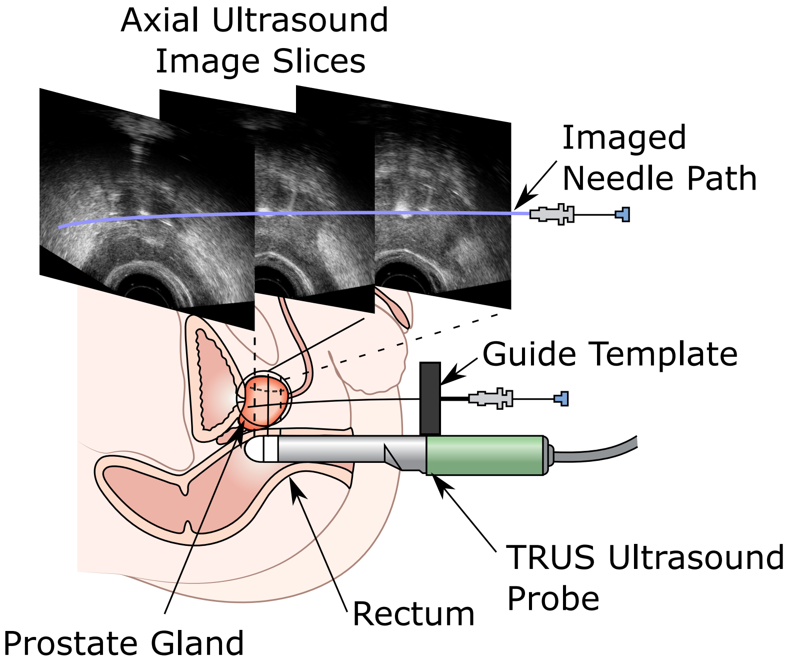


Figure 4.1: Standard brachytherapy setup with TRUS probe and template. Figure modified from original courtesy of Cancer Research UK/Wikimedia Commons.

image slice. Particle filters are widely used in signal processing [90] and image processing [91] [92] to fit parameters and characterize noise in model based systems. As the kinematic parameters are being updated the predicted needle shape is used to augment the needle tracking routine by determining a likely location for the position of the needle cross-section in the current and future axial slices. The error in the predicted needle cross section location is the primary method through which the particle filter updates the parameters of the kinematic model. The kinematic model parameters are updated until a pre-specified insertion depth and then the kinematic model is used to predict the final shape of the fully inserted needle.

This chapter will be organized as follows, Section 4.1 will go over the needle segmentation and tracking routine. Section 4.2 will cover the kinematic bicycle model of a needle, which will be used for needle shape prediction. Section 4.3 contains the implementation of the particle filter that is used to update parameters of the kinematic model of the needle in real time. Section 4.4 gives

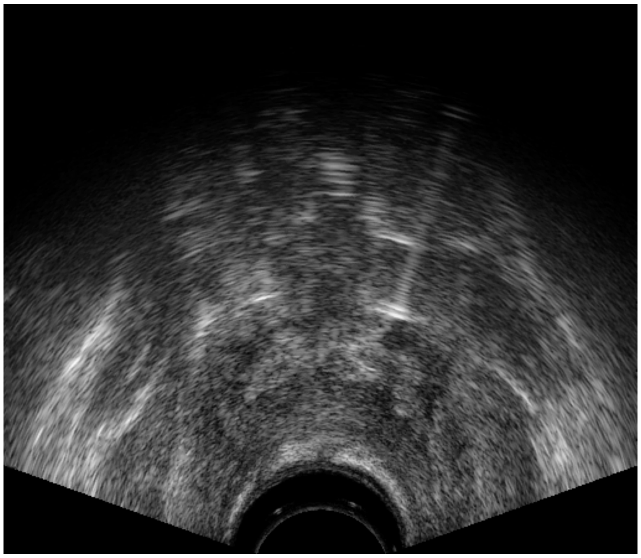


Figure 4.2: Original patient image

the experimental setup to be used for phantom tissue and clinical[1] images, and outlines the results of the algorithm.

4.1 Image Processing

During the clinical and experimental procedures, ultrasound images are taken normal to the needle's insertion direction; see Fig 4.1. The method chosen to process the images is based on the Lucas-Kanade tracking method [93]. Using the results of [94] and [95], an active tracking method was devised in order to incorporate both the current imaging data as well as information from the model prediction.

The clinical axial images, Fig. 4.2, are taken exactly 5 mm apart and the entire image set was captured after the needle was fully inserted into the prostate. For the experimental images, the probe was translated along the direction of needle insertion in order to capture the needle tip cross section at 0.5 mm intervals. Given the assumptions in [42] that the needle path perfectly follows the tip position as it is inserted, the two imaging cases return the same information about needle shape or equivalently needle tip location.

[1] Approval for study granted from Alberta Cancer Research Ethics Committee under reference 25837

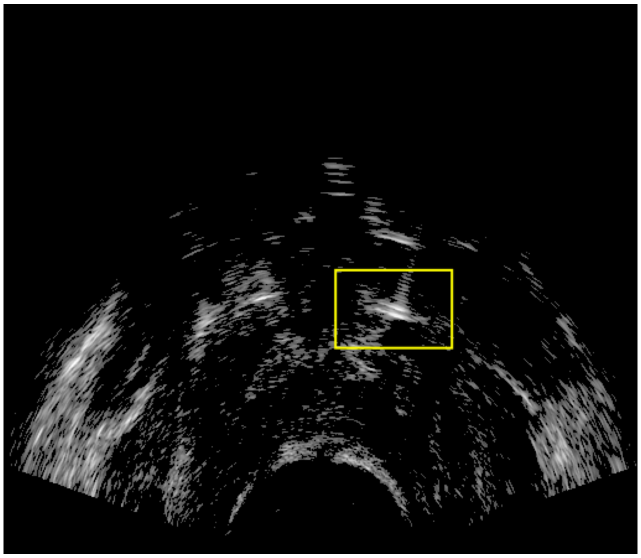


Figure 4.3: Variance image with selected template patch

Our algorithm divides the image processing into a pre-processing phase and a tracking phase. In the pre-processing phase, the entire image is enhanced in order to increase the signal-to-noise ratio. The desired signal is the brightness of the image pixels corresponding to the needle cross section and the noise is the brightness of the background pixels representing the surrounding tissue. After pre-processing the filtered image goes on to the tracking phase. The tracking phase uses a template patch of pixels around the needle cross section in the current image slice to determine the location of the needle cross section in the subsequent image slice. The region of interest that is searched to find the needle cross section is informed by the predicted location of the needle to limit the template matching search area.

Each image I_k is captured at a discrete time step k. Here we will define the pixel brightness, or intensity, to be $I_k(px, py)$ at each point px, py in the image. The pre-processing stage calculates the variance image V_k in order to increase the brightness of the needle cross section pixels with respect to the average background pixel brightness. The formula for intensity of the variance image pixels $V_k(px, py)$ is given as follows,

$$V_k(px, py) = \left(I_k(px, py) - \overline{I_k}\right)^2 \tag{4.1}$$

where $\overline{I_k}$ is the average pixel intensity of the image. The intensity of the variance image was normalized to be within the range from 0 to 1 in our implementation by dividing each of the variance pixels $V_k(px, py)$ by largest value of $V_k(px, py)$ in the variance image. The result of this operation on Fig 4.2 is shown in Fig 4.3.

The next phase of the image processing routine is to localize the needle in each variance image. The center point of the needle cross section will be used as the needle location in the axial image slice and is referred to as I_{ck}. For the first axial image slice, at time step k_0, a user will manually select a point I_{c0} that corresponds to the center of the needle cross section in that slice. A template patch of pixels I_{t0} is created around the needle cross section with height h and width w; see Fig. 4.3. This template I_t is compared to the variance image of the subsequent axial slice using normalized cross correlation (NCC) [95]. The best match, with respect to NCC value, is used to localize the center of the needle cross section, I_{ck}, in that image. Due to the variation in needle cross section intensity and appearance across all of the axial slices the template patch I_t is updated for each frame and consists of the values of the variance image around the center of the needle cross section.

In order to speed up template matching we restrict the NCC calculations to be within a region of interest of height $2h$ and width $2w$. Nominally the region of interest is centred around the kinematic model prediction of the needle cross section location. For the first 10 mm of the needle insertion, however, the region of interest is centred around the previous needle cross section center I_{ck-1}. After the needle is inserted past the 10 mm depth the mean value from the particle filter is used in the kinematic bicycle model to predict the needle shape. The predicted location of the needle cross section in the current axial slice is used as the center of the region of interest. Note that the position of I_{ck} in px, py is scaled using a predetermined $\frac{pixel}{mm}$ ratio for the x and y axis before being input into the particle filtering routine, to be described in the Sec 4.3.

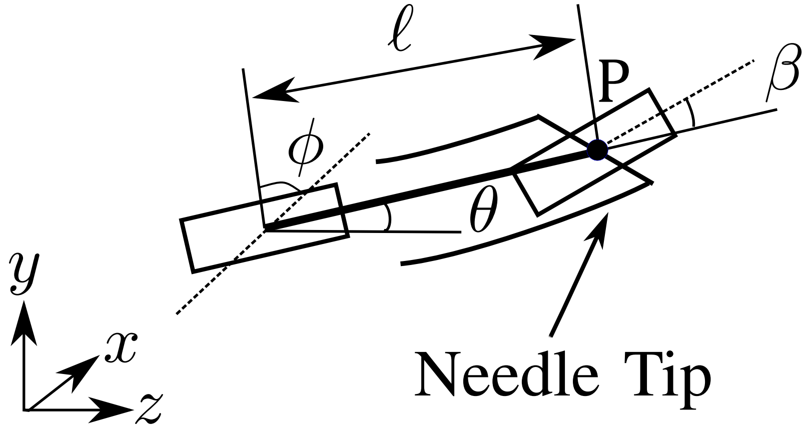


Figure 4.4: Nonholonmic bicycle model of needle motion in 3D

4.2 Kinematic Model

The model to be used in this chapter is the kinematic bicycle model presented in the context of needle modeling in [42] and [46]. This model approximates the motion of a needle cutting through tissue as a constant curvature motion that is dependent on the angle of the bicycles' front wheel β and the distance ℓ between the two bicycle wheels, see Fig 4.4. For use in this chapter, the parameters of the bicycle model will be initialized and updated as described in Section 4.3. Using the state space formulation of the kinematic bicycle model, the states of the system are given by $X_{k|k} = [x, y, z, \theta, \phi]^T$, where x, y and z are the coordinates of the needle tip at time step k, θ is the angle of the needle tip at time k, and ϕ is the rotation angle about the z axis for the plane that contains the needle deflection. The value of ϕ will be considered to be constant throughout the insertion procedure. The control inputs to the state space system are given by $U_{k|k} = [\alpha, \beta]^T$, where α is the needle insertion distance along the z direction per time step. The value of β is the rate of change of needle tip angle θ per time step. We consider the needle bevel angle β to be alternating between a positive and negative constant, corresponding to a rotation of the needle bevel of 180° about the z axis.

The state-space equation of the non-holonomic bicycle model in 3D is given as

$$X_{k+1|k} = I_5 X_{k|k} + \begin{bmatrix} \sin(\theta)\sin(\phi) & 0 \\ \sin(\theta)\cos(\phi) & 0 \\ \cos(\theta) & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} U_{k|k} \tag{4.2}$$

where I_5 is the five-by-five identity matrix. The values of ℓ and β can be used to determine a radius of curvature R about a center of motion defined in general as x_c, y_c, z_c in 3D space, where $R = \frac{\ell}{\sin(\beta)}$. Given the value of ϕ, we are able to define a new 2D coordinate system u, v, Fig 4.5, that will be used to represent the in-plane motion of the needle tip. Here u will be the coordinate that represents the in-plane deflection of the needle shape and v will be parallel to the insertion direction of the needle, making it equivalent to z in the general coordinate system.

The particle filter, outlined in Sec 4.3, requires that the needle shape from the bicycle model be simulated many times per frame. In order to reduce the state-space model into a form that is more computationally efficient we derive a piece-wise solution to the state space equation. This piece-wise solution allows us to predict the needle shape based on the parameters of the bicycle model. The needle shape is given by tracing the path of the needle tip point P as the needle is inserted into tissue. The piece-wise nature of the solution is used to allow simulation of the needle path for an arbitrary number of needle rotations. The equation of motion of the needle tip point P is given by

$$u = u_c(i) \pm \sqrt{R^2 - (v - v_c(i))^2} \tag{4.3}$$

in the u, v coordinate system. Where the values of $u_c(i)$ and $v_c(i)$ are the center of rotation for the circular needle motion. The value i is used to allow for multiple centers of rotation that correspond to rotating the needle by 180° about the v, or equivalently z axis. The values of $u_c(i)$ and $v_c(i)$ can be found through the following formula

$$\begin{aligned} u_c(i) &= \pm R \times \cos(\theta) + u_{turn}(i) \\ v_c(i) &= \pm R \times \sin(\theta) + v_{turn}(i) \end{aligned} \tag{4.4}$$

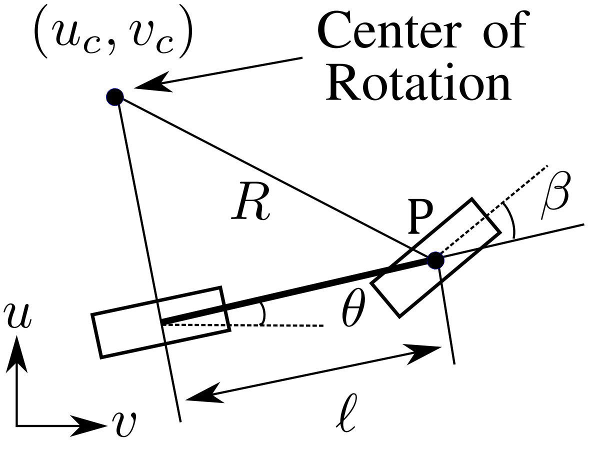


Figure 4.5: Needle Motion around Center of Rotation in u, v Plane

where $u_{turn}(i)$ and $v_{turn}(i)$ are the values of u and v at the moment when the needle is rotated 180°. For $u_c(0)$ and $v_c(0)$ corresponding to center of rotation at needle insertion, and before rotation, the values of $u_{turn}(0) = 0$ and $v_{turn}(0) = 0$. Note that only a singular value of $\pm R$ is used in the formulation, i.e. either a positive or negative R, and that this corresponds to an convex or concave curvature of the needle which will be solved for in the subsection 4.3. The position of the needle tip in the x, y, z coordinate system is

$$\begin{aligned} x &= \cos(\phi) \times u \\ y &= \sin(\phi) \times u \\ z &= v \end{aligned} \tag{4.5}$$

Using this equation we can simplify solving for all of the model parameters into solving for the center of rotation for each time step k, which will be done through the use of the previous described image based needle tracking routine and the particle filter, as will be described in the next section.

4.3 Particle Filtering

A particle filter will be used to solve for the model parameters in Sec. 4.2. Given the low needle-to-speckle contrast inherent in axial TRUS ultrasound imaging, primarily due to poor spatial resolution, the data returned from the image processing algorithm (see Sec 4.1) contains significant noise in the measured values of x and y in each image plane. This noise, or jitter, affects the ability to predict the values of β and θ in the state-space bicycle model and has a large impact on the predicted needle shape. The particle filter uses the piece-wise linear solution of the bicycle model to compare multiple predictions of the needle shape to noisy observations of the needle shape from the image processing routine. The mean value of all of the particle filter predictions is then used to predict the entire shape of the needle at each time step k.

The goal of our estimator is to find the parameters of the model before needle rotation at a specified depth D in the z axis (or equivalently the v axis). From (4.5), it is clear that once we have the value of ϕ, which we solve for using principal component analysis, we are able to formulate a method to fit those parameters on the 2D plane (u, v). We will define the first point the needle passes through as the origin of the coordinate system for the (u, v) plane, and therefore $u_0 = 0$ and $v_0 = 0$. Given that the depth of needle rotation is known, we are able to use (4.3) and (4.4) to formulate the following stochastic model of motion for the path of the needle

$$u = u_c(0) \pm \sqrt{R^2 - (v - v_c(0))^2} \tag{4.6}$$

where

$$\begin{aligned} R &= \sqrt{(u_c(0)^2 + v_c(0)^2)} + \omega_1 \\ \theta_0 &= \tan^{-1}\left(\frac{v_c(0)}{u_c(0)}\right) + \omega_2 \\ \omega_1 &= \mathcal{N}(0, \sigma_1) \\ \omega_2 &= \mathcal{N}(0, \sigma_2) \end{aligned} \tag{4.7}$$

The values of u and v describe the motion of the needle tip point P in the (u, v) plane. Note that $v \leq D$ is the depth of needle rotation, R is the center of circular motion of the needle, θ_0 is the initial value of θ when the needle is first inserted and ω_1 and ω_2 are Gaussian noise with zero mean and standard deviations σ_1 and σ_2 respectively.

The idea behind particle filtering is to use a large number of particles, n, in order to estimate the probability density function of a noise source. A weight is applied to each of the particles $w(n)$ in order to estimate the output of a function at a time k in the presence of noise. Thus here we used particle filtering to find the values of R and θ_0 based on the noisy observations along the needle path at time k. The mean values of R and θ_0 for all of the particles are used to perform needle path prediction.

As before, k is defined as the time that an image is taken and we will consider all points k_0, corresponding to the first image in our set, to k_D, the point in time at which the needle is rotated. The particle filter is updated with every image that is processed (refer to Sec. 4.1) which defines a nominal sample frequency of 20 Hz. Given that the goal of the filtering is to determine the parameters for the needle shape we allow the needle to be inserted by 10 mm, denoted by time k_{10mm} before updating the particle weights so that the initial curvature can be detected. This 10 mm section of points also allows for the value of ϕ to be calculated through PCA so that the all of the observed points in (x, y, z) can be transformed into points p in (u, v). The implementation of our particle filter is as follows:

1. Initialize n particles randomly with each particle containing a pair of values $R(n)$ and $\theta_0(n)$. A Gaussian random distribution for the particles shall be used where $R(n) \sim \mathcal{N}(\mu_R, \sigma_R)$ and $\theta_0(n) \sim \mathcal{N}(\mu_{\theta_0}, \sigma_{\theta_0})$, for user defined values of μ_R, σ_R, μ_{θ_0}, and σ_{θ_0}

2. At time step k_{10mm}, choose p equidistant points, with respect to their insertion depth v, that are in the set $\{k_0 : k_{10mm}\}$

3. For each of the n particles determine the predicted value of u, denoted by $\hat{u}$, for the corresponding depth v for all of the points p

4. For each particle n determine the L^1 normed distance between the predicted value and the actual value, $L^1(u) = |u - \hat{u}|$

5. Find the values of $R(n)$ and $\theta_0(n)$ that have the minimum L^1 norm for each point p

6. Take the mean value of those minimum $\overline{R(p)}$ and $\overline{\theta_0(p)}$

7. Using a gain factor γ, update all other particles

$$R(n) = R(n) - \gamma(R(n) - \overline{R(p)})$$

$$\theta_0(n) = \theta_0(n) - \gamma(\theta_0(n) - \overline{\theta_0(p)})$$

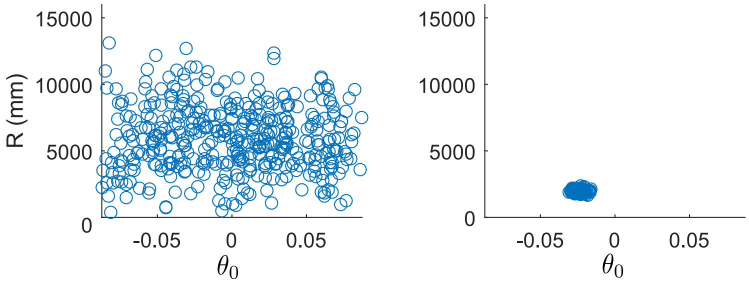


Figure 4.6: Inital distribution of R and θ_0 particles and particle convergence after 5 iterations.

8. Calculate the mean and standard deviation of $R(n)$ and $\theta_0(n)$,these are the new values of values of μ_R, σ_R, μ_{θ_0}, and σ_{θ_0}

9. Remove c particles and replace them with c new particles distributed according to the new mean and standard deviation of $R(n)$ and $\theta_0(n)$

10. Take the data from the next step k when it is available and from the set $\{k_0 : k\}$ choose p equidistant points and return to line 3; if $k = k_D$ then terminate returning the mean values, μ_R and μ_{θ_0}

For both the experimental and clinical image sets the values of $n = 500$, $p = 5$, and $c = 3$ were used. The Gaussian distributions for the intial randomization of the particles were chosen to be $R(n) \sim \mathcal{N}(1500, 600)$ and $\theta_0(n) \sim \mathcal{N}(0, 5)$. For this algorithm, we have based our weighting updating methodology on the work of [92] such that we avoid calculating the a priori and a posteriori probability distributions of ω_1 and ω_2 in order to update the particle weights. Instead, we modify the distributions of $R(n) \sim \mathcal{N}(\mu_R, \sigma_R)$ and $\theta_0(n) \sim \mathcal{N}(\mu_{\theta_0}, \sigma_{\theta_0})$ directly from needle shape observation. One of the particular advantages to this method is that by calculating the means, μ_R and μ_{θ_0}, and standard deviations, σ_R and σ_{θ_0}, of our particle set we can evaluate a confidence in our prediction in that the smaller the standard deviation the closer to an ideal value the mean is. Figure 4.6 shows the convergence of the filter particles at 5 iterations after k_{10mm}, Fig 4.7 shows the decreasing error of the tip prediction as the filter is iterated from k_{10mm} to the rotation depth $k_D = 80mm$.

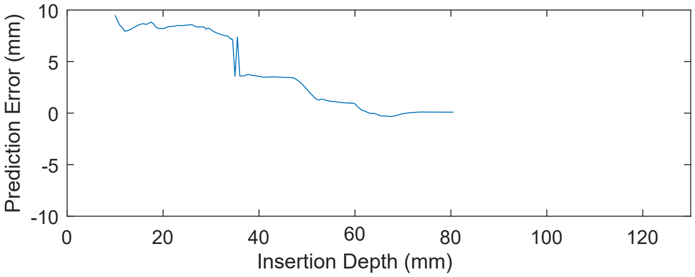


Figure 4.7: Tip prediction error decreases as needle is inserted into tissue.

4.4 Results

The ultrasound used for phantom tissue was an Ultrasonix Touch with a 4DL14-5/38 Linear 4D transducer (Ultrasonix Corp, Richmond, BC, Canada). The clinical ultrasound used for the intra-procedure scans is a Sonoline Adara TRUS scanner with an Endo PII probe (Siemens Medical Solutions USA Inc., Malvern, PA, USA). For both the clinical procedure and the phantom tissue insertions the needles used were 18-gauge 200 mm prostate seeding needles. (Eckert & Zielger BEBIG GmbH, Berlin, Germany).

The ex-vivo tissue phantom was created using beef chuck encased in gelatin. Beef was used to create a non-homogeneous tissue phantom that more closely represents the in-vivo human tissue in both mechanical properties as well as ultrasound imaging characteristics.

The first set of results corresponds to the validation of the particle filtering routine and model after a pre-defined rotation depth (which was not captured in the clinical data); to this end, the needle was inserted into the phantom tissue and rotated at a specified depth. Fig 4.8 shows the measured needle points and needle shape prediction for 6 needles, which were rotated at a depths of 60 mm and 80 mm.

The second set of results, shown in Fig. 4.9, corresponds to the validation of the image processing routine along with particle filtering in clinical data. 6 insertions were used to test that the image processing in particular was successful in tracking the needle position in TRUS images as well as to test the ability for the model to perform successful prediction in real human tissue. The same number of particles and noise parameters for particle initialization

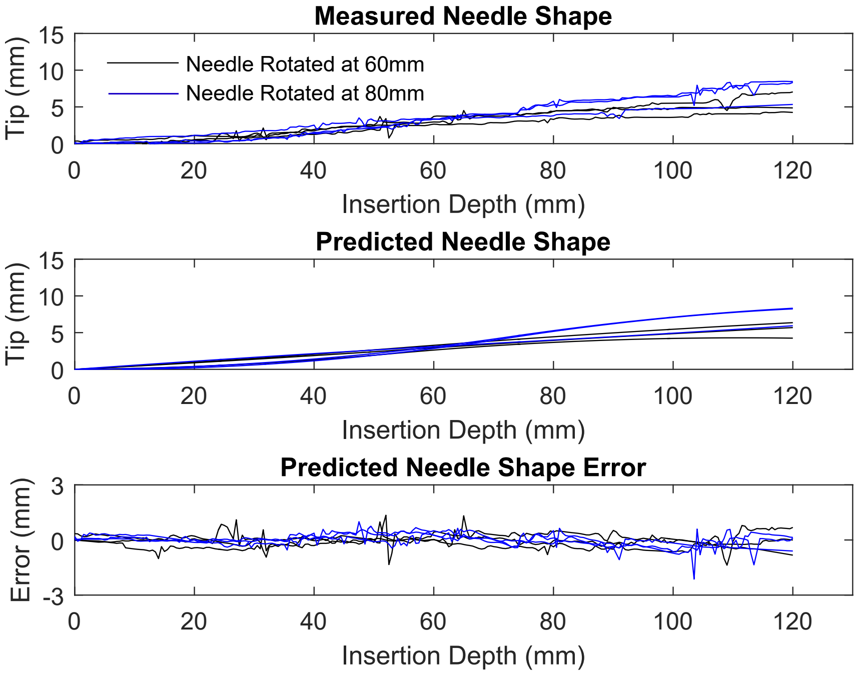


Figure 4.8: Needle shape prediction vs. acutal needle shape for needles rotated at 60 mm and 80 mm in beef phantom.

were used for both set of results. Being as there was no rotation in the clinical data set the first 35 mm on insertion data was used to predict the entire needle shape.

Two metrics were used to compare the accuracy of the prediction. The first metric was tip error, the difference between the final predicted tip displacement $\hat{u}$ and the measured tip displacement in u, where $\text{TipError} = |u - \hat{u}|$. The second metric to evaluate the shape prediction accuracy is the shape error that compares the absolute areas of the measured and predicted needle shapes in mm^2. The shape error is calculated using the following formula

$$\text{ShapeError} = \int |u - \hat{u}| dv \tag{4.8}$$

Using our two metrics of prediction accuracy, in Table 4.1 we can see that the more axial image information the particle filter incorporates, corresponding to a larger rotation depth, the more accurate the needle shape prediction. The needle shape errors in the clinical data show that the kinematic model

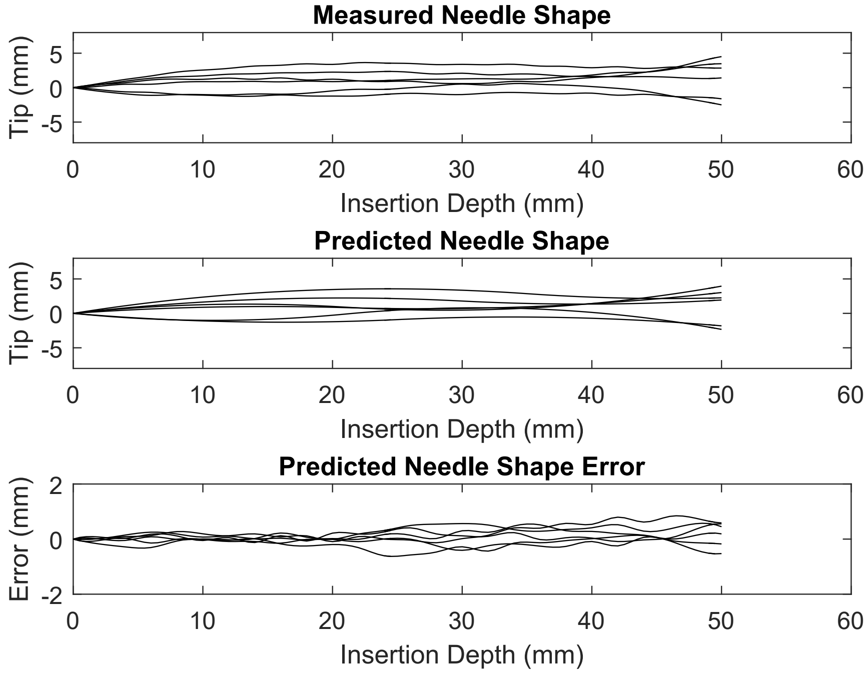


Figure 4.9: Needle shape prediction vs. acutal needle shape for clinical image dataset.

Table 4.1: Needle Path Prediction Accuracy

Data Set	Tip Prediction Error (mm)	Shape Prediction Error (mm‘^{2})
Beef D60	0.497 ± 0.38	105.0 ± 15.1
Beef D80	0.125 ± 0.02	18.4 ± 5.0
Patient Data	0.44 ± 0.15	19.1 ± 9.6

Table 4.2: Image segmentation and particle filtering update time

Data Set	Image Processing Time (ms/frame)	Particle Filter Update Time (ms/frame)	Total Time (ms/frame)
Beef D60	3.1 ± 0.2	2.8 ± 0.3	5.9 ± 0.4
Beef D80	3.1 ± 0.2	2.6 ± 0.4	5.7 ± 0.4
Patient Data	4.1 ± 0.3	2.4 ± 0.3	6.5 ± 0.3

and particle filtering routine work as well on real human tissue as they do on phantom tissue.

The image processing and prediction routines were both coded in Matlab 2015b (The Mathworks Inc, Natwick, MA, USA) and ran on a single core of a Intel Core i7-3930K running at 3.20 GHz (Intel Corporation, Santa Clara, CA, USA). The values of Table 4.2 show that the image segmentation and particle filter update routines run in real-time in both the ex-vivo and in-vivo datasets. The peak total processing time, which corresponded to the clinical image set[2], was 9.5 ms. This peak value is well under the 50 ms total time available for image processing and particle filtering, corresponding to the 20 Hz image update frequency. As can be seen from the combined results, our proposed image guided particle filtering routine was able to successfully predict needle shape of a needle inserted into either phantom or ex-vivo tissue in real-time with average accuracies of 0.31 (±0.27) mm and 0.44 (±0.15) mm respectively.

4.5 Conclusion

In this chapter, we have shown that combining a needle-segmentation routine with a kinematic bicycle model allows for the entire needle shape prediction based only on the observation of a portion of the needle as it is inserted. This routine works equally well on ex-vivo beef based phantom tissue and in-vivo clinical images. Normalized cross correlation is used in a template matching route to capture needle position information in axial image slices. This needle position information is input into a particle filtering routine in order to extract the parameters needed to predict the needle shape with a kinematic bicycle model. Both the final needle tip position and needle shape are compared with the measured needle shape to validate the prediction. In general the needle tip position can be predicted with an average error of less than 0.5 mm in both phantom and ex-vivo tissue. This needle shape prediction is also used on-line to augment the needle segmentation process by defining the center of a region of interest for the template matching routine. In the next chapter

[2] The input US images used for the results were 444 x 512 pixels in size

this work will be used to inform a needle steering control routine in order to reduce needle tip deflection as the needle is inserted.

Chapter 5

Needle Deflection Control in Surgeon-in-the-loop Paradigm based on real-time Ultrasound Image Feedback

Incorporating the needle-tip tracking algorithm given in the previous chapter, the next step focuses on reducing the deflection of the needle tip during the entire insertion; this is often referred to as needle deflection regulation in the literature [7]. Incorporating a 3D version of the bicycle model and by using a handheld motorized needle steering device [6] that acts as a robotic assistant to the surgeon, the base of the needle is automatically rotated during insertion to correct for needle deflection. This rotation changes the orientation of the needle bevel and thus the direction of needle deflection. In the context of this chapter, regulation of needle tip deflection to zero across the entire length of needle insertion is desirable. Eliminating deflection during the entire insertion length is ideal for procedures such as prostate brachytherapy because it ensures both that the needle tip will accurately reach the desired target location and that all seeds loaded into each needle are deposited on or near the desired straight line.

In this chapter, we will propose and evaluate this needle deflection controller designed using a reduced-order 3D nonholonomic bicycle model; see Fig. 5.1. The controller is based on a switching continuous-time control law and is shown to asymptotically reduce the needle tip deflection to zero. This

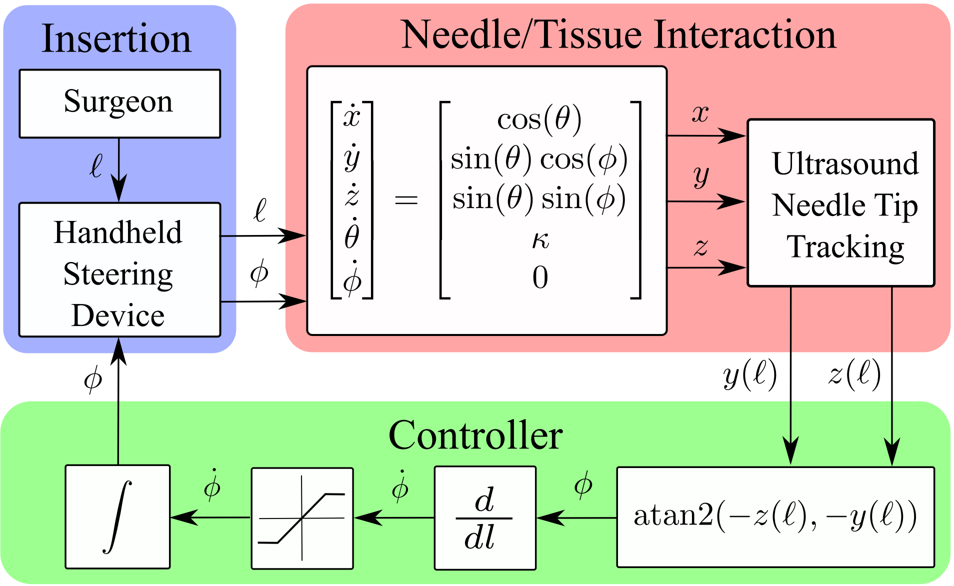

Insertion
Surgeon
ℓ
Handheld Steering Device
ℓ
ϕ
Needle/Tissue Interaction
$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\\ \dot{\theta}\\ \dot{\phi}\end{bmatrix} = \begin{bmatrix}\cos(\theta)\\ \sin(\theta)\cos(\phi)\\ \sin(\theta)\sin(\phi)\\ \kappa\\ 0\end{bmatrix}$
x
y
z
Ultrasound Needle Tip Tracking
$y(\ell)$
$z(\ell)$
ϕ
Controller
$\int$
$\dot{\phi}$
$\dot{\phi}$
$\frac{d}{dl}$
ϕ
atan2$(-z(\ell), -y(\ell))$

Figure 5.1: Control loop diagram showing the controller outlined in this chapter. Note that the value of $\dot{\phi}$ is tested to see if it is above the saturation limit; if it is beneath this limit, then the value of ϕ is sent directly to the handheld device to avoid differentiation noise

controller is designed within the context of surgeon-in-the-loop needle insertion, where the clinician manually inserts the needle and the controller reduces the needle deflection away from the target in 3D. The performance of the controller is evaluated with a constraint on the maximum output needle rotation velocity to show that acceptable performance is achieved under practical considerations of patient safety which would otherwise force a clinician to slow down the insertion.

The rest of this chapter is organized as follows. Related work and an overview of the reduced nonholonomic bicycle model is outlined in Sec. 5.1. The controller is developed in Sec. 5.2. The experimental setup for evaluating the controller is presented in Sec. 5.3 with the results of the ex-vivo tissue trials using the handheld device given in Sec. 5.4. The conclusions and future work are summarized in Sec. 5.5.

5.1 Background

The bicycle model is typically applied to the control of mobile robots. However, [41] and [42] modified it to describe the motion of a beveled-tip needle as it is inserted into tissue. This model has been used quite often in the literature for needle trajectory control. Research in [42] applied this needle deflection model to control a needle directed toward a target in a single 2D plane. One complication of using the bicycle model is that it is non-trivial to estimate the system states in real-time control. Our previous work [2] developed a particle filter based technique for state estimation of a 2D variable plane bicycle model from clinical ultrasound images in real-time and [45] showed that an Unscented Kalman Filter may be used for 3D state estimation. Using modern control techniques, the researchers in [46] developed a linearizing transform for a 2D version of the bicycle model and implemented it in an observer-controller combination. Sliding mode control has also proven very effective at the combination of state-estimation and control applied to the bicycle model with [48] implementing 2D and [49] implementing 3D sliding mode controllers. One contribution of the controller presented in this chapter is that it is proved to asymptotically converge based only on needle deflection measurements and does not require any state estimation.

The bicycle model has also been used for trajectory control in 3D by modeling the needle deflection over small insertion lengths as being contained in a number of 2D planes with [37] and [50] developing trajectory controllers to compensate for 3D target motion and account for 3D obstacle motion inside tissue. This is similar to the work of [96] where ultrasound image tracking was used along with the bicycle model for needle steering in the presence of tissue motion. The constant curvature model has also been augmented with mechanical needle characteristics in [97] for control of the needle trajectory. This work was extended to consider the interaction of needle and tissue mechanics by [61] and [44]. While not unique to this work, the controller presented here is capable of 3D needle control without constraining the needle motion to be in planar segments and with negligible computational time to calculate the

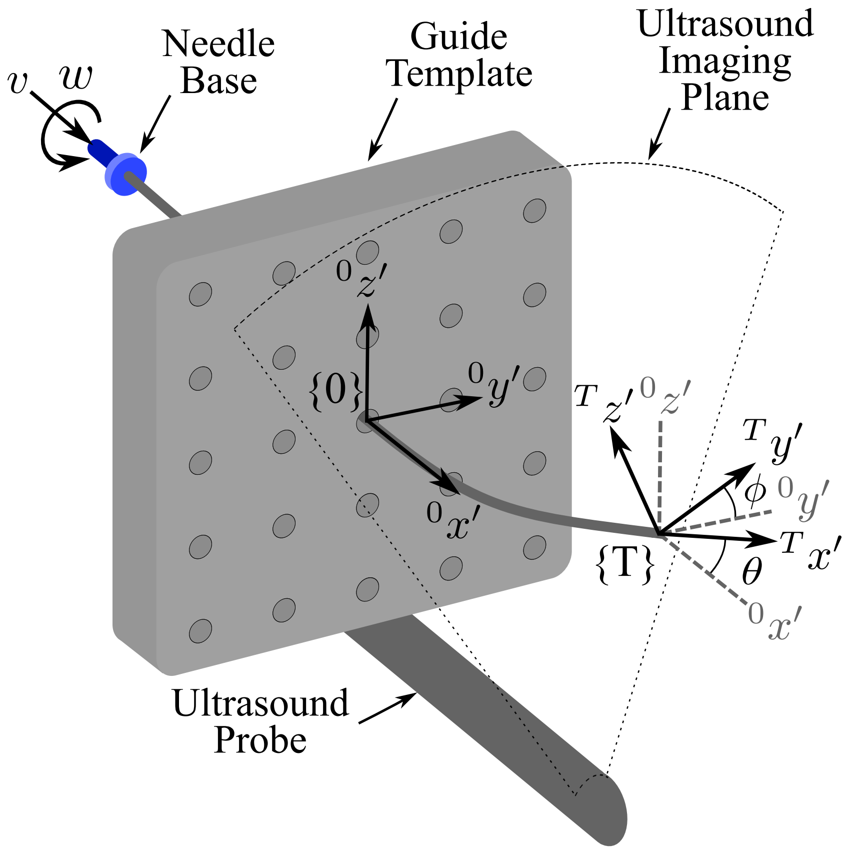


Figure 5.2: Coordinate system used for kinematic model of needle motion, showing TRUS probe, axial imaging plane, control inputs and needle base, and guide template.

control output when compared to mechanical model-based control.

This chapter will use a reduced-order nonholonomic model originally presented in [5] for control of a needle in 3D by feedback-linearization in a Frenet-Serret frame. This work will build on this model and use it for feedback control of the needle tip deflection in a general 3D frame. This model assumes that the needle is stiff in both torsional and compressional directions such that insertion and rotation at the base of the needle are conveyed directly to the needle tip.

5.1.1 Reduced Order Bicycle Model

From the work of [5], the derivation of the reduced-order model uses an inertial frame $\{0\}$ anchored at the point of needle insertion into tissue and a frame

$\{T\}$ that is rigidly attached to the tip of the needle as shown in Fig. 5.2. At the beginning of insertion, the two frames coincide. The needle base velocity along the direction of insertion is denoted as v, and rotational velocity on the needle base is indicated by w, where $v, w \in \mathbb{R}$. The axes of the inertial frame, labeled $[{}^0x', {}^0y', {}^0z']$, are stationary at the point of needle exit out of the guide template and define the coordinates of the 3D space. The tip-attached frame $\{T\}$ is placed such that the origin of the frame is at the needle tip, with the axes of this frame denoted by $[{}^Tx', {}^Ty', {}^Tz']$. During insertion, the insertion base velocity v will translate the needle tip along the ${}^Tx'$ axis. The position of the needle tip (and equivalently the origin of frame $\{T\}$), with respect to the base frame $\{0\}$, is given by $p = [x, y, z] \in \mathbb{R}^3$. The Tait-Bryan angle definition of rotation about the Z-Y-X axes, involving the angles ψ, θ, ϕ, is used to represent the rotational orientation of frame $\{T\}$ with respect to frame $\{0\}$.

In this model, we assume that the needle is torsionally stiff such that rotation of the needle base is conveyed directly to the needle tip. From this torsional stiffness, a rotation at the base of the needle with an input rotational velocity of w, about the ${}^0x'$ axis, is the same as directly rotating the needle bevel implying the tip-attached frame rotates about ${}^Tx'$ with the same input w velocity. The angle of rotation about ${}^Tx'$, with respect to the orientation of $\{0\}$, is defined as ϕ where $\{\phi \in \mathbb{R} : -\pi \leq \phi < \pi\}$.

The angle ϕ is related to the physical orientation of the asymmetric bevel on the needle-tip and, in an analogous manner to the work of [41] and [42], we model the needle-tissue interaction as causing a constant "turning" action of the needle tip frame. Here, we define $\{T\}$ to be attached such that the needle bending is modeled as a rotation of the frame $\{T\}$ about the axis ${}^Ty'$. The angle θ, where $\{\theta \in \mathbb{R} : 0 \leq \theta < \pi\}$, represents the angle of this rotation. The rate of change in the angle θ is defined as the variable κ, where κ represents the inverse of the radius of curvature of the needle and is constant during insertion ($\kappa = \frac{1}{R}$). Thus controlling the angle of ϕ changes the orientation of the ${}^Ty'$ axis and steers the needle.

With the torsional stiffness assumption implying direct control of the needle bevel about ${}^Tx'$ and by defining needle bending as a rotation about ${}^Ty'$ it follows that there is no action which will cause the needle tip to be rotated about the ${}^{T}z'$ axis. This stiffness assumption is valid, as shown in [5], for the 18-gauge stainless steel needles used in this work as well as in clinical brachytherapy. This assumption may not hold for substantially thinner or more flexible needles such as those made out of nitinol (where effects such as torsional windup can be modeled as causing a rotation about ${}^{T}z'$). We thus consider the angle of rotation about the ${}^{T}z'$ axis to be zero in this model, therefore $\psi = 0$, and so the system reduces to 5-DoF. The states of the reduced-order system are then given as $X = [x, y, z, \theta, \phi]$.

A full derivation of the model is given in [5] along with proofs of accessibility and controllability. As the goal is to use a handheld device for surgeon-in-the-loop cooperative needle insertion, we would prefer to have our model, and therefore the control action, written as a function of the inserted length of the needle ℓ rather than insertion time. To do this, motivated by [46], a division of the entire system by the needle base velocity v is performed, noting that $v = \frac{d\ell}{dt}$. The result is a depth-dependent nonholonomic system described by

$$\dot{X} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos(\theta) \\ \sin(\theta)\cos(\phi) \\ \sin(\theta)\sin(\phi) \\ \kappa \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} w \tag{5.1}$$

Here, in a slight abuse of notation which we shall use throughout the rest of this chapter, we use the dot operator to denote the derivative with respect to inserted needle length rather than time, such that $\dot{X} = \frac{dX}{d\ell}$. Using this depth-dependent model, we will design a controller to minimize the needle tip deflection during insertion.

5.2 Switching Controller

With the depth-dependent kinematic needle model outlined in Sec. 5.1, we will design a switching control law to regulate the needle deflection away from the target ${}^{0}x'$ axis to zero, thus minimizing $e(\ell)$ at all depths during insertion

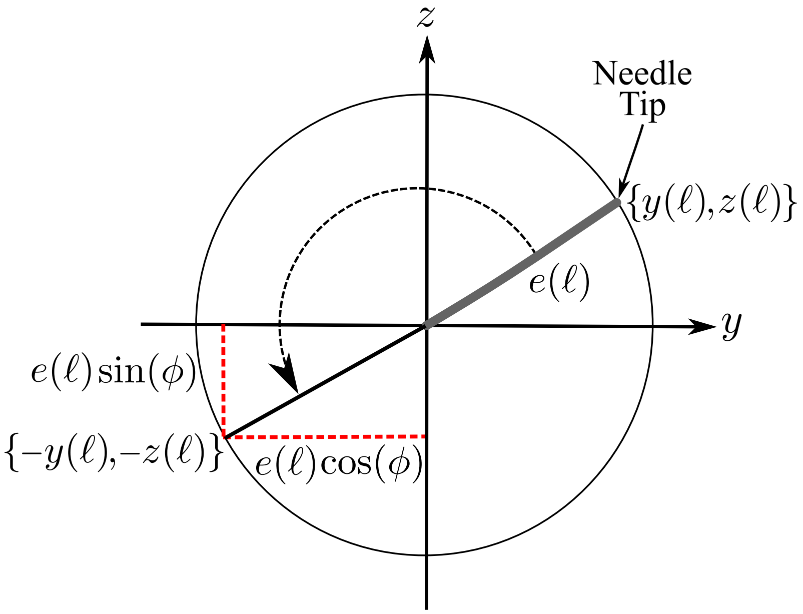


Figure 5.3: Axial view (along x-axis viewed from the needle tip) of needle deflection with switching control action.

where

$$e(\ell) = \sqrt{y(\ell)^2 + z(\ell)^2} \tag{5.2}$$

Here, $y(\ell)$ and $z(\ell)$ are the needle tip deflections along the ${}^0y'$ and ${}^0z'$ axes respectively. This deflection measure will be used to design and evaluate the controller performance in the system. Here, the surgeon is manually inserting the needle and therefore directly controlling ℓ. With reference to Sec. 5.1.1, as the needle is being inserted it translates the needle tip-attached frame $\{T\}$ forward along the ${}^Tz'$ axis. The surgeon will stop the insertion when the needle tip has reached the desired target depth on the ${}^0x'$ axis, thus completely controlling the position of the needle tip p in 3D space.

From (5.1), it is clear that one of the major advantages of this model over the standard 6-DoF model presented in the literature is that we have the ability to directly design the value of $\phi(\ell)$ (as the control action) at a particular insertion depth, rather than having to design its derivative $\dot{\phi}(\ell)$. Consider the control law

$$\phi(\ell) = \text{atan2}(-z(\ell), -y(\ell)) \tag{5.3}$$

where atan2 is the four-quadrant inverse tangent function. One important note is that the reduced-order model, like the general 6-DoF model, violates Brockett's condition [98] and so cannot be stabilized with a smooth continuous controller. Thus, the atan2 function acts in this context as a switching controller. With the goal of bringing the needle tip deflection in the ${}^{0}y' \, {}^{0}z'$-plane to zero, we shall look at a positive-definite Lyapunov function candidate $V(y, z)$ given as

$$V(y, z) = \frac{1}{2}y(\ell)^2 + \frac{1}{2}z(\ell)^2 \tag{5.4}$$

The derivative of this function is

$$\dot{V}(y, z) = \dot{y}(\ell)y(\ell) + \dot{z}(\ell)z(\ell) \tag{5.5}$$

Which, with the control law given in (5.3), will be shown to be negative-definite, i.e., $\dot{V}(0, 0) = 0$ and $\dot{V}(x, y) < 0$, $\forall y, z \in \mathbb{R}$. By substituting our control law ϕ into the above, we have

$$\dot{V}(y, z) = sin(\theta(\ell)) \cos(\phi(\ell))y(\ell) + sin(\theta(\ell)) \sin(\phi(\ell))z(\ell) \tag{5.6}$$

Here $sin(\theta(\ell))$ is a positive number during insertion as $0 \leqslant \theta(\ell) < \pi$ from the coordinate system definition and with $\dot{\theta} = \kappa$, where κ is a positive number. Taking the current needle tip deflection as $e(\ell) = \sqrt{y(\ell)^2 + z(\ell)^2}$ we can use the trigonometric definitions

$$\cos(\phi(\ell)) = \frac{-y(\ell)}{e(\ell)} \tag{5.7}$$

$$\sin(\phi(\ell)) = \frac{-z(\ell)}{e(\ell)} \tag{5.8}$$

and substitute these into (5.6) resulting in

$$\begin{aligned} \dot{V}(y, z) &= sin(\theta(\ell)) \left(\frac{-y(\ell)^2}{e(\ell)} + \frac{-z(\ell)^2}{e(\ell)} \right) \\ &= sin(\theta(\ell)) \left(-\frac{y(\ell)^2 + z(\ell)^2}{e(\ell)} \right) \\ &= -sin(\theta(\ell)) \sqrt{y(\ell)^2 + z(\ell)^2} \end{aligned} \tag{5.9}$$

which is a negative definite function. Therefore, using this controller, the tip needle deflection asymptotically converges to zero. As shown in the experimental results, Sec. 5.4, this proof of convergence is conservative as in practice the system can reduce and regulate needle deflection over short needle insertion distances.

5.2.1 Practical Switching Controller

It is clear from the formulation of the switching controller that the control output $\phi(\ell)$ instantaneously turns the needle bevel 180 degrees. Implementing the controller would then require a sufficiently high rotational velocity $\dot{\phi}(\ell)$ to achieve near-ideal performance. The handheld needle steering device, or any other device used to implement the control action, will have some maximum rotation speed limit. Here, and throughout this section when referring to rotation we use speed to imply rotation in the time domain ($\frac{rad}{s}$) and velocity to refer to rotation in the insertion-depth domain ($\frac{rad}{mm}$). To implement large rotation velocities required by the ideal controller the needle insertion can be stopped or slowed during the rotation action. Thus the rotation speed of the system with respect to change in insertion depth will result in a large, or infinite, rotation velocity $\dot{\phi}(\ell)$. For this, the surgeon could be signaled to temporarily stop or slow insertion to meet the rotational velocity requirements. This would have the effect of lengthening the procedure time and would be relativity impractical.

Instead of relying on the surgeon to control insertion speed, we will modify the controller to limit the required rotation velocity. This is beneficial for practical implementation of the system as the reduced rotational velocity requirement will allow for much higher insertion velocities without being limited by the rotation speed of the needle steering device. More importantly, we want to limit the rotation velocity for patient safety. This is to prevent the controller from performing high velocity rotations, corresponding to a large $\dot{\phi}(\ell)$, over the entire insertion length. This would result in a "drilling motion", which would induce unnecessary tissue trauma. Thus we will modify the controller to limit the rotation velocity. In an attempt to bound $\dot{\phi}(\ell)$, we shall incorporate the following control limiting rule where v_{max} is the maximum admissible rotation velocity:

$$\begin{aligned} \dot{\phi}(\ell) &= \mathrm{sat}(v_{max}, \frac{d}{dl}\mathrm{atan2}(-z(\ell), -y(\ell))) \\ &= \mathrm{sat}(v_{max}, \frac{\dot{z}(\ell)y(\ell) - \dot{y}(\ell)z(\ell)}{y(\ell)^2 + z(\ell)^2}) \end{aligned} \tag{5.10}$$

with the definition of the sat function given by $\mathrm{sat}(x, y) = \mathrm{sign}(y)\min(x, \mathrm{abs}(y))$.

Integrating $\dot{\phi}(\ell)$ will give the actual control input to be applied to the system. Note that in the case that $|\dot{\phi}(\ell)| < v_{max}$ then control signal $\phi(\ell) = \text{atan2}(-z(\ell), -y(\ell))$ will be feed directly into the handheld controller to avoid differentiation noise.

To confirm that the controller performance is still acceptable even when $\dot{\phi}(\ell)$ is saturated (the "practical controller"), the closed-loop system was simulated in 3D for various values of v_{max}. One performance measure that was evaluated is the additional insertion length required for the controller to bring the deflection to zero when compared to the insertion length required by the controller in which $\dot{\phi}(\ell)$ was not saturated (the "ideal controller"); this measure is denoted by Δl. The other metric used was the increase in total needle deflection, $e(\ell) = \sqrt{y(\ell)^2 + z(\ell)^2}$, when compared to the ideal controller, denoted by Δe_{max}. Both of these metrics are illustrated in Fig. 5.4 for a simulated needle insertion.

To test the performance of the saturated controller, needle insertion was simulated 10,000 times while varying the values of both v_{max} and θ_0. Here we define the variable θ_0 to be the initial value of θ when the needle is first inserted into tissue, i.e. $\ell = 0$, such that $\theta_0 \equiv \theta(0)$. The value of v_{max} was varied from $0.1\frac{rad}{mm}$ to $0.9\frac{rad}{mm}$ and θ_0 was varied from 0° to 10°. Note that the unit of rotational velocity, $\frac{rad}{mm}$, refers to the amount of needle rotation per mm of inserted needle length as the system model, (5.1), was derived to depend only on insertion depth. For the simulations, the value of κ was kept constant at $\kappa = \frac{1}{1000}$. Previously and during experiments, we had found the values of θ_0 and κ to be inside the above-reported ranges used in simulations. The results of these simulations, shown in Fig. 5.5, indicate that the practical controller performance approaches the performance of the ideal controller as v_{max} is increased. The results also show that the relative performance of the practical controller is not sensitive to the changing values of θ_0. Additionally, this modified controller offers acceptable performance while being limited to low rotation velocities. This implies a manner of insertion-velocity independence where a practical needle steering system, with a finite maximum rotation speed, can provide the necessary rotation velocity for control over a

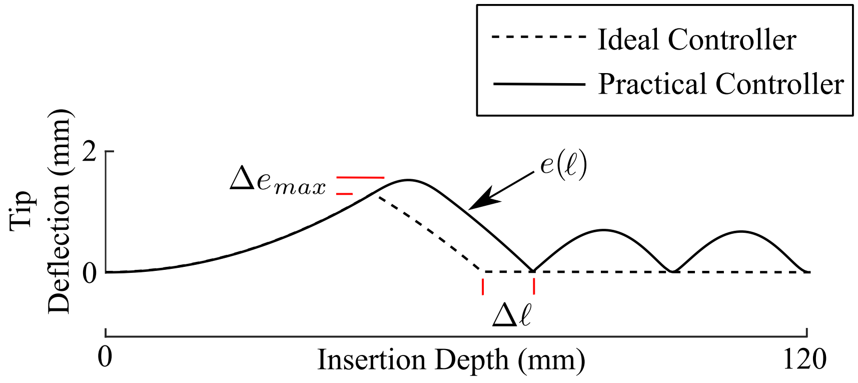


Figure 5.4: Diagram of two performance measures used to evaluate the saturated controller.

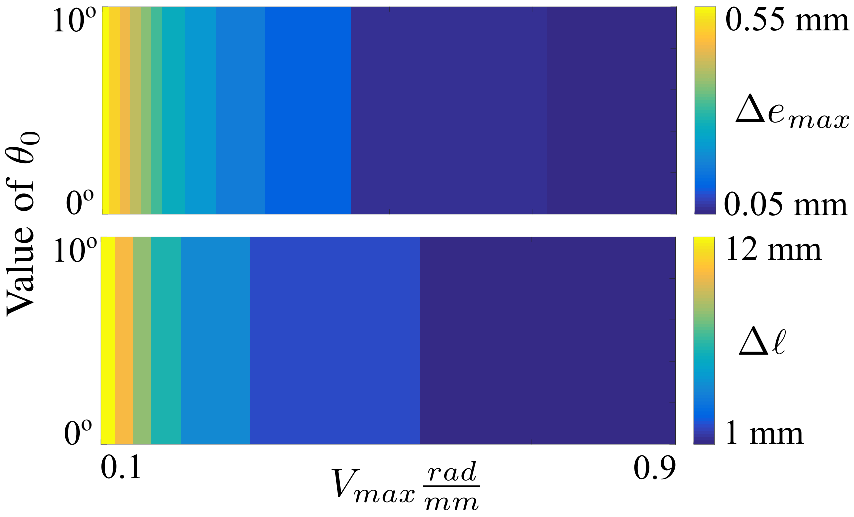


Figure 5.5: Performance of controller when rotational velocity is limited.

broad range of insertion velocities.

5.3 Experimental Setup and Ultrasound Image Processing

Needle insertions into phantom tissue were used to experimentally validate the saturated controller performance. The experimental setup, shown in Fig. 5.6, consisted of an ultrasound probe, prostate brachytherapy needles, three phantom tissues, and the handheld needle steering device originally presented in [6]. For the needles in the experiments, standard 200 mm long prostate brachytherapy 18-gauge seeding needles (Eckert & Zielger BEBIG GmbH, Berlin, Ger-

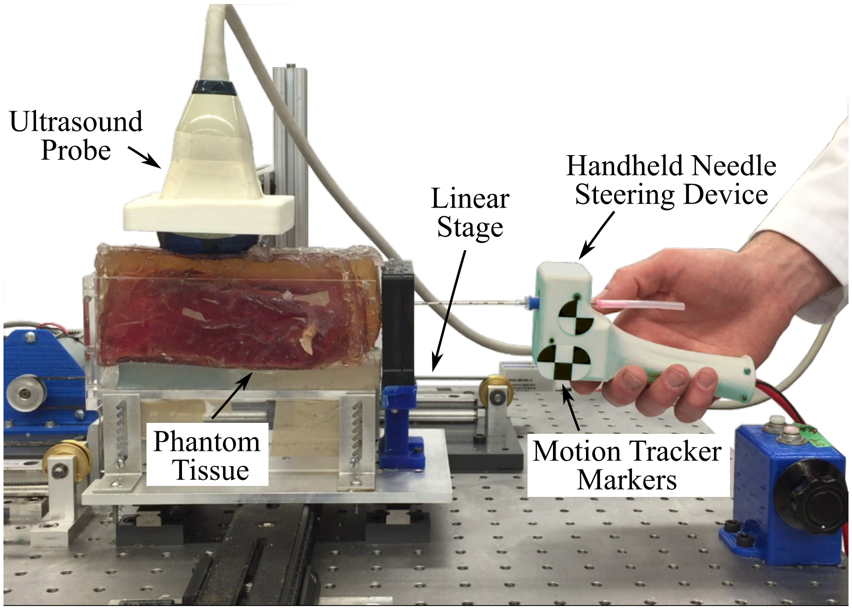
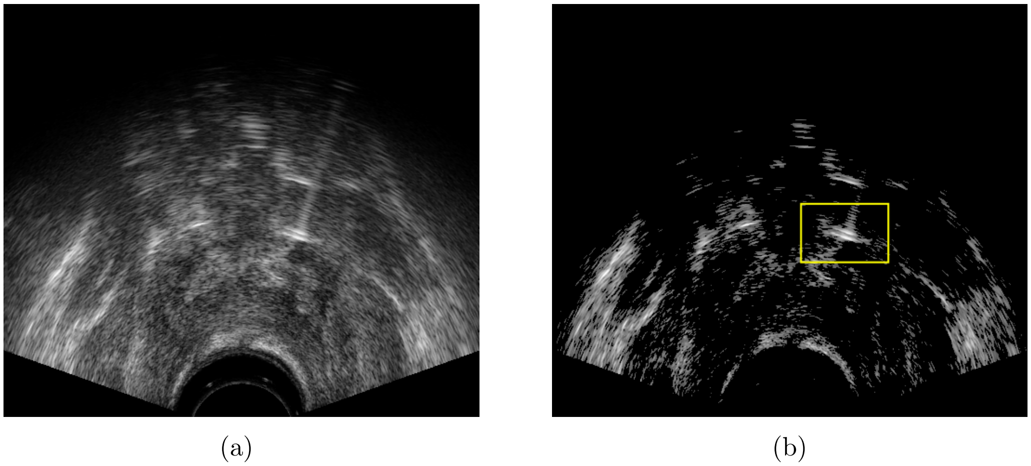


Figure 5.6: Photo of experimental setup showing phantom tissue, handheld needle steering device with motion tracker markers, and ultrasound probe on motorized linear stage.

many) were used.

The ultrasound machine used was an Ultrasonix Touch with a 4DL14-5/38 Linear 4D transducer (Ultrasonix Corp, Richmond, BC, Canada). Measuring the inserted length of the needle is done by using a Micron Tracker (HX60 from Claron Technology Inc., Toronto, ON, Canada) via optical tracking markers placed on the handheld device. The needle base was rotated during insertion by the use of the handheld steering device.

For the ex-vivo tissue, phantoms incorporating three different materials were used. Two biological tissue phantoms contained bovine rump tissue and porcine loin tissue embedded in gelatin (Knox from Kraft Inc., Northfield, IL, USA) to represent non-homogeneous tissue that closely resembles in-vivo human tissue in both mechanical properties as well as ultrasound imaging characteristics. The other tissue phantom is made from a plastisol gel (M-F Manufacturing Co, Fort Worth, USA) that was created to test the response of the controller in materials that are stiffer and have more friction than human tissues (which makes the needle steering more challenging). The image processing and needle control routines were both coded in Matlab 2016a (The



(a) (b)

Figure 5.7: Ultrasound image processing showing original input image and variance image with tracking area.

Mathworks Inc, Natwick, MA, USA) and were run on a single core of an Intel Core i7-3930K processor running at 3.20 GHz (Intel Corporation, Santa Clara, CA, USA).

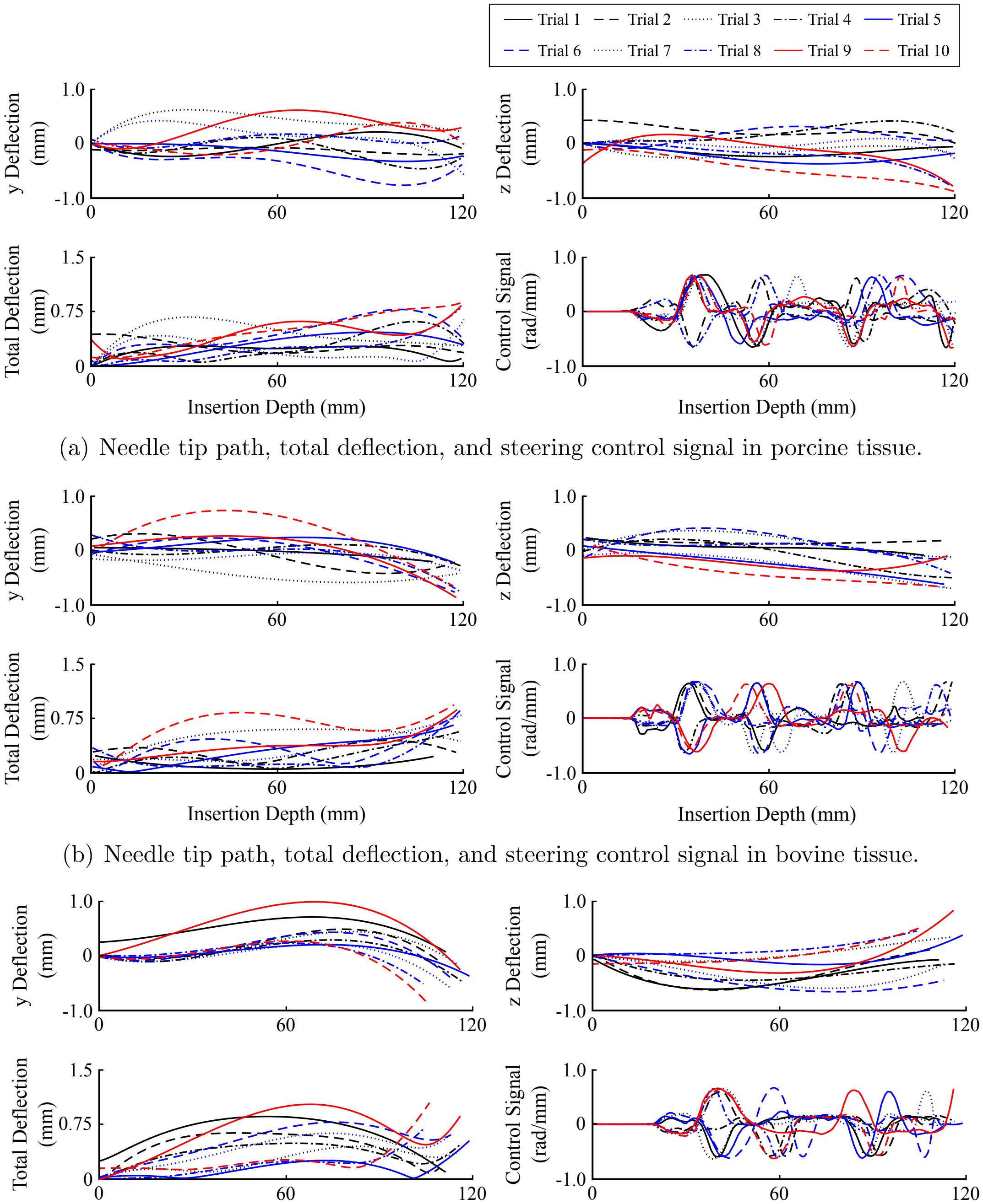
For these experiments, 2D ultrasound images were used to track the needle tip during insertion. The needle tracking was done using a slightly modified version of the needle tracking algorithm presented in [1]. During insertion, ultrasound images are captured and processed in real-time (see Fig. 5.7(a)) at a frame rate of 20 Hz. Each ultrasound image is enhanced to increase the brightness of the image pixels corresponding to the needle cross section and to reduce the intensity of the background pixels representing the surrounding tissue.

In the first frame of the live ultrasound image sequence, corresponding to when the needle is just inserted, a user clicks on the center of the needle cross-section in the ultrasound image. The needle cross-section, in this work capturing the needle tip, is tracked using a template patch of pixels around it. Based on concepts from video tracking the cross-section location in the next ultrasound image frame is calculated by using normalized cross correlation (NCC) along with the assumption that the needle tip motion between frames is small. The result of the image processing and needle tip tracking patch is seen in Fig. 5.7(b). During insertion, the ultrasound imaging probe translates forward along the axis of needle insertion ${}^0x'$, using the motion tracker measurement of ℓ, so that the needle tip is always captured in the image plane. The pixel coordinate output of the tracking routine is then converted to the needle tip location in the ${}^0y'\,{}^0z'$-plane in real-world mm coordinates. The ultrasound image pixel coordinates are calibrated to provide accurate real-world measurements throughout the entire insertion. The calibration is performed by taking four image points with measured real-world coordinates, manually selected in two ultrasound image slices at insertion depths of 10 mm and 100 mm respectively. This image-metric calibration is done individually for each of the three tissues to compensate for speed of sound differences.

For the controller, only information about the current needle tip coordinates $y(\ell)$ and $z(\ell)$ is required. One complication, however, is that the needle tip position as returned from the ultrasound images will contain a large amount of positional noise (or jitter) due to the low spatial resolution of ultrasound images and signal-to-noise similarities between the needle and background tissue. In order to combat this, we employ a simple noise filtering routine using a linear Kalman filter on the positional data to estimate the true needle position in the current frame.

5.4 Results

For each of the three previously described tissue phantoms, ten insertion trials were completed, and the efficacy of the controller was evaluated by looking at two metrics. For each trial the target location was chosen to correspond to real-world ${}^0y'\,{}^0z'$-plane coordinates of the center of a hole in the guide template, see Fig. 5.2, at a depth of 120 mm. The needle was allowed to deflect in an uncontrolled manner for the first 20 mm of insertion to evaluate the efficacy of the controller at regulating needle deflection in the presence of some initial deflection from the target axis. Additionally, the results presented here show the real-world deflection in mm, measured via calibrated ultrasound images. The needle deflection is shown relative to the target axis and any initial needle tip offset from the target axis, caused by tissue motion during insertion, was



(a) Needle tip path, total deflection, and steering control signal in porcine tissue.

(b) Needle tip path, total deflection, and steering control signal in bovine tissue.

(c) Needle tip path, total deflection, and steering control signal in plastisol tissue.

Figure 5.8: Results of needle insertion experiments in the three phantom tissues.

not compensated for to provide a fair comparison with the insertion errors seen in current clinical practice.

Once the needle had been inserted to a depth of 20 mm the control output was applied and the handheld device rotated the needle during the remaining 100 mm of insertion. For these results, the rotational velocity of the system was constrained to be under ±0.65 rad/mm using the methodology developed in Sec. 5.2.1. A buzzer was used to indicate to the user that the target depth had been reached. The insertion speed of the needle was not controlled and the insertion depth was captured in real-time using the Micron Tracker.

The needle tip position in the ${}^{0}y'\,{}^{0}z'$-plane, measured in ultrasound images, was used as the feedback to the controller. The two metrics used to evaluate the efficacy of the controller are based on the Euclidean distance of the needle tip away from the target ${}^{0}x'$-axis. The first metric is based on the definition of $e(\ell)$, given in (5.2), and evaluated the deflection of the needle tip when fully inserted into tissue, given by

$$\text{Final Tip Deflection} = \sqrt{y(L)^2 + z(L)^2} \tag{5.11}$$

where L is the depth of the needle tip at the end of the insertion. Given that the goal of the controller is to limit needle tip deflection during the entire insertion length, the second measure is defined as

$$\text{Mean Tip Deflection} = \sum_{i=0}^{L} \frac{\sqrt{y(\ell_i)^2 + z(\ell_i)^2}}{n} \tag{5.12}$$

where $y(i)$ and $z(i)$ denote the needle tip deflection measured in an ultrasound image frame at a discrete depth ℓ_i during insertion and n is the total number of discrete depth points captured during insertion. For the Mean Tip Deflection results given in Table 5.1 the $y(\ell_i)$ and $z(\ell_i)$ points found in the ultrasound image slice nearest to 100 equally spaced points along the ${}^{0}x'$-axis (from 0 mm to final insertion depth) were used, such that $n = 100$. The result of the needle insertion experiments is given in Table 5.1, with the values in each row corresponding to the averaged final tip deflection and averaged mean tip deflection, along with respective standard deviations for the ten trials in each

Table 5.1: Experimental Results

Tissue Phantom	Final Tip Deflection (mm)	Mean Tip Deflection (mm)	Maximum Final Tip Deflection (mm)	Maximum Mean Tip Deflection (mm)
Porcine	0.47 (±0.29)	0.35 (±0.10)	0.90	0.55
Bovine	0.64 (±0.25)	0.51 (±0.19)	0.93	0.52
Plastisol	0.51 (±0.27)	0.39 (±0.15)	1.04	0.57

tissue. The largest values across the 10 trials are given as the maximum final tip deflection and maximum mean insertion tip deflection. The measured needle deflection for each of the insertions was plotted with respect to insertion depth. The experimental data was smoothed by fitting the measured deflection for the entire insertion to a 3rd order polynomial which was constrained such that the polynomial curve exactly matched experimental deflection measured at the initial insertion depth (0 mm) and the final insertion depth. The results for the porcine tissue phantom are given in Fig. 5.8(a), the bovine tissue phantom in Fig. 5.8(b), and the plastisol tissue phantom in Fig. 5.8(c).

In the context of surgeon-in-the-loop needle steering, an important criterion is that the controller is robust to changes in insertion speed. To this end, insertion speed was not controlled during the experimental trials but was measured for each trail. For the porcine tissue insertion the maximum insertion velocity measured was 41 mm/s with an average insertion velocity of 9.5 mm/s, for the bovine tissue the maximum insertion velocity measured was 52 mm/s with an average insertion velocity of 9.4 mm/s, and for the plastisol tissue the maximum insertion velocity measured was 35 mm/s with an average insertion velocity of 8.5 mm/s.

From results in Table 5.1, we can see that the performance of the controller is relatively insensitive to the tissue characteristics with all 30 insertions having an average final tip deflection of 0.54 ± 0.27 mm and an average mean tip deflection of 0.42 ± 0.15 mm. Of note is the modestly higher mean tip deflection and maximum final tip deflection in plastisol, which is primarily due to the unrealistically high values of needle-tissue friction and stiffness in this tissue that increases the needle deflection during insertion. The maximum target error of 1.04 mm is still below the average targeting error of 1.22 reported in a survery of 26 other 2D and 3D needle steering algorithms in the literature [7].

5.5 Conclusion

In this chapter, we have presented a surgeon-in-the-loop needle steering controller for percutaneous procedures that is designed to regulate to zero the needle tip deflection. The controller was based on a reduced-order kinematic bicycle model and has been modified to depend only on the inserted depth of the needle, which can readily be measured externally, using needle tip position feedback from an ultrasound imaging device that will not requiring significant changes to the current clinical operating procedure. The rotation of the needle tip, corresponding to the model control input, was done using a light-weight and clinician-friendly handheld device that allows the surgeon to fully carry out the needle insertion and decide the final insertion depth.

The controller has been shown to asymptotically bring the needle deflection error to zero in the ideal case. A practical variant of the controller that limits the needle rotation speed for both mechanical and tissue damage reduction effects was implemented. The stability and performance of the practical controller was evaluated both in simulation as well as experimentally in ex-vivo tissue phantoms. The experimental results from the ex-vivo phantoms, created from bovine, porcine, and plastisol tissues, show that the controller is robust with respect to both non-homogeneous tissue as well as varying tissue types. The average targeting error in across all of the needle insertion experiments in the three tissues resulted in an average final needle tip deflection of 0.54 ± 0.27 mm and an average total tip deflection of 0.42 ± 0.15 mm throughout the entire insertion. While this work was presented in the context of prostate brachytherapy, it could be useful for any percutaneous procedure that requires precise needle placement with ultrasound image-guided feedback.

The concepts covered in this chapter will be expanded on in the next chapter, with the development of an improved needle steering controller.

Chapter 6

Surgeon-in-the-loop Needle Steering using a Predictive Event-triggered Controller

This chapter presents a semi-autonomous (surgeon-in-the-loop) ultrasound-image-guided system, which can steer the needle during insertion to increase targeting accuracy. The system is designed such that the clinician directly controls the insertion velocity and depth, in this case by inserting the needle attached to a hand-held device (shown in Fig. 6.1) originally presented in [6]. Real-time ultrasound-image based needle tracking is used to estimate the parameters of a reduced-order nonholonomic kinematic bicycle model. The needle deflection is minimized through the use of an event-triggered control system which optimizes needle rotation during insertion, thus steering the needle. The control system is designed such that, with the addition of the hand-held steering device, it can be incorporated into a standard clinical setup used during prostate brachytherapy without requiring additional sensors or changes to the clinical setup.

The main contribution of this chapter is an analytic solution to the reduced-order bicycle model which allows for calculation of the needle-tip trajectory, for the entire needle insertion, without requiring numerical integration. This analytic solution is shown to be more computationally efficient than traditional discrete-step numerical integration methods for needle-tip trajectory calculation and allows the proposed event-triggered control system to reduce needle

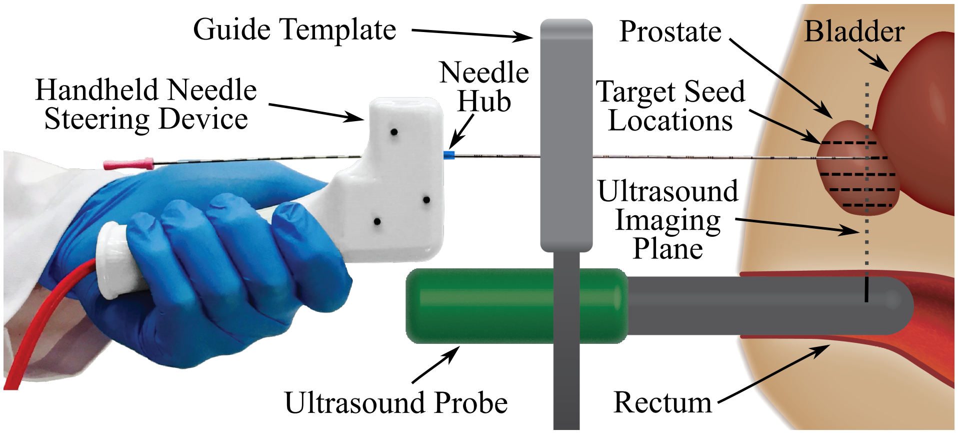


Figure 6.1: Our proposed hand-held device being used in conjunction with a standard clinical brachytherapy setup with transrectal ultrasound (TRUS) probe, axial imaging plane, and guide template.

deflection in real-time.

An overview of related work on ultrasound-image-based needle tracking and control is given in Sec. 6.1. The event-triggered controller utilizing the analytic solution to the reduced-order kinematic bicycle model along with a detailed comparison of the computational speed of the solution are shown in Sec. 6.2. In Sec. 6.3 the setup used to validate the controller, including the hand-held device, along with results of needle insertion trials in three different tissue phantoms is presented. Sec. 6.4 covers the algorithm for needle tracking in axial ultrasound images along with the model parameter fitting. The last Sec. 6.5 gives a discussion of the results obtained along with future challenges to be solved.

6.1 Background

From the last chapter, the reduced order kinematic bicycle model was shown to be capable of reducing needle deflection during insertion in a surgeon-in-the-loop system. This chapter will improve on the control design and will incorporate a predictive needle-tip path component simliar to the ones presented in Chapters 3 and 4.

Several different control paradigms have been devised within the context of needle steering for percutaneous procedures, with a thorough review of

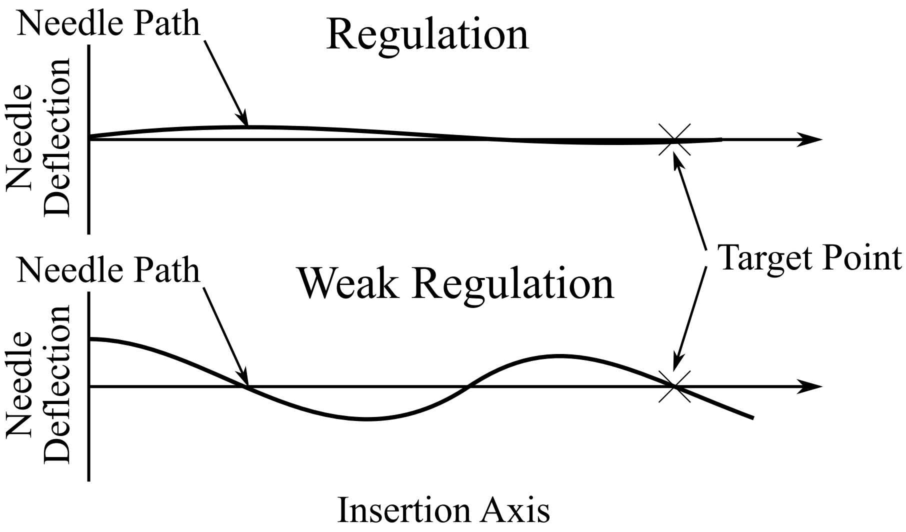


Figure 6.2: Comparison of the regulation and weak regulation control paradigms seen in needle steering control, as described in [7].

the control paradigms and their applications given in[7]. One such control paradigm is referred to in the literature as regulation. For regulation, the controller is designed to eliminate needle-tip deflection away from a target axis throughout the entire insertion and is ideal for procedures such as prostate brachytherapy. Another control paradigm is weak regulation, whereby the needle-tip is only controlled to pass through a desired target point, allowing for larger needle-tip deflection away from the target axis during insertion. Weak regulation is primarily used for procedures such as biopsy but also applicable to brachytherapy. For brachytherapy, weak regulating controllers can be designed to reduce the required number of total needle rotations, and therefore tissue trauma, while maintaining acceptable needle deflection around the target axis. Figure 6.2 illustrates the difference in deflection minimization between weak regulation and regulation.

Event-triggering is a modern control technique, initially used for distributed control systems, that uses discrete events as a trigger to initiate a control action for both continuous-time and discrete-time systems[99]. Event-triggered control theory presents a cohesive framework to design, and analyze the performance of, aperiodic controllers, where stability of a distributed system can be proven using only knowledge of previous control actions and an estimate of the system state[100]. Using the ideas presented in event-triggered control theory, we will derive an analytic solution to a reduced order bicycle model incorporating discrete changes to the needle rotation velocity (the control output) that are triggered based on inserted needle length (during manual needle insertion by a clinician). One contribution of the controller presented in this chapter is that deflection minimization performance can be tuned, through increasing or decreasing the number of event-trigger points, between the weak regulation and regulation paradigms. The number of event-trigger points is directly related to the total number of needle rotations performed by the controller. Thus, by tuning the number of event-trigger points, tissue trauma can be reduced while keeping the needle-tip deflection during insertion at an acceptable level. The event-triggered controller presented in this chapter can be tuned between regulation type performance, for a procedure such as prostate brachytherapy, or weak-regulation performance, for biopsy procedures.

To minimize the needle-tip deflection during insertion, the event-triggered controller requires an estimate of the initial state of the reduced-order bicycle model. This state estimate will be found by tracking the needle-tip path in ultrasound images. Needle tracking, or segmentation, has been performed in the literature on 3D ultrasound volumes but this chapter focuses strictly on needle tracking in a series of 2D images to be more representative of current clinical practice. In 2D ultrasound images, the needle is typically imaged either in the axial plane, a plane orthogonal to the direction of needle insertion (see Fig. 5.2), or the sagittal plane, a plane containing the needle and parallel to the direction of needle insertion. In sagittal plane ultrasound images, the needle appears as a distinct white blob or thick curved line, and segmentation can be performed through the use of Gabor Filtering [20] along with the Hough Transform for straight needles [17] and curved needles [19].

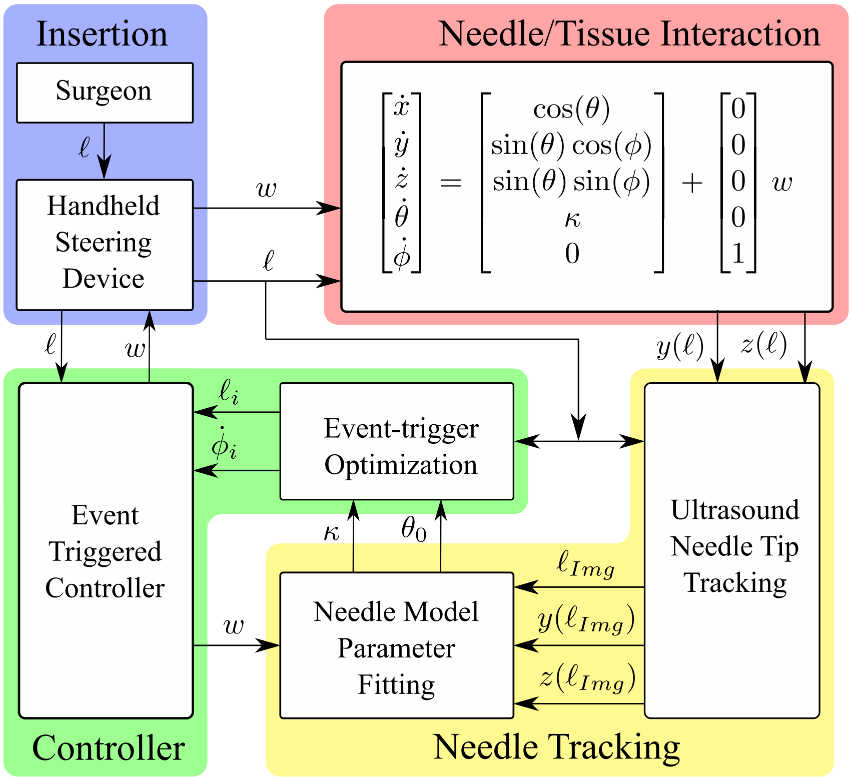

Insertion
Surgeon
ℓ
Handheld Steering Device
w
ℓ
Needle/Tissue Interaction
$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\\ \dot{\theta}\\ \dot{\phi}\end{bmatrix} = \begin{bmatrix}\cos(\theta)\\ \sin(\theta)\cos(\phi)\\ \sin(\theta)\sin(\phi)\\ \kappa\\ 0\end{bmatrix} + \begin{bmatrix}0\\0\\0\\0\\1\end{bmatrix} w$
$y(\ell)$
$z(\ell)$
ℓ
w
ℓ_i
$\dot{\phi}_i$
Event-trigger Optimization
Event Triggered Controller
κ
θ_0
ℓ_{Img}
$y(\ell_{Img})$
$z(\ell_{Img})$
w
Needle Model Parameter Fitting
Ultrasound Needle Tip Tracking
Controller
Needle Tracking

Figure 6.3: Control diagram showing semi-autonomous insertion, needle tip tracking in ultrasound images, model fitting, and event-triggered controller.

6.2 Event-triggered Control

Needle steering in a surgeon-in-the-loop context requires a control algorithm that is both real-time and robust to changes in the needle insertion velocity. This section will present an analytic solution to the 5-DoF reduced order bicycle model and the implementation of an event-triggered controller for needle steering. The control loop for the system is diagrammed in Fig. 6.3. The derived analytic solution of the 5-DoF reduced-order model allows for fast simulation of needle insertion and is ideal in the sense that it does not require a trade-off between numerical accuracy and computational time, as PDE solvers or discrete time-step variants of the kinematic bicycle model typically do.

To implement the 5-DoF bicycle model in a practical control system, while attempting to limit tissue damage, the controller will be designed to reduce needle deflection through slowly varying the needle bevel angle ϕ. Here, in a manner analogous to standard clinical practice, the target location is defined to lie at a specified depth on the ${}^{0}x'$-axis, in line with the direction of insertion. Thus the needle model gives the needle-tip deflection in both the ${}^{0}y'$ and ${}^{0}z'$ axes. To solve the closed-loop system equations, based on event-trigger theory, we constrain the output of the controller, ω, such that it only changes once the insertion length of the needle passes an event-trigger point. The method to optimize the locations of the event-trigger points, with respect to insertion length, is given in Sec. 6.2.2. For this controller, the number of event-trigger points can be chosen empirically to provide the desired controller performance. Using a small number of event-trigger points will minimize needle rotation velocity at the cost of greater needle-tip deflection away from the ${}^{0}x'$-axis during insertion. Using more event-trigger points will minimize needle-tip deflection during insertion but will require larger needle rotation velocities.

From the state-space representation of the model, in (5.1), we start by evaluating the differential state equation $\dot{\theta}$ to derive $\theta(\ell)$, noting that this state is independent of the control input $\dot{\phi} = \omega$. The needle insertion depth is denoted by ℓ, with the needle being inserted through the guide template along the ${}^{0}x'$-axis. The initial insertion length is then given as $\ell = 0$ with a desired final insertion length defined as $\ell = \ell_f$, where $\ell_f > 0$. From (5.1) we can see that $\dot{\kappa} = 0$, thus the solution to this first-order differential equation is

$$\theta(\ell) = \int_0^\ell \dot{\theta}(\ell)\, d\ell = \int_0^\ell \kappa = \kappa\ell\Big|_0^\ell + \theta_0 \tag{6.1}$$

where θ_0 is the initial value of $\theta(\ell)$ at the point of insertion, i.e. $\theta_0 = \theta(0)$. With the use of the ultrasound image processing the value of θ_0 can be found. Thus, using this, we can find the state of the system $\theta(\ell)$ to be

$$\theta(\ell) = \kappa\ell + \theta_0 \tag{6.2}$$

where $\{\ell \in \mathbb{R} : 0 \leq \ell \leq \ell_f\}$. With the solution to $\theta(\ell)$ known, the system state $x(\ell)$ can then be found, such that

$$\begin{aligned} x(\ell) &= \int_0^\ell \cos(\theta)\, d\ell = \int_0^\ell \cos(\kappa\ell)\, d\ell \\ x(\ell) &= \frac{\sin(\kappa\ell + \theta_0)}{\kappa}\Big|_0^\ell + x_0 \end{aligned} \tag{6.3}$$

where, in an identical manner to (6.2), the limit of integration is the current insertion depth ℓ. From the definition of the base frame $\{0\}$, i.e. that it is located at the initial point of insertion, we have $x(0) \equiv 0$, thus $x_0 = 0$ and $x(\ell)$ is then given as

$$x(\ell) = \frac{\sin(\kappa\ell + \theta_0)}{\kappa} - \frac{\sin(\theta_0)}{\kappa} \tag{6.4}$$

throughout the insertion interval $0 \leq \ell \leq \ell_f$.

With solutions to both $\theta(\ell)$ and $x(\ell)$ derived, the next step is to solve for $y(\ell)$ and $z(\ell)$. Unlike $\theta(\ell)$ and $x(\ell)$, our control input $\dot{\phi}(\ell)$ clearly affects the response of $y(\ell)$ and $z(\ell)$. To formulate the response of the system using event-trigger control points we consider the total insertion length to be divided into n sections of arbitrary length where, through controller design, $\dot{\phi}(\ell)$ is constant in each section (i.e. use of zero-order hold). Using subscript notation, we will define the space of $l = \{\ell_0 : \ell_1, \ell_1 : \ell_2, ..., \ell_{n-1} : \ell_n\}$. Here $\ell_0 = 0$, and ℓ_n is our target insertion length ℓ_f. We will allow the value of $\dot{\phi}_i$ to change by some finite amount at the event points ℓ_i between each segment. This then defines our event-triggers as these insertion depth points ℓ_i with corresponding control outputs after the trigger as $\omega = \dot{\phi}_i$; see Fig. 6.4.

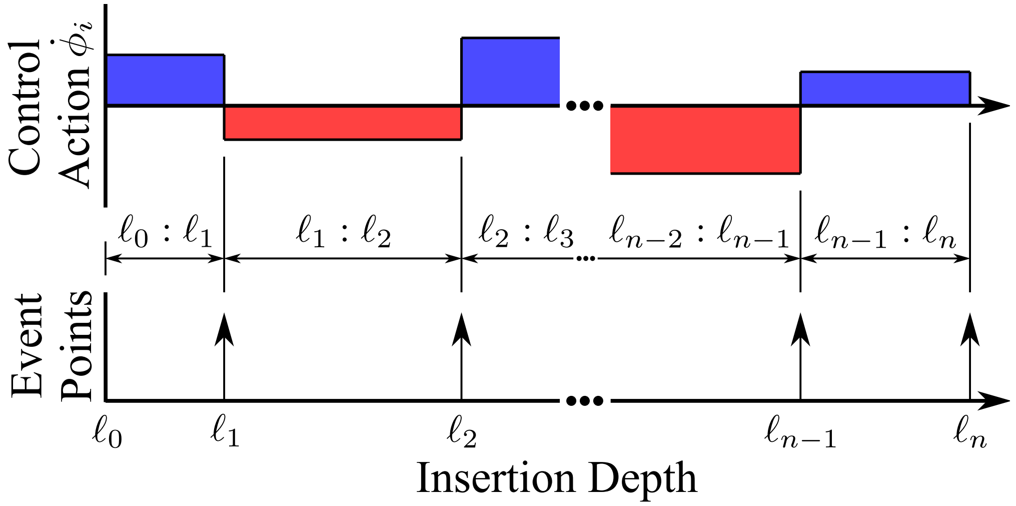


Figure 6.4: Event-trigger points partitioning space into sections and zero-order hold on control signal between adjacent points.

The notation used here and throughout the following sections assigns particular meaning to the subscripts used; when used in a function such as $x(\ell)$ it is implied that ℓ can be any value, such that $\ell \in \mathbb{R} : 0 \leq \ell \leq \ell_f$. In contrast, when a subscript appears explicitly, such as $\phi(\ell_i)$, this indicates the value of $\phi()$ at the specific depth ℓ_i, with corresponding event-trigger point i. The subscript notion $\dot{\phi}_i$ is used to indicate a control signal value that remains constant after the event point i up until the next event point $i + 1$.

With these definitions, we can evaluate the value of $\phi(\ell)$ for the entire insertion through summation of piece-wise integrals over each section, such that

$$\phi(\ell) = \sum_{i=0}^{n-1} f_\phi^i(\ell_i, \ell) + \phi_0 \tag{6.5}$$

where

$$\begin{aligned} f_\phi^i(\ell_i, \ell) &= \begin{cases} \int_{\ell_i}^{\ell} \dot{\phi}(\ell_i)\, d\ell & \text{if } \ell_i \leqslant \ell < \ell_{i+1} \\ \int_{\ell_i}^{\ell_{i+1}} \dot{\phi}(\ell_i)\, d\ell & \text{if } \ell_{i+1} \leqslant \ell \\ 0 & \text{if } \ell < \ell_i \end{cases} \\ &= \begin{cases} \dot{\phi}(\ell_i) \cdot (\ell - \ell_i) & \text{if } \ell_i \leqslant l < \ell_{i+1} \\ \dot{\phi}(\ell_i) \cdot (\ell_{i+1} - \ell_i) & \text{if } \ell_{i+1} \leqslant \ell \\ 0 & \text{if } \ell < \ell_i \end{cases} \end{aligned} \tag{6.6}$$

given the initial needle bevel angle of $\phi(0) \equiv \phi_0$.

The solution to $y(\ell)$ is given by

$$y(\ell) = \sum_{i=0}^{n-1} f_y^i(\ell_i, \phi(\ell), \ell) \tag{6.7}$$

where the piece-wise solution for the needle deflection during each interval, $f_y^i(\ell_i, \phi(\ell), \ell)$, is solved using equation (6.6) for $\phi(\ell)$ in (6.9). From the initial placement of frame $\{T\}$ the deflection of the needle at $\ell = 0$ is defined to be zero, such that $y(0) = 0$.

Similarly the corresponding solution to $z(\ell)$ is then

$$z(\ell) = \sum_{i=0}^{n-1} f_z^i(\ell_i, \phi(\ell), \ell) \tag{6.8}$$

where the piece-wise solution for the needle deflection during each interval, $f_z^i(\ell_i, \phi(\ell), \ell)$, is given in (6.10) where $z(0) = 0$. Thus, we now have analytic solutions for $y(\ell)$ and $z(\ell)$ giving the deflection of the needle tip over the entire insertion without any need for numerical integration. Using these equations to calculate the needle deflection, the controller will be implemented by optimizing the location of the event-trigger points, i.e. their respective depths, and the value of the zero-order hold rotation velocity after each trigger.

$$
\begin{aligned}
f_y^i(\ell_i, \phi(\ell), \ell) &= \begin{cases} \displaystyle\int_{\ell_i}^{\ell} \sin(\theta(\ell)) \cos(\phi(\ell))\, d\ell & \text{if } \ell_i \leqslant \ell < \ell_{i+1} \\ \displaystyle\int_{\ell_i}^{\ell_{i+1}} \sin(\theta(\ell)) \cos(\phi(\ell))\, d\ell & \text{if } \ell_{i+1} \leqslant \ell \\ 0 & \text{if } \ell < \ell_i \end{cases} \\
&= \begin{cases} \displaystyle\int_{\ell_i}^{\ell} \frac{\sin(\theta(\ell) + \phi(\ell)) + \sin(\theta(\ell) - \phi(\ell))}{2}\, d\ell & \text{if } \ell_i \leqslant \ell < \ell_{i+1} \\ \displaystyle\int_{\ell_i}^{\ell_{i+1}} \frac{\sin(\theta(\ell) + \phi(\ell)) + \sin(\theta(\ell) - \phi(\ell))}{2}\, d\ell & \text{if } \ell_{i+1} \leqslant \ell \\ 0 & \text{if } \ell < \ell_i \end{cases} \\
&= \begin{cases} -\dfrac{1}{2}\left(\dfrac{\cos(\kappa\ell + \theta_0 + \phi(\ell))}{\kappa + \dot{\phi}(\ell_i)} + \dfrac{\cos(\kappa\ell + \theta_0 - \phi(\ell))}{\kappa - \dot{\phi}(\ell_i)} \right)\Bigg|_{\ell_i}^{\ell} & \text{if } \ell_i \leqslant \ell < \ell_{i+1} \\ -\dfrac{1}{2}\left(\dfrac{\cos(\kappa\ell + \theta_0 + \phi(\ell))}{\kappa + \dot{\phi}(\ell_i)} + \dfrac{\cos(\kappa\ell + \theta_0 - \phi(\ell))}{\kappa - \dot{\phi}(\ell_i)} \right)\Bigg|_{\ell_i}^{\ell_{i+1}} & \text{if } \ell_{i+1} \leqslant \ell \\ 0 & \text{if } \ell < \ell_i \end{cases}
\end{aligned}
\tag{6.9}
$$

$$
\begin{aligned}
f_z^i(\ell_i, \phi(\ell), \ell) &= \begin{cases} \displaystyle\int_{\ell_i}^{\ell} \sin(\theta(\ell)) \sin(\phi(\ell))\, d\ell & \text{if } \ell_i \leqslant \ell < \ell_{i+1} \\ \displaystyle\int_{\ell_i}^{\ell_{i+1}} \sin(\theta(\ell)) \sin(\phi(\ell))\, d\ell & \text{if } \ell_{i+1} \leqslant \ell \\ 0 & \text{if } \ell < \ell_i \end{cases} \\
&= \begin{cases} \displaystyle\int_{\ell_i}^{\ell} \frac{\cos(\theta(\ell) + \phi(\ell)) - \cos(\theta(\ell) - \phi(\ell))}{2}\, d\ell & \text{if } \ell_i \leqslant \ell < \ell_{i+1} \\ \displaystyle\int_{\ell_i}^{\ell_{i+1}} \frac{\cos(\theta(\ell) + \phi(\ell)) - \cos(\theta(\ell) - \phi(\ell))}{2}\, d\ell & \text{if } \ell_{i+1} \leqslant \ell \\ 0 & \text{if } \ell < \ell_i \end{cases} \\
&= \begin{cases} \dfrac{1}{2}\left(\dfrac{\sin(\kappa\ell + \theta_0 + \phi(\ell))}{\kappa + \dot{\phi}(\ell_i)} - \dfrac{\sin(\kappa\ell + \theta_0 - \phi(\ell))}{\kappa - \dot{\phi}(\ell_i)} \right)\Bigg|_{\ell_i}^{\ell} & \text{if } \ell_i \leqslant \ell < \ell_{i+1} \\ \dfrac{1}{2}\left(\dfrac{\sin(\kappa\ell + \theta_0 - \phi(\ell))}{\kappa + \dot{\phi}(\ell_i)} - \dfrac{\sin(\kappa\ell + \theta_0 - \phi(\ell))}{\kappa - \dot{\phi}(\ell_i)} \right)\Bigg|_{\ell_i}^{\ell_{i+1}} & \text{if } \ell_{i+1} \leqslant \ell \\ 0 & \text{if } \ell < \ell_i \end{cases}
\end{aligned}
\tag{6.10}
$$

Note in (6.9) and (6.10) if $\dot{\phi(\ell_i)} = \pm\kappa$, from the definition in (6.6), it indicates that $\phi(\ell) = \kappa\ell + \phi(\ell_i)$ in the region $\ell_i \leqslant \ell < \ell_{i+1}$ such that one of the terms $\kappa\ell + \theta_0 \pm \phi(\ell)$ will reduce to $\theta_0 \pm \phi(\ell_i)$. Thus the corresponding sin() or cos() function will be a constant over that interval. Being as $\dot{\phi_i}$ is the control output the optimization routine can also be constrained such that $\dot{\phi}(\ell_i) \neq \pm\kappa$.

6.2.1 Insertion simulation computational performance

As stated, one of the advantages of the analytic solution is its reduced computational complexity when compared to the discrete step simulation of the reduced-order kinematic bicycle model. Due to the nonholonomic constraints in the system and the single control input ω, the needle-tip cannot move in an arbitrary direction. If given a set of control actions, for instance, the event-triggering points and rotation values, then the needle-tip position throughout insertion can be found through simulation. The inverse operation, finding the control actions that will lead to the needle tip passing through a desired point or set of points, is not straightforward and requires the use of an optimization routine, like the one covered in Sec. 6.2.2. In general, these optimization routines simulate the insertion process many times while modifying the control actions taken during insertion to find a set of control actions which result in a needle-tip path that passes through the target point(s). In the context of needle steering, where the target location is located at the final insertion depth, then the entire insertion must then be simulated which can have a significant computational cost. This section will give a brief overview of how the reduced-order model can be simulated versus the event-triggered model. For both of these methods, the number of computational steps will be shown and a theoretical performance comparison will be made.

Normally, to simulate a needle insertion using the reduced-order formulation of the bicycle model, given in (5.1), the Euler method is used to solve the state-space differential equation at each time step via the discrete-step form of the equation $X_{K+1} = X_K + \frac{\partial}{\partial x} f(X_K)\Delta\ell + \omega_K$. Here K is the discrete-step depth during the simulation and w_K is the control action per step, where

$\omega_K = \omega\Delta\ell$. This expansion results in the following discrete system

$$X_{K+1} = \begin{bmatrix} x_K \\ y_K \\ z_K \\ \theta_K \\ \phi_K \end{bmatrix} + \begin{bmatrix} \cos(\theta_K) \\ \sin(\theta_K)\cos(\phi_K) \\ \sin(\theta_K)\sin(\phi_K) \\ \kappa \\ 0 \end{bmatrix} \Delta\ell + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} w_K \tag{6.11}$$

where the state within the system X_K consists of the values of $X_K = [x_K, y_K, z_K, \theta_K, \phi_K]$ at the discrete insertion depth K. To evaluate the computational cost of both the discrete-time reduced-order model and the analytic solution used for the event-triggered model the cost of each simulation step will be found. For both algorithms, the step-wise computational cost is defined to be the number of mathematical operations required, defined as n_{calc}. These algorithms will calculate the needle tip deflection at a desired final insertion depth. Here, we define mathematical operations to be the functions cos(), sin(), add(), sub(), mul(), div() and will ignore any computations with respect to the loop counters. As an example, in line 4 of Alg. 6.1, the calculation $y_{K+1} \leftarrow \sin(\theta_K) \cdot \cos(\phi_K) \cdot \Delta\ell + y_K$ would be computed as $y_{K+1} = \text{add}(y_K, \text{mul}(\Delta\ell, \text{mul}(\sin(\theta_K), \cos(\phi_K))))$ requiring 5 calculations, thus $n_{calc} = 5$ as indicated.

For the discrete-step reduced-order model, each step updates the states of the system model using the Euler method above (6.11). Starting at the initial simulation step, $K = 0$, the initial state of the system is defined as $X_0 = [x_0, y_0, z_0, \theta_0, \phi_0]$. The simulation proceeds until the desired final insertion step calculated, defining the last time simulation step to be K_{end}. The algorithm for simulating the discrete-time needle can then be described as follows, with the symbols ‘ $\cdot$ ’ and ‘ / ’ used to indicate multiplication and division respectively.

As shown in Algorithm 6.1, the computation cost of each simulation step K is $n_{calc} = 17$. Assuming the simulation consists of d simulation steps discretized along the insertion depth a total computational complexity of $n_{calc} = 17d$ will result, which grows linearly with the number of insertion steps to be simulated.

The analytic solution for the event-triggered needle model allows for a computational speed increase for insertion simulation. Using the results of equations (6.9) and (6.10) we can simulate the entire needle insertion in a more

Algorithm 6.1 Discrete-time step simulation

```
Require: $X_0, \Delta\ell, \omega_k, \kappa$
$K = 0$
while $K < K_{end}$
    $x_{K+1} \leftarrow \cos(\theta_K) \cdot \Delta\ell + x_K$                              ▷ $n_{calc} = 3$
    $y_{K+1} \leftarrow \sin(\theta_K) \cdot \cos(\phi_K) \cdot \Delta\ell + y_K$            ▷ $n_{calc} = 5$
    $z_{K+1} \leftarrow \sin(\theta_K) \cdot \sin(\phi_K) \cdot \Delta\ell + z_K$            ▷ $n_{calc} = 5$
    $\theta_{K+1} \leftarrow \kappa \cdot \Delta\ell + \theta_K$                            ▷ $n_{calc} = 2$
    $\phi_{K+1} \leftarrow \phi_K + \omega_K$                                             ▷ $n_{calc} = 2$
    $K \leftarrow K + 1$
```

efficient manner. The model can be evaluated by calculating the changes to the needle state at the insertion depths corresponding to the chosen number of event points, n, including the initial and final insertion depths, defined above as l_0 and l_n. As above, our control signal ω is held constant between event-trigger points giving a zero-order hold for $\dot{\phi}(\ell_i)$.

For brevity, we will define the state of the needle to be $X_i = [x_i, y_i, z_i, \theta_i, \phi_i]$ at a depth ℓ_i corresponding to the event-trigger point i. At the depth of each event-trigger point, ℓ_i, the rotation speed and direction of the needle are changed; here we indicate the velocity after the event point as $\dot{\phi}_i$. Again we used subscript notation from Sec 6.2, where we defined our space ℓ to be broken into sections, such that $\ell = \{\ell_0 : \ell_1, \ell_1 : \ell_2, ..., \ell_{n-1} : \ell_n\}$. The initial state of the needle, at the point of insertion, is then defined as $X_0 = [x_0, y_0, z_0, \theta_0, \phi_0]$. The final insertion state of the needle is denoted as $X_n = [x_n, y_n, z_n, \theta_n, \phi_n]$. The algorithm for simulating the needle is then the following.

For each event point, we have a computational cost of $n_{calc} = 49$ for calculating the deflection in both the y and z axis, as shown in Algorithm 6.2. For an insertion to a length ℓ_n, where there are $n - 1$ control actions taken, there will be a total of n steps to calculate, thus the computational complexity is $n_{calc} = 49(n)$ which also grows linearly, but with respect to the number of rotations only. Even though the per-event computation cost is higher, simulating a small number of rotations during the insertion will be less costly than simulating the discrete-time system.

One of the primary disadvantages is that the discrete-time simulation ac-

Algorithm 6.2 Event-triggered needle simulation

```
Require: $X_0, \dot{\phi}_i, \kappa$
i = 0
while $i < n$
    $\theta_{i+1} \leftarrow \kappa \cdot \ell_{i+1} + \theta_i$                                              ▷ $n_{calc} = 2$
    $\phi_{i+1} \leftarrow \phi_i + \dot{\phi}_i \cdot (l_{i+1} - l_i)$                                    ▷ $n_{calc} = 3$
    $x_{i+1} \leftarrow (\sin(\theta_{i+1}) - \sin(\theta_0))/\kappa$                                     ▷ $n_{calc} = 4$

    Steps to calculate $y_{\ell_{i+1}}$
    $y_{p1} \leftarrow \cos(\theta_i + \phi_i)/(\kappa + \dot{\phi}_i)$                                    ▷ $n_{calc} = 4$
    $y_{p2} \leftarrow \cos(\theta_i - \phi_i)/(\kappa - \dot{\phi}_i)$                                    ▷ $n_{calc} = 4$
    $y_{p3} \leftarrow \cos(\theta_{i+1} + \phi_{i+1})/(\kappa + \dot{\phi}_{i+1})$                         ▷ $n_{calc} = 4$
    $y_{p4} \leftarrow \cos(\theta_{i+1} - \phi_{i+1})/(\kappa - \dot{\phi}_{i+1})$                         ▷ $n_{calc} = 4$
    $y_{i+1} \leftarrow (-y_{p3} - y_{p4} + y_{p1} + y_{p2})/2$                                         ▷ $n_{calc} = 4$

    Steps to calculate $z_{\ell_{i+1}}$
    $z_{p1} \leftarrow \sin(\theta_i + \phi_i)/(\kappa + \dot{\phi}_i)$                                    ▷ $n_{calc} = 4$
    $z_{p2} \leftarrow \sin(\theta_i - \phi_i)/(\kappa - \dot{\phi}_i)$                                    ▷ $n_{calc} = 4$
    $z_{p3} \leftarrow \sin(\theta_{i+1} + \phi_{i+1})/(\kappa + \dot{\phi}_{i+1})$                         ▷ $n_{calc} = 4$
    $z_{p4} \leftarrow \sin(\theta_{i+1} - \phi_{i+1})/(\kappa - \dot{\phi}_{i+1})$                         ▷ $n_{calc} = 4$
    $z_{i+1} \leftarrow (z_{p3} + z_{p4} - z_{p1} - z_{p2})/2$                                          ▷ $n_{calc} = 4$
    $\ell \leftarrow \ell + 1$
```

curacy is dependent on the number of steps used, with accuracy increasing with the number of steps. Empirically, the insertion depth must be discretized into steps of 0.5 mm or smaller to result in a reasonable simulation accuracy at the final insertion depth, such that the 120 mm insertion depth used in this chapter requires 241 or more simulation steps. For the implementation of the controller used in Sec. 6.4.2, for 3 event-triggered rotations, the analytic solution has an absolute cost of 196 operations whereas the discrete-time step solution would require 4097 math operations to fully simulate each insertion. The analytic event-mode algorithm allows a full 17 simulations to be run in the same amount of time as a single insertion simulation through the discrete-time step method. This computational speed-up implies that the control algorithms built on this analytic solution can be implemented in real-time even in low-end hardware. In addition, the analytic solution does not use any approximation of the underlying differential equations and therefore, offers optimal accuracy without requiring additional simulation points, providing one of the contributions of this chapter.

6.2.2 Event-triggered control optimization

The analytic solution of the reduced-order kinematic model facilitates the development of an online optimization routine for the event-triggered controller. This routine will be designed to achieve the stated goal of minimizing both needle deflection and the total number of needle rotations. Minimizing needle deflection will be achieved through optimizing the location of event-trigger points, along $\dot{\phi}(\ell)$, and the associated changes in needle rotation velocity at those points. With the target location defined to be on the ${}^{0}x'$-axis, parallel to the direction of needle insertion, needle deflection will be defined as the Euclidean distance

$$e(\ell) = \sqrt{y(\ell)^2 + z(\ell)^2} \tag{6.12}$$

for a particular insertion depth ℓ. With this definition of needle deflection, a cost function used for minimizing needle deflection can be given as

$$\mathcal{L} = \min\left(\alpha\sqrt{y(\ell_f)^2 + z(\ell_f)^2} + \beta \max\left(\sqrt{y(\ell)^2 + z(\ell)^2}\right)\right) \tag{6.13}$$

where α and β are weighting constants and $\max(\sqrt{y(\ell)^2 + z(\ell)^2})$ represents the maximum deflection during insertion, $\ell \in \{0 : \ell_f\}$. This allows the controller to be tuned from weak regulation, where $\alpha > 0$ and $\beta = 0$, towards the deflection minimization performance of a regulating controller when $\alpha = 0$ and $\beta > 0$. The controller can also implement a weighted minimization of both objectives when $\alpha, \beta > 0$.

With the needle steering control actions being performed through event-triggering, the online optimization is implemented through finding event points, ℓ_i, along with the required change in needle rotation velocity, $\Delta\dot{\phi}_i$, at those points which minimize the cost. For this controller implementation, the number of event points must be chosen before the insertion. This results in a segmentation of the insertion space $\ell = \{\ell_0 : \ell_1, \ell_1 : \ell_2, ..., \ell_{n-1} : \ell_n\}$ with each segment having a desired needle base rotation velocity $\omega = \dot{\phi}(\ell_i)$, where $\{i \in 0, 1, ..., n-1\}$. As a first step in optimization, the result given in (6.6) is rearranged to limit the total change of the value of $\phi(\ell)$ in each segment. This change in $\phi(\ell)$ in each segment is limited by a constant $\Delta\dot{\phi}_i$, and is used in

$$\left|\phi(\dot{\ell_i})\right| \leqslant \frac{(\ell_{i+1} - \ell_i)}{\Delta\dot{\phi}_i} \text{ where } \ell_i \leqslant \ell < \ell_{i+1} \tag{6.14}$$

so that the value of $\dot{\phi}_i$ for each segment can be limited.

Using (6.14) allows tissue trauma to be limited by a choice of $\Delta\dot{\phi}_i$ independently of the needle deflection cost function. For this chapter, the value of $\Delta\dot{\phi}_i$ was empirically chosen to be π, such that the total number of rotations is directly limited to be equal or less than half of the number of event points chosen. Using this limit, the cost function minimization now is independent of any tissue effects for a given number of event-trigger points.

The minimization of the cost function is done by a genetic algorithm[101]. Here a traditional genetic algorithm was modified to have performance closer to simulated annealing by implementing a large number of cross-overs and large uniform random distribution used to seed each generation. The algorithm was “greedy” in that it returned only the single best-performing individual which was then propagated and compared and crossed-over with new individuals created from the uniform distribution. The two needle model parameters, found through ultrasound imaging, θ_0 and κ, were updated at the beginning of each iteration. The number of individuals tested at each iteration was held constant, and the number of cross-over individuals created was chosen to be equal to the number of regular individuals.

Each individual consisted of C_i coupled event-trigger points and control signal values $< \ell_i, \dot{\phi}_i >$, where $i = 1, 2...C_i$; to describe the control actions for a complete insertion, the cross-overs were made by swapping a random number of coupled values between two individuals. The cross-over and random individual generator were constrained such that the algorithm only optimized control actions which could be taken in the future as the needle is inserted deeper into tissue. To do this, the values of $< \ell_i, \dot{\phi}_i >$ from the best individual of the previous generation were used for any point i where the current insertion depth was greater than ℓ_i. The result from the simulation of each individual was multiplied by α and stored, as the simulation returns the needle-tip deflection at the target insertion depth, $\sqrt{y(\ell_f)^2 + z(\ell_f)^2}$, and incorporates the minimal rotation constraint.

To calculate the second component of the cost function $\beta \max \left(\sqrt{y(\ell)^2 + z(\ell)^2} \right)$, each pair of adjacent trigger-points in the individual were used to check if the values of $\dot{y}(\ell)$ or $\dot{z}(\ell)$, defined in (5.1), had crossed zero in that section. This zero crossing implies the needle deflection was at a maximum or minimum. The values of $y(\ell)$ and $z(\ell)$ at the points where $\dot{y}(\ell)$ or $\dot{z}(\ell)$ are equal to zero were evaluated by inserting the depth, ℓ, where the derivatives are zero into equations (6.9) and (6.10). The largest result of $\sqrt{y(\ell)^2 + z(\ell)^2}$ across all of the zero-crossing points was multiplied by β and used to calculate that individual's score with respect to the cost function; if there was no zero-crossing in $\dot{y}(\ell)$ or $\dot{z}(\ell)$ then the largest deflection is the final-tip deflection, and this was multiplied by β and used instead. The average computation time of the controller optimization routine during the insertion experiments is given in Table 6.4.2 where each generation of the genetic algorithm consisted of 60 individuals, 30 of which were created through cross-overs.

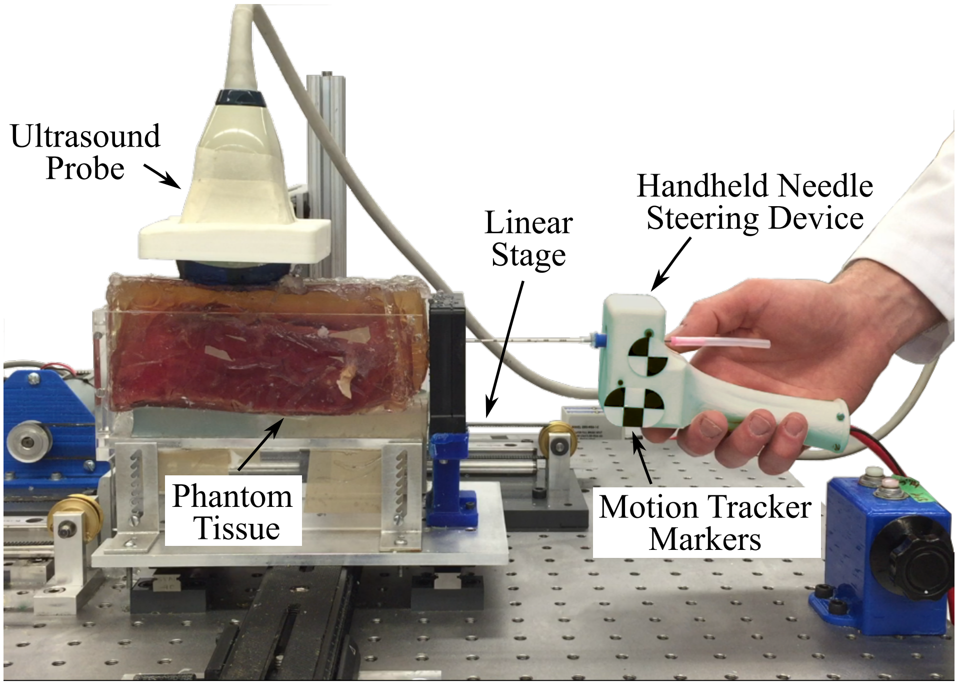


Figure 6.5: Experimental setup with ultrasound probe, hand-held needle insertion device, and tissue phantom.

6.3 Experimental Setup

During the experiments, ultrasound images are captured and processed in real-time to return the needle deflection. The ultrasound transducer, see Fig. 6.5, was mounted on a motorized linear rail which moved the transducer such that the needle-tip was always contained in the imaging plane. The ultrasound machine used for the experiments was an Ultrasonix Touch with a 4DL14-5/38 Linear 4D transducer (Ultrasonix Corp, Richmond, BC, Canada). For these experiments, only the 2D imaging functionality of the ultrasound probe was used. The needle was controlled during insertion using a hand-held steering device, Fig 6.5, originally developed in [6]. Optical tracking markers were placed on the hand-held device and a Micron Tracker (HX60 from Claron Technology Inc., Toronto, ON, Canada) was used to measure the length of the needle inserted into tissue. For the phantom tissue insertions, the needles used were standard 18-gauge 200 mm prostate seeding needles (Eckert & Ziegler BEBIG GmbH, Berlin, Germany). The image processing, controller optimization, and event-triggered controller were all programmed in Matlab

2016a (The Mathworks Inc, Natwick, MA, USA) and ran using the Simulink Real-Time environment, on an Intel Core i7-3930K running at 3.20 GHz (Intel Corporation, Santa Clara, CA, USA).

6.4 Ultrasound Image Processing

For each of the insertion experiments, 2D ultrasound image slices are processed in real-time at a frame-rate of 20 Hz. The 2D functionality of the probe, rather than 3D/4D functionality, was used to replicate the imaging capabilities of the transrectal ultrasound (TRUS) probes used clinically. As shown in Fig. 6.5, during needle insertion the ultrasound probe is translated along the direction of needle insertion, the ${}^{0}x'$-axis, such that the needle tip is always captured in the ultrasound image slice. These ultrasound images are processed in real-time using an altered version of the algorithm presented in [2] to track the needle-tip deflection. The images are captured at discrete time intervals, corresponding to the 20 Hz frame rate of the ultrasound machine, with the imaging time step denoted by k_{US}. With the ultrasound probe moving along the ${}^{0}x'$-axis, the image frame, corresponding to the ${}^{0}y'$ and ${}^{0}z'$ axes, captures the needle tip deflection (with respect to the target ${}^{0}x'$-axis). The pixel coordinates of the image are indicated by py and pz and the pixel intensity of the image, at time step k_{US}, is defined as $I_{k_{US}}(py, pz)$. The insertion depth of the needle is also measured for each of the ultrasound images, defined as $\ell_{k_{US}}$, and this information is used along with the tracked needle-tip location to fit parameters of the reduced-order bicycle model, to be covered in Sec. 6.4.1. The image processing routine for each frame consists of two stages which are shown in Fig. 6.6. The first stage is a preprocessing stage that enhances the visibility of the needle tip to make it more distinct from the tissue background in the image and to make the tracking invariant to changes in the needle tip pixel intensity. The second stage uses template matching to perform the needle tip tracking. At the beginning of insertion, a user clicks on the needle tip in the first frame of the ultrasound image and the image enhancement and tracking are done using a region-of-interest (ROI) around this point. For preprocessing,

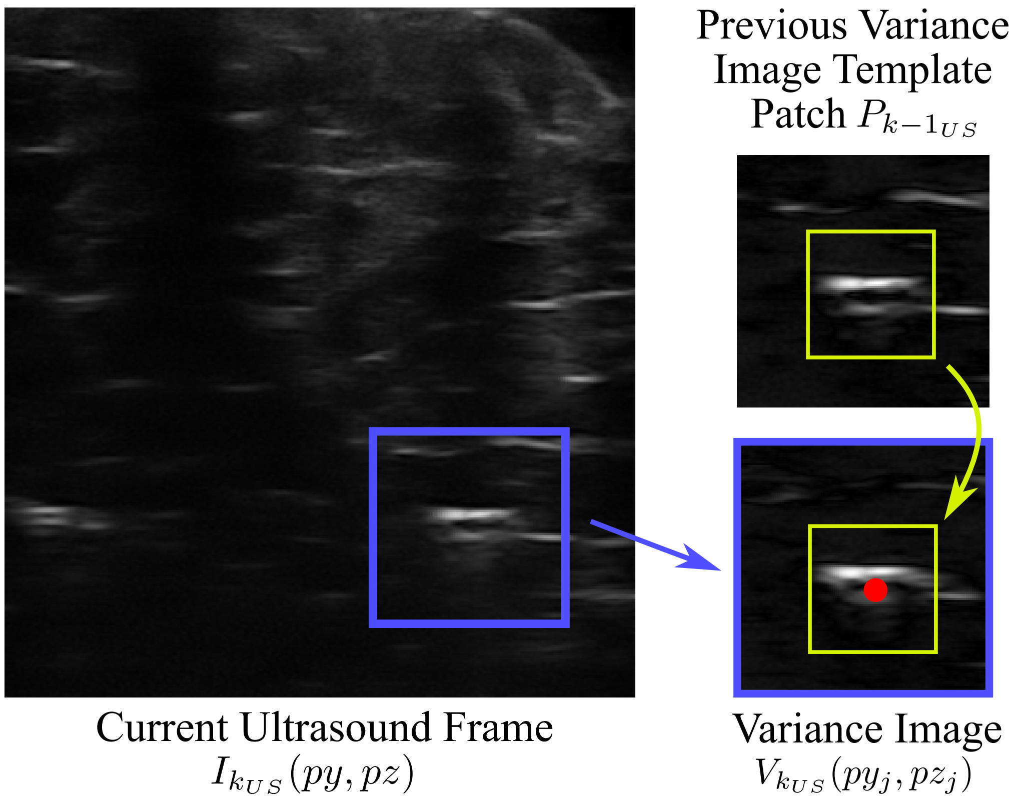


Figure 6.6: Ultrasound image processing showing original input image, variance image, template patch, and resulting needle-tip location (indicated by red dot).

the needle-tip visibility is improved by the creation of a variance image of the ROI. This variance image, when used for tracking, is similar in theory to the sum of conditional variance method presented in [102]. To calculate the variance image, statistics of the pixel intensity in the ROI are used, given

$$\begin{aligned} \mu_{k_{US}} &= \frac{1}{n_{px}} \sum_{j=1}^{n_{px}} I_{k_{US}}(py_j, pz_j) \\ \sigma_{k_{US}} &= \frac{1}{n_{px}} \sqrt{\sum_{j=1}^{n_{px}} \left(I_{k_{US}}(py_j, pz_j) - \mu_{k_{US}}\right)^2} \end{aligned} \tag{6.15}$$

where n_{px} is the number of pixels in the ROI, py_j and pz_j are the pixel locations within the ROI, $\mu_{k_{US}}$ and $\sigma_{k_{US}}$ are the resulting mean and standard deviation of the pixel intensities. The variance image is then calculated as

$$V_{k_{US}}(py_j, pz_j) = \frac{\| I_{k_{US}}(py_j, pz_j) - \mu_{k_{US}} \|}{\sigma_{k_{US}}} \tag{6.16}$$

where $V_{k_{US}}$ is the resulting variance image intensity for each pixel py_j, pz_j in the ROI. This results in an increase of contrast between the needle and the background tissue.

To perform the needle tip tracking, a method similar to [93] is used, under the assumption that the needle-tip location only changes by a small amount between subsequent ultrasound images due to the mechanical characteristics of the needle. Needle tip tracking is done using sum-absolute-difference template matching between a small patch of the variance image in the previous frame, $P_{k-1_{US}} \subset V_{k-1_{US}}(py_j, pz_j)$, centered around the needle tip, py_{TIP} and pz_{TIP}, and the region of interest in the current frame $V_{k_{US}}(py_j, pz_j)$. Due to the use of variance images, this sum-absolute-difference template matching is equivalent to performing normalized cross-correlation based template matching. The center position of the best template match is the needle-tip position, $py_{TIP|k_{US}}$ and $pz_{TIP|k_{US}}$, which is converted into a metric deflection, in mm, in the ${}^0y'$ and ${}^0z'$ axes. The metric needle tip deflections for each image frame are denoted by $y_{US}(\ell_{k_{US}})$ and $z_{US}(\ell_{k_{US}})$, where the pixel-to-millimeter ratios were measured experimentally with a ratio 0.10 $\frac{\text{mm}}{\text{px}}$ for the ${}^0y'$-axis and 0.12 $\frac{\text{mm}}{\text{px}}$ for the ${}^0z'$-axis. The tracked needle tip is then used for estimation of θ_0

and κ during the needle insertion. For the experimental results, presented in Sec. 6.4.2 with the image processing time given in Table. 6.4.2, the size of the region-of-interest was a 129px×129px square and the size of the needle-tip template was 65px×65px.

6.4.1 Image based parameter fitting

Two of the parameters of the reduced-order bicycle model, θ_0 and κ are required for the event-triggered control optimization. These parameters correspond to the initial angle of needle insertion and the (inverse of) the radius of curvature. The ultrasound processing algorithm returns the location of the needle tip and the corresponding insertion depth, resulting in $\ell_{k_{US}}, y_{US}(\ell_{k_{US}}), z_{US}(\ell_{k_{US}})$ in metric coordinates. During insertion, the location of the needle tip is recorded for every frame, and once the needle has been inserted past a depth of 10 mm, the values of θ_0 and κ are estimated, with an updated estimate returned with every frame. Here the value of 10 mm was chosen empirically to allow for a small amount of needle deflection to occur before estimation such that the value of κ can be measured. Note that the values of θ_0 and κ are defined to remain constant for a single insertion but may change from insertion to insertion. During insertion, as the parameter estimates are updated, the current best estimate is used in the controller optimization routine. The controller optimization routine simulates the needle insertion using previous control actions (if any event-trigger depths have been passed) on the current system estimate, thus keeping θ_0 and κ constant for that insertion simulation. The simulation of the needle parameter fitting algorithm, given at the end of this section, shows that the estimate settles to the correct values very quickly, within 40 mm of the needle entry point, so in practice, the model parameters being used for control optimization remain essentially constant after that depth.

As with the optimization for the event triggered controller, the parameter fitting for the model was done using a modified genetic algorithm. The same percentage of cross-overs and generation of random individuals to refill the population per iteration were used. In contrast to the control optimization algorithm, for this algorithm the best 30% of the population was propagated forward at each iteration. Another difference was the way that the individuals were constructed; each individual consisted of only a single set of coupled values $<\tilde{\theta_0}, \tilde{\kappa}>$. Due the limit on $\theta(\ell)$ in the model, $\{\theta \in \mathbb{R} : 0 \leq \theta < \pi\}$, and the use of the guide template which mechanically constrains the needle during insertion into tissue such that θ_0 is small, the value of $\tilde{\theta_0}$ was limited in the algorithm to the range 0 rad $\leq \tilde{\theta_0} \leq$ 0.174 rad (0.174 rad $\approx 10°$). The value of $\tilde{\kappa}$ was also constrained to the range $\frac{1}{10000} \leq \tilde{\kappa} \leq \frac{1}{500}$ which is larger than required from data gathered from previous insertion experiments and the literature. To solve for these parameters, the needle path is simulated for a the set of initial values $<\tilde{\theta_0}, \tilde{\kappa}>$ resulting in deflections $\tilde{y}(\ell)$ and $\tilde{z}(\ell)$. The best individual is chosen such that it minimizes the following cost function

$$\mathcal{L}_{Fit} = \min_{<\tilde{\theta_0}, \tilde{\kappa}>} \sum_{j=1}^{n} \frac{\sqrt{[\tilde{y}(\ell_j) - y_{US}(\ell_j)]^2 + [\tilde{z}(\ell_j) - z_{US}(\ell_j)]^2}}{n} \tag{6.17}$$

where the score $\mathcal{L}_{Fit}$ is the minimized simulated needle tip path closest to the observed tip path at measurement test points $j = 1, 2, ..., n$. Multiple measurement points j are used, rather than just the current needle-tip location, to make the parameter estimation routine more robust to image noise and occlusion-based needle-tip localization errors, which occur intermittently during needle-tip tracking. For this implementation, these points were chosen to measure the needle location a equidistant insertion depths spanning from the insertion point, $\ell = 0$, to the current needle-tip point, ℓ_{Img}. For the experimental results 11 test points were empirically determined to be sufficient for parameter fitting. To convert from $\ell_{k_{US}}$ to $x(\ell_{k_{US}})$, (6.3) was inverted using each individual's $\tilde{\kappa}$. Finally, for each incoming frame an iteration of the genetic algorithm was run, where the system was simulated using the analytic solution given in Sec. 6.2, incorporating the control actions that had been taken up to the current insertion depth. L_{Fit} was evaluated using the resulting needle-tip path from simulation and the measured needle-tip locations from the frames closest to each j test point depth. The $<\tilde{\theta_0}, \tilde{\kappa}>$ of the best performing individual was returned as the values θ_0 and κ used for the control optimization.

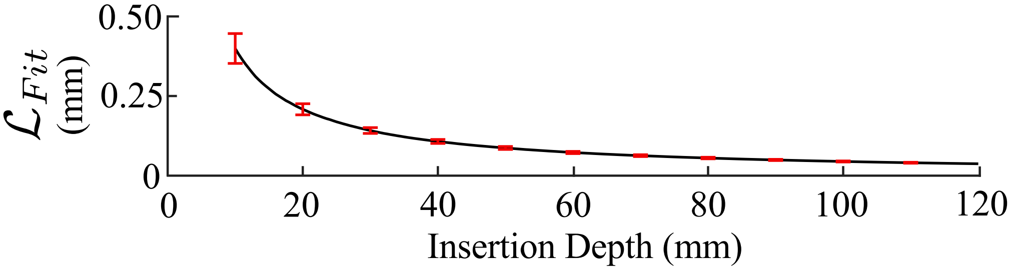


Figure 6.7: Needle parameter estimitation score with respect to insertion depth.

The computation time of the parameter estimation routine is given in Table. 6.4.2 where 50 individuals were used for each generation.

The needle parameter optimization was validated in simulation by comparing the needle-tip path of a simulated insertion with known $< \theta_0, \kappa >$ values to the needle-tip path predicted with the estimated parameters $< \tilde{\theta_0}, \tilde{\kappa} >$ from the genetic algorithm. For the validation, 10000 insertions were simulated with uniformly random selections of θ_0 and κ_0, where the value of θ_0 was tested in the range $0 \leq \theta_0 \leq \frac{\pi}{4}$ and κ was tested in the range $\frac{1}{10000} \leq \kappa \leq \frac{1}{500}$. To simulate a "measured" needle-tip path, zero-mean Gaussian noise was added to the simulated needle deflection (derived from equations (6.9) and (6.10), such that $y_{US}(\ell) = y(\ell) + N$ and $z_{US}(\ell) = z(\ell) + N$ with $N \sim \mathcal{N}(0, 0.5)$). The parameters $< \tilde{\theta_0}, \tilde{\kappa} >$ were estimated using simulated values with ℓ being incremented in 1.0 mm steps. The plot of the average estimator score, $\mathcal{L}_{Fit}$, is given in Fig. 6.7 with the error bars showing the standard deviation of the estimator score across the 10000 simulations. These results show that the parameter fitting is able to estimate the needle shape with a mean-squared error of 0.1 mm after the needle has been inserted to a depth of 40 mm.

6.4.2 Experimental Results

Three different tissue phantoms were fabricated to evaluate the performance of the needle steering system. Two non-homogeneous tissue phantoms were made from bovine and porcine tissue embedded in gelatin (Knox from Kraft Inc., Northfield, IL, USA). These bovine and porcine phantoms were created to closely resemble the ultrasound imaging and mechanical properties of human tissue. The gelatin was used to hold the tissue in the experimental setup and to provide a flat surface for the ultrasound probe to scan along, such that there was sufficient contact between the probe and the tissue phantom to maintain ultrasound image quality throughout the insertion (see Fig. 6.5). A third tissue phantom was made entirely from plastisol (M-F Manufacturing Co, Fort Worth, USA) which has friction and stiffness properties higher than seen in human tissue. The elevated friction and stiffness create a more challenging scenario for the controller as the needle deflection during insertion is increased.

A total of 10 insertion trials was performed for each of the three phantom tissues, with each insertion having a desired target depth of 120 mm. The needle base rotation was performed by the hand-held device using the optimal event-trigger points and rotation values found by the genetic algorithm. The optimization parameters, α and β, were chosen to be 10 and 1 respectively. For these results, only three event-trigger points were used and the number of total needle rotations was limited to be less than 3. For most of the trials, the optimized event-triggered control output resulted in fewer than two full rotations. The needle insertion length, ℓ, required for the controller and ultrasound image processing was measured in real time using the micron device; see Fig. 6.5.

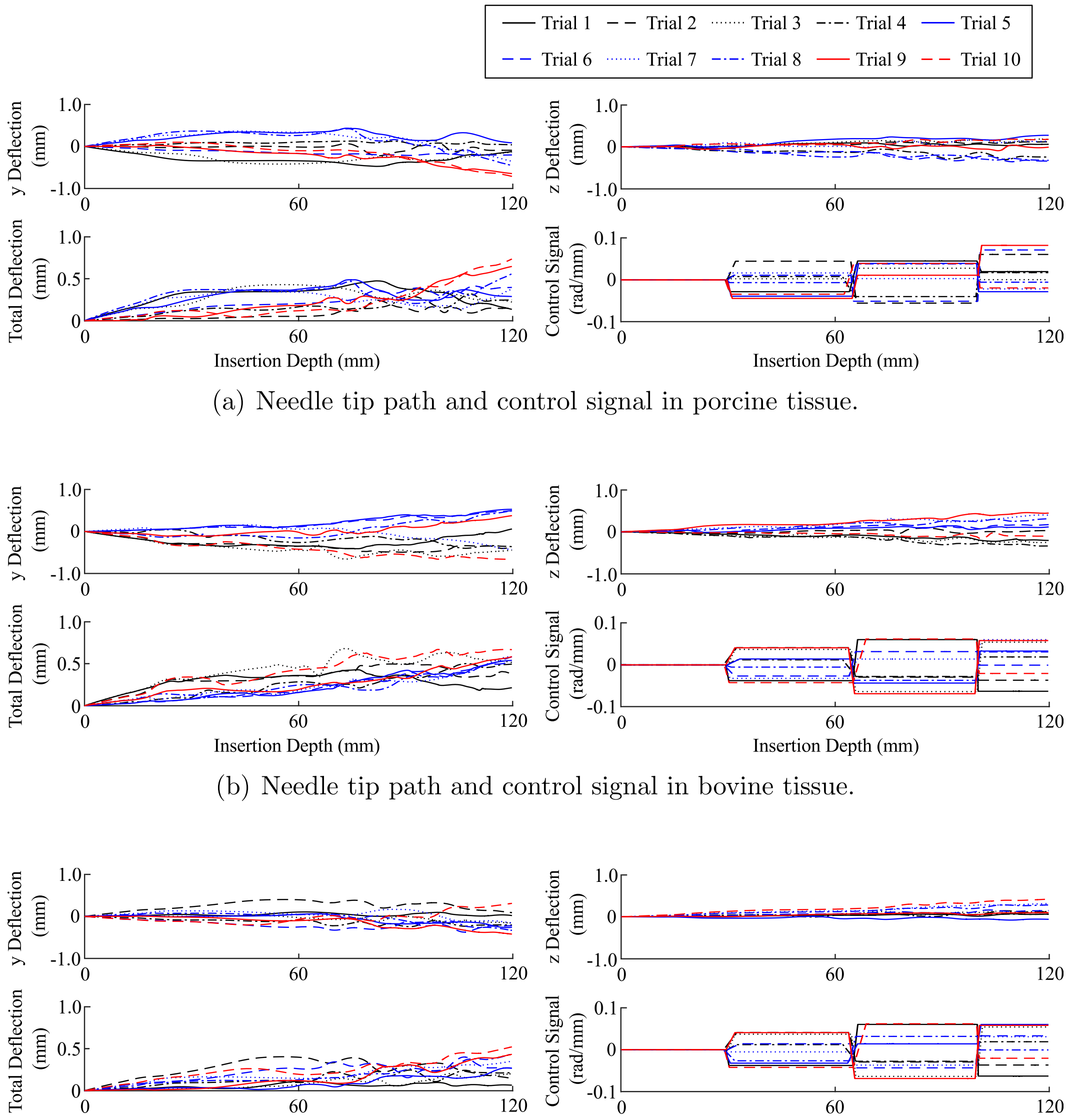
The deflection of the needle tip away from the ${}^{0}x'$-axis was measured through ultrasound imaging. The needle tip location at the target insertion depth was used to evaluate the performance of the controller. The Final Tip Deflection is calculated as

$$\text{Final Tip Deflection} = \sqrt{y(\ell_f)^2 + z(\ell_f)^2} \tag{6.18}$$

where ℓ_f represents the target insertion depth. The controller was tuned to minimize deflection over the entire insertion. To evaluate this, the other performance measure that was used was the Mean Tip Deflection, given by

$$\text{Mean Tip Deflection} = \int_{\ell=0}^{\ell_f} \frac{\sqrt{y(\ell)^2 + z(\ell)^2}}{\ell_f} d\ell \tag{6.19}$$

which calculates the average needle tip deflection away from the ${}^{0}x'$-axis



(a) Needle tip path and control signal in porcine tissue.

(b) Needle tip path and control signal in bovine tissue.

(c) Needle tip path and control signal in plastisol tissue.

Figure 6.8: Experimental results for three different tissue phantoms.

Table 6.1: Average Needle Insertion Results

Phantom Tissue	Final Tip Deflection Average (mm)	Mean Tip Deflection Average (mm)	Total Rotation Amount (degrees)
Porcine	0.40 ± 0.24	0.25 ± 0.06	174 ± 86
Bovine	0.51 ± 0.14	0.31 ± 0.08	177 ± 84
Plastisol	0.48 ± 0.25	0.30 ± 0.12	137 ± 68

Table 6.2: Maximum Needle Insertion Results

Phantom Tissue	Max Final Tip Deflection (mm)	Max Total Rotation Amount (degrees)
Porcine	0.86	306
Bovine	0.69	306
Plastisol	0.90	242

throughout the insertion. The last metric that was evaluated was the total rotation amount that the needle made. To calculate this, the absolute value of each control action $\dot{\phi}_i$ was taken and the resulting value of $\phi(\ell)$ evaluated, where the initial bevel angle, ϕ_0, is neglected. This simplified to the following

$$\text{Total Rotation} = |\phi(\ell_f)| = \sum_{i=0}^{n-1} \left| f_\phi^i(\ell_i, \ell_f) \right| \tag{6.20}$$

where $f_\phi^i(\ell_i, \ell)$ is evaluated using the method in (6.6). The values for the Final Tip Deflection and Mean Tip Deflection, and Total Rotation were averaged over the results for the 10 insertions per tissue phantom in Table 6.4.2. Maxiumum Total Rotation and maximum Tip Deflection at any length during insertion are given in Table 6.4.2. The needle tip path captured for the porcine, bovine, and plastisol phantoms are displayed in Fig. 6.8(a), Fig. 6.8(b), Fig. 6.8(c) respectively. The computational time of the image processing algorithm, the parameter estimation algorithm, and the control optimization algorithm are presented in Table 6.4.2 with the template patch sizes and number of individuals per generation given in Sec. 6.4, Sec. 6.4.1, and Sec. 6.2.1 respectively.

From the tabular data, the controller is shown to perform approximately the same across the three tissue types. The Final Tip Defection average across

Table 6.3: Processing Time per Image Frame

Procedure	Average Time (ms)	Max Time (ms)
Image Processing	36.0 ± 0.4	40.0
Parameter Estimation	1.6 ± 0.7	6.3
Control Optimization	4.6 ± 1.9	8.16

all 30 trials was 0.47(±0.21) mm with an averaged Mean Tip Deflection of 0.29(±0.09) mm. The maximum needle tip deflection at the target was 0.90 mm, which compares favorably to the literature surveyed in [7], improved on the results of the previous chapter, and greatly exceeds the current clinical accuracy demonstrated in [13].

6.5 Conclusion

We have demonstrated a system designed for use in either biopsy or brachytherapy percutaneous procedures. This system steers the needle using a hand-held device driven by an event-triggered controller designed to reduce needle deflection during insertion. The controller incorporates a nonholonomic reduced-order bicycle model with required model parameters being estimated online from ultrasound images. The parameter estimation and control signal planning use the presented solution to the kinematic bicycle model which is shown to increase model simulation performance by 21x when compared to a discrete-time step implementation of the model. From insertion trials in ex-vivo tissue phantoms, the controller is shown to decrease Final Tip Deflection to an average of 0.47(±0.21) mm and Mean Tip Deflection to an average of 0.29(±0.09) mm, significantly better than seen clinically at present. The total amount of needle rotation was constrained during insertion with the maximum total needle rotation measured to be 306^o across all trials. The next chapter will move on to develop a prostate segmentation routine which can localize the target position, such that the controller presented here is able to steer a needle towards it.

Chapter 7

Autonomous Prostate Segmentation in 2D B-Mode Ultrasound Images

In current practice, prostate brachytherapy requires a planning stage where the entire volume of the prostate is imaged and the 3D contour of the prostate is created. The 3D prostate contour is used for dosimetric calculation and, traditionally, is manually segmented by a clinician in software. While either MR or CT imaging modalities can be used for the dosimetric planning, the typical imaging modality used is ultrasound, both for cost-effectiveness and to match the imaging modality used by the clinician during the insertion procedure. This manual segmentation of the prostate from axial image slices is time-consuming, and thus algorithmic prostate segmentation poses an attractive option to reduce the amount of time where a clinician is required during prostate brachytherapy pre-planning. In addition to therapeutic intervention planning, prostate gland contours can also be used for medical diagnostics, through measuring parameters such as gland asymmetry [103] or gland image appearance [104].

The prostate contouring system presented here is based on the use of “superpixels”[105], consisting of individual B-mode ultrasound image pixels that have been optimized into large clusters. This work presents a novel extension of the superpixel method, whereby the superpixel regions are optimized based on statistical similarity measures such that the various structures

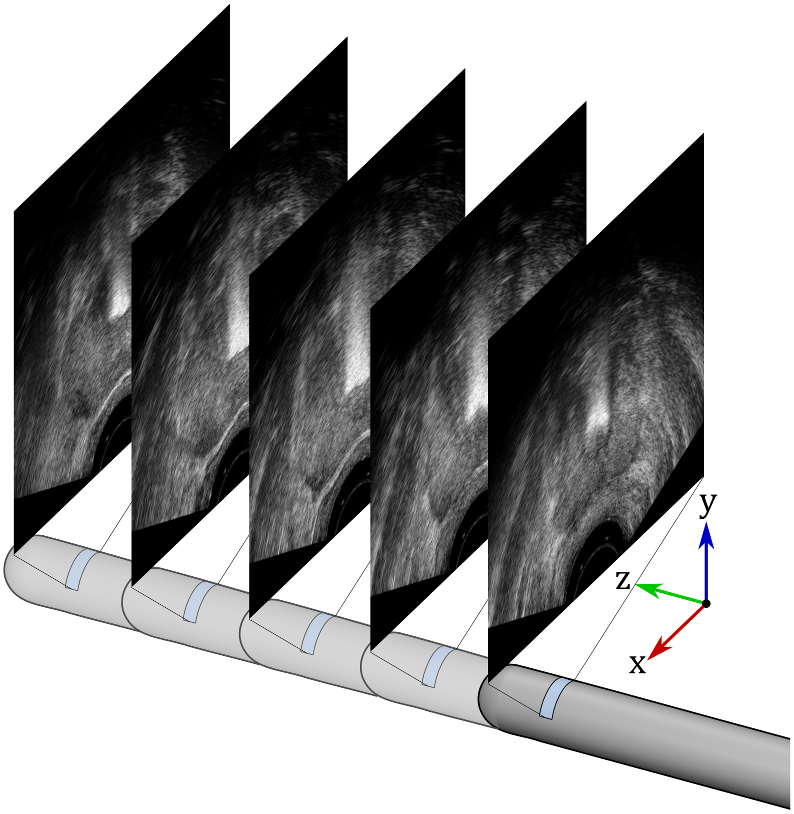


Figure 7.1: Ultrasound axial image slices being taken during the pre-planning phase of prostate brachytherapy. Each axial slice is taken 5 mm apart as the probe is translated along the z-axis.

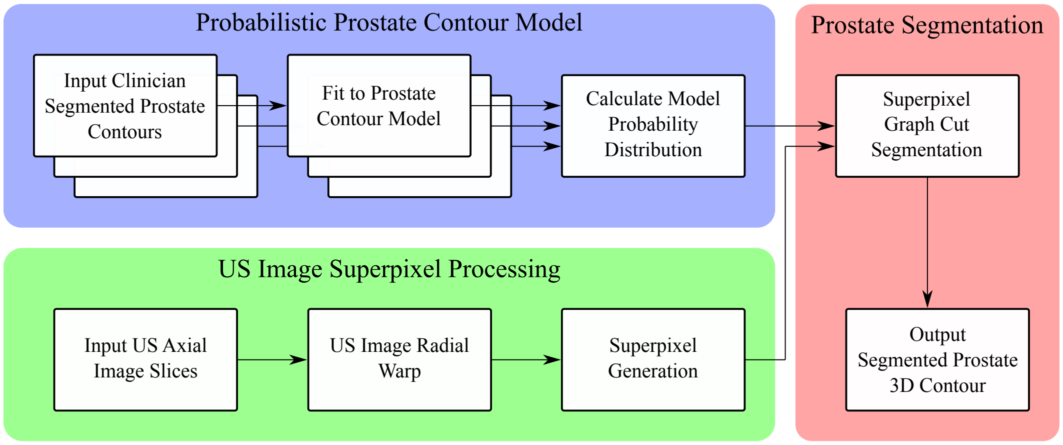

Probabilistic Prostate Contour Model
Input Clinician Segmented Prostate Contours
Fit to Prostate Contour Model
Calculate Model Probability Distribution
Prostate Segmentation
Superpixel Graph Cut Segmentation
US Image Superpixel Processing
Input US Axial Image Slices
US Image Radial Warp
Superpixel Generation
Output Segmented Prostate 3D Contour

Figure 7.2: Overview of prostate segmentation algorithm.

within the ultrasound image can be differentiated. A statistical-shape (or active-shape) prostate contour model is then used to delineate the prostate within the image based on the superpixel regions. Before segmentation, the model is fit to a series of point-based clinician-segmented prostate contours exported from conventional prostate brachytherapy planning software. The algorithm is fully-autonomous, incorporating only the a priori knowledge from the statistical shape model for segmentation of the prostate within B-mode images.

Within the context of this thesis, algorithmic prostate segmentation represents another step towards full computer integration and assistance throughout the pre-operative planning and intra-operative needle insertion stages comprising prostate brachytherapy procedures. This chapter is organized as follows. The prostate contour model and statistical prostate shape model are presented in Sec.7.2. The proposed superpixel algorithm for the ultrasound image processing step is covered in Sec. 7.3. The experimental setup and evaluation of the proposed algorithm as applied to in-vivo patient images are given in Sec. 7.4. The conclusions and results are summarized in Sec. 7.5.

7.1 Prostate Segmentation Overview

Our proposed prostate segmentation algorithm, shown in Fig. 7.2, consists of three main parts: a statistics-based active shape model, superpixel image clustering, and graph-cut based segmentation. In Sec. 7.2, a 3D prostate contour model is proposed, and, in the active shape paradigm, this contour model is fit to a series of 3D clinician-segmented prostate contours. After fitting, the a-posteriori statistics of a Gaussian distribution based model are found. This Gaussian distribution model thus forms the active shape prostate model and is used to estimate the range of values (and corresponding probability) for each parameter in the model.

With the active shape model found, the next step in the prostate segmentation algorithm involves the image processing of a full set of axial images for a patient (capturing the 3D shape of the prostate). As a preprocessing step, outlined in Sec 7.1.2, the input images are warped to compensate for the high radial curvature caused by the TRUS probe. Then a novel superpixel algorithm, given in Sec. 7.3, is used on each image in the set to cluster contiguous pixels together, based on statistical similarity, into large regions of superpixels. By utilizing the statistical dissimilarity between neighboring superpixel regions after processing the edges of the prostate in each axial image can be found.

Conceptually, measuring the statistical dissimilarity between superpixel regions is a form of texture discrimination and edge feature extraction. In [106], a model incorporating pixel cluster covariance was used to create a line and edge feature extractor for generic images. This concept was incorporated in the work of [107], where a line/edge detector was formulated based only on the gradient of the pixel region covariance (change between covariance in neighboring pixel regions). Expanding on the use of covariance for edge extraction, [108] showed that a statistical distance measurement, the Bhattacharyya space, allowed for the discrimination of various textures within an image. The Bhattacharyya space was applied to large-scale region-of-interest (ROI) finding and feature segmentation in [109]. [110, 111] demonstrate that comparable

ROI and feature segmentation can be performed on ultrasound images using speckle and image statistics.

For the final 3D prostate contour segmentation, a graph-cut based method is used to find the most likely prostate contour, based on the active shape model and informed by edge features detected by the superpixel algorithm. This algorithm is thus a hybrid method which incorporates information from an active shape model into a graph-cut based segmentation system. The work of [112] proposed a similar style of hybrid system, where an active contour model is progressively moved and deformed through graph-cut based optimization. This hybrid method was shown to be robust to image noise and large edge discontinuities. This robustness to edge discontinuities is of particular interest in medical imaging, where [113] applied a similar hybrid active contour graph-cut model, and used a series of graph-cut based expansion and contraction steps to move the contour and segment out a kidney in CT image slices. In a similar vein to our proposed method, [114] proposed augmenting an active shape - active appearance model (incorporating both contour and image statistics for the organ of interest) with a graph-cut based optimization to delineate 3D contours of livers, kidneys, and spleens in CT image volumes.

7.1.1 Patient-Attached Coordinate System

For planning in a clinical prostate brachytherapy procedure, a series of ultrasound images are taken as a transrectal ultrasound (TRUS) probe is translated forward along the z-axis (towards the head of the patient)[1]; see Fig. 7.1 and Fig. 7.3. The prostate images are taken at regular distances apart to obtain a 3D volume and shape of the prostate. In standard clinical practice, a clinician manually outlines the contour of the prostate in each image slice in order to delineate the 3D contour of the prostate when the contours across all of the image slices are used together. This contour is then used in dosimetry calculations and a surgical plan, consisting of needle insertion trajectories and target

[1] Please note that, in order to match the terminology and coordinate systems typically seen in the literature, the coordinate axes used here differ from the coordinate systems used in preceding chapters of this thesis.

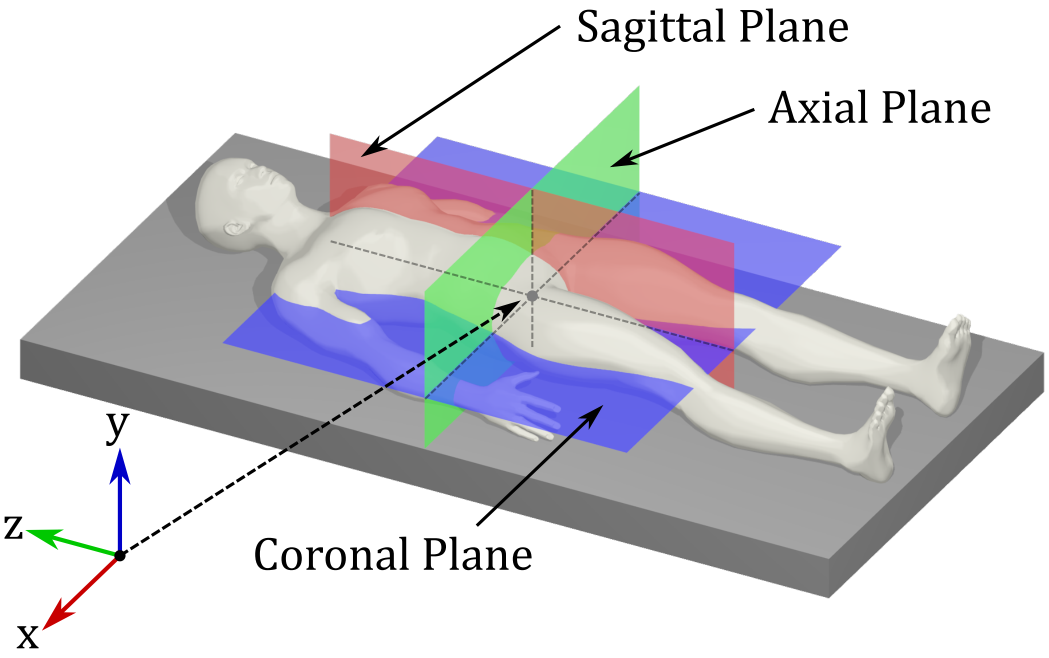


Figure 7.3: Diagram indicating axial, sagittal, and coronal anatomical planes. In this work the patient-attached coordinate system is defined with the x,y,z axes being normal to the sagittal, coronal, and axial planes respectively.

locations for the placement of radioactive seeds within the prostate, is created.

The goal of this work is to automate the 3D prostate contouring process through segmentation of the prostate in all of the TRUS image slices, ${}^{P}\mathbf{I}$, for a particular patient. Each TRUS image, denoted by ${}^{P}\mathcal{I}_i$, within the image set captures an axial slice of the prostate (in the ${}^{P}x$-${}^{P}y$ plane). The image set is being described in terms of its placement and size in the patient attached frame, $\{P\}$, with distances measured using millimetres. Each axial image slice is taken at a depth ${}^{P}z_i$, where, for the patient image sets used in this work, the slices are spaced at 5 mm intervals. The placement of the patient-attached coordinate frame $\{P\}$ is chosen such that the first image in the set ${}^{P}\mathcal{I}_0$ is located at ${}^{P}z_0 = 0$. The full set of image slices taken for a particular patient is given by ${}^{U}\mathbf{I} = \left\{{}^{U}\mathcal{I}_0, {}^{U}\mathcal{I}_1, ..., {}^{U}\mathcal{I}_{nImg}\right\}$, and consists of a number, $nImg$, of image slices taken at depths ${}^{P}d = \{0 \text{ mm}, 5 \text{ mm}, ..., nImg \cdot 5 \text{ mm}\}$, where ${}^{P}z_i$ is the i-th element of ${}^{P}d$.

7.1.2 Ultrasound Image Preprocessing

For convenience, a second frame, $\{U\}$, will be defined for the patient axial TRUS image set, where the measurement units for $\{U\}$ are based on the individual pixel positions in the axial ${}^{U}x$-${}^{U}y$ plane for the ultrasound image; See Fig. 7.4(c). The depth of an image in the ultrasound frame, ${}^{U}z_i$, is given in millimetres and is equivalent to the depth ${}^{P}z_i$. A point within the patient attached frame, ${}^{P}\vec{p} = [\,{}^{P}x,\,{}^{P}y,\,{}^{P}z, 1]^{\top}$, can be transformed by a matrix, ${}^{U}_{P}T$, to a point in the the ultrasound frame, ${}^{U}\vec{p}$, given by

$$\begin{gathered} {}^{U}\vec{p} = {}^{U}_{P}T\,{}^{P}\vec{p} \\ \begin{bmatrix} {}^{U}px \\ {}^{U}py \\ {}^{U}z_i \\ 1 \end{bmatrix} = \begin{bmatrix} s_{px} & 0 & 0 & t_{px} \\ 0 & s_{py} & 0 & t_{py} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{P}x \\ {}^{P}y \\ {}^{P}z_i \\ 1 \end{bmatrix}, \end{gathered} \tag{7.1}$$

with the pixel-to-millimetre scaling of the US image[2] given by s_{px} and s_{px} and a translational offset between the origin of $\{U\}$ with respect to the origin of $\{P\}$ expressed as t_{px} and t_{py}.

Being as the frame $\{U\}$ is used to describe the coordinates of a pixel within an image, the position vector ${}^{U}\vec{p}$ of the pixel will instead be denoted by $({}^{U}px,\,{}^{U}py)$. The ${}^{U}z_i$ coordinate in this notation is omitted for brevity as the image warp, and subsequent superpixel image processing, work on a single 2D image slice at a time. Since the ${}^{U}px$ and ${}^{U}py$ coordinates describe image pixel locations, for an image ${}^{U}\mathcal{I}_i$, the values of ${}^{U}px$ and ${}^{U}py$ will be restricted to being positive whole numbers and limited by the height, ${}^{U}\mathcal{I}_{height}$, and width, ${}^{U}\mathcal{I}_{width}$, of the image, such that $\{{}^{U}px \in \mathbb{N} : 1 \leq {}^{U}px \leq {}^{U}\mathcal{I}_{width}\}$ and $\{{}^{U}py \in \mathbb{N} : 1 \leq {}^{R}py \leq {}^{U}\mathcal{I}_{height}\}$. A function $({}^{U}px,\,{}^{U}py) = \mathrm{Trunc}({}^{U}px,\,{}^{U}py,\,{}^{U}\mathcal{I}_{height},\,{}^{U}\mathcal{I}_{width})$ will first limit and then round any input position value such that it is returned as a valid $({}^{U}px,\,{}^{U}py)$ pixel position. The grayscale intensity (brightness) of an individual pixel in the TRUS image at the location $({}^{U}px,\,{}^{U}py)$ is denoted as ${}^{U}\mathcal{I}_i({}^{U}px,\,{}^{U}py)$.

[2] For the ultrasound images used in this work, the pixel-to-millimetre scaling values are $s_{px} = 5.5769\frac{px}{mm}$ and $s_{py} = -5.5769\frac{px}{mm}$. The origin of $\{P\}$ was defined to be the origin of the ROI coordinate frame, $({}^{U}px_c,\,{}^{U}py_c)$, giving a translation offset from the origin of $\{U\}$ to $\{P\}$ of $t_{px} = 33.4615$ and $t_{py} = 441.1758$.

For computational efficiency, we will use an image warp as a preprocessing step on all of the ultrasound images. This preprocessing step is done to limit the prostate segmentation algorithm to operate within in a region-of-interest (ROI) in the TRUS image. The ROI includes only the actual US image (and excludes any background pixels outside of the ultrasound image field-of-view). The superpixel algorithm, described in Sec. 7.3, will then operate only on pixels within this ROI. Due to the small diameter of the TRUS probe, the ultrasound transducer elements have significant curvature, and the desired ROI consists of a large arc of pixels in the TRUS image; see Fig. 7.4(a).

To isolate the ROI, the input B-mode TRUS image is warped through the use of a modified 2D Cartesian-coordinate to polar-coordinate transform. The resulting image, ${}^{R}\mathcal{I}_d$, will be referred to as the ROI image throughout this chapter. The pixels of the ROI image are referred to by ${}^{R}\mathcal{I}_d({}^{R}px,\, {}^{R}py)$, where a new coordinate frame $\{R\}$ is defined for the pixels with the ROI image. The transformation between a pixel in the TRUS image $\{U\}$ frame and the ROI image frame $\{R\}$ is given by

$$\begin{aligned} {}^{R}px &= s_{rx}\left(\text{atan2}({}^{U}py - {}^{U}py_c,\, {}^{U}px - {}^{U}px_c) + t_{rx}\right) \\ {}^{R}py &= s_{ry}\left(\sqrt{({}^{U}py - {}^{U}py_c)^2 + ({}^{U}px - {}^{U}px_c)^2} + t_{ry}\right), \end{aligned} \tag{7.2}$$

with the point $({}^{U}px_c,\, {}^{U}py_c)$ being the origin of the new ROI (polar-coordinate system). The output ${}^{R}px$ and ${}^{R}py$ values of the transformation are converted to whole pixel index values that lie within a desired ROI image width, ${}^{R}\mathcal{I}_{width}$, and height, ${}^{R}\mathcal{I}_{height}$, using the function $({}^{R}px,\, {}^{R}py) = \text{Trunc}({}^{R}px,\, {}^{R}py,\, {}^{R}\mathcal{I}_{height},\, {}^{R}\mathcal{I}_{width})$. Thus the coordinates of a pixel in the ROI image $({}^{R}px,\, {}^{R}py)$ are in the sets $\{{}^{R}px \in \mathbb{N}, 1 \leq {}^{R}px \leq {}^{R}I_{width}\}$ and $\{{}^{R}py \in \mathbb{N}, 1 \leq {}^{R}py \leq {}^{R}I_{height}\}$. The constants s_{rx} and s_{ry} implement a scaling component within the transformation, so that the output ROI image will be of a desired size, and in the same manner, the t_{rx} and t_{ry} constants are chosen to translate the warped pixels to be within valid ROI image locations. The result of this warping on pixels within the TRUS image ROI can be seen in Fig. 7.4(b) with the coordinate frame $\{R\}$ shown in Fig. 7.4(d). The warped TRUS images will be input into the superpixel algorithm, covered in 7.3, to identify edges and salient features that will be used for prostate segmentation.

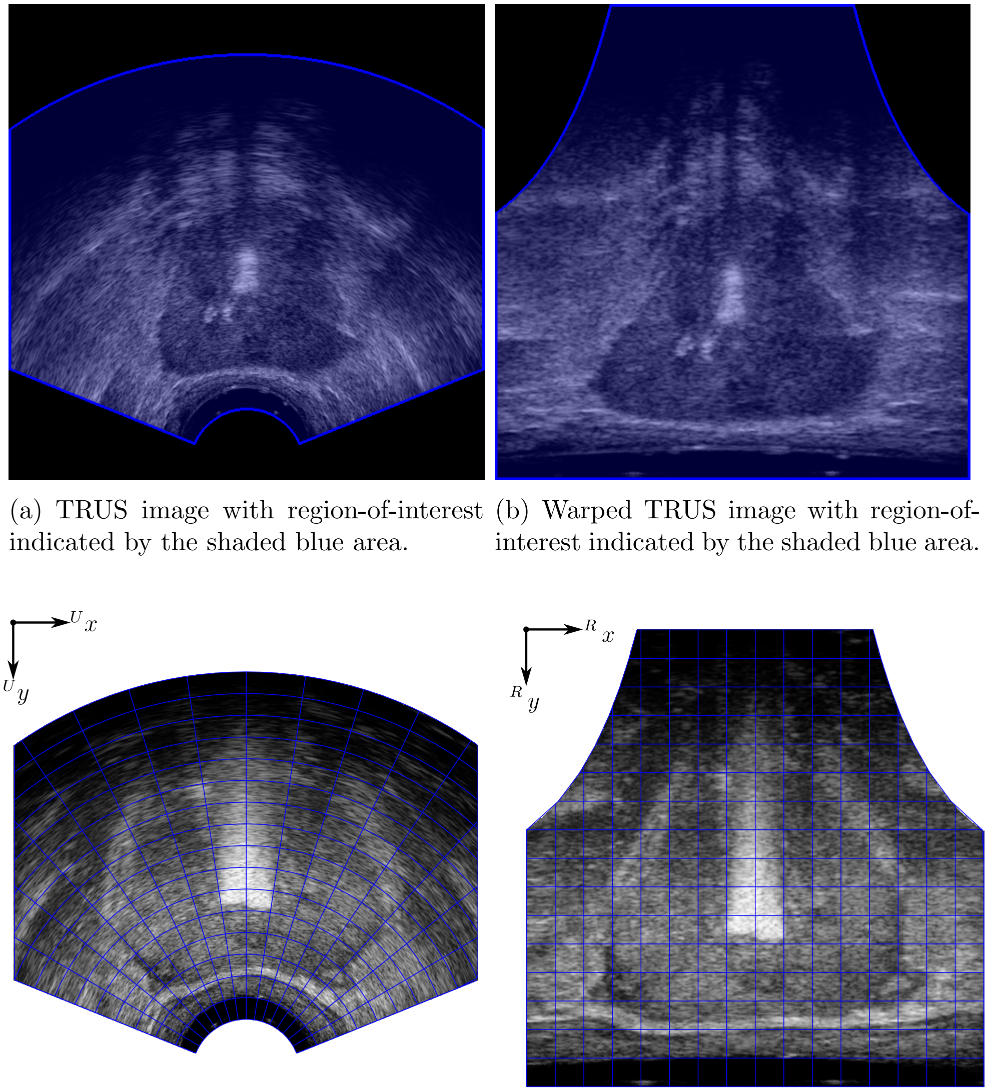
7.2 Active Shape Prostate Contour Model

To evaluate the radiation dosimetry and create a surgical plan for a prostate brachytherapy procedure, a clinician is required to manually segment, or delineate the 3D contour, of the prostate. When using axial US images to create the surgical plan, planning software will allow the clinician to select points in each axial TRUS image which correspond to the edge of the prostate, thereby contouring the prostate. This prostate contour can, in general, be represented by a polygon that is comprised of a number of vertices, $p_{(v,i)} = [{}^{P}x_{(v,i)}, {}^{P}y_{(v,i)}, {}^{P}z_{(v,i)}]^{\top}$, with v being an index for the vertices, in the axial image slice i (from the index of the TRUS image ${}^{P}\mathcal{I}_i$). For each axial slice the polygon can be represented by the vectors

$$\begin{aligned} {}^{P}\vec{x}_i &= [{}^{P}x_{(0,i)}, {}^{P}x_{(1,i)}, ..., {}^{P}x_{(n_v,i)}] \\ {}^{P}\vec{y}_i &= [{}^{P}y_{(0,i)}, {}^{P}y_{(1,i)}, ..., {}^{P}y_{(n_v,i)}] \\ {}^{P}\vec{z}_i &= [{}^{P}z_{(0,i)}, {}^{P}z_{(1,i)}, ..., {}^{P}z_{(n_v,i)}], \end{aligned} \tag{7.3}$$

where n_v is the number of the vertices in the polygon. The position point vector is defined in terms of the patient-attached frame, $\{P\}$, and can be converted, for a particular image slice i, to the pixel locations in the US image by using the transform ${}^{U}_{P}T$, given in (7.1). The pixels inside of the prostate contour polygon in the US pixel frame, $\{U\}$, are defined as being inside the prostate.

To smooth out the point-based contour, we will propose a prostate contour model in Sec. 7.2.1 to describe the shape of the prostate in axial image slices. This contour model is designed to allow for the creation of the prostate shape model and ensures that algorithmic prostate segmentation will result in a closed prostate contour with no self-intersections. The probabilistic prostate shape model, described in Sec. 7.2.3, forms the basis of the probabilistic prostate shape model that will be incorporated, by the graph-cut procedure, into the prostate segmentation algorithm.



(a) TRUS image with region-of-interest indicated by the shaded blue area.

(b) Warped TRUS image with region-of-interest indicated by the shaded blue area.

(c) TRUS image $^{U}x - {}^{U}y$ axes.

(d) Warped TRUS region-of-interest image $^{R}x - {}^{R}y$ axes.

Figure 7.4: Ultrasound image preprocessing region-of-interest and warping.

7.2.1 Axial-image Prostate Contour Model

In this section, we propose a model which describes the shape of the prostate in an axial TRUS image. This sliced-based contour model is designed to incorporate physical and geometric attributes of the prostate in axial images, such that the contour of a the prostate should be a smooth continuous curve with no self-intersections. To achieve this, the model is implemented using a weighted sum-of-sines series within a cylindrical coordinate system[3]. Given a center point ${}^{P}c = [\,{}^{P}x_c,\, {}^{P}y_c]^{\top}$, the standard form of the axes in the cylindrical coordinate system[4] is defined with respect the patient attached frame, such that

$$\begin{aligned} \theta &= \mathrm{atan2}(\,{}^{P}y - \,{}^{P}y_c,\, {}^{P}x - \,{}^{P}x_c) \\ r &= \sqrt{(\,{}^{P}y - \,{}^{P}y_c)^2 + (\,{}^{P}x - \,{}^{P}x_c)^2} \\ z &= \,{}^{P}z, \end{aligned} \tag{7.4}$$

where the angle, θ, is constrained to the range $-\pi \leq \theta < \pi$, by definition r is positive semi-definite, and z is the image slice depth (${}^{P}z = \,{}^{P}z_i$). For each image slice i, The prostate contour model, $R_i(\theta_i)$, describes the radius of a curve, within the cylindrical coordinate system, as a weighted sum-of-sines (a finite length Fourier Series). The general form of the model is given by

$$R_i(\theta) = \alpha_0 + \sum_{j=1}^{n_\omega} \alpha_j \sin(j\theta + \phi_j), \tag{7.5}$$

where n_ω is the number of sine components in the model, j controls the frequency of sine components in the model ($j = [1, 2, ..., n_\omega]$), α_0 is a positive constant ($\alpha_0 > 0$) , the α_j terms weight the magnitude of each of the sine functions, and the ϕ_j terms are phase offsets for the sine functions. The model consists of a number of modes, where the j-th mode refers either to the constant, α_0 for mode 0, or weighted sine component of the model ($\alpha_j \sin(j\theta + \phi_j)$).

[3] The left-hand superscript has been deliberately omitted for axes of the cylindrical coordinate system to indicate that the origin of the system (the center point) is not fixed and changes from prostate to prostate. An optimal center point is found in (7.11) which minimizes the SSD of the curve with respect to the input contour points.

[4] The contour model is based on a Cartesian-to-cylindrical transform of points in the patient-attached frame, $\{P\}$, and should not be confused with the image warping transform given in Sec 7.1.2

In the patient frame, the model describes the curve

$$\begin{aligned} {}^{P}x &= R_i(\theta) \cdot \cos(\theta) + {}^{P}x_c \\ {}^{P}y &= R_i(\theta) \cdot \sin(\theta) + {}^{P}y_c \\ {}^{P}z &= {}^{P}z_i, \end{aligned} \tag{7.6}$$

for a particular image slice i, and the model can thus be seen as imposing a θ dependence onto the radius of a planar curve. The output value of $R_i(\theta)$ must be strictly positive over the domain of theta, where $R_i(\theta) > 0$, $\forall -\pi \leq \theta < \pi$. Enforcing the constraint that $R_i(\theta)$ is strictly positive ensures that the contour model is both smooth, continuous and has no self-intersections in the patient-attached frame.

The modes of the model, j, modify the frequency of the sine components. The first four modes of the model, $j = [1, 2, 3, 4]$, are shown in the patient-attached frame in Fig. 7.5, where the sine component's frequency dependence on j allows for increasing contour curvatures to be described by the model. Of note, the first mode of the model, mode 1, appears to just shift the center point of the model as the weights, α_j, for all of the modes of the model were chosen to be unity; however when $\alpha_0 \neq \alpha_1$ the first mode will result in elliptical curves instead.

In this work, the parameters to be estimated when fitting the model to the manual clinician contours consist of the weight constants α_j, where $j = [0, 1, ..., n_\omega]$. The phase offset values ϕ_j, with $j = [1, ..., n_\omega]$, are chosen before parameter estimation and remain constant during optimization. These phase offsets are kept constant for a number of practical reasons, with the primary reason involving the creation of the statistical prostate shape model, discussed in Sec. 7.2.3, where the parameter values of α_j would not be statistically independent of the value ϕ_j. This parameter dependence would significantly increase the amount of input data (patient contour datasets) and statistical analysis required before the active shape model would be able to perform any useful inference.

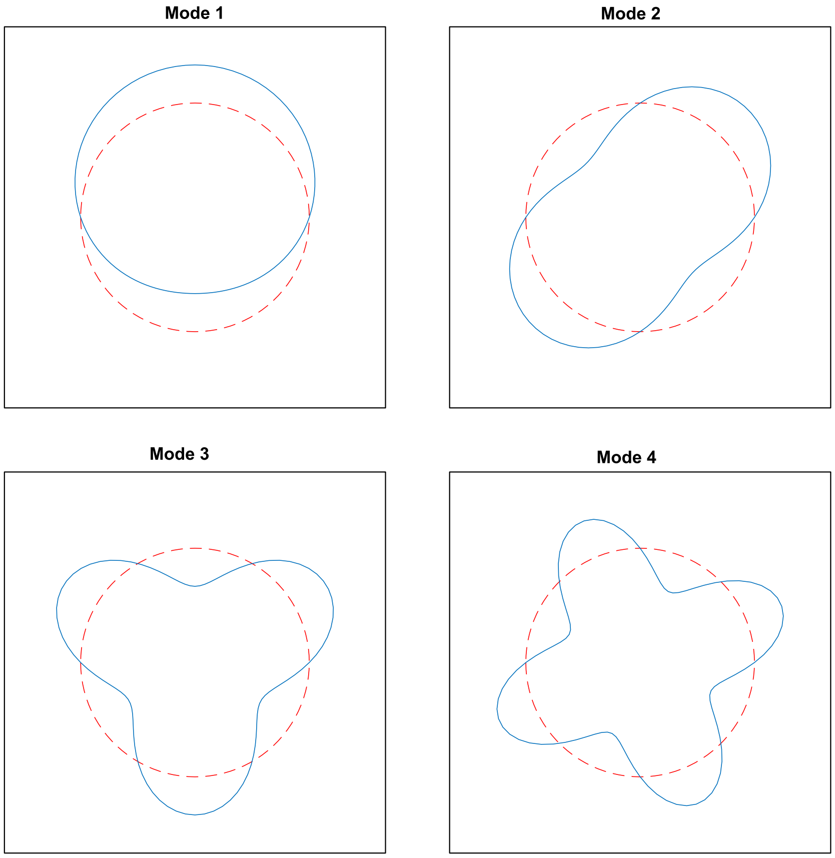


Figure 7.5: Modes one through four of the prostate model overlayed on mode 0 (dashed red line) at $j = 0$.

7.2.2 3D Prostate Contour Model Fitting and Results

A constrained least-squares optimization will be used to fit the manually segmented prostate contour points to the contour model. While the model is defined with respect to the prostate in a single image slice, ${}^{P}\mathcal{I}_i$, the model fitting optimization incorporates information from the entire patient data set, where all $nImg$ slice-based contours in ${}^{P}\mathbf{I}$ will be used. Each slice's contour will be represented through a model, $R_i(\theta)$, with the parameters of the model being estimated for that particular slice. By evaluating the goodness-of-fit for all of the slice models across the patient image set an optimal, patient specific, center point and rotation offset for the model will found.

From the introduction to Sec. 7.2, the contour points of the prostate, segmented by the clinician, are given in the point vector form described by (7.3), with a particular contour, in slice i, having n_v contour points. A rotation offset will be incorporated into the center point vector, such that ${}^{P}c = [{}^{P}x_c, {}^{P}y_c, {}^{P}\phi_c]^{\top}$. This rotation offset aims to reduce any fitting artefacts caused by constant ϕ_j values in the model and forms an important part of prostate shape model generation given in Sec. 7.2.3. Being as the model is formulated in cylindrical coordinates, the prostate contour points, for image slice i, are defined in terms of the patient-attached frame and must be transformed by

$$\begin{aligned} \vec{\theta}_i &= \text{atan2}({}^{P}\vec{y}_i - {}^{P}y_c, {}^{P}\vec{x}_i - {}^{P}x_c) - {}^{P}\phi_c \\ \vec{r}_i &= \sqrt{({}^{P}\vec{y}_i - {}^{P}y_c)^2 + ({}^{P}\vec{x}_i - {}^{P}x_c)^2} \\ \vec{z}_i &= {}^{P}\vec{z}_i, \end{aligned} \tag{7.7}$$

with ${}^{P}\vec{x}_i, {}^{P}\vec{x}_i$, and ${}^{P}\vec{z}_i$ being the clinician-segmented points; the values of ${}^{P}x_c$, ${}^{P}y_c$, and ${}^{P}\phi_c$ come from the augmented center point vector, and $\vec{\theta}_i$, $\vec{r}_i$, and $\vec{z}_i$ are the transformed contour points. The individual elements of the transformed contour point vectors are given by

$$\begin{aligned} \vec{\theta}_i &= [\theta_{(0,i)}, \theta_{(1,i)}, ..., \theta_{(n_v,i)}] \\ \vec{r}_i &= [r_{(0,i)}, r_{(1,i)}, ..., r_{(n_v,i)}] \\ \vec{z}_i &= [z_{(0,i)}, z_{(1,i)}, ..., z_{(n_v,i)}], \end{aligned} \tag{7.8}$$

with each $\left\{\theta_{(v,i)}, r_{(v,i)}, z_{(v,i)}\right\}$ set of points being mapped from the set $\left\{ {}^{P}x_{(v,i)},\ {}^{P}y_{(v,i)},\ {}^{P}z_{(v,i)}\right\}$ of contour points.

The parameters of the model are denoted by $\boldsymbol{\beta}_i = [\alpha_0, \alpha_1, ..., \alpha_{n_\omega}]$ and are found through the least squares optimization

$$\underset{\boldsymbol{\beta}_i}{\operatorname{argmin}} \left\| \vec{r_i} - R_i(\vec{\theta_i}|\boldsymbol{\beta}_i) \right\| \tag{7.9}$$

where $R_i(\vec{\theta_i}|\boldsymbol{\beta}_i)$ is the model, incorporating parameters $\boldsymbol{\beta}_i$, and the optimization is constrained, such that

$$R_i(\vec{\theta_i}|\boldsymbol{\beta}_i) > 0,\ \forall\ -\pi \leq \theta < \pi, \tag{7.10}$$

holds for the fit $\boldsymbol{\beta}_i$ parameters. A second least-squares optimization will be performed in order to find the optimal augmented center point, ${}^{P}c = [{}^{P}x_c,\ {}^{P}y_c,\ {}^{P}\phi_c]^\top$, for fitting the 3D model. The augmented center point corresponds to the center point of the prostate for each patient. Having an optimal ${}^{P}c$ allows for the prostate shape model to be found directly, as finding ${}^{P}c$ removes the relative translation and rotation between two prostate contours in different image sets. Here the parameters to be optimized are $\boldsymbol{\beta}_c = [{}^{P}x_c,\ {}^{P}y_c,\ {}^{P}\phi_c]$. The least-squares optimization, under the constraint given in (7.10), is then

$$\underset{\boldsymbol{\beta}_c}{\operatorname{argmin}} \sum_{i=0}^{nImg} \underset{\boldsymbol{\beta}_i}{\operatorname{argmin}} \left\| \{\vec{r_i}|\boldsymbol{\beta}_c\} - R_i(\left\{\vec{\theta_i}|\boldsymbol{\beta}_c\right\} |\boldsymbol{\beta}_i) \right\| \tag{7.11}$$

where, $\{\vec{r_i}|\boldsymbol{\beta}_c\}$ and $\left\{\vec{\theta_i}|\boldsymbol{\beta}_c\right\}$ are the transformed points calculated from (7.7). Thus the optimal augmented center point is found which minimizes the slice contour model error across all slices, i, in the image set ($i = [0, 1, ..., nImg]$). This fitted contour model, incorporating the prostate contour information from all 2D slices thus describes the 3D contour of the prostate. The resulting 3D contour model, after fitting to all of the patient contour slices, is shown in Fig. 7.6, where input contour points are the red dots and the blue curves are the resulting prostate contour model.

To evaluate the accuracy of the prostate contour model, the mean-squared difference and maximum absolute error between the output model and the

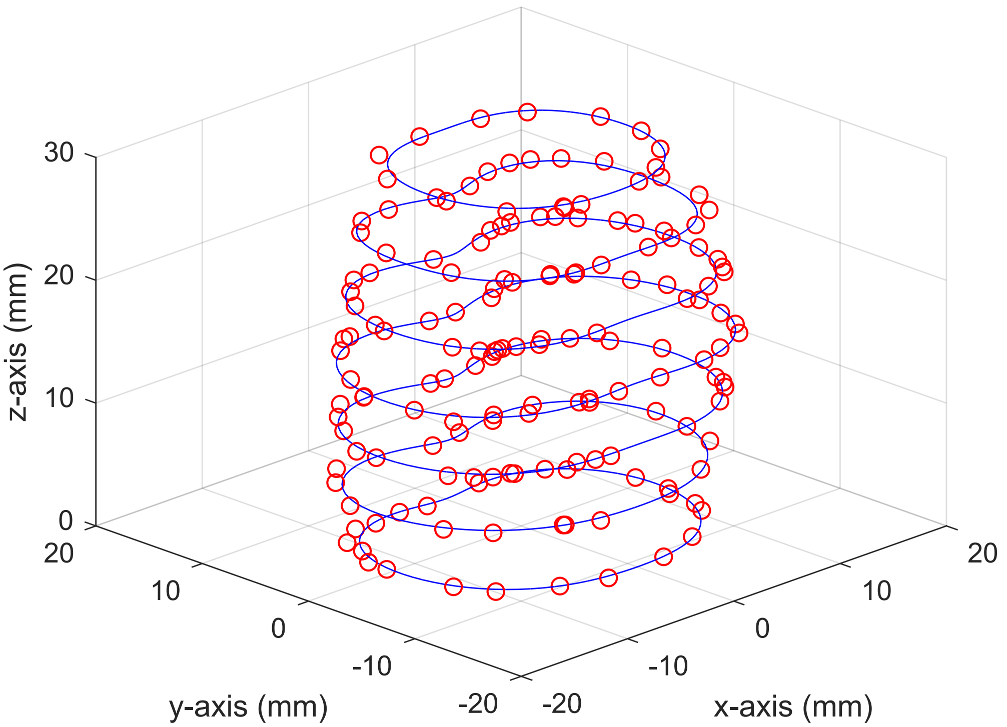


Figure 7.6: Input clinician contour points and resulting 3D prostate contour model curves.

input contour points will be used. The mean-squared error is given as

$$\text{MSE} = \frac{1}{n_{pt}} \sum_{i=0}^{nImg} \{\vec{r}_i|\boldsymbol{\beta}_c\} - R_i(\left\{\vec{\theta}_i|\boldsymbol{\beta}_c\right\}|\boldsymbol{\beta}_i), \tag{7.12}$$

where n_{pt} is the total number of clinician-segmented contour points that the model was fit to. The maximum absolute error is given as

$$\text{MAE} = \max_i \max_v \left| \{\vec{r}_i|\boldsymbol{\beta}_c\} - R_i(\left\{\vec{\theta}_i|\boldsymbol{\beta}_c\right\}|\boldsymbol{\beta}_i) \right|, \tag{7.13}$$

with v being the index of a vertex point within the input slice contour points; hence the MAE is the maximum absolute error between the model and all of the input contour points. These two metrics were evaluated for a set of 100 3D clinician-segmented contours (consisting of 7 to 13 2D slice contours each) based on a TRUS images of the prostate and a set of 100 3D clinician-segmented contours (consisting of 7 to 16 2D contours each) based on MR images of the prostate. While this chapter is focused on US prostate segmentation, the second MR data set was evaluated to see if the model had any bias with respect to input imaging modality. In these results, and for the prostate contour after graph cutting segmentation, only four modes were used in the model ($n_\omega = 4$), and the phase offset values, ϕ_j, were chosen to be $\phi_1 = 0, \phi_2 = \pi/2, \phi_3 = 0$, and $\phi_4 = \pi/2$; as shown in Fig. 7.5. For each contour slice, the four mode model is given by

$$R_i(\theta) = \alpha_0 + \alpha_1 \sin(\theta) + \alpha_2 \cos(2\theta) + \alpha_3 \sin(3\theta) + \alpha_4 \cos(4\theta) \tag{7.14}$$

where the parameters to be fit for slice i are $\boldsymbol{\beta}_i = [\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4]$. The mean value and standard deviation of the MSE and MAE across the 100 contours and the range (minimum-to-maximum) of the MAE for the two data sets is presented in Table 7.1. The results show that the contour model has an average sub-millimetre mean-square error accuracy and a mean maximum absolute error less than 1.6 mm, regardless of imaging modality. The results also indicate that, by using only four modes, the clinician-segmented contours are modelled with acceptable accuracy, where the largest discrepancy (maximum absolute error) between the input clinician contour point and the contour model across the two data sets was under 2.2 mm.

Table 7.1: Prostate Model Fitting Results

Data Set	Average MSE (mm)	Average Max Abs Error (mm)	Range of Max Abs Error (mm)
US Contours	0.611 (±0.165)	1.519 (±0.310)	0.679 − 2.031
MR Contours	0.616 (±0.167)	1.565 (±0.305)	0.794 − 2.124

7.2.3 Statistical Prostate Shape Model

For the prostate segmentation algorithm a 3D probabilistic shape model of the prostate will be created. This shape model is used to find the likelihood of the edge of the prostate being at a particular point. To generate the prostate shape model, a statistical distribution of the prostate contour locations is found by analysis of the prostate shape information contained in a number of 3D contour models from different patients (image sets). The resulting shape model characterizes the average prostate shape, and variance of the prostate shape, measured across a large number of 3D contour models.

After fitting in (7.11), a particular 3D contour model, given an index m, will be denoted by

$$R(\theta, i|\boldsymbol{\beta}_m) = R_i(\theta|\boldsymbol{\beta}_i) \tag{7.15}$$

where $\boldsymbol{\beta}_m$ contains the model parameters of all the slices, m_{nImg} for a particular prostate, such that $\boldsymbol{\beta}_m = [\boldsymbol{\beta}_0, \boldsymbol{\beta}_i, ..., \boldsymbol{\beta}_{m_{nImg}}]$.

The probabilistic prostate shape model will incorporate many, n_{Model}, 3D contour models from different patient TRUS images, and is created using linear interpolation of Gaussian distributions. The radius R, from the 3D contour models, will be evaluated at a discrete number of θ_c points to build the probability model for each slice. The probabilistic prostate shape model will then be in the same slice-based 3D form as the contour model, in the cylindrical coordinate system. Each Gaussian component of the shape model will be centered at $\vec{\theta} = [\theta_1, \theta_2, ..., \theta_{n_\theta}]$ points with the vector

$$\vec{\theta} = \left[-\pi, -\pi + 2\pi\frac{1}{n_\theta - 1}, -\pi + 2\pi\frac{2}{\theta - 1}, ..., -\pi + 2\pi\frac{n_\theta - 1}{n_\theta - 1}\right] \tag{7.16}$$

giving n_θ points sampled at regular intervals in the range $(-\pi, \pi]$. The Gaussian normal distribution for a single point, θ_c, in contour slice i, is

$$p_c(r|\theta_c, i) = \frac{1}{w_{(c,i)}\sqrt{2\pi\sigma^2_{(c,i)}}} \exp\left(-\frac{(r-\mu_{(c,i)})^2}{2\sigma^2_{(c,i)}}\right), \tag{7.17}$$

where $w_{c,i}$ is a normalization constant, $\mu_{(c,i)}$ is the mean radius, and $\sigma_{(c,i)}$ is the standard deviation for all of the shape model at the θ_c point. Note that the θ_c points are used for all slices of the shape model. The CDF of this distribution is given by

$$\begin{aligned} P_c(R \leq r|\theta, i) &= \int_0^r p(r|\theta, i)dr - \int_{-\infty}^0 p(r|\theta, i)dr \\ P_c(R \leq r|\theta, i) &= \frac{1}{2}\left[\text{erf}\left(\frac{(r-\mu_{(c,i)})}{\sigma_{(c,i)}\sqrt{2}}\right) - \text{erf}\left(\frac{(0-\mu_{(c,i)})}{\sigma_{(c,i)}\sqrt{2}}\right)\right] \end{aligned} \tag{7.18}$$

where erf() is the error function, and the normalization constant is given by

$$w_{(c,i)} = \frac{1}{2}\left[1 - \text{erf}\left(\frac{(0-\mu_{(c,i)})}{\sigma_{(c,i)}\sqrt{2}}\right)\right] \tag{7.19}$$

such that the probability of $P_c(0 \leq R \leq \infty|\theta_c, i) = 1$ as desired. With this normalization constant, the mean and standard deviation of the radius at θ_c, across all n_{Model} contour models, are calculated by

$$\mu_{(c,i)} = \frac{1}{n_{Model}} \sum_{m=1}^{n_{Model}} R(\theta_c, i|\boldsymbol{\beta}_m) \tag{7.20}$$

and

$$\sigma_{(c,i)} = \frac{1}{n_{Model}+1} \sum_{m=1}^{n_{Model}} \left(R(\theta_c, i|\boldsymbol{\beta}_m) - \mu_{(c,i)}\right)^2 \tag{7.21}$$

in the same manner as a standard Gaussian distribution.

To find the contour edge probability for any value of θ in the slice, linear interpolation between the two closest θ_c distributions is used. For a point θ, let θ_c^- be the closest θ_c point that is less than θ and θ_c^+ be the closest θ_c which is greater than θ. If θ is between the largest θ_c and π ($\theta_{n_\theta} < \theta \leq \pi$), then $\theta_c^- = \theta_{n_\theta}$ and $\theta_c^+ = -\pi$, so that the contour probabilities at $-\pi$ and π are the same. The interpolated probability for all θ values is then

$$p(r|\theta, i) = \left(\frac{\theta - \theta_c^-}{n_\theta - 1}\right) p_c(r|\theta_c^-, i) + \left(1 - \frac{\theta - \theta_c^-}{n_\theta - 1}\right) p_c(r|\theta_c^+, i) \tag{7.22}$$

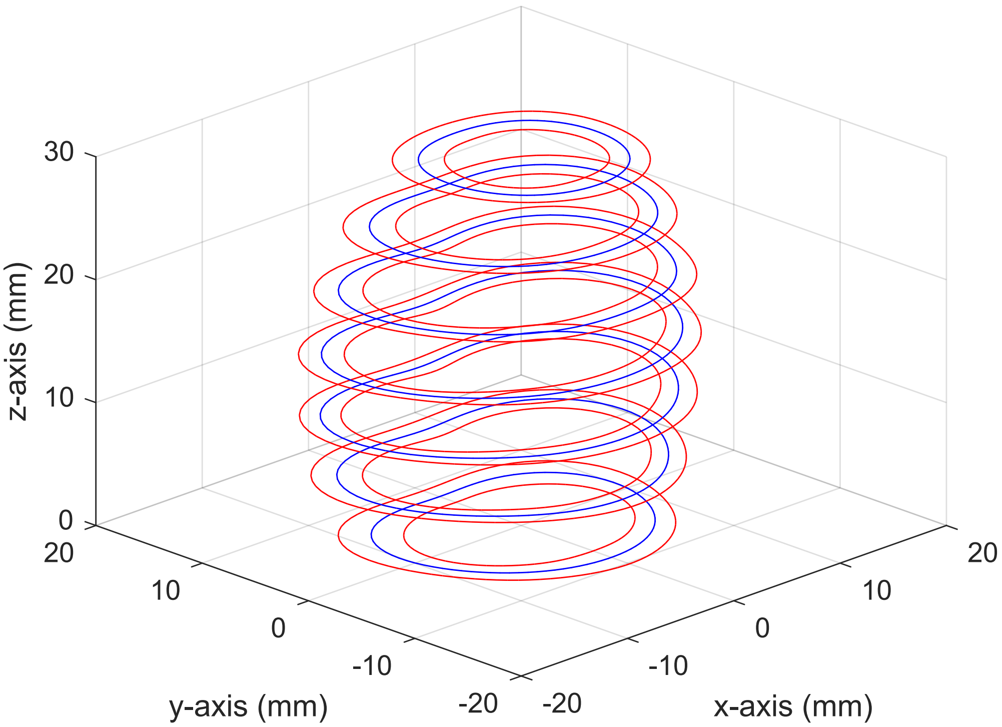


Figure 7.7: Probabilistic prostate shape model for entire prostate with the blue lines representing the mean contour location and the red lines showing one standard deviation away from the mean.

where normalization of (7.17) means that the interpolated shape model value is also a normalized (standard) probability. The CDF of the probability for the shape model for any point θ is given by

$$\begin{aligned} P(R \leq r|\theta, i) = \left(\frac{\theta - \theta_c^-}{n_\theta - 1}\right) P_c(R \leq r|\theta_c^-, i) + \\ \left(1 - \frac{\theta - \theta_c^-}{n_\theta - 1}\right) P_c(R \leq r|\theta_c^+, i) \end{aligned} \tag{7.23}$$

where $R \leq r$ is the probability that the edge of the prostate is located at some point less than r for a given value of θ. The resulting mean and standard deviation of the probabilistic prostate shape model, from a set of 100 prostate contours segmented by clinicians from TRUS images is shown in Fig. 7.7.

When creating the shape model, the number of contour slices between 3D models may vary; $nImg_0$ may or may not equal $nImg_1$ for instance. The number of slices, $nSlice$, in the prostate shape model will be limited to the smallest number of slices, $nImg$, across all of the 3D contour models. To line up the slices, the prostate shape model components are calculated for all models with $nSlice$ images. For any contour model with more than $nSlice$ images, $nSlice$ components of the 3D contour model are compared to the mean prostate shape components, such that

$$MD(i_z) = \sum_{i=0}^{nSlice} \sum_{c=1}^{n_\theta} \left| R(\theta_c, i + i_z | \boldsymbol{\beta}_m) - \mu_{(c,i)} \right| \tag{7.24}$$

with i_z being a slice index offset. The model difference is evaluated starting at index $i_z = 0$, the first $nSlice$ components are compared to the mean 3D contour in the prostate shape model, then the next $nSlice$ components, $i_z = 1$, are evaluated, continuing until $i_z = nImg_m - nSlice$. The set of slices with the lowest model difference, starting at i_z, will be incorporated into the prostate shape model by updating $\mu_{(c,i)}$ and $\sigma_{(c,i)}$. Essentially, the $nSlice$ components from each 3D contour model which best match the prostate shape model are used.

7.3 Superpixel Optimization Incorporating Image Statistics

In this section, we propose a superpixel algorithm which incorporates statistical information from an image to detect edges and salient features. Superpixel algorithms, such as the SLIC algorithm[105], group individual pixels within an image into large clusters (or superpixels). An iterative method is used to update which pixels are grouped into each cluster to minimize a score or distance function. This work will follow a similar design for pixel-based clustering and proposes an additional optimization (or clustering) layer that is built on top of, and groups together, the pixel-based clusters. The regions in the input image delineated by this second layer will be considered the output superpixels of the algorithm, and will be used as the basis for the prostate contouring and segmentation covered in Sec. 7.2. The algorithm will update and optimize the pixel clusters and superpixel regions in an iterative manner, where in each iteration the pixel clusters will first be updated (Sec. 7.3.1) and then the superpixel regions will be updated (7.3.2). An outline of the complete algorithm along with an example of the processing and output of our proposed superpixel algorithm will be given in Sec. 7.3.3.

While the target application of our proposed superpixel algorithm is ultrasound image segmentation, it can be applied in theory to aid in the segmentation of any generic grayscale image. To more closely match the notation used in the literature[5], in this section $\mathcal{I}(i)$ refers to the pixel intensities (brightness values) of an input image and individual pixels are denoted by an index i or equivalently a 2D position (x_i, y_i).

7.3.1 Pixel Clustering

At the pixel level, superpixel clustering will be performed on an input image $\mathcal{I}(i)$, where $\mathcal{I}_{width}$ and $\mathcal{I}_{height}$ are the width and height of the image. An iterative method will be used to group individual pixels, i, in $\mathcal{I}(i)$ into clusters C_j. A pixel may only belong to a single cluster, and from the work of [105], a label image[6], $\mathcal{L}(i)$, is created to indicate which cluster that the pixel is grouped into, where $\mathcal{L}(i) = j$ signifies pixel i is a member of cluster C_j. The pixel-based clustering will be described in two separate stages, the initialization stage, where initial pixel clusters are created from the input image, and the update stage, which is done iteratively to optimize which cluster a particular pixel belongs to.

At initialization, the label image is divided into a regular grid of squares (or approximately a regular grid for a non-square input image). The chosen height and width of each grid, s_{grid}, breaks the input image into n_s square sections, where $n_s = \frac{\mathcal{I}_{width} \cdot \mathcal{I}_{height}}{s^2_{grid}}$ (or $n_s \leq \frac{\mathcal{I}_{width} \cdot \mathcal{I}_{height}}{c^2_{grid}}$ when the input image is non-square). To assign the pixels to an initial set of clusters, each grid square is given a unique index, j, where $j = \{1, 2, .., n_s\}$. The label for all of the pixels within a grid square is then updated, $\mathcal{L}(i) = j$, to match.

With the label image created and the initial pixel clustering performed, an

[5] The superpixel algorithm will be applied to the warped ROI images after preprocessing, where $\mathcal{I}(i) \equiv {}^{R}\mathcal{I}_d({}^{R}px, {}^{R}py)$ with the pixel index i and its corresponding location $(x_i, y_i) \equiv ({}^{R}px, {}^{R}py)$

[6] The label image $\mathcal{L}(i)$ is identical in size (width and height) to the input image $\mathcal{I}(i)$.

array of cluster elements, $\mathbf{C}$, will be created. The array takes the form

$$\mathbf{C} = [C_0, C_1, ..., C_{n_c}]^\top, \tag{7.25}$$

where each element is a pixel cluster, C_j, containing a set of parameters to be calculated. The cluster parameters are expressed as

$$C_j = [\mu_j, x_j, y_j], \tag{7.26}$$

with (x_j, y_j) being the centroid of the cluster and μ_j denoting the average intensity of the pixels in the cluster. The values of μ_j,x_j, and y_j are computed through

$$\begin{aligned} \mu_j &= \sum_{i \in C_j} \frac{\mathcal{I}(i)}{n_{jpx}} \\ x_j &= \sum_{i \in C_j} \frac{x_i}{n_{jpx}} \\ y_j &= \sum_{i \in C_j} \frac{y_i}{n_{jpx}}, \end{aligned} \tag{7.27}$$

where x_i and y_i are the x-y coordinates of the pixel i and n_{jpx} is the number of pixels $i \in C_j$ (the set of all pixels i in the cluster C_j).

The distance of an individual pixel i with respect to the cluster to which it belongs, C_j, will be stored in a distance image, $\mathcal{D}(i)$, which is of the same size as the input image, $\mathcal{I}(i)$, and which will be used when updating the pixel label image $\mathcal{L}(i)$. To ignore regions in the US image which contain invalid pixels, for instance those pixels in the warped US image that are outside the desired ROI ($i \notin \mathcal{I}_{ROI}$, see Fig. 7.4(b)), the distance image will be initialized to allow only valid pixels to be considered during cluster updating. The pixel value in the distance image will be set to infinity (or a large positive value) for pixels inside of the ROI and -1 for pixels outside of the ROI, such that

$$\mathcal{D}(i) = \begin{cases} \infty & \text{for } i \in \mathcal{I}_{ROI} \\ -1 & \text{for } i \notin \mathcal{I}_{ROI} \end{cases} \tag{7.28}$$

with $i \in \mathcal{I}_{ROI}$ and $i \notin \mathcal{I}_{ROI}$ being the pixels inside of and outside of the ROI, respectively. The invalid pixels are removed from their respective clusters through modifying the label image, such that $\mathcal{L}(i) = 0$, $\forall \, \{i \notin \mathcal{I}_{ROI}\}$. If there

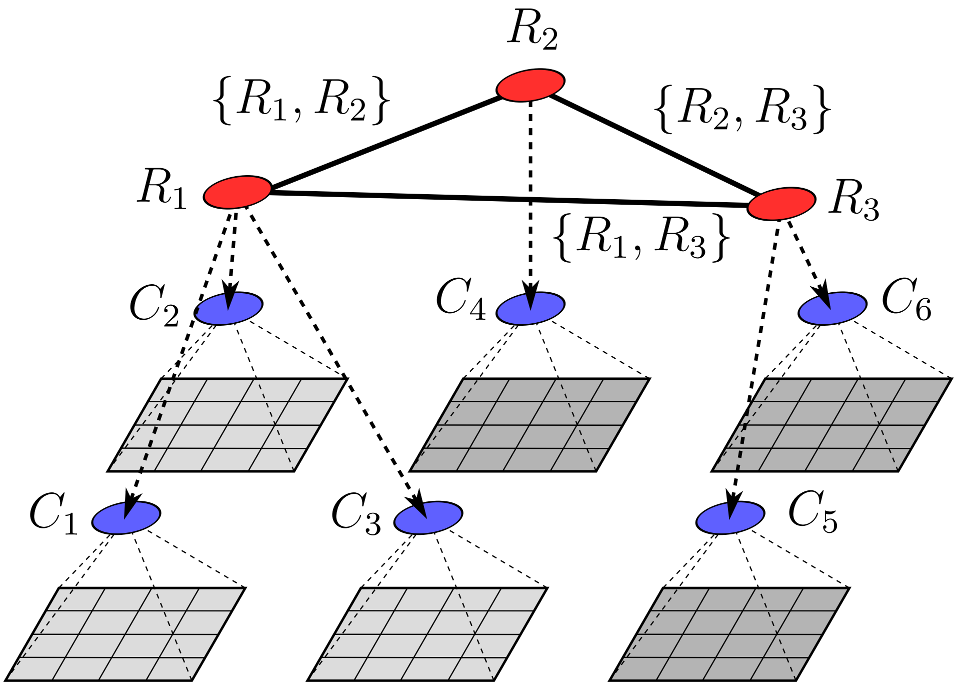


Figure 7.8: Hierarchy created by the superpixel algorithm, showing individual pixels, pixel clusters (C_1, C_2, C_3, C_4, C_5 and,C_6), and superpixel regions (R_1, R_2, and R_3), where the regions are the graph verticies interconnected with graph edges.

are large areas outside of the ROI then there may be several clusters which no longer contain any pixels. In this case, the clusters without any pixels are removed from the cluster array so that **C** only contains valid pixel clusters.

7.3.2 Superpixel Regions

The input image pixels are now grouped into SLIC style clusters as proposed in [105], through the creation of the label image. We now propose our extension to the SLIC superpixel algorithm with a second optimization layer. This additional layer is the mechanism through which image statistics will be incorporated into the superpixel algorithm. The underlying implementation of the second layer incorporates a graph component, which will allow for the output of the superpixel algorithm to be segmented by graph cutting in a straightforward manner. The components of this second layer will be referred to as superpixel regions, which will be stored in a list **R**, given by

$$\mathbf{R} = [R_0, R_1, ..., R_n],^\top \tag{7.29}$$

with R_k denoting the individual regions. In a similar manner to a pixel cluster, C_j, being comprised of a number of individual pixels, a region R_k is comprised of a number of pixel clusters (C_j), thus building a second optimization layer on top of the pixel clusters. This second layer creates a pyramid-type hierarchy in this superpixel implementation, as diagrammed in Fig. 7.8.

Each region consists of a series of calculated values, given by

$$R_k = [\hat{\mathbf{F}}_{\mathbf{k}}, x_k, y_k, j_k^\star], \tag{7.30}$$

where $j_k^\star$ is the list of indexes to the pixel clusters (C_j) which make up the superpixel region, (x_k, y_k) is the centroid of the region, and $\hat{\mathbf{F}}_{\mathbf{k}}$ is the empirical cumulative density function (CDF) of all of the pixels contained within the region. Being as a region is made up of a number of pixel clusters, all of the pixels belonging to the region's subordinate clusters are defined to be pixels of the region. Formally this is described as the union of all of the pixels in the component clusters of R_k, given as

$$i \in R_k = \bigcup_{j \in k} i \in C_j, \tag{7.31}$$

where $j \in R_k$ are the component clusters of the region. The centroid values of the region, x_k and y_k, are calculated as follows

$$\begin{aligned} n_{kpx} &= \sum_{j \in R_k} n_{jpx} \\ x_k &= \sum_{j \in R_k} \frac{n_{jpx} \cdot x_j}{n_{kpx}} \\ y_k &= \sum_{j \in R_k} \frac{n_{jpx} \cdot y_j}{n_{kpx}}, \end{aligned} \tag{7.32}$$

with n_{jpx} being the number of pixels in a cluster and n_{kpx} being the total number of pixels within the region. The empirical CDF, $\hat{\mathbf{F}}_{\mathbf{k}}$, allows for a metric based on pixel intensity (image) statistics within a region; the CDF is calculated through

$$\begin{aligned} \hat{F}_k(x) &= \text{Count}\left(\mathcal{L}(i \in R_k) \leq x\right) \\ \hat{\mathbf{F}}_{\mathbf{k}} &= [\hat{F}_k(0), \hat{F}_k(1), ..., \hat{F}_k(255)], \end{aligned} \tag{7.33}$$

given a counting function, Count(), which returns the number of pixels within a set, in this case, the set of pixels in the region R_k with an intensity value,$\mathcal{L}(i)$, less than I. This is essentially the cumulative summation of all the elements in the image intensity histogram, for all of the pixels in the region $i \in R_k$.

The superpixel region optimization uses a graph-based component, $\mathcal{G}$ to keep track of which regions are adjacent to one another, called neighbours. For this work, an undirected graph of the form

$$\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}, \tag{7.34}$$

is used. Here $\mathcal{V}$ are the vertices of the graph which are interconnected by edges $\mathcal{E}$. The vertices of the graph will be the superpixel regions and edges define which regions are neighbours to one another; the vertices and edges have the form

$$\mathcal{V} = \{0, 1, ...n_R\} \tag{7.35}$$

$$\mathcal{E} = \{k^{\star}_{v1}, k^{\star}_{v2}\}, \tag{7.36}$$

with $\mathcal{V}$ being a set of all of the region indexes (k). The edges in the graph $\mathcal{E}$ are described by a set of double-valued elements, k_{v1} and k_{v2}, with an individual line edge element connecting two vertices given by $E_v = (k_{v1}, k_{v2})$, where k_{v1} and k_{v2} are the indexes of two neighbouring regions R_{k1} and R_{k2}. With the graph being undirected, the ordering of the elements in an edge is ignored, such that $(k_{v1}, k_{v2}) \equiv (k_{v2}, k_{v1})$. Additionally, under this independence from element ordering, the connection between two vertices k_{v1} and k_{v2} is described by a single edge without duplication. For example, only one of the two edge elements $E_a = (k_{v1}, k_{v2})$ and $E_b = (k_{v2}, k_{v1})$ are allowed to be a member of the set $\mathcal{E}$.

The adjacency graph is used to definite a region's neighbours and will be used when optimizing region placement and size. The neighbourhood of a region will also be used as part of the weighting to determine the optimal prostate segmentation graph-cut. The neighbours to a region, R_k, are defined as a set of all of the regions, R_m, which have a graph edge connected to R_k. The set of neighbours, $\mathcal{N}_k$, is derived from the elements $E_v = (k_{v1}, k_{v2})$ in the edge set $\mathcal{E}$, where $\mathcal{N}_k$ is given by

$$\begin{aligned} N_{k1} &= \{k_{v1}, \forall E_v \text{ where } k_{v2} = k\} \\ N_{k2} &= \{k_{v2}, \forall E_v \text{ where } k_{v1} = k\} \\ \mathcal{N}_k &= N_{k1} \cup N_{k2} \end{aligned} \tag{7.37}$$

such that $\mathcal{N}_k$ contains all of the k_{v1} and k_{v2} regions that are paired with (connected to) k in any edges of the graph $\mathcal{E}$.

To initialize the regions and the graph, a region R_k is created for each cluster C_j such that $k = j$, resulting in an initial one-to-one mapping between the clusters and regions. Being as pixel clusters are created in a grid pattern (see Sec. 7.3.1 and Fig. 7.11(a)), the graph $\mathcal{G}$ can be easily created by finding the (4-connected) grid squares that adjoin each other; see Fig. 7.11(b) and Fig. 7.11(c). With the pixel clusters, superpixel regions, and graph structure created, the algorithm will then work to reduce the number of superpixel regions while simultaneously enlarging the area (number of pixels) contained within those regions. The relationship between the individual pixels, the pixel clusters, and superpixel regions along with the graph structure are shown in Fig. 7.8. Explicitly, in Fig. 7.8, R_1 is a superpixel region comprised of three pixel clusters C_1, C_2, and C_3, with neighbouring regions R_2 and R_3 connected via graph edges.

7.3.3 Superpixel Algorithm

After both the pixel clusters and superpixel regions have been initialized, the algorithm will step through a series of update steps, such that large superpixel regions will be created. These update steps work to reduce the number of regions in an image by increasing the area (number of pixels) belonging to each region. Using the region layer and its associated graph, the algorithm works to decrease the number of regions by joining together statistically similar regions, returning a smaller number of larger regions each process iteration. This statistics-based approach creates regions from which edges and salient features can be found in a straightforward manner. This section will provide an explanation of the update process, with final segmentation of the resulting superpixel regions described in Sec. 7.4.

Each update step can be broken into two sub-steps, the first of which works at the individual pixel level to determine which pixels belong to a particular cluster, acting to increase or decrease the size of the clusters with each step. The second sub-step controls the growth of the superpixel regions by increasing the number of clusters belonging to each region and grouping together pixel clusters to maximize region size. This region update step incorporates the region statistics to only group together regions which have comparable image statistics, thus separating regions which vary in average pixel intensity and standard deviation (or covariance) of pixel intensity. This statistical separation uses the premise developed in [107], where edge and corner features in a generic image have been shown to have distinct statistical properties from smooth features in the image. During the region update step a hard limit is placed on how dissimilar two regions can be when joined together, with small regions being retained if they are distinct enough from neighbouring regions.

The algorithm uses a few parameters which can be tuned to control how the pixel clusters and superpixel regions are modified during updating. The use of these parameters will be described in full in the text for the cluster update and region update sub-steps. One of the parameters of the algorithm is a weighting vector, λ, is used to determine the goodness-of-fit for a particular pixel to a cluster and to limit region-to-region statistical similarity. The values within this vector are

$$\lambda = [\lambda_\mu, \lambda_{L2}, \lambda_{KS}]\,, \tag{7.38}$$

with λ_μ and λ_{L2} being used in the cluster update step to weight a distance function (score) between a pixel and nearby clusters. The value λ_{KS} will be described in the region update step and defines the limit on how dissimilar two regions must be before they cannot be merged.

There are a few constraints on the output superpixels of the algorithm which will either be enforced by the algorithm or be checked at the end of each iteration to determine when the optimization is complete (to be discussed below). The desired size of the output superpixel regions is defined as r_{darea}

with constraints on the minimum pixel cluster and region area, c_{min} and r_{min} respectively. The minimum acceptable size for the regions is chosen to be equivalent to or larger than the size of the initial grid squares such that $r_{min} \geq s_{grid}$ with the clusters being required to be larger than c_{min}, where $c_{min} = \alpha_c r_{min}$ with α_c being a scaling constant taking values within the range $0 < \alpha_c \leq 1$. For this work a value of $\alpha_c = 0.75$ was found to be sufficient. A limit is also placed on the maximum number of pixel clusters, c_{max}, allowed within each region. The desired number of regions returned by the algorithm is controlled by the parameter r_{dnum}, where, nominally, there will be r_{dnum} or fewer regions when the algorithm terminates.

7.3.4 Cluster Update

Two distance functions are used during the pixel cluster update operations which score which cluster, C_j, an individual pixel, i, should belong to. The calculation of this score is based on two different distance metrics, $D_\mu(i,j)$ and $D_{L2}(i,j)$. The metric $D_\mu(i,j)$ is the absolute difference between the pixel's intensity (in the input image $\mathcal{I}(i)$) with respect to the average pixel intensity, μ_j, of a cluster. The other metric $D_{L2}(i,j)$ is defined as the Euclidean distance from the pixel's coordinates, (x_i, y_i), to the centroid, (x_j, y_j), of the cluster. A distance score, $D_{Score}(i,j)$, for a pixel is calculated as

$$\begin{aligned} D_\mu(i,j) &= |\mu_i - \mathcal{I}(i)| \\ D_{L2}(i,j) &= \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \\ D_{Score}(i,j) &= \left(\frac{D_\mu(i,j)}{\lambda_\mu}\right) + \left(\frac{D_{L2}(i,j)}{\lambda_{L2}}\right) \end{aligned} \tag{7.39}$$

with λ_μ and λ_{L2} being two tunable parameters that weight the importance of the average intensity distance and Euclidean distance. The distance value of each pixel, with respect to the cluster to which it currently belongs, is stored in the distance image $\mathcal{D}(i)$.

The goal of the cluster update step is to minimize the total of the distance score for all (valid) pixels in the image. Here the method for updating the clusters is similar in most respects to the SLIC superpixel algorithm [105]. Minimization of the total distance score is achieved by moving a pixel to a different cluster if the pixel's score, with respect to that new cluster, is less than its current score value $\mathcal{D}(i)$. To determine if a pixel should be moved, the distance score for each pixel, i, is calculated with respect to each cluster C_j. This calculation is evaluated individually for all of the clusters within the cluster array $\mathbf{C}$ in a loop. If, during the loop, the distance score, $D_{Score}(i,m)$, for a cluster, C_m, is less the current value in $\mathcal{D}(i)$, then the pixel is relabelled through the function

$$\mathcal{L}(i) = m \text{ if } D_{Score}(i,m) < \mathcal{D}(i), \tag{7.40}$$

so that the pixel is now a member of the cluster C_m. To speed up processing, only the distance values for pixels within a square region (of size $2r_{desarea} \times 2r_{desarea}$) around each cluster's centroid are evaluated, based on the suggested method in [105]. After all of the clusters have been iterated through, with pixels relabelled to minimize the distance scores, the values in the cluster array, $\mathbf{C}$, are updated using (7.27). Note that the distance image value for invalid pixels, those outside of the ROI, was set to -1 when the distance image was created. Being as the distance score is positive semi-definite ($D_{Score}(i,j) \geq 0$) invalid pixels will never be incorporated into a cluster and do not need to be considered separately.

Region Update

With the cluster update sub-step completed, the region update sub-step is then performed. Region updating incorporates pixel cluster information and the calculation of image statistics to modify the individual regions R_k and the region adjacency graph $\mathcal{G}$. The goal of the region update sub-step is to join neighbouring regions together so that, at the termination of the superpixel algorithm, large statistically similar regions in the input image will be delineated. In a similar manner to the pixel cluster update, a distance metric, $D_{KS}(k_1,k_2)$, will be used to inform the region update process. The distance metric used is the Kolmogorov–Smirnov (KS) test, which gives a distance between two empirical CDFs [115]. The KS test is the mechanism through which changes in image statistics in the input image will be incorporated into the region update process. Using the KS test, regions with similar statistical properties (or equivalently similar empirical CDFs), will be joined together, and conversely, dissimilar regions will remain separated, thus creating regions with distinct image statistics. The region update step consists of two separate optimizations, a region-statistic optimization through performing a region union operation and a subsequent optimization of the pixel cluster through cluster merging.

Region Union

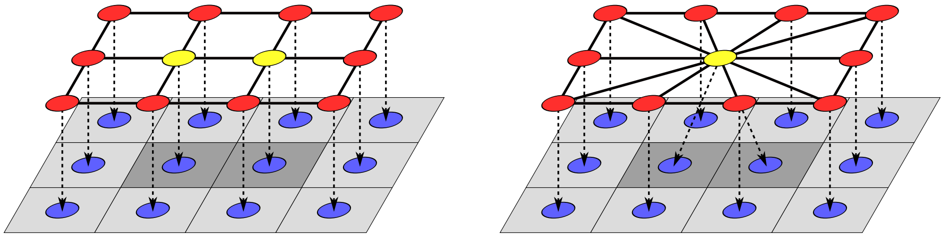
The first operation, the region union, will incorporate the adjacency information stored in the graph $\mathcal{G}$ to evaluate the KS distance between neighbouring regions and to choose which regions a union operation should be performed on. As each region consists of a pixel cluster (or pixel clusters after the first iteration of the algorithm) the values in the region list $\mathbf{R}$ must be updated to incorporate the changes to the pixel clusters during their update step. Values of the centroid of each region, given in (7.32), and the empirical CDF of each region, given in (7.33), need to be calculated. Using these new region list values, the distance metric is computed for each edge $E_v = (k_{v1}, k_{v2})$ in the graph ($E_v \in \mathcal{E}$). The edges describe the interconnection between the region elements, a region's neighbours, and the score measures how statistically similar neighbouring regions are. The score is stored in an array,

$$\mathbf{S} = [s_0, s_1, ..., s_\ell]^\top, \tag{7.41}$$

which contains the same number of entries as the edge structure, $\ell = v_{max}$ where v_{max} is the largest index of edges $E_v \in \mathcal{E}$. The distance score is then calculated along each edge via the function

$$\begin{aligned} D_{KS}(k_i, k_j) &= \sup_x \left| \hat{F}_i(x) - \hat{F}_j(x) \right| \\ s_v &= D_{KS}(k_{v1}, k_{v2}) \end{aligned} \tag{7.42}$$

which returns the largest mismatch between the values of the empirical CDFs, $\hat{F}_k(x)$ (stored in $\hat{\mathbf{F}}_\mathbf{k}$), across all of the input pixel intensities, $x = \{0, 1, ..., 255\}$, seen in the input image $\mathcal{I}(i)$. The KS test has several useful properties, in that it is symmetric $D_{KS}(\hat{F}_i(x), \hat{F}_j(x)) \equiv D_{KS}(\hat{F}_j(x), \hat{F}_i(x))$ and doesn't require the



(a) Superpixel regions before union operation.

(b) Superpixel region after union operation.

Figure 7.9: Union on yellow superpixel regions, showing regions and connections before and after operation.

data to be modelled as belonging to some underlying distribution. The symmetry property of the KS test allows for the distance values to be evaluated only once per graph edge irrespective of the order of elements in an edge E_v. The second property, the independence from an underlying distribution, is critical in the context of ultrasound image processing as the pixel intensities seen in B-mode images have been shown to follow a number of different statistical distributions varying with tissue type, location in the body, ultrasound intensity, and transducer focusing [116].

A low KS score between two regions indicates that they are statistically similar. In a loop, each region R_k and its neighbours $R_m \in \mathcal{N}_k$ will be compared to see if they are statistically similar enough, and if so a union operation will be performed. For each R_k the neighbour with the minimum KS distance,

$$D_{KSmin}(R_k) = \min_{k_m} D_{KS}(k, k_m) \forall k_m \in \mathcal{N}_k, \tag{7.43}$$

giving the index m of the neighbouring region R_m with the minimum statistical difference $D_{KSmin}(R_k)$ to R_k is identified. A binary threshold is then used to determine if a union will be performed,

$$R'_k = R_k \cup R_i \text{ if } \left(D_{KSmin}(k) \leq \lambda_{KS} \mathrm{or} n_{kpx} < r_{min}\right), \tag{7.44}$$

which results in a new region R'_k when R_k and R_m are either similar to one another or, forcing a union, when the area of R_k is smaller than the minimum allowable region size r_{min}. When $n_{kpx} \geq r_{min}$ the threshold λ_{KS} prevents unions between statistically dissimilar regions. If neither condition is met then R_k will be ignored and the next region in the set $\mathbf{R}$ will be tested.

When either of the the union conditions are met, the union results in the two regions being combined to create one larger region; $R'_k = R_k \cup R_m$. This new region, R'_k, takes ownership over all of the pixel clusters that originally belonged to R_k and R_m. From (7.30), the region is described by its parameters, $R'_k = [\hat{\mathbf{F}}'_{\mathbf{k}}, x'_k, y'_k, j'^{\star}_k]$. The indexes for the all of the pixel clusters in this new region, $j'^{\star}_k$, are given by

$$j'^{\star}_k = j^{\star}_k \cup j^{\star}_m, \tag{7.45}$$

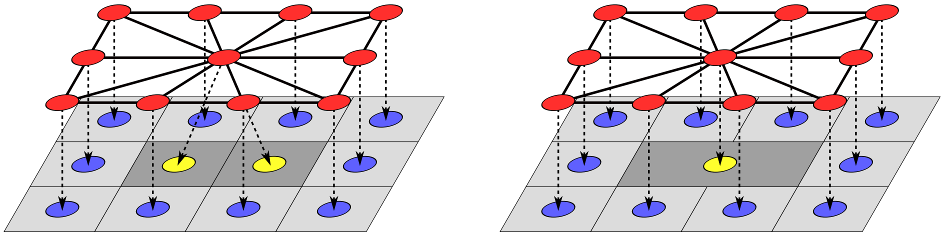
where $j^{\star}_k$ were indexes for the clusters $\{C_j, j \in R_k\}$ and, likewise, $j^{\star}_m$ were indexes for the clusters $\{C_j, j \in R_m\}$. The values in the rest of R_k are then updated, from (7.32) and (7.33), to incorporate all of the information from the pixel clusters in the new region. A diagram of the region union operation is given in Fig. 7.9, where a union is performed on the two regions in Fig. 7.9(a) (indicated in yellow) combining them into the single region in Fig. 7.9(b).

As shown in Fig. 7.9 after the union operation the region list R and the adjacency graph, $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, need to be updated. For simplicity, the newly created region R'_k will replace the parameters of R_k (at row k) in the region list, $\mathbf{R}$, and the row m, corresponding to R_m, is removed from $\mathbf{R}$. To update the graph, the m vertex V_m is deleted and the edge $E_v = (k, m)$ is removed. Any additional edges, $E_v = (k_{v1}, k_{v2})$, which originally connected to m, $\{k_{v1}, k_{v2} = m\}$, will be updated to connect to k. Any duplicate edges in $\mathcal{E}$ will also be cleaned up to ensure that there is only a single edge between vertices connected to one another.

This procedure can continue until all of the regions R_k have been tested. From empirical observation however, it is advantageous to limit the number of unions performed during one algorithm iteration. This allows more time for the pixel cluster layer to converge, where the base SLIC algorithm in [105] was shown to require a number of iterations before satisfactory cluster convergence. For this work, a subset of all of the regions is evaluated for union operations in each iteration. The subset of regions to analyse was chosen to encourage regions to grow in size and to guarantee that small regions, those with an area less than r_{min}, had a union performed on them. To do this, the size values n_{kpx} in the superpixel region list $\mathbf{R}$ were sorted such that the smallest regions are investigated first. A limit, n_{union}, was placed on the number of unions performed at each iteration and, empirically, the superpixels were found to converge well when $n_{union} = 50$ regions per iteration. Using the list of region indexes sorted according to size, all regions smaller than r_{min} are evaluated. If removing those small regions did not exceed the n_{union} unions-per-iteration limit then the algorithm continues working through the sorted list to evaluate regions until n_{union} union operations have been performed. This ensures all regions have an area larger that r_{min} while achieving adequate convergence of the pixel clusters and growth of the superpixel regions.

Cluster Merge

After the cluster update and region union operations, the optimization constraints for the pixel cluster size c_{min} and number of clusters per region c_{max} must be enforced. After the region union procedure, all regions smaller than r_{min} have been placed into nearby regions and the following cluster merge operation works on the pixel cluster components inside of the updated regions. Regions may also now contain more than the maximum allowable number of clusters which will be corrected through the use of the cluster merge operation. In a similar manner to the region update method, the cluster merge method works to join together neighbouring pixel clusters to create new larger, and contiguous, pixel clusters. The cluster merge operation respects the pixel cluster-superpixel region hierarchy, such that a pixel cluster belongs only to a single region, and the merge operation does not move clusters between regions. The cluster merge update loops through all of the individual pixel clusters, C_j, with the size of the cluster, n_{jpx}, being calculated. Any regions smaller than c_{min} are merged with their nearest neighbouring clusters inside of their respective regions. The neighbours of a cluster C_j belonging to a region R_k are defined to be all of the other clusters within that region, $\{C_i \in R_k \neq C_j\}$. The nearest neighbour to the cluster C_j is defined as the neighbouring cluster with



(a) Pixel clusters before merge operation. (b) Pixel cluster after union operation.

Figure 7.10: Union on pixel clusters, showing features before and after operation.

the minimum Euclidean distance between their respective centroids,

$$N_N(C_j) = \min_i \sqrt{(x_j - x_i)^2 + (x_j - x_i)^2} \forall C_i \in\in R_k \neq C_j, \tag{7.46}$$

where x_i, x_j, y_i, y_k are pixel coordinates of the centroids and $N_N(C_j)$ returns the index i of the nearest neighbour to C_j. The merge operation is performed by updating the label image $\mathcal{L}(i)$ such that,

$$\mathcal{L}(j) = i, \; \forall \text{ pixels } j \in C_j, \tag{7.47}$$

which relabels all of the pixels in the cluster $j \in C_j$ to belong to the cluster C_i. Lastly, the row j in the pixel cluster list $\mathcal{C}$ and the pointer $j_k^\star = j$ in the parameters of region R_k, are then removed. Note that the scaling constant α_c between the minimum region size and minimum cluster size $c_{min} = \alpha_c r_{min}$ is within the region $0 < \alpha_c \leq 1$. This ensures that, at the ouput of the region union operation, any region containing a cluster of size $n_{jpx} < c_{min}$ will necessarily have a second cluster that it can be merged with. An example of the cluster merge operation is shown in Fig. 7.10, with the two clusters in Fig. 7.10(a) (indicated in yellow) being combined into the single cluster in Fig. 7.10(b).

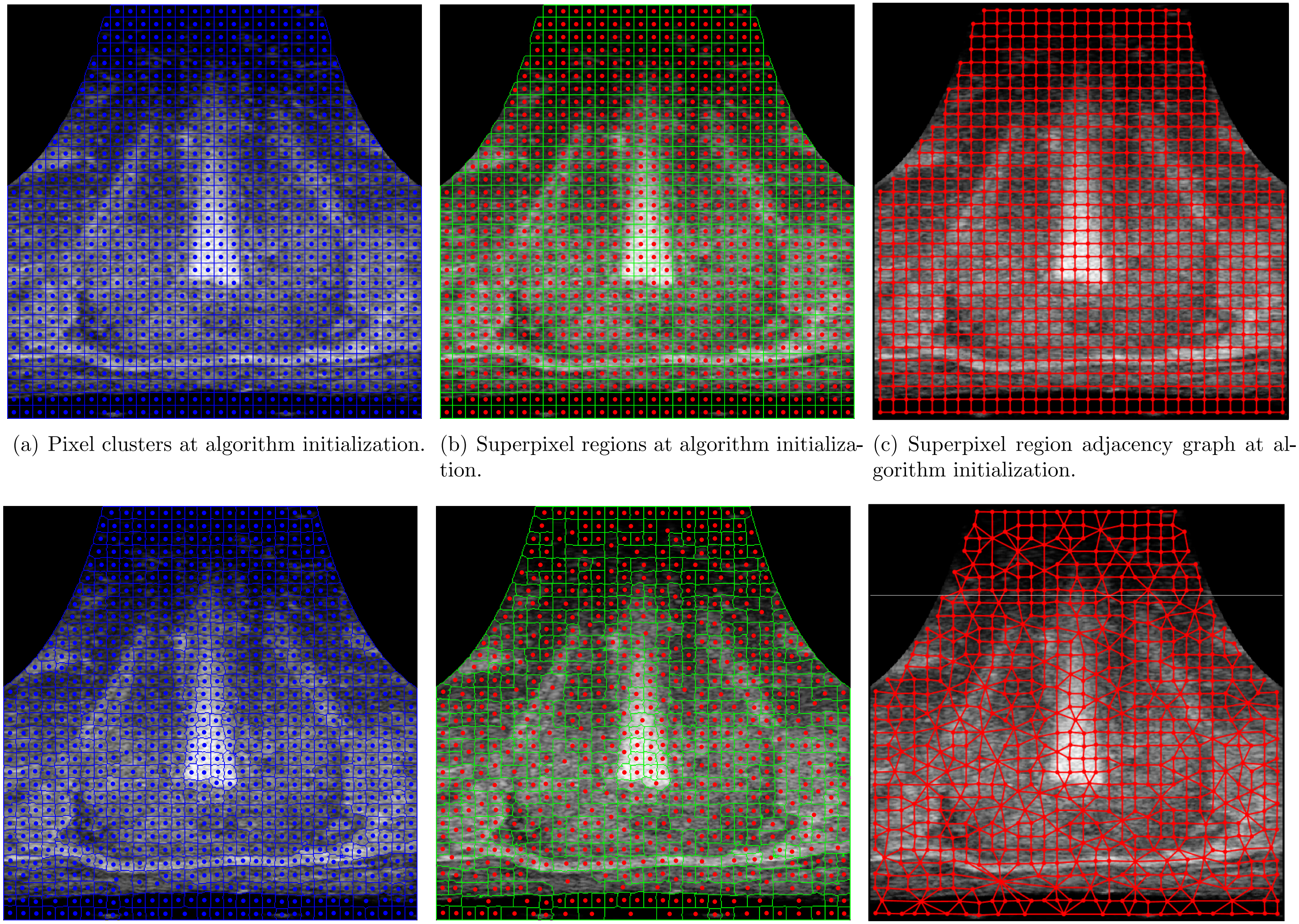
After all the pixel clusters smaller than c_{min} have been merged, a second loop is used which calculates the number of cluster components n_{ck} within a region R_k, where $n_{ck} = \text{Count}(j_k^\star)$, the number of indexes in $j_k^\star$. If n_{ck} is greater than c_{max} then the $n_{rc} - c_{max}$ smallest clusters will be merged with their nearest neighboring cluster. After evaluating the loop for all of the superpixel regions in the image, the last step is to update the parameters of each of the pixel clusters C_j in the cluster list $\mathbf{C}$ using (7.27). The parameter update recalculates the position of the centroids and average pixel intensity of the cluster for use in the distance-image based cluster optimization in the next iteration of the algorithm.

7.3.5 Algorithm Iteration and Output

With all of the cluster merging having been performed and the parameters of the pixel clusters updated, the current iteration step of the superpixel algorithm is completed. The next and subsequent algorithm iterations will start again at the cluster update step, followed by the region update step (consisting of the region union and cluster merge operations). The algorithm will continue iterating until a stopping condition occurs, thus terminating the algorithm. There are three cases under which the algorithm will be stopped, with the first case occurring when the statistical dissimilarity between all regions is greater than the threshold λ_{KS}, as no further region union operations can be performed. The second case arises when all of the superpixel regions within the image are larger than the desired superpixel size r_{max}, and the third case happens when the current number of superpixel regions (in $\mathbf{R}$) is less than or equal to r_{ndes}. The algorithm will be stopped whenever any of the cases occur and the condition can be given as a boolean value

$$E_{stop} = \begin{cases} \text{true} & \text{if } \min_{\mathcal{E}} \left(D_{KS}(k_{v1}, k_{v2}) \right) > \lambda_{KS} \; \forall \; E_v \\ \text{true} & \text{if } n_{kpx} > r_{max} \; \forall \; R_k \\ \text{true} & \text{if } \mathrm{Count}(R_k \in \mathbf{R}) \leq r_{ndes} \\ \text{false} & \text{otherwise} \end{cases} \tag{7.48}$$

with the first case being evaluated for all of the edges E_v in the graph, and the second and third cases being evaluated for all of the regions in the image. When the algorithm terminates, the resulting output is a label image $\mathcal{L}_R(i)$ that indicates the corresponding superpixel region for each pixel in the input image and the superpixel adjacency graph. The region image is created by initializing a new image of the same size as the input image, initializing all pixel values to zero, and then updating the label (value) of all of the pixels



(a) Pixel clusters at algorithm initialization.

(b) Superpixel regions at algorithm initialization.

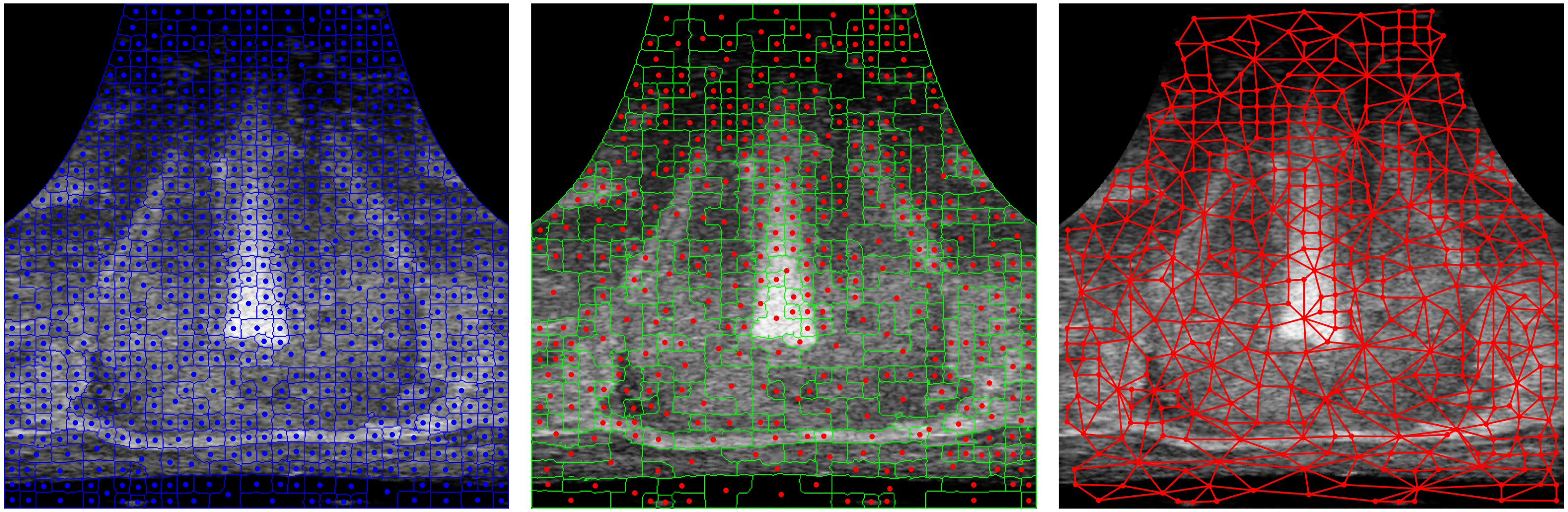
(c) Superpixel region adjacency graph at algorithm initialization.

(d) Pixel clusters after 5 update iterations.

(e) Superpixel regions after 5 update iterations.

(f) Superpixel region adjacency graph after 5 update iterations.

Figure 7.11: Pixel clusters, superpixel regions, and adjacency graph at initialization of superpixel algorithm and after 5 cluster update and region update iterations.



(a) Pixel clusters at termination of algorithm. (b) Superpixel regions at termination of algorithm. (c) Superpixel region adjacency graph at termination of algorithm.

Figure 7.12: Result of superpixel algorithm after the final update iteration, where 11 cluster update and region update steps have been performed on TRUS image.

based on which superpixel (region) they are contained within, with the labeling given by

$$\mathcal{L}_R(i) = k,\ \forall\,\{i \in C_j, \text{for each } C_j \in R_k\}, \tag{7.49}$$

such that all of the pixels i in the component clusters C_j of the superpixel R_k are labelled with an index k.

The results of running the superpixel algorithm on an input TRUS image, showing the pixel cluster, superpixel regions, and adjacency graph during multiple algorithm iterations are given in Fig. 7.11 and Fig. 7.12. The boundaries of the pixel clusters are indicated with blue lines and the centroids of the pixel clusters are shown as blue dots in Figs. 7.11(a), 7.11(d), and 7.12(a). The superpixel region boundaries are designated with green lines and the region centroids are shown in Figs. 7.11(b), 7.11(e), and 7.12(b). Red dots are used to indicate the centroids of each of the regions (which are also the graph vertices) in Figs. 7.11(b), 7.11(c), 7.11(e), 7.11(f), 7.12(b), and 7.12(c). The region adjacency graph is drawn in Figs. 7.11(c), 7.11(f), and 7.12(c) where a red line between two region centroids represents a graph edge. The output of the superpixel algorithm would be the label image based on the remaining regions in Fig. 7.12(b) and its adjacency graph shown in Fig. 7.12(c).

7.4 Graph-Cut Segmentation and Results

The last step in the prostate segmentation algorithm, as shown in Fig. 7.2, combines the output of the superpixel processing with the statistical prostate shape model. The prostate shape model, described in Sec. 7.2.3, can be used to find the likelihood that a particular pixel will be on the edge of the prostate. The superpixel image processing algorithm returns an image which has been divided into a large number of superpixel regions, with each superpixel region being a large area of pixels which have particular pixel intensity distributions. The superpixel algorithm also returns an associated graph describing the adjacency between the superpixel regions. A graph-cut algorithm will be used to merge the contour position information with statistical properties of the superpixel regions in order to segment the prostate.

Given an input set of TRUS ultrasound images slices, ${}^{U}\mathbf{I}$, containing $nImg$ images, the first step in segmentation is to pre-process each image, as covered in Sec. 7.1.2, to obtain the ROI for segmentation. After pre-processing, each of the ROI images in the set, ${}^{R}\mathbf{I}$, will be input into the superpixel processing algorithm. Each image i in the set will then have a corresponding superpixel region image ${}^{R}\mathcal{L}_{Ri}$ and graphs describing the neighbours of each region $\mathcal{G}_i = \{\mathcal{V}_i, \mathcal{E}_i\}$. For final segmentation, the superpixel region image is warped back into the ultrasound image frame $\{U\}$ by inverting the transform given in (7.1). To do this, the location of each pixel inside of the ROI in ${}^{U}\mathcal{I}({}^{U}px,\, {}^{U}py)_i$ will be used to find the corresponding pixel in ${}^{R}\mathcal{I}({}^{R}px,\, {}^{R}py)_i$, and the region label for that pixel, at ${}^{R}\mathcal{L}_{Ri}({}^{R}px,\, {}^{R}py)$, will be transferred to a new region image ${}^{U}\mathcal{L}_{Ri}({}^{U}px,\, {}^{U}py)$ and any pixels outside of the ROI are given a label of 0. For a region R_k, the pixels inside of that region are defined to be $\{{}^{U}\mathcal{L}_{Ri}({}^{U}px,\, {}^{U}py) = k : \forall({}^{R}px,\, {}^{R}py) \in R_k\}$; the result of this warping procedure is shown in Fig. 7.13. Warping the region images back in the $\{U\}$ frame is required in order to use the prostate shape model, as it was derived based on the prostate contour segmented by clinicians in standard TRUS images. The probabilistic prostate shape model is generated offline before segmentation and so the prostate can be segmented in all of the input images.

For segmentation, an energy-based cost function [117], will be minimized to perform the final segmentation (contouring) of the prostate in the TRUS images. The segmentation and contouring will be performed by labelling pixel regions R_k as being inside of the prostate, $\mathbf{L}_{In}$, or outside of the prostate, $\mathbf{L}_{Out}$. For a single image i the cost function is given by

$$E(\mathbf{L})_i = \lambda_D \sum_{k \in \mathcal{V}_i} \psi_k(\mathbf{L}_k) + \lambda_S \sum_{k,m \in \mathcal{E}_i} \psi_{km}(\mathbf{L}_k, \mathbf{L}_m) \tag{7.50}$$

where $\mathbf{L}$ is a labelling for the images. The cost function contains a data term

$$\sum_{k \in \mathcal{V}_i} \psi_k(\mathbf{L}_k) \tag{7.51}$$

for each region R_k in the image corresponding to the vertices in $\mathcal{V}_i$ and a smoothness term

$$\sum_{k,m \in \mathcal{E}_i} \psi_{km}(\mathbf{L}_k, \mathbf{L}_m), \tag{7.52}$$

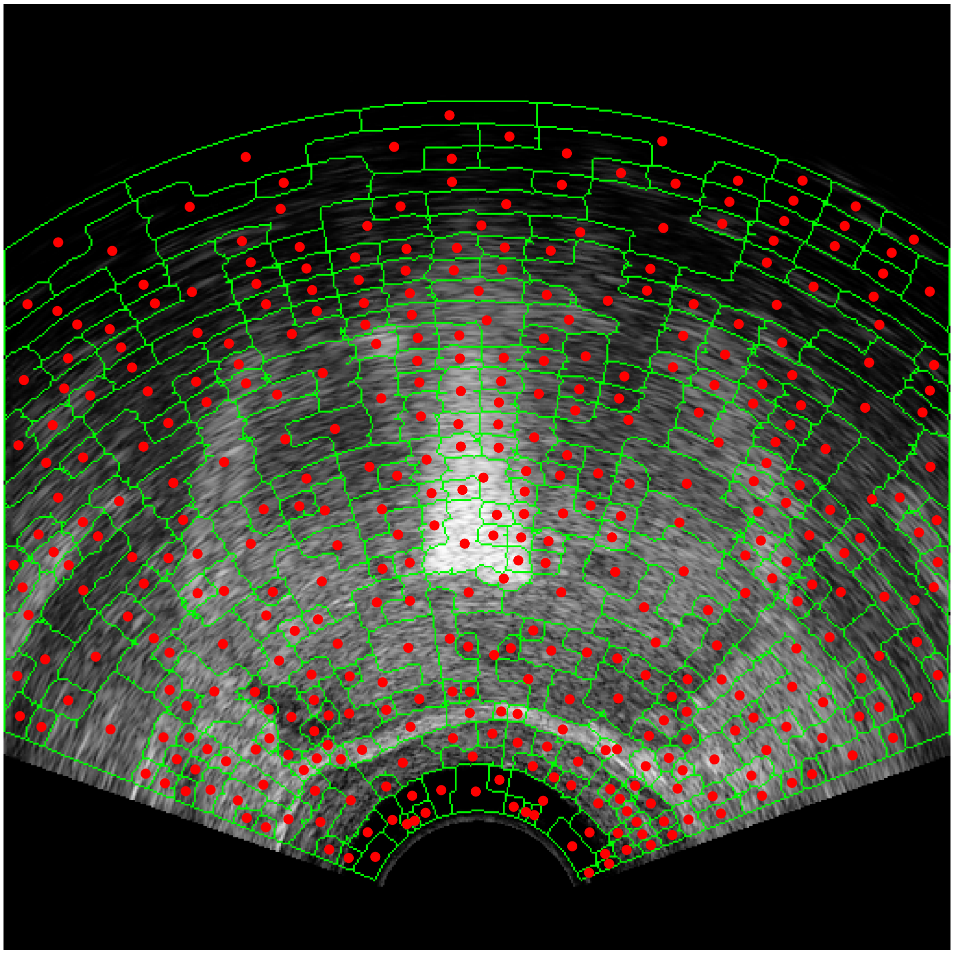


Figure 7.13: Output of superpixel algorithm in TRUS image frame $\{U\}$ with region borders shown in green and region centroids indicated wtih red dots.

which incorporates a score based on the neighbours connected by an edge $\mathcal{E}_i$, of the region R_k. The values λ_D and λ_S weight the data terms and smoothing terms, respectively, and this cost function will be minimized through the graph-cut procedure.

The data term will incorporate the contour location probabilities from the prostate shape model and a prostate appearance term into the graph-cut score. For this, the probability from the prostate shape model, given in (7.22), will be discretized to give a pixel-based probability score which will be stored in an image, ${}^{U}\mathcal{P}_i$, of the same size as the TRUS images (with a height ${}^{U}\mathcal{I}_{height}$ and width ${}^{U}\mathcal{I}_{width}$). For mapping between the ultrasound image set and prostate a center point, ${}^{P}c = [{}^{P}x_c, {}^{P}y_c, {}^{P}\phi_c]^{\top}$, will be used. Only a single contour point is used for the entire image set and, in a similar manner to the contour fitting procedure, this center point will be optimized during the graph-cut procedure. Using a given center point, the location of a pixel in the TRUS image, $({}^{U}px, {}^{U}py)$, is converted to the contour coordinate system through the transform

$$\begin{aligned}
{}^{P}px &= \frac{{}^{U}px - t_{px}}{s_{px}} \\
{}^{P}py &= \frac{{}^{U}py - t_{py}}{s_{py}} \\
\theta_{px} &= \text{atan2}({}^{P}py - {}^{P}y_c, {}^{P}px - {}^{P}x_c) - {}^{P}\phi_c \\
r_{px} &= \sqrt{({}^{P}py - {}^{P}y_c)^2 + ({}^{P}px - {}^{P}x_c)^2}
\end{aligned} \tag{7.53}$$

where ${}^{P}px$ and ${}^{P}py$ are the pixel coordinates in the patient attached frame, using the constants in the transform (7.1), giving the output values θ_{px} and r_{px} in the contour coordinate system. Using this transform, the probability image, ${}^{U}\mathcal{P}$, is created by evaluating the CDF of the prostate shape, given in (7.23), for every pixel within the ROI of in the input image, such that

$${}^{U}\mathcal{P}({}^{U}px, {}^{U}py)_i = \begin{cases} P(R \leq r_{px} | \theta_{px}, i) & \text{for } ({}^{U}px, {}^{U}py) \in {}^{U}\mathcal{I}_{ROI} \\ 1 & \text{for } ({}^{U}px, {}^{U}py) \notin {}^{U}\mathcal{I}_{ROI} \end{cases} \tag{7.54}$$

where $P(R \leq r_{px} | \theta_{px}, i)$ is the probability that the prostate contour is smaller than the pixel's radial distance r_{px}, relative to the center point, and the value of 1 will prevent the prostate contour from being outside the ROI.

With the probability image defined, the data term components $\psi_k(\mathbf{L}_k)$ for the cost function (7.50), are given as

$$\begin{aligned}\psi_k(R_k \in \mathbf{L}_{In}) = \lambda_{DP}\left(\sum_{({}^Upx,\,{}^Upyk)\in k} \frac{{}^U\mathcal{P}({}^Upx,\,{}^Upy)_i}{n_{kpx}}\right) + \\ \lambda_{SD}\left(\frac{1}{n_{px}}\sum_{({}^Upx,\,{}^Upyk)\in k} \frac{\left|{}^U\mathcal{L}({}^Upx,\,{}^Upy)_i - \mu_L\right|}{n_{kpx}}\right)\end{aligned} \tag{7.55}$$

and

$$\begin{aligned}\psi_k(R_k \in \mathbf{L}_{Out}) = \lambda_{P}\left(\sum_{({}^Upx,\,{}^Upyk)\in k} \frac{1-{}^U\mathcal{P}({}^Upx,\,{}^Upy)_i}{n_{kpx}}\right) + \\ \lambda_{L}\left(\frac{1}{n_{px}}\sum_{({}^Upx,\,{}^Upyk)\in k} \frac{255/\mu_L - \left|{}^U\mathcal{L}({}^Upx,\,{}^Upy)_i - \mu_L\right|}{n_{kpx}}\right)\end{aligned} \tag{7.56}$$

with n_{kpx} being the number of pixels in the region k, and λ_{DP} weighting contour probabilities. The μ_L value is an input constant corresponding to the average pixel intensity of pixels inside the prostate, and λ_{DL} weights this pixel intensity term. The two weighting components are represented in (7.50) as λ_D, such that $\lambda_D = [\lambda_{DP}, \lambda_{DL}]$. The smoothness terms within the cost function use the KS distance to weight the statistical distances between regions with the same and different labels, given by

$$\psi_{km}(\mathbf{L}_k, \mathbf{L}_m) = \begin{cases}\left(\sup_x \left|\hat{F}_k(x) - \hat{F}_m(x)\right|\right) & \text{where } \mathbf{L}_k = \mathbf{L}_m \\ \left(1 - \sup_x \left|\hat{F}_k(x) - \hat{F}_m(x)\right|\right) & \text{where } \mathbf{L}_k \neq \mathbf{L}_m\end{cases}, \tag{7.57}$$

with $\hat{F}_k(x)$ and $\hat{F}_m(x)$ being the empirical CDFs of the neighbouring regions R_k and R_m.

Each TRUS image slice i is then labelled to optimize the cost function using graph cutting [117]. The cost of the particular labelling L is calculated using (7.50) for all of the slices. To get the most accurate contour probability in the images ${}^U\mathcal{P}_i$ from the prostate shape model, the center point ${}^Pc = [{}^Px_c,\ {}^Py_c,\ {}^P\phi_c]^\top$ is optimized to remove any translational and rotational offset between the TRUS image and the prostate shape model. The center point is found through the optimization

$$\underset{P_c}{\operatorname{argmin}} \sum_{i=0}^{nImg} \underset{L}{\operatorname{argmin}}\, E(\mathbf{L})_i \tag{7.58}$$

where the $\operatorname{argmin}_L E(\mathbf{L})_i$ corresponds to the graph-cut procedure and the $\operatorname{argmin}_{P_c}$ optimization is done through a pattern search algorithm. Once the optimal center point has been found, with a minimal labelling cost across all of the slices, the prostate segmentation is complete, as shown in Fig. 7.14. All regions labelled with $\mathbf{L}_{In}$ after graph cutting are inside of the prostate, which could be represented in a binary image where all of the pixels inside of the regions $R_k \in \mathbf{L}_{In}$ have a value of one and all other pixels are zero. To smooth out the segmented prostate contour, the contour model is fit to the pixels on the perimeter of the binary region $R_k \in \mathbf{L}_{In}$. The perimeter pixels are transformed to the contour model coordinate system using (7.53), and then the model is fit to these pixel points from the least-squares fitting given in (7.9), thus creating a 3D contour model for the segmented prostate; see Fig. 7.15. The results of the segmentation on a number of patient images sets is given in Sec. 7.4.1.

7.4.1 Prostate Contour Results

The input dataset used to create the probabilistic prostate contour model was a series of contours exported from Variseed (Varian Medical Systems Inc, Palo Alto, CA, USA) after pre-planning for clinical prostate brachytherapy [7]. For this work the prostate segmentation algorithm was tested on the clinical ultrasound images collected from 9 patients during a prostate brachytherapy procedure. The ultrasound images were taken using a B&K Medical Pro Focus 2202 ultrasound machine with a B&K Medical Type 8808 Biplane Transducer (B&K Medical Systems, Peabody, MA, USA). The image processing, model fitting, and graph-cut algorithms were all programmed or implemented in Matlab 2018a (The Mathworks Inc, Natwick, MA, USA) and run using the

[7] Approval for study granted from Alberta Cancer Research Ethics Committee under reference 25837

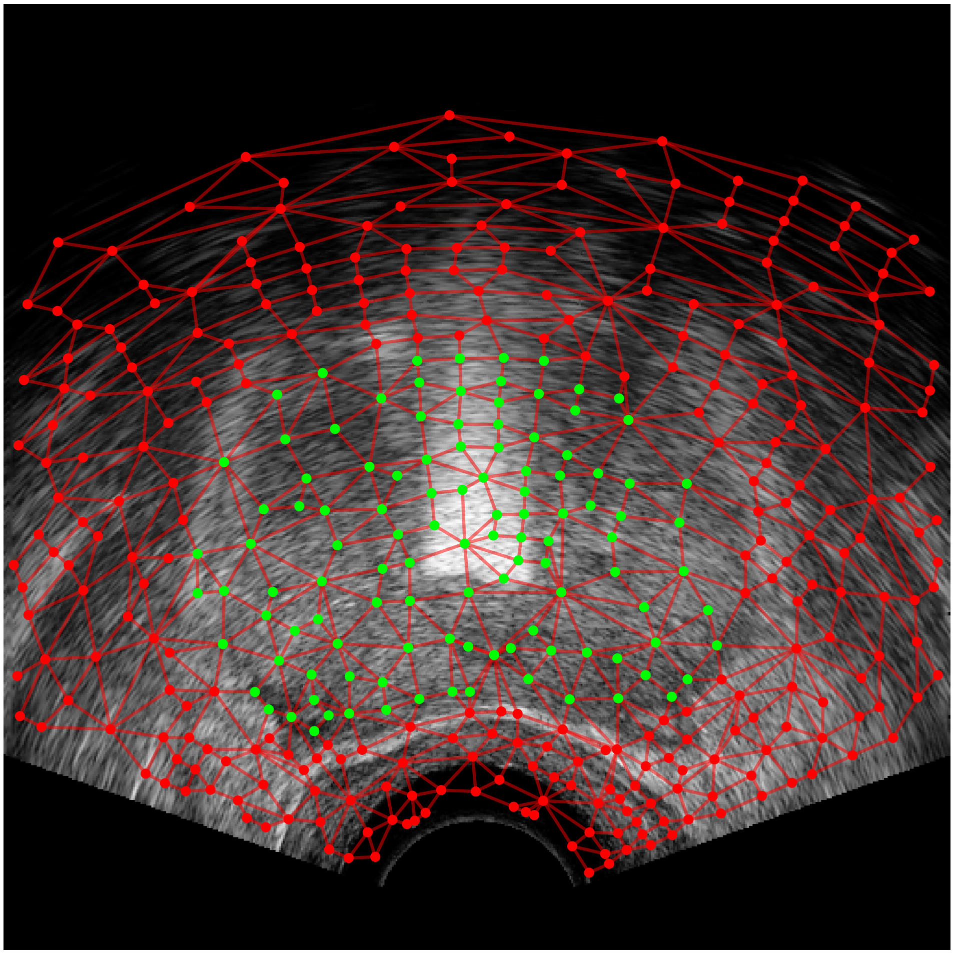
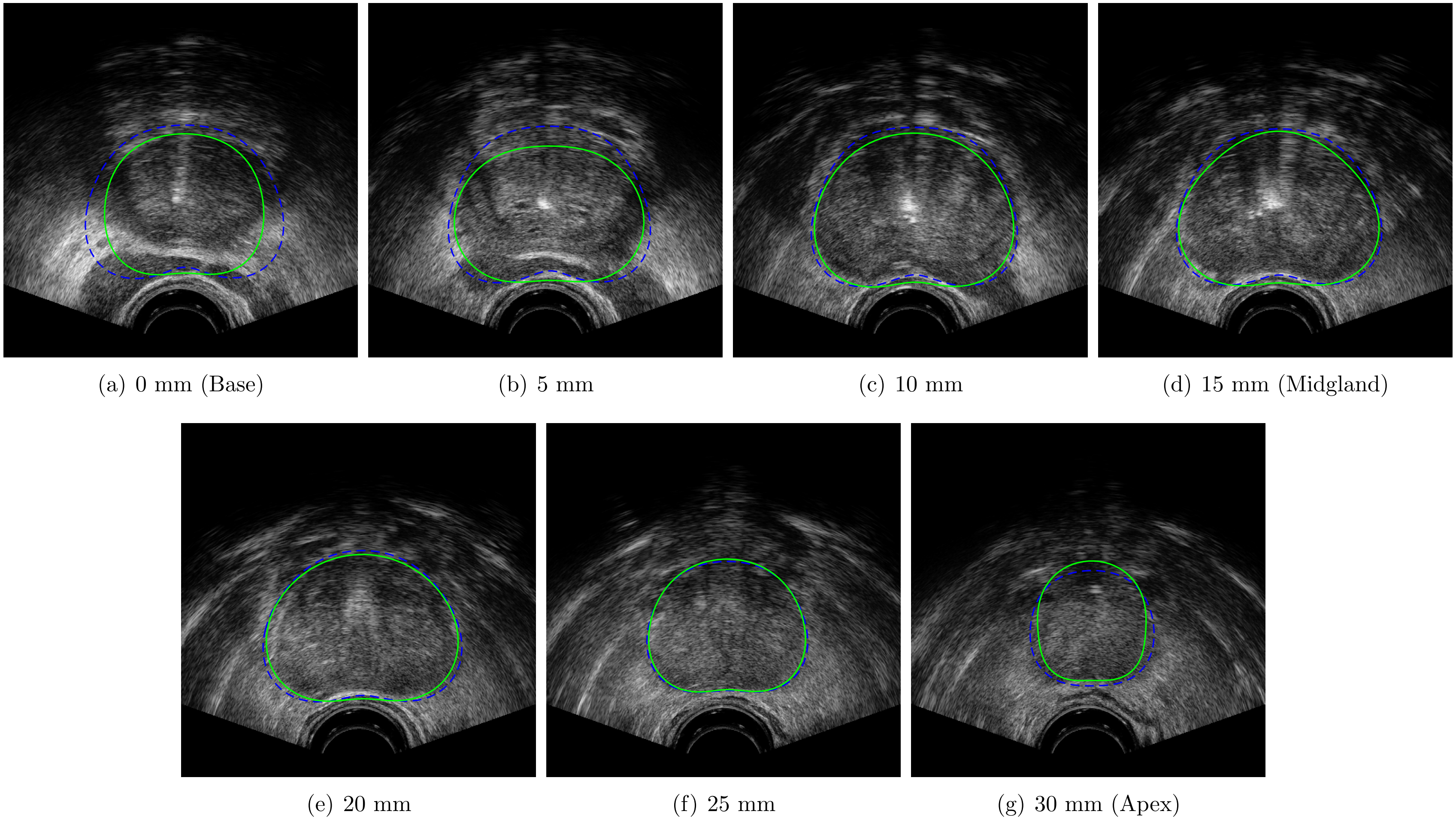


Figure 7.14: Superpixel regions after graph-cut procedure. The centroids of regions labelled as being inside of the prostate, $R_k \in \mathbf{L}_{In}$, are displayed with green dots and the centroids of the superpixel regions outside of the prostate, $R_k \in \mathbf{L}_{Out}$, are displayed with red dots.



(a) 0 mm (Base)

(b) 5 mm

(c) 10 mm

(d) 15 mm (Midgland)

(e) 20 mm

(f) 25 mm

(g) 30 mm (Apex)

Figure 7.15: Segmented prostate contours in TRUS image slice from our proposed algorithm, $R(\theta, i|\boldsymbol{\beta}_A)$, shown as solid green line and compared with input clinician-segmented contour, $R(\theta, i|\boldsymbol{\beta}_C)$, indicated with dashed blue line.

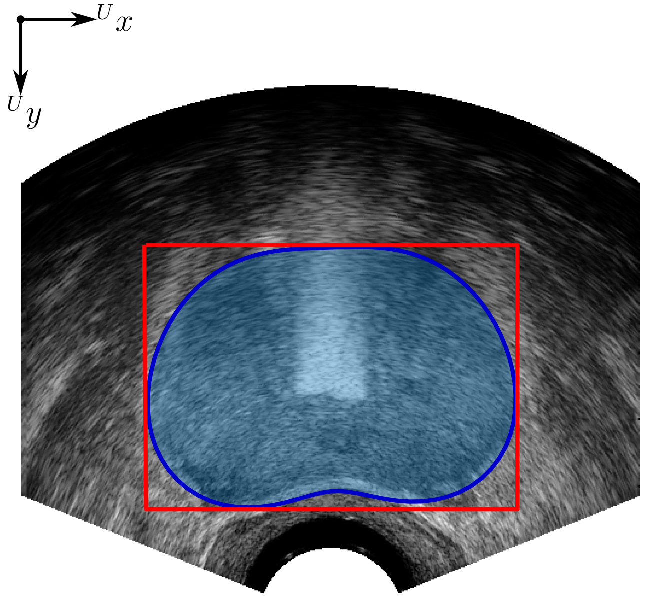


Figure 7.16: Bounding box, indicated in red, and shaded-blue polygon contour area used for prostate contour height,width, and volume measurements.

Simulink Real-Time environment, on an Intel Core i7-3930K running at 3.20 GHz (Intel Corporation, Santa Clara, CA, USA).

The the mean absolute difference (MAD), max absolute difference (MaxD), segmented prostate volume, Jaccard index, and prostate contour width-height will be used to compare the 3D contour volumes returned by algorithm, $R(\theta, i|\boldsymbol{\beta}_A)$, to the target clinician segmented 3D contours, $R(\theta, i|\boldsymbol{\beta}_C)$, to evaluate the effectiveness and accuracy of the algorithm. The mean absolute difference (MAD) and max absolute difference (MaxD) are commonly used performance metrics in the literature which provide a measure of how closely the component 2D algorithm and clinician segmented contours match across all slices in the 3D contour. The difference between corresponding points, with the same i and θ values, on the algorithm segmented contour and clinician segmented contours are given by

$$D_{CA}(i,\theta) = \sqrt{\left({}^{P}x_C(i,\theta) - {}^{P}x_A(i,\theta)\right)^2 + \left({}^{P}y_C(i,\theta) - {}^{P}y_A(i,\theta)\right)^2} \qquad (7.59)$$

where $\{{}^{P}x_C(i,\theta),\, {}^{P}y_C(i,\theta)\}$ and $\{{}^{P}x_A(i,\theta)\, {}^{P}y_A(i,\theta)\}$ are the corresponding points between the algorithm and clinician contours in Cartesian coordinates, respectively; where the Cartesian coordinates are found from (7.6). A set of θ_j points sampled at regular intervals in the range $(-\pi, \pi]$, where $theta_j = -\pi + 2\pi\frac{j}{n_\theta - 1}$, will be used for calculating the MAD and MaxD values; the value n_θ is the number of points to be compared[8]. The mean absolute difference and max absolute difference metrics are given by

$$\text{MAD} = \frac{1}{nSlice \times n_\theta} \sum_i \sum_j |D(i, \theta_j)| \, \text{MaxD} = \max |D(i, \theta_j)| \; \forall \{i, j\} \quad (7.60)$$

where $i = \{0, 1, ..., nSlice\}$ and $j = \{0, 1, ..., n_\theta\}$. The height and width of the prostate contour, measured in mm in the ultrasound image frame ${}^{U}y$ and ${}^{U}x$ axes, is found from the height and width of a bounding-box containing the prostate contour; see Fig. 7.16. The average width and height of the algorithmic and clinician segmented prostate contours, with respect to image depth, are presented in Fig 7.18(b). For measuring the area and volume of the prostate a polygon is fit to points on the prostate contour model $R(\theta, i|\boldsymbol{\beta}_A)$ or clinician-segmented contour $R(\theta, i|\boldsymbol{\beta}_C)$ for each slice i, as shown in Fig. 7.16, with the resulting polygon areas, $Area_A(i)$ and $Area_C(i)$ for the algorithmic and clinician contours respectively, calculated in metric units (mm ^{2}). To calculate the volume of the prostate for the algorithm Vol_A and clinician contours Vol_C the area of each slice was multiplied by the 5 mm TRUS image slice stepping distance and summed for all of the slices in the image set, such that

$$\begin{aligned} Vol_A &= \sum_{i=0}^{nSlice} \frac{5\text{mm} * Area_A(i)}{1000} \\ Vol_C &= \sum_{i=0}^{nSlice} \frac{5\text{mm} * Area_C(i)}{1000} \end{aligned} \quad (7.61)$$

with the factor of 1000 used to scale the output volumes to mL. The calculated prostate volumes, for each of the 9 prostates incorporated in the experimental results, are shown in Fig. 7.18(a). For the results presented in Table 7.2, the volume difference in mL and % were calculated by

$$\begin{aligned} \text{VolumeDifference(mL)} &= Vol_C - Vol_A \\ \text{VolumeDifference(\%)} &= \frac{|Vol_A - Vol_C|}{Vol_C} \end{aligned} \quad (7.62)$$

[8] $n_\theta = 90$ for the presented results.

where a positive value for the prostate contour volume difference indicates that the algorithm segmented prostate contour was larger than the clinician-segmented contour and a negative difference indicates that the algorithmic segmented contour was smaller.

As an additional metric to evaluate the performance of the prostate contouring algorithm the Jaccard index was also analyzed. The Jaccard index gives a measure between 0 and 1 that indicates how well the algorithmic output of the prostate contour matches with the clinician-segmented contour, with a value of 1 indicating a perfect match. The Jaccard index is given by

$$JI = \frac{|R(\theta, i|\boldsymbol{\beta}_A) \cap R(\theta, i|\boldsymbol{\beta}_C)|}{|R(\theta, i|\boldsymbol{\beta}_A) \cup R(\theta, i|\boldsymbol{\beta}_C)|} \tag{7.63}$$

where $R(\theta, i|\boldsymbol{\beta}_A)$ is the output contour of our proposed prostate segmentation algorithm, for a particular TRUS image set, and $R(\theta, i|\boldsymbol{\beta}_C)$ is the clinician-segmented contour for the same image set. Figure 7.17 shows the average Jaccard index score values with respect to the z-axis depth of the TRUS image where the first image slice at a depth of 0 mm corresponds to the base of the prostate, the midgland of the prostate is located at 15 mm, and the apex of the prostate is at a depth of 30 mm; as indicated in Fig. 7.15.

Table 7.2 and Table 7.3 show the resulting values of the metrics comparing the output contours from our proposed prostate segmentation algorithm[9] for 9 patient TRUS image sets, where each image set consisted of 7 images slices. Table 7.2 presents the mean absolute difference, max absolute difference, and prostate volume difference and Table 7.3 present the contour difference, Jaccard index, and prostate volume.

The average mean absolute difference for all of the trials was 2.52 (±1.66) mm and the average maximum absolute difference was 7.19 (±1.22) mm. While the mean absolute difference is slightly larger than comparable results presented in the literature the maximum absolute difference is in line those reported in the literature (see Table 2.1 for a summary of related literature).

[9] The input parameters used for the superpixel algorithm for the results given in Table 7.2, were $\lambda_\mu = 100$, $\lambda_{L2} = 125$, $\lambda_{KS} = 0.2$, $s_{grid} = 256$px, $r_{darea} = 512$px, $r_{min} = 256$px, , $c_{min} = 192$px, $c_{max} = 4$, and the parameters for the graph-cut segmentation were $\lambda_{DP} = 3.1, \lambda_{DL} = 1.8, \mu_l = 40, \lambda_S = 1.5$.

Table 7.2: Prostate Segmentation Contour Results

Data Set	Mean Absolute Difference (mm)	Max Absolute Difference (mm)	Volume Difference (mL)	Volume Difference (%)
Prostate 1	2.01 (±1.30)	5.89	4.65	8.85
Prostate 2	1.87 (±1.29)	6.91	2.27	4.37
Prostate 3	3.03 (±2.21)	8.50	6.75	13.77
Prostate 4	2.54 (±1.48)	6.00	5.77	11.88
Prostate 5	2.46 (±1.60)	6.82	2.80	5.76
Prostate 6	2.24 (±1.43)	6.70	6.20	12.68
Prostate 7	3.08 (±2.06)	8.82	7.86	17.56
Prostate 8	2.73 (±1.71)	8.90	-3.39	6.07
Prostate 9	2.76 (±1.65)	6.20	7.47	17.11
Average	2.52 (±1.66)	7.19 (±1.22)	4.49 (±3.53)	10.89 (±4.90)

Table 7.3: Prostate Segmentation Jaccard Index Results

Data Set	Average Jaccard Index	Min Jaccard Index	Max Jaccard Index
Prostate 1	0.84 (±0.04)	0.79	0.92
Prostate 2	0.85 (±0.06)	0.75	0.91
Prostate 3	0.77 (±0.02)	0.73	0.79
Prostate 4	0.79 (±0.06)	0.73	0.87
Prostate 5	0.79 (±0.04)	0.72	0.83
Prostate 6	0.82 (±0.06)	0.71	0.90
Prostate 7	0.74 (±0.07)	0.66	0.85
Prostate 8	0.77 (±0.07)	0.66	0.84
Prostate 9	0.77 (±0.09)	0.61	0.89
Average	0.79 (±0.07)	0.71 (±0.05)	0.87 (±0.04)

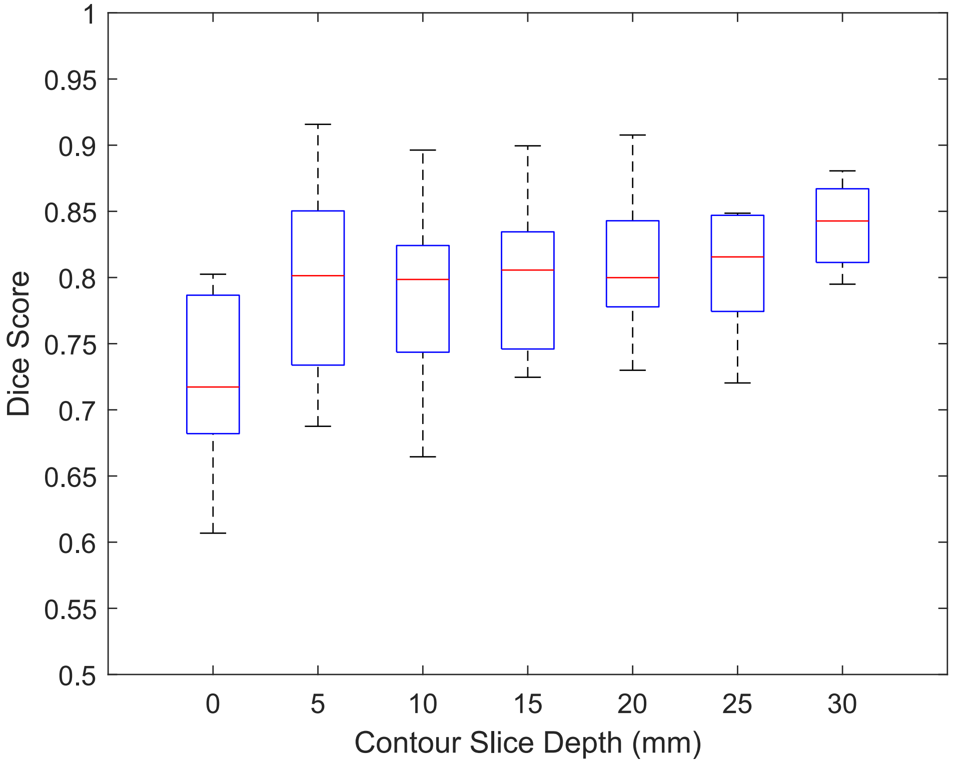
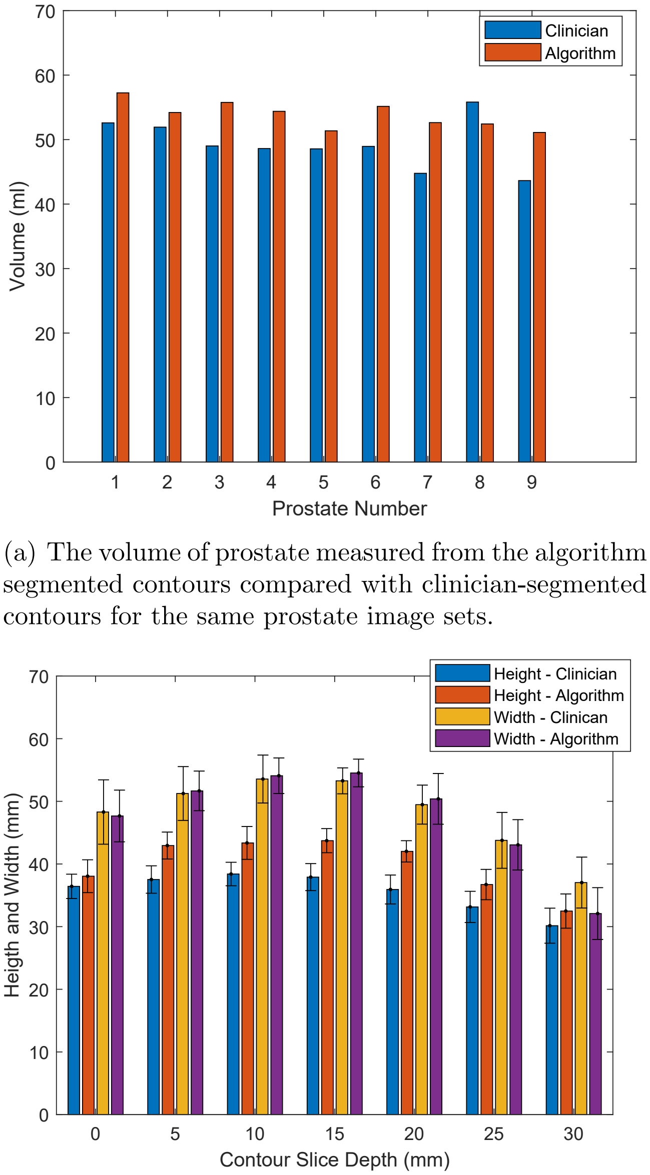


Figure 7.17: Jaccard index score of prostate contouring algorithm with respect to contour depth in the z-axis.



(a) The volume of prostate measured from the algorithm segmented contours compared with clinician-segmented contours for the same prostate image sets.

(b) Average width and height of the algorithmic and clinician segmented prostate contours, measured in the ${}^{U}x$ and ${}^{U}y$ axes respectively.

Figure 7.18: Union on yellow superpixel regions, showing regions and connections before and after operation.

A major factor which may account for this discrepancy in mean absolute difference performance between our approach and those in the literature is that many methods require a clinician to manually correct the algorithmic prostate contours after segmentation [68, 69, 70], and report only the post-corrected segmentation values. In many cases in the literature this manual clinician correction is required in up to 30% of the input set TRUS images for segmentation.

In the literature, it is well known that there is a high amount of variation between clinicians when manually segmenting the prostate in TRUS images[15, 16]. Therefore the performance of the algorithm can also be compared to clinician segmentation of TRUS images, where [15] and [16] analyzed the interobserver clinician segmentation accuracy between observers (clinicians) and intraobserver repeatability from a single observer (a clinician segmenting the same TRUS image set multiple times). Interestingly, the difference between the prostate volume measured from the algorithm and clinician-segmented contours is directly comparable to the interobserver variance reported in [15, 16] indicating that the segmented contours from the proposed algorithm are within the accuracy range accepted by clinicians in practice.

The average Jaccard index values across all images slices are slightly lower than those reported for clinicians [16] indicating some additional work to increase contouring accuracy may be required. The per-slice Jaccard index values presented in Fig. 7.17 indicate that the contouring has roughly the same contouring performance across all prostate image slices with the exceptions of the base image slice (0 mm) which has slightly lower than average accuracy and the apex image slice (30 mm) which has higher than average accuracy. These per slice results compare favorably with the variances reported in [16] with the base slice segmentation variance being higher for both the algorithm and between clinicians. The accuracy of the prostate midgland contouring is the primary area where the algorithm performs slightly worse than clinicians [16] and indicates the primary area where the algorithm accuracy can be improved.

7.5 Conclusion

In this chapter, we have outlined a method for prostate segmentation which incorporates a prostate contour model, active shape probabilistic model of the prostate, a superpixel image processing algorithm, and graph-cut-based contour optimization. The proposed contour model was validated independently of the prostate segmentation algorithm by comparing the mean-squared error and maximum absolute error between input, clinician-segmented, prostate contours and the resulting contour models after parameter fitting. Two sets of 100 3D clinician-segmented prostate contours were used for validation, with one set consisting of 3D contours containing 7 to 13 2D TRUS image derived contours of the prostate and the second set having 3D contours with each containing 7 to 16 2D contours segmented using MR images of the prostate. The results of this validation, given in Table 7.1, show that the clinician-segmented contours, from either imaging modality, can be represented in the contour model with sub-millimeter accuracy on average. The average mean-squared error across the two input contour sets was 0.61($\pm$ 0.16) mm and the average maximum absolute error was 1.54 ($\pm$ 0.31) mm. The largest absolute error observed in parameter fitting of all 200 contours was 2.12 mm. With these results, the prostate contour model was shown to be unaffected by imaging modality and capable of modeling a variety of prostate shapes with an acceptable accuracy.

The prostate contour model was then incorporated, using interpolation between Gaussian normal distribution models, into an active shape statistical contour model of the prostate. This statistical shape model describes the probability of the edge of the prostate being at a particular location in a TRUS image slice. The CDF for the statistical shape model was incorporated as a parameter within the graph-cutting cost function to utilize prior prostate contour information for algorithmic prostate segmentation. The graph to be segmented was output by a novel superpixel image processing algorithm which uses a hierarchical structure to simultaneously optimize pixel clusters, with respect to a pixel intensity and Euclidean-based distance metrics, and superpixel regions, which are comprised of a number of pixel clusters. These superpixel regions separate an input image into areas with similar image statistics, in a type of edge and texture discrimination, with the statistical similarity between two superpixel regions being another parameter within the graph-cutting cost function.

The superpixel image processing algorithm created and updated a graph structure, with the superpixel regions corresponding to the vertices of the graph and neighboring superpixel regions being interconnected by a graph edge. Graph-cuts are performed on the resulting superpixel adjacency graph, after image processing, in order to segment out superpixel regions which are inside of the prostate in the TRUS image. This prostate segmentation algorithm was tested on 9 prostate TRUS image sets, consisting of 6 or 7 prostate image slices, to assess the algorithmic segmentation accuracy. The output of the prostate segmentation algorithm was compared to prostate segmentation done by a clinician, in those TRUS image sets, with an average mean absolute difference of 2.52 (± 1.66) mm and an average maximum absolute difference was 7.19 (± 1.22) mm. The average Jaccard index of the algorithm was 0.79 (± 0.07). The average Jaccard index result is satisfactory for the limited number of TRUS image sets that were tested, however, the continuation of this work will require a significantly larger number of the TRUS prostate image sets to fine-tune the optimization parameters used during superpixel image processing and segmentation. Additional research within the area of algorithmic prostate segmentation will be discussed as part of Chapter 8.

Chapter 8

Conclusion

This thesis has presented research in the area of surgeon-in-the-loop robotic assistance for percutaneous procedures, with a primary focus on image processing algorithms and needle steering controllers for the prostate brachytherapy procedure. There are many aspects of prostate brachytherapy, in current clinical practice, which can utilize robotic assistance or computer guidance to aid a clinician in increasing treatment accuracy and efficacy while reducing the time required for surgical planning and seed implantation. The work presented in this thesis has aimed to utilize equipment currently used by clinicians during a prostate brachytherapy procedure, for instance, ultrasound image feedback, when developing these surgical assistant systems. This work has developed needle segmentation and tracking algorithms which have been incorporated into two surgeon-in-the-loop needle steering controllers, which assist a clinician during seed implantation by increasing needle placement accuracy. To assist a clinician during the planning phase of a prostate brachytherapy procedure, a method for fully-autonomous prostate segmentation has also been presented.

The first needle segmentation routine presented in this work utilized needle deflection information which can be measured in sagittal plane ultrasound images. While only a portion of the needle can be observed in a single sagittal image, the algorithm presented in Chapter 3, used this partial observation of needle deflection to implement a needle shape predictor, which estimates the shape of the entire needle when it is fully inserted into tissue. This needle shape prediction is based around a mechanical spring-beam needle model. The needle shape predicted by the model was compared to a reference camera image of the same needle to validate the proposed approach. The results show that the entire needle shape can be accurately predicted in tissues of varying stiffness based on observation of parts of the needle in an ultrasound image.

In Chapter 4 a needle-tip tracking routine for axial TRUS images was developed. Unlike needle shape observation in the sagittal plane, estimation of needle model parameters requires deflection information from multiple axial image slices and necessitates the use of a computationally efficient model which allows for the parameters of the model to be updated in real-time as each axial image frame is captured. In this chapter, particle filtering was used to estimate the parameters of a kinematic bicycle model based on needle-tip tracking between subsequent axial image frames. The tracking method and needle-tip deflection prediction were evaluated in both ex-vivo beef phantom tissue and in-vivo clinical images, displaying an average tip prediction error of less than 0.5 mm in both image sets.

Using the axial image needle-tip tracking algorithm, a surgeon-in-the-loop controller was devised in Chapter 5 based on a switching-based continuous control law. The controller was formulated based on a reduced-order 3D non-holonomic model of needle tip motion and was proven to reduce needle deflection to zero asymptotically. This controller was implemented with a hand-held steering device in a surgeon-in-the-loop system and tested on multiple ex-vivo tissue phantoms. The ex-vivo results show an average absolute needle tip deflection of 0.54 mm away from the target location at a depth of 120 mm, with an average needle tip deflection of 0.36 mm throughout the entire insertion length. This controller incorporated real-time measurement of needle-tip deflection but did not utilize needle tip path prediction and therefore required that acceptable but relatively large changes in the needle-tip orientation be allowed in order to achieve this level of deflection control.

An improved surgeon-in-the-loop system, implemented with an event-trigger controller was presented in Chapter 6 which incorporated needle path prediction based on needle-tip tracking in axial ultrasound images. An analytic solution to the kinematic bicycle model was derived and is shown to be more computationally efficient than discrete-step simulations of the model. Using this analytic solution, the event-trigger points of the controller, corresponding to changes in the control output, were optimized. The use of the analytic model and the event-triggered controller allowed for limiting the number and extent of needle rotations, if required to reduce tissue trauma or for practical reasons, in a constrained optimization framework. The system was tested in three different ex-vivo tissue phantoms and was able to keep the average needle deflection within 0.47 (± 0.21) mm at the final insertion depth of 120 mm.

This work also presented a fully-autonomous prostate contouring algorithm using a novel superpixel based approach. The contouring algorithm used superpixel image processing to find statistically similar areas within an input TRUS image and used a statistical shape model along with graph-cuts to segment the prostate. The algorithm was evaluated on 10 sets of prostate images and compared with clinician-segmented contours for those same prostate sets, with the output of the contouring algorithm having an average mean absolute difference of 2.52 (± 1.66) mm, an average maximum absolute difference of 7.19 (± 1.22) mm, and an average Jaccard index accuracy of 0.79 (± 0.07). The Jaccard index, prostate volume, and height-width differences between the proposed algorithm and clinician results were also analyzed and show that the prostate contour algorithm segmented the prostate within the variance seen when different clinicians segment the same TRUS image set.

8.1 Future Work

The work presented in this thesis has focused on needle deflection minimization and has analyzed the accuracy with which the needle tip can be placed with respect to a target position. This needle placement accuracy is one metric through which needle steering accuracy can be evaluated but for the clinical brachytherapy procedure, there are some additional factors that should be analyzed in future work. One such aspect is seed-placement accuracy after the needle has been withdrawn, to see if there are significant deviations between the needle path observed during insertion and the final placed seed locations due to tissue compression/relaxation or other forces. Another factor to consider is prostate motion caused by needle insertion and ultrasound probe movement. The desired seed target location, within the prostate, should be tracked and used for control feedback. The following two research areas can be used to address and analyze these factors in future research:

- Prostate Tracking in TRUS Images
  - Our proposed prostate segmentation algorithm and other segmentation systems presented in the literature can be used to measure the location of the prostate, and associated needle insertion targets, before needle insertion.
  - The prostate moves during needle insertion and if prostate segmentation can be done in real-time, or if a real-time tracking algorithm can be developed which deforms the segmented prostate contours based on real-time axial image feedback, then prostate motion could be incorporated as a control input.
  - Prostate motion tracking could be used as part of a surgeon-in-the-loop system, where the 2D motion of the prostate and associated target locations could be used to compensate for prostate motion when steering the needle towards the target locations.
  - The control algorithms presented in this thesis assume that the desired target location is along the axis of needle insertion, so additional work would need to be done to evaluate the stability and needle steering performance when compensating for prostate motion.
- Prostate Motion Prediction and Modeling
  - Beyond tracking the 2D in-plane motion of the prostate, there is a need for a 3D mechanical model of prostate motion and interaction with tissue, where such a model would incorporate 3D torques and

forces to predict the linear and rotational movement of the prostate during needle insertion.

- Several mechanical needle-tissue interaction models have been developed in the literature, but modeling the motion of the prostate as a needle is inserted remains an open problem.
- Modelling the motion of the prostate during insertion is useful for predictive needle steering control.
- The relaxation motion of the prostate after needle the needle is withdrawn from the patient will allow for prediction of post-needle-retraction movement of the implanted seeds, which would allow for better needle steering algorithm development to improve final implant seed locations.
- One of the limiting factors in developing a prostate motion model is that a live 3D, intraoperative, volume of the prostate is needed in order to accurately measure prostate motion for model development and fitting.
- Several US companies now produce 2D matrix array transducers which can capture entire 3D US volumes in real-time, and recently, a few ultrasound manufacturers have made small endoscopic matrix array probes commercially available.

References

[1] J. Carriere, C. Rossa, N. Usmani, R. Sloboda, and M. Tavakoli, “Needle shape estimation in soft tissue based on partial ultrasound image observation,” in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, May 2015, pp. 2277–2282.

[2] J. Carriere, C. Rossa, R. Sloboda, N. Usmani, and M. Tavakoli, “Real-time needle shape prediction in soft-tissue based on image segmentation and particle filtering,” in *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, July 2016, pp. 1204–1209.

[3] J. Carriere, M. Khadem, C. Rossa, N. Usmani, R. Sloboda, and M. Tavakoli, “Surgeon-in-the-Loop 3-D Needle Steering Through Ultrasound-Guided Feedback Control,” *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 469–476, Jan 2018.

[4] ——, “Event-Triggered 3D Needle Control Using a Reduced-Order Computationally Efficient Bicycle Model in a Constrained Optimization Framework,” *Journal of Medical Robotics Research*, vol. 0, no. 0, p. 1842004, 2018.

[5] M. Khadem, C. Rossa, N. Usmani, R. S. Sloboda, and M. Tavakoli, “Feedback-linearization-based 3D needle steering in a frenet-serret frame using a reduced order bicycle model,” in *2017 American Control Conference (ACC)*, May 2017, pp. 1438–1443.

[6] C. Rossa, N. Usmani, R. Sloboda, and M. Tavakoli, “A hand-held assistant for semiautomated percutaneous needle steering,” *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 3, pp. 637–648, March 2017.

[7] C. Rossa and M. Tavakoli, “Issues in closed-loop needle steering,” *Control Engineering Practice*, vol. 62, pp. 55 – 69, 2017.

[8] L. Potters, C. Morgenstern, E. Calugaru, P. Fearn, A. Jassal, J. Presser, and E. Mullen, “12-Year outcomes following permanent prostate brachytherapy in patients with clinically localized prostate cancer,” *The Journal of Urology*, vol. 173, pp. 1562 – 1566, 2005.

[9] D. Bowes and J. Crook, “A critical analysis of the long-term impact of brachytherapy for prostate cancer: a review of the recent literature.” *Current Opinion in Urology*, vol. 21, no. 3, pp. 219 – 224, 2011.

[10] 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.

[11] J. E. Dawson, T. Wu, T. Roy, J. Y. Gu, and H. Kim, “Dose effects of seeds placement deviations from pre-planned positions in ultrasound guided prostate implants.” *Radiotherapy And Oncology: Journal Of The European Society For Therapeutic Radiology And Oncology*, vol. 32, no. 3, pp. 268 – 270, 1994.

[12] J. E. Sylvester, P. D. Grimm, S. M. Eulau, R. K. Takamiya, and D. Naidoo, “Permanent prostate brachytherapy preplanned technique: The modern seattle method step-by-step and dosimetric outcomes,” *Brachytherapy*, vol. 8, no. 2, pp. 197 – 206, 2009.

[13] M. F. Jamaluddin, S. Ghosh, M. P. Waine, R. S. Sloboda, M. Tavakoli, J. Amanie, A. D. Murtha, D. Yee, and N. Usmani, “Quantifying ^{125}I placement accuracy in prostate brachytherapy using postimplant transrectal ultrasound images,” *Brachytherapy*, vol. 16, no. 2, pp. 306 – 312, 2017.

[14] Canadian Cancer Society’s Steering Committee, “Canadian Cancer Statistics,” 2015, available: http://www.cancer.ca/statistics.

[15] W. L. Smith, C. Lewis, G. Bauman, G. Rodrigues, D. D’Souza, R. Ash, D. Ho, V. Venkatesan, D. Downey, and A. Fenster, “Prostate volume contouring: A 3d analysis of segmentation using 3dtrus, ct, and mr,” *International Journal of Radiation Oncology*Biology*Physics*, vol. 67, no. 4, pp. 1238 – 1247, 2007.

[16] D. Liu, N. Usmani, S. Ghosh, W. Kamal, J. Pedersen, N. Pervez, D. Yee, B. Danielson, A. Murtha, J. Amanie, and R. S. Sloboda, “Comparison of prostate volume, shape, and contouring variability determined from preimplant magnetic resonance and transrectal ultrasound images,” *Brachytherapy*, vol. 11, no. 4, pp. 284 – 291, 2012.

[17] M. Ding and A. Fenster, “A real-time biopsy needle segmentation technique using Hough transform.” *Medical Physics*, vol. 30, no. 8, pp. 2222 – 2233, 2003.

[18] K. Draper, C. Blake, L. Gowman, D. Downey, and A. Fenster, “An algorithm for automatic needle localization in ultrasound-guided breast biopsies.” *Medical Physics*, vol. 27, no. 8, pp. 1971 – 1979, 2000.

[19] S. Okazawa, R. Ebrahimi, J. Chuang, R. Rohling, and S. Salcudean, “Methods for segmenting curved needles in ultrasound images.” *Medical Image Analysis*, vol. 10, no. 3, pp. 330 – 342, 2006.

[20] M. Kaya and O. Bebek, “Needle localization using gabor filtering in 2D ultrasound images,” in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, May 2014, pp. 4881–4886.

[21] ——, “Gabor filter based localization of needles in ultrasound guided robotic interventions,” in *2014 IEEE International Conference on Imaging Systems and Techniques (IST) Proceedings*, Oct 2014, pp. 112–117.

[22] W. Kremkau F and J. Taylor K, “Artifacts in ultrasound imaging.” *Journal of Ultrasound in Medicine*, vol. 5, no. 4, pp. 227–237, 1986.

[23] J. Huang, J. K. Triedman, N. V. Vasilyev, Y. Suematsu, R. O. Cleveland, and P. E. Dupont, "Imaging artifacts of medical instruments in ultrasound-guided interventions," *Journal of Ultrasound in Medicine*, vol. 26, no. 10, pp. 1303–1322, 2007.

[24] G. Reusz, P. Sarkany, J. Gal, and A. Csomos, "Needle-related ultrasound artifacts and their importance in anaesthetic practice," *British Journal of Anaesthesia*, vol. 112, no. 5, p. 794, 2014.

[25] M. Ding and A. Fenster, "Projection-based needle segmentation in 3D ultrasound images." *Computer Aided Surgery*, vol. 9, no. 5, pp. 193 – 201, 2004.

[26] M. Ding, Z. Wei, L. Gardi, D. B. Downey, and A. Fenster, "Needle and seed segmentation in intra-operative 3D ultrasound-guided prostate brachytherapy," *Ultrasonics*, vol. 44, Supplement, no. 0, pp. e331 – e336, 2006, proceedings of Ultrasonics International (UI'05) and World Congress on Ultrasonics (WCU).

[27] W. Qiu, M. Yuchi, M. Ding, D. Tessier, and A. Fenster, "Needle segmentation using 3d hough transform in 3d trus guided prostate transperineal therapy," *Medical Physics*, vol. 40, no. 4, pp. 042 902–n/a, 2013, 042902.

[28] H. Neshat and R. Patel, "Real-time parametric curved needle segmentation in 3D ultrasound images," in *Biomedical Robotics and Biomechatronics, 2008. BioRob 2008. 2nd IEEE RAS EMBS International Conference on*, Oct 2008, pp. 670–675.

[29] T. Adebar, S. Salcudean, S. Mahdavi, M. Moradi, C. Nguan, and L. Goldenberg, *A Robotic System for Intra-operative Trans-Rectal Ultrasound and Ultrasound Elastography in Radical Prostatectomy.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 79–89.

[30] A. Pourtaherian, S. Zinger, P. H. N. de With, H. H. M. Korsten, and N. Mihajlovic, "Gabor-based needle detection and tracking in three-dimensional ultrasound data volumes," in *2014 IEEE International Conference on Image Processing (ICIP)*, Oct 2014, pp. 3602–3606.

[31] Z. Wei, L. Gardi, D. Downey, and A. Fenster, "Oblique needle segmentation for 3d trus-guided robot-aided transperineal prostate brachytherapy," in *Biomedical Imaging: Nano to Macro, 2004. IEEE International Symposium on*, April 2004, pp. 960–963 Vol. 1.

[32] W. Qiu, M. Yuchi, and M. Ding, "Phase grouping-based needle segmentation in 3-D trans-rectal ultrasound-guided prostate trans-perineal therapy," *Ultrasound in Medicine & Biology*, vol. 40, no. 4, pp. 804 – 816, 2014.

[33] M. Waine, C. Rossa, R. Sloboda, N. Usmani, and M. Tavakoli, "3D shape visualization of curved needles in tissue from 2D ultrasound images using ransac," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, May 2015, pp. 4723–4728.

[34] G. A. Chapman, D. Johnson, and A. R. Bodenham, "Visualisation of needle position using ultrasonography." *Anaesthesia*, vol. 61, no. 2, pp. 148 – 158, 2006.

[35] K. Mathiassen, D. Dall'Alba, R. Muradore, P. Fiorini, and O. Elle, "Real-time biopsy needle tip estimation in 2d ultrasound images," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013, pp. 4363–4369.

[36] G. J. Vrooijink, M. Abayazid, and S. Misra, "Real-time three-dimensional flexible needle tracking using two-dimensional ultrasound," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 1688–1693.

[37] G. J. Vrooijink, M. Abayazid, S. Patil, R. Alterovitz, and S. Misra, "Needle path planning and steering in a three-dimensional non-static environment using two-dimensional ultrasound images," *The International Journal of Robotics Research*, vol. 33, no. 10, pp. 1361–1374, 2014.

[38] P. Chatelain, A. Krupa, and M. Marchal, "Real-time needle detection and tracking using a visually servoed 3d ultrasound probe," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 1676–1681.

[39] K. Mathiassen, D. Dall'Alba, R. Muradore, P. Fiorini, and O. J. Elle, "Robust real-time needle tracking in 2-d ultrasound images using statistical filtering," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 3, pp. 966–978, May 2017.

[40] K. Mathiassen, K. Glette, and O. J. Elle, "Visual servoing of a medical ultrasound probe for needle insertion," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 3426–3433.

[41] 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 *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, April 2005, pp. 4600–4605.

[42] R. J. Webster, J. S. Kim, N. J. Cowan, G. S. Chirikjian, and A. M. Okamura, "Nonholonomic modeling of needle steering," *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 509–525, 2006.

[43] M. Abayazid, R. J. Roesthuis, R. Reilink, and S. Misra, "Integrating deflection models and image feedback for real-time flexible needle steering," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 542–553, April 2013.

[44] P. Moreira and S. Misra, "Biomechanics-based curvature estimation for ultrasound-guided flexible needle steering in biological tissues," *Annals of Biomedical Engineering*, vol. 43, no. 8, pp. 1716–1726, 2015.

[45] T. K. Adebar and A. M. Okamura, "Recursive estimation of needle pose for control of 3d-ultrasound-guided robotic needle steering," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2014, pp. 4303–4308.

[46] V. Kallem and N. Cowan, "Image guidance of flexible tip-steerable needles," *Robotics, IEEE Transactions on*, vol. 25, no. 1, pp. 191–196, Feb 2009.

[47] B. Fallahi, C. Rossa, R. S. Sloboda, N. Usmani, and M. Tavakoli, "Sliding-based switching control for image-guided needle steering in soft tissue," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 860–867, July 2016.

[48] ——, "Sliding-based image-guided 3D needle steering in soft tissue," *Control Engineering Practice*, vol. 63, pp. 34 – 43, 2017.

[49] D. C. Rucker, J. Das, H. B. Gilbert, P. J. Swaney, M. I. Miga, N. Sarkar, and R. J. Webster, "Sliding mode control of steerable needles," *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1289–1299, Oct 2013.

[50] S. Patil, J. Burgner, R. J. Webster, and R. Alterovitz, "Needle steering in 3-D via rapid replanning," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 853–864, Aug 2014.

[51] H. Kataoka, T. Washio, M. Audette, and K. Mizuhara, "A model for relations between needle deflection, force, and thickness on needle penetration," in *Proceedings of the 4th International Conference on Medical Image Computing and Computer-Assisted Intervention*, ser. MICCAI '01. Springer-Verlag, 2001, pp. 966–974.

[52] A. M. Okamura, C. Simone, and M. D. O'Leary, "Force modeling for needle insertion into soft tissue," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 10, pp. 1707–1716, Oct 2004.

[53] N. Abolhassani and R. V. Patel, "Deflection of a flexible needle during insertion into soft tissue," in *Proceedings of the 2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, 2006, pp. 3858–3861.

[54] S. Misra, K. B. Reed, A. S. Douglas, K. T. Ramesh, and A. M. Okamura, "Needle-tissue interaction forces for bevel-tip steerable needles," in *2008 2nd IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics*, Oct 2008, pp. 224–231.

[55] M. Khadem, C. Rossa, R. S. Sloboda, N. Usmani, and M. Tavakoli, "Mechanics of Tissue Cutting During Needle Insertion in Biological Tissue," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 800–807, July 2016.

[56] ——, "Ultrasound-guided model predictive control of needle steering in biological tissue," *Journal of Medical Robotics Research*, vol. 01, no. 01, p. 1640007, 2016.

[57] M. Khadem, C. Rossa, N. Usmani, R. S. Sloboda, and M. Tavakoli, "Semi-automated needle steering in biological tissue using an ultrasound-based deflection predictor," *Annals of Biomedical Engineering*, vol. 45, no. 4, pp. 924–938, 2017.

[58] C. Rossa, M. Khadem, R. Sloboda, N. Usmani, and M. Tavakoli, "Adaptive quasi-static modelling of needle deflection during steering in soft tissue," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 916–923, July 2016.

[59] N. U. Thomas Lehmann, Mahdi Tavakoli and R. Sloboda, "Force-sensor-based estimation of needle tip deflection in brachytherapy," *Journal of Sensors*, vol. 2013, 2013.

[60] T. Lehmann, C. Rossa, N. Usmani, R. S. Sloboda, and M. Tavakoli, "Intraoperative tissue young's modulus identification during needle insertion using a laterally actuated needle," *IEEE Transactions on Instrumentation and Measurement*, vol. 67, no. 2, pp. 371–381, Feb 2018.

[61] S. Misra, K. Reed, B. Schafer, K. Ramesh, and A. Okamura, "Mechanics of flexible needles robotically steered through soft tissue." *International Journal of Robotics Research*, vol. 29, no. 13, pp. 1640 – 1660, 2010.

[62] D. De Lorenzo, Y. Koseki, E. De Momi, K. Chinzei, and A. Okamura, "Coaxial needle insertion assistant with enhanced force feedback." *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 2, pp. 379 – 389, 2013.

[63] S. DiMaio and S. Salcudean, "Needle steering and motion planning in soft tissues," *Biomedical Engineering, IEEE Transactions on*, vol. 52, no. 6, pp. 965–974, June 2005.

[64] M. Waine, C. Rossa, R. Sloboda, N. Usmani, and M. Tavakoli, "Needle tracking and deflection prediction for robot-assisted needle insertion using 2d ultrasound images," *Journal of Medical Robotics Research*, vol. 01, no. 01, p. 1640001, 2016.

[65] Z. Neubach and M. Shoham, "Ultrasound-guided robot for flexible needle steering," *Biomedical Engineering, IEEE Transactions on*, vol. 57, no. 4, pp. 799–805, April 2010.

[66] O. Goksel, E. Dehghan, and S. E. Salcudean, "Modeling and simulation of flexible needles," *Medical engineering & physics*, vol. 31, no. 9, pp. 1069–1078, 2009.

[67] C. Rossa, R. Sloboda, N. Usmani, and M. Tavakoli, "Estimating needle tip deflection in biological tissue from a single transverse ultrasound image: application to brachytherapy," *International Journal of Computer Assisted Radiology and Surgery*, vol. 11, no. 7, pp. 1347–1359, Jul 2016.

[68] J. Noble and D. Boukerroui, "Ultrasound image segmentation: a survey," *Medical Imaging, IEEE Transactions on*, vol. 25, no. 8, pp. 987–1010, Aug 2006.

[69] S. Ghose, A. Oliver, R. Martí, X. Lladó, J. C. Vilanova, J. Freixenet, J. Mitra, D. Sidibé, and F. Meriaudeau, "A survey of prostate segmentation methodologies in ultrasound, magnetic resonance and computed tomography images," *Computer Methods and Programs in Biomedicine*, vol. 108, no. 1, pp. 262 – 287, 2012.

[70] R. P. Singh, S. Gupta, and U. R. Acharya, "Segmentation of prostate contours for automated diagnosis using ultrasound images: A survey," *Journal of Computational Science*, vol. 21, pp. 223 – 231, 2017.

[71] H. M. Ladak, F. Mao, Y. Wang, D. B. Downey, D. A. Steinman, and A. Fenster, "Prostate boundary segmentation from 2D ultrasound images," *Medical Physics*, vol. 27, no. 8, pp. 1777–1788, 2000.

[72] Y. Wang, H. N. Cardinal, D. B. Downey, and A. Fenster, "Semiautomatic three-dimensional segmentation of the prostate using two-dimensional ultrasound images," *Medical Physics*, vol. 30, no. 5, pp. 887–897, 2003.

[73] N. Hu, D. B. Downey, A. Fenster, and H. M. Ladak, "Prostate boundary segmentation from 3D ultrasound images," *Medical Physics*, vol. 30, no. 7, pp. 1648–1659, 2003.

[74] P. Abolmaesumi and M. R. Sirouspour, "An interacting multiple model probabilistic data association filter for cavity boundary extraction from ultrasound images," *IEEE Transactions on Medical Imaging*, vol. 23, no. 6, pp. 772–784, June 2004.

[75] S. Badiei, S. E. Salcudean, J. Varah, and W. Morris, "Prostate segmentation in 2D ultrasound images using image warping and ellipse fitting," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2006*, ser. Lecture Notes in Computer Science, R. Larsen, M. Nielsen, and J. Sporring, Eds. Springer Berlin Heidelberg, 2006, vol. 4191, pp. 17–24.

[76] L. Gong, S. D. Pathak, D. R. Haynor, P. S. Cho, and Y. Kim, "Parametric shape modeling using deformable superellipses for prostate segmentation," *IEEE Transactions on Medical Imaging*, vol. 23, no. 3, pp. 340–349, March 2004.

[77] S. S. Mahdavi, N. Chng, I. Spadinger, W. J. Morris, and S. E. Salcudean, "Semi-automatic segmentation for prostate interventions," *Medical Image Analysis*, vol. 15, no. 2, pp. 226 – 237, 2011.

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

[79] S. D. Pathak, D. R. Haynor, and Y. Kim, "Edge-guided boundary delineation in prostate ultrasound images," *IEEE Transactions on Medical Imaging*, vol. 19, no. 12, pp. 1211–1219, Dec 2000.

[80] B. Chiu, G. H. Freeman, M. M. A. Salama, and A. Fenster, "Prostate segmentation algorithm using dyadic wavelet transform and discrete dynamic contour," *Physics in Medicine and Biology*, vol. 49, no. 21, pp. 4943–4960, oct 2004.

[81] A. C. Hodge, A. Fenster, D. B. Downey, and H. M. Ladak, "Prostate boundary segmentation from ultrasound images using 2D active shape models: Optimisation and extension to 3D," *Computer Methods and Programs in Biomedicine*, vol. 84, no. 2–3, pp. 99 – 113, 2006, medical Image Segmentation Special Issue.

[82] I. B. Tutar, S. D. Pathak, L. Gong, P. S. Cho, K. Wallner, and Y. Kim, "Semiautomatic 3-d prostate segmentation from trus images using spherical harmonics," *IEEE Transactions on Medical Imaging*, vol. 25, no. 12, pp. 1645–1654, Dec 2006.

[83] S. Nouranian, S. Mahdavi, I. Spadinger, W. Morris, S. Salcudean, and P. Abolmaesumi, "A multi-atlas-based segmentation framework for prostate brachytherapy," *Medical Imaging, IEEE Transactions on*, vol. 34, no. 4, pp. 950–961, April 2015.

[84] S. Nouranian, M. Ramezani, I. Spadinger, W. J. Morris, S. E. Salcudean, and P. Abolmaesumi, "Learning-based multi-label segmentation of transrectal ultrasound images for prostate brachytherapy," *IEEE Transactions on Medical Imaging*, vol. 35, no. 3, pp. 921–932, March 2016.

[85] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[86] P. Novotny, J. Stoll, N. Vasilyev, P. del Nido, P. Dupont, T. Zickler, and R. Howe, “Gpu based real-time instrument tracking with three-dimensional ultrasound.” *Medical Image Analysis*, vol. 11, no. 5, pp. 458 – 464, 2007.

[87] M. Uherčík, J. Kybic, H. Liebgott, and C. Cachard, “Model fitting using RANSAC for surgical tool localization in 3-D ultrasound images.” *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 8, pp. 1907 – 1916, 2010.

[88] N. Otsu, “A threshold selection method from gray-level histograms,” *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 9, no. 1, pp. 62–66, Jan 1979.

[89] R. C. Gonzalez and R. E. Woods, *Digital image processing.* Upper Saddle River, N.J. : Prentice Hall, 2002.

[90] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-Gaussian bayesian tracking,” *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, Feb 2002.

[91] N. Zheng and J. Xue, *Statistical Learning and Pattern Analysis for Image and Video Processing.* London: Springer London, 2009, ch. Probabilistic Data Fusion for Robust Visual Tracking, pp. 245–285.

[92] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. V. Gool, “Robust tracking-by-detection using a detector confidence particle filter,” in *Computer Vision, 2009 IEEE 12th International Conference on*, Sept 2009, pp. 1515–1522.

[93] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, ser. IJCAI’81. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1981, pp. 674–679.

[94] S. Baker and I. Matthews, “Lucas-kanade 20 years on: A unifying framework.” *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221 – 255, 2004.

[95] E. Antonakos, J. Alabort-i Medina, G. Tzimiropoulos, and S. Zafeiriou, “Feature-based lucas-kanade and active appearance models,” *Image Processing, IEEE Transactions on*, vol. 24, no. 9, pp. 2617–2632, Sept 2015.

[96] R. Alterovitz, M. Branicky, and K. Goldberg, “Motion planning under uncertainty for image-guided medical needle steering,” *The International Journal of Robotics Research*, vol. 27, no. 11-12, pp. 1361–1374, 2008.

[97] N. Abolhassani, R. Patel, and F. Ayazi, “Needle control along desired tracks in robotic prostate brachytherapy,” in *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, Oct 2007, pp. 3361–3366.

[98] D. Liberzon, *Switching in Systems and Control*, 1st ed. Boston, MA: Birkhäuser Basel, 2003.

[99] P. Tabuada, “Event-triggered real-time scheduling of stabilizing control tasks,” *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1680–1685, Sept 2007.

[100] W. P. M. H. Heemels, K. H. Johansson, and P. Tabuada, “An introduction to event-triggered and self-triggered control,” in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, Dec 2012, pp. 3270–3285.

[101] D. Ashlock, *Evolutionary Computation for Modeling and Optimization.* New York: Springer, 2010.

[102] R. Richa, R. Sznitman, R. Taylor, and G. Hager, “Visual tracking using the sum of conditional variance,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2011, pp. 2953–2958.

[103] C. Salembier, P. Lavagnini, P. Nickers, P. Mangili, A. Rijnders, A. Polo, J. Venselaar, and P. Hoskin, “Tumour and target volumes in permanent prostate brachytherapy: A supplement to the estro/eau/eortc recommendations on prostate brachytherapy,” *Radiotherapy and Oncology*, vol. 83, no. 1, pp. 3 – 10, 2007.

[104] A. Rampun, B. Tiddeman, R. Zwiggelaar, and P. Malcolm, “Computer aided diagnosis of prostate cancer: A texton based approach.” *Medical physics*, vol. 43, p. 5412, Oct 2016.

[105] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “Slic superpixels compared to state-of-the-art superpixel methods,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, Nov 2012.

[106] F. van der Heijden, “Edge and line feature extraction based on covariance models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 1, pp. 16–33, Jan 1995.

[107] S. Ando, “Image field categorization and edge/corner detection from gradient covariance,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 2, pp. 179–190, Feb 2000.

[108] C. Reyes-Aldasoro and A. Bhalerao, “The Bhattacharyya space for feature selection and its application to texture segmentation,” *Pattern Recognition*, vol. 39, no. 5, pp. 812 – 826, 2006.

[109] M. Donoser and H. Bischof, “Using covariance matrices for unsupervised texture segmentation,” in *2008 19th International Conference on Pattern Recognition*, Dec 2008, pp. 1–4.

[110] P.-H. Tsui, Y.-L. Wan, D.-I. Tai, and Y.-C. Shu, “Effects of estimators on ultrasound nakagami imaging in visualizing the change in the backscattered statistics from a rayleigh distribution to a pre-rayleigh distribution,” *Ultrasound in Medicine & Biology*, vol. 41, no. 8, pp. 2240 – 2251, 2015.

[111] P.-H. Tsui, “Current status and future prospects of scattering statistics in ultrasound imaging,” *Journal of Medical Ultrasound*, vol. 24, no. 3, pp. 83 – 85, 2016.

[112] N. Xu, N. Ahuja, and R. Bansal, “Object segmentation using graph cuts based active contours,” *Computer Vision and Image Understanding*, vol. 107, no. 3, pp. 210 – 224, 2007.

[113] P. Zhang, Y. Liang, S. Chang, and H. Fan, “Kidney segmentation in CT sequences using graph cuts based active contours model and contextual continuity,” *Medical Physics*, vol. 40, no. 8, p. 081905, 2013.

[114] X. Chen, J. K. Udupa, A. Alavi, and D. A. Torigian, “GC-ASM: Synergistic integration of graph-cut and active shape model strategies for medical image segmentation,” *Computer vision and image understanding*, vol. 117, pp. 513–24, May 2013.

[115] H.-M. Kaltenbach, *A Concise Guide to Statistics.* Basel, Switzerland: Springer-Verlag Berlin Heidelberg, 2012.

[116] F. Destrempes and G. Cloutier, “A critical review and uniformized representation of statistical distributions modeling the ultrasound echo envelope,” *Ultrasound in Medicine & Biology*, vol. 36, no. 7, pp. 1037 – 1051, 2010.

[117] Y. Boykov and O. Veksler, “Graph cuts in vision and graphics:theories and applications,” in *Handbook of Mathematical Models in Computer Vision*, 2006.