AN AUGMENTED REALITY HAPTIC TRAINING SIMULATOR FOR SPINAL NEEDLE PROCEDURES

by

COLIN SUTHERLAND

A thesis submitted to the
Department of Electrical and Computer Engineering
in conformity with the requirements for
the degree of Master of Applied Science

Queen's University
Kingston, Ontario, Canada

November 2011

Copyright © Colin Sutherland, 2011

Abstract

Medical simulators have become commonly used to teach new procedures to medical students and clinicians. Their accessibility allows trainees to perform training whenever they desire, and their flexibility allows for various patient body types and conditions to be simulated. This is in contrast to *in-vivo* training, which requires direct supervision from a trained clinician, and access to a live patient or cadaver, both of which have restrictions.

This thesis proposes a novel prototype system for spinal anesthesia procedures which combines the use of both a haptic device for virtual, ultrasound-guided (US) needle simulations, and a physical mannequin registered to a patient specific computed tomography (CT) volume in order to create an augmented reality (AR) overlay. The mannequin will provide the user with a greater sense of spatial awareness that is not present in a purely virtual simulation, as well as providing physical visual clues to navigate the patient. Another novel aspect is the simulation of US images from CT images deformed via a finite element model (FEM).

The system is composed of a torso mannequin from Sawbones Inc., a Micron-Tracker2 optical tracking system from Claron Technology, a Sensable PHANToM Premium 1.5A haptic device and a graphical user interface (GUI) to display relevant visual feedback. The GUI allows the user to view the AR overlaid on the video feed,

and the CT slice and simulated US image based the position/orientation of a dummy US probe.

Forces during the insertion are created via the FEM and sent to the haptic device. These forces include force from needle tip insertion, friction along the length of the needle inside the body, and from displacing the needle off its original insertion axis. Input to the system consists of a patient CT volume.

The system is able to create forces that closely match those reported in the literature. A user study consisting of subjects with expertise ranging from familiarity with medical imaging to clinical experience with needle insertion procedures, was performed to qualitatively analyze the performance of the system. Three experienced physicians were also consulted for input and improvements. The feedback received from the questionnaire, and comments from the subjects and physicians, showed the system is able to simulate a real needle insertion quite well, and the graphical aids added were helpful during the training procedure.

Acknowledgments

Without the guidance and support of my supervisors, Keyvan Hashtrudi-Zaad, Parvin Mousavi and Purang Abolmaesumi, there would be no thesis for this work and to them I would like to extend a most sincere thank you. I appreciate all the help along the way, especially during the tougher times.

Secondly, I would like to thank Dr. Rick Sellens for providing me with the Micron-Tracker2 camera system, which was an integral part of the simulator and Dr. Ramtin Shams for the MATLAB ultrasound simulation codes, which were the basis for the CUDA implementation. A special thank you goes to my co-worker Amir Haddadi for providing me with codes that were the basis for the FEM computation. I would like to thank all the subjects who took part in the user study, those from the Perk lab and the medical students and residents at Kingston General Hospital in particular for their insight into areas of the system that could be improved. Also, I would specifically like to thank Drs. Bob McGraw, Dan Borschneck and Melanie Jaeger for taking time out of their busy schedules to give feedback on the system. As well, I would like to thank NSERC for supporting this research.

Finally, I would like to thank Kat Romatowski for always being there for me and sticking through the hard times.

Contents

Abstract	i
Acknowledgments	iii
Contents	iv
List of Tables	viii
List of Figures	ix
Chapter 1: Introduction	1
1.1 Motivation	4
1.2 System Overview	6
1.3 Objective	7
1.4 Contributions	8
1.5 Outline	9
Chapter 2: Background on Medical Simulators	11
2.1 Technical Background	13
2.1.1 Medical Imaging	13
2.1.2 Hardware and Software	14

2.1.3	Needle Insertion and Bending Methods	16
2.1.4	Augmented Reality	18
2.2	Medical Simulators	20
2.2.1	Virtual Reality	20
2.2.2	Mannequin Based	25
2.3	Summary	28
Chapter 3: System Overview and Description		29
3.1	Overview	29
3.2	Components	30
3.2.1	Camera System	31
3.2.2	Mannequin	32
3.2.3	Haptic Device	33
3.2.4	Finite Element Model	35
3.2.5	Ultrasound Simulation	35
3.3	Program Flow and Description	36
3.4	Summary	40
Chapter 4: Tissue Deformation and Force Feedback		41
4.1	Overview	41
4.2	Finite Element Construction	43
4.3	Mesh deformations	48
4.4	Transverse Needle Resistance	55
4.5	Summary	58
Chapter 5: Visualization		59

5.1	Overview	59
5.2	Visual Tracking	60
5.3	Printed Spine Model Substitute	62
5.3.1	Overlay Method	62
5.4	Mannequin Based Substitute	64
5.4.1	Mannequin	66
5.4.2	Overlay Method	67
5.5	Ultrasound Simulation	74
5.5.1	Probe Tracking	75
5.5.2	Volume Preparation	78
5.5.3	Simulation	79
5.6	Summary	82
Chapter 6:	System Performance and Evaluation	83
6.1	Overview	83
6.2	Hardware and Software Performance	84
6.3	User Impact Study	87
6.3.1	Questionnaire Design	88
6.3.2	Evaluation of the Questionnaire	88
6.4	Summary	93
Chapter 7:	Summary and Conclusions	94
7.1	Summary of Contributions	94
7.2	Contributions	95
7.2.1	System as a simulation environment	95

7.2.2	Tracking system	95
7.2.3	Graphical User Interface	95
7.2.4	Overlay of CT volume	96
7.2.5	Finite Element Model and Force Feedback	96
7.2.6	CUDA-enabled Ultrasound Simulation	97
7.2.7	User Evaluation Study	97
7.3	Future Work	98
Bibliography		102
Appendix A: Evaluation Questionnaire		110

List of Tables

5.1	US Simulation comparison for a 300×300 pixel image	81
6.1	Questions on System Functionality. S1-6 correspond to inexperienced users, S7*-8* to experienced users.	89
6.2	Questions on the Graphical User Interface (GUI). S1-6 correspond to inexperienced users, S7*-8* to experienced users.	90
6.3	Questions for Haptic Feedback. S1-6 correspond to inexperienced users, S7*-8* to experienced users.	91

List of Figures

1.1	An overview of needle insertion for spinal anesthesia: <i>Left:</i> Needle direction for midline approach. <i>Right:</i> Needle direction for lateral approach.	3
1.2	Overview of System with data flow. MicronTracker2 camera tracks markers on mannequin, sending the data, along with video to the PC which displays an augmented reality overlay. The PHANToM haptic device sends positional/rotational data to the PC and receives force signals calculated from the model in return	5
2.1	<i>Left:</i> An example of CT scanning showing rotating X-Ray source and receivers, and a mechanical table. <i>Right:</i> An sample CT scan from the abdomen of a patient.	14
2.2	A PHANToM Haptic Device. <i>Left:</i> Schematic showing the linkages and angles of the device. <i>Right:</i> PHANToM Premium 1.5A.	16
2.3	An example of depth perception issues created by simply overlaying on image on another. The overlaid image appears to be in front of the background.	19
2.4	Virtual Reality training simulators.	21
3.1	Optional caption for list of figures	31

3.2	System schematics showing the data flow between components. MicronTracker2 sends video feed and tracking info to PC, which receives position info from PHANToM and returns force data.	32
3.3	Two components of the system. <i>Left:</i> The MicronTracker2 camera system <i>Right:</i> The torso mannequin from Sawbones, Inc.	33
3.4	The retractable needle guide attachment. <i>Left:</i> The needle is at full length just prior to insertion. <i>Right:</i> The guide retracts up the stylus as the virtual needle is inserted into body.	34
3.5	A CT image of lumbar spine <i>left</i> with simulated US image <i>right</i>	36
3.6	The data flow in program. Turquoise boxes are program functions that input/output data. The type of data is described on the arrow. . . .	36
3.7	The program loops with data flow between steps.	37
3.8	The classes in the program.	39
4.1	Physical needle guide for modeled needle.	42
4.2	<i>Left:</i> Force $\vec{f}$ acting on the node at $\vec{x}$ of the mesh, prior to calculation of displacements. <i>Right:</i> The mesh after displacement, with displacement $\vec{u}$ resulting from $\vec{f}$ being applied to the node at $\vec{x}$	43
4.3	<i>Left:</i> Mesh prior to adaptation. <i>Right:</i> Mesh post adaptation. Mesh elements i, j, k, l share the node n being displaced	49
4.4	Needle inserted into mesh at element e , with closest vertex n being displaced	52
4.5	State machine for latching between force directions. Machine begins at state A when needle is inserted into the body	54

4.6	Needle inserted into body and bent. Original insertion vector $\vec{v}_i$, current needle base b_n , closest point along original insertion vector to current base p_b direction of bend $\vec{v}_b$ with end deflection $\vec{\delta}_{end}$, end force from bending f_{end} in direction $\vec{v}_{end}$, forces from friction and tip $\vec{f}_{tip}, f_{fric}$	57
5.1	Facet tracking information displayed by MicronTracker2. <i>Left:</i> An example of a cross shaped face with four points. <i>Right:</i> An example of two L-shaped facets with three points each.	60
5.2	A MicronTracker2 Camera with coordinate frame attached. Looking in the direction of the camera viewpoint, the x direction is to the right, the y direction down, and z direction outwards in the direction the camera is facing	61
5.3	A composition of a camera image and a VTK scene snapshot to create an image overlay.	63
5.4	A comparison of a direct overlay (<i>Top</i>) with an overlay using edge blending (<i>Bottom</i>).	64
5.5	<i>Top:</i> Torso mannequin with spine model shown in the cutout region. CT beads are visible on the right, with MicronTracker markers on left. <i>Bottom:</i> A slice of the mannequin CT volume, with bead visible at top left.	65
5.6	A registration example for the mannequin CT (red bounding box and blue dots) with the MicronTracker markers (green dots).	70
5.7	An example of the final output of CT overlay procedure.	73

5.8	Overview of US simulation procedure. <i>Left</i> : Slicing the CT volume at the probe position. <i>Centre</i> : Ray tracing down a fan shape region of the CT slice. <i>Right</i> : Final simulated US image.	75
6.1	A screenshot of the Graphical User Interface. Main video feed is on the left, the auxiliary view, simulated US image and CT slice are on the right.	84
6.2	A plot of the forces during simulated needle insertion versus time for an insertion piercing the ligamentum flavum (LF).	85
6.3	A plot of the output from the finite state machine. Force (red) switches direction when velocity is over or under the threshold ($\pm 0.03\text{mm/s}$) for the latch time of 100ms. The binary switch is shown in green.	86

Chapter 1

Introduction

Medical training systems and simulators are becoming more common for training medical students, clinicians and technicians in the procedures they will need to perform in operating rooms [1]. Generally, *in vivo* training for these procedures would require the direct supervision of a trained clinician, as well as access to either a live patient, or a cadaver, to perform trial operations on. The combination of these factors leads to limited training availability, as the time of trained clinicians is both valuable and in high demand, and patients and cadavers are not available at all hours of the day. In addition, trainees generally do not have the skill set that trained clinicians do, and having them perform their training on live patients during operations could lead to inconvenience or complications for the patient [2]. Medical training systems can aid in reducing the cost, increasing the availability of training and providing the ability to access a multitude of patient specific data in an off-line environment, particularly for rare cases that may not be experienced on a regular basis. In addition, they provide a platform for the technical assessment of trainees, through the ability to collect multi-sensory data and store it for future review by the trainee or assessment

by the supervisor.

The two main types of training systems in the literature are Virtual Reality (VR) based and mannequin based. VR systems (such as [3],[4],[5],[6],[7] and [8]) are by far the most common type of medical training systems in the literature. These systems allow for easy creation of many different virtual environments (or scenes) by taking some form of medical imaging modality as an input (such as patient CT, MRI volumes, or US scans) and enabling the user to interact with it through the use of virtual tools. The output of VR systems is most commonly solely a computer monitor. Additionally, some systems use stereoscopic goggles to give the illusion of depth, or project the image from the monitor onto a slightly mirrored yet still transparent screen, or in rare cases use actual three dimensional (3D) vision goggles. While the majority of medical training systems are based on VR, mannequins are also used as the basis for training ([9], [8], [10]). Compared to VR systems, mannequins provide trainees with solid spatial orientation when they are performing trials, however a mannequin itself does not provide a very good platform for examining varied patients and providing force feedback, as the force would be dependent on the internal structure of the mannequin, which would not change from trial to trial.

Both types of simulation systems may use multi-modal medical imaging to provide visual feedback, and to construct haptic and visual scenes. The types of medical images include ultrasound (US), magnetic resonance imaging (MRI) and computed tomography (CT) and they can be either complete data sets that are pre-scanned from patients, or simulated from each other. These medical imaging data sets can be used to construct the 3D scenes used in VR simulators and displayed through the use of volume or surface rendering techniques. They can also be used to create

Figure 1.1: An overview of needle insertion for spinal anesthesia: *Left*: Needle direction for midline approach. *Right*: Needle direction for lateral approach [11].

structures for haptic devices to interact with. This can be done by assigning stiffness values to ranges of data via lookup tables, or by segmenting out a certain region to be considered a hard surface.

This thesis will focus on the development of a prototype simulator for percutaneous spinal interventions, specifically spinal anesthesia, with potential to be used as a training simulator. Spinal anesthesia is a procedure performed prior to operations below the navel, commonly hernia surgeries as well as operations on the reproductive organs and legs, of a patient to numb the lower extremities. It is performed by inserting a thin needle (commonly 22-25 gauge) into the lumbar region of the spine, below the level that the spinal cord reaches. The insertion takes place between two spinous processes of the lumbar spine, and can be inserted straight in, or at an angle. In Figure 1.1, two different approaches commonly used in spinal anesthesia are shown:

the midline and lateral approaches. The arrows in the figure represent the angles from the direct midline the needle is moved prior to insertion. Often a hollow, larger gauge, needle called an introducer is inserted into the patient prior to the thin anesthesia needle. This is done to stabilize the thinner needle and prevent it from bending on entry. The spinal anesthesia procedure is characterized by a “loss of resistance” that occurs when the needle pushes through the tough ligamentum flavum. This sudden drop in force felt would signal the clinician that he/she has reached the spinal column. Once the spinal column is reached, the clinician should see cerebrospinal fluid in the needle. At this point, the anesthetic agent is injected into the spinal column.

1.1 Motivation

There are several motivations behind this work. As mentioned in the introduction, training time for clinical procedures is limited based on the availability of supervisors and training subjects (whether live patients or cadavers). Even when trainees have access to patients, they might not encounter patients with rare medical conditions. Therefore we wish to develop a system that will allow a trainee to perform a simulation of the procedure, at any time, on a virtual patient of any body type or medical condition. This would help reduce costs and relieve the burden on the precious time of trained clinicians. In addition, since *in-vivo* training is limited to the patient currently available, trainees would seldom be able to experience training on patients with rare anatomy or conditions.

As previously discussed, VR simulators can suffer from a lack of spatial orientation because they only provide visuals on the two dimensional (2D) platform of a computer monitor (sometimes enhanced to “3D” by stereoscopic goggles) which is away from

Figure 1.2: Overview of System with data flow. MicronTracker2 camera tracks markers on mannequin, sending the data, along with video to the PC which displays an augmented reality overlay. The PHANTOM haptic device sends positional/rotational data to the PC and receives force signals calculated from the model in return

the area of operation. Mannequin based systems that do not use haptic feedback are unable to simulate varied patient body types and conditions, while those systems that solely use mannequins as guides and stand-ins cannot provide extra functionality that can be gained from incorporating mannequins more heavily into a system. To that end, a goal of this work is to create a simulation environment that encompasses the benefits of both types of systems, to attempt to counteract the drawbacks of each, and to take full advantage of the use of a mannequin.

1.2 System Overview

The proposed system will allow a user to perform needle insertions and US scanning on a specific virtual patient projected on a physical, generic mannequin. The patient is rendered graphically through a personal computer (PC) and physically through a haptic robotic device. An overview of the system can be seen in Figure 1.2. The system uses a Sensable PHANToM Premium 1.5A haptic device as the force feedback unit with three active, translational, degrees of freedom (DOFs) and additional three passive DOFs. The PHANToM device was chosen mainly because it is the most commonly used device in research labs, and the work developed for this thesis could be implemented by other research groups. Visual tracking and video feedback is provided by a Claron Technologies MicronTracker2 camera system. The MicronTracker2 is the foundation of the augmented reality CT overlay, allowing a user to view a cutaway of the patient anatomy projected onto the camera feed. A torso mannequin from Sawbones Inc. is used as the tracked body. The mannequin is visually tracked and registered to a patient specific 3D CT data volume, from which the US images are simulated. Tracking also allows for the CT volume to be overlaid on the video feed of the camera, and blended to create a realistic cutaway. An FEM created from the CT volume is deformed based on the position of the virtual needle tip (from the haptic device). These deformations are applied to the CT volume, in order to deform the US images during simulation, and they also provide the force feedback to the user through the haptic device.

The video feedback from the camera, CT slice and simulated US image are displayed on the PC monitor for the user to view. The video feedback has the CT volume overlaid on it based on the position of the mannequin; it also displays a wire-frame

outline of the US slice through the volume based on the position of the virtual needle.

The system is written in C++ using Microsoft Visual Studio 2005 on a Quad Core Intel Q9550 2.83GHz PC with 3.25GB of RAM. The PHANToM device is accessed through the use of Sensable's OpenHaptics2.0 architecture, and the MicronTracker2 camera through its SDK [12]. The Visualization Toolkit (VTK) [13] version 5.6.0 and Insight Toolkit (ITK) [14] version 3.18.0 are used for some image processing, and QT [15] version 4.6.2 is used for the graphical user interface.

1.3 Objective

The main objective of this thesis is to design and develop an effective and easy to use simulator, and to create a comprehensive simulation environment that incorporates the benefits of both VR and mannequin based simulators. The specific objectives are to:

1. Develop a prototype system with potential to assist students and clinicians in training for spinal needle insertions (specifically spinal anesthesia).
2. Incorporate a mannequin to provide the trainee with visual landmarks to guide the training.
3. Incorporate a variety of patients for input to the system - using CT scans as the data type.
4. Provide more information to the user through an overlaid CT volume of the patient on a video feed of the procedure, aligned with the mannequin.

5. Simulate an US image from the CT volume for the US guided portion since not all CT scans of patients are accompanied by US images.
6. Provide the user with force feedback representing a needle insertion
7. Evaluate the system with a user study of clinicians

1.4 Contributions

The work presented in this thesis will have the following contributions:

1. Development of the first simulator system for percutaneous spinal procedures that incorporates a tracked mannequin and registered CT data with video overlay to guide the training procedure. The system also displays real-time deformation of the tissue under needle pressure using a FEM and haptic device.
2. Development of a real-time US simulation algorithm that incorporates simulated deformations of the tissue. The algorithm has been implemented on GPU through the use of CUDA, an improvement over the original MATLAB codes provided.
3. Conduct a user study to investigate the effectiveness of the functionality of the system and its potential for training purposes using two subject groups of differing skills.

1.5 Outline

This thesis will outline the development and evaluation of a prototype system for spinal anesthesia procedures. It has the following chapters

Chapter 2 Background: This chapter explores current medical training systems in the literature, detailing the type of the simulator and its use as well as the components of each. The benefits and drawbacks of different simulator designs are discussed. In addition, background on some technical topics is presented.

Chapter 3 System Overview and Description: This chapter discusses in more detail the system overview and purpose, detailing the hardware and software used, as well as the individual components of the system.

Chapter 4 Visual Tracking and Display Overlay: This chapter explains the methods behind the visual tracking hardware and software used in the system. It also explains how a patient's CT volume is overlaid onto the video feedback from the camera.

Chapter 5 Finite Element Model: The Finite Element Model (FEM) is discussed in detail in this chapter, including construction of the model and its use in providing force feedback and image deformations.

Chapter 6 Ultrasound Simulation: The algorithm used to simulate Ultrasound images from patient CT volumes is presented in this chapter.

Chapter 7 System Evaluation: This chapter presents an evaluation of the performance of the system, including a user study performed with clinicians familiar with the spinal anesthesia procedure.

Chapter 8 Discussion: In this chapter, a discussion of the system and its performance is presented.

Chapter 9 Conclusions and Future Work: This chapter will draw conclusions from the work and present avenues for future work to improve on the developed system.

Chapter 2

Background on Medical Simulators

Training for surgical procedures is being increasingly performed on medical simulators in order to provide trainees with greater flexibility and freedom in training times, as well as access to a multitude of different patient data. In comparison, *in vivo* training for these procedures generally requires direct supervision from trained clinicians, who have limited time, and access to either live patients or cadavers, which have limited availability in number, type of case, or examination time. Medical training systems can allow supervisors to technically assess the progress of trainees by collecting data from many sources and storing it for future review.

There are two main types of medical training systems in the literature: virtual reality (VR) systems and mannequin-based systems. VR based systems ([3],[4],[5],[6],[7],[8],[16],[17],[18],[19],[20],[21]) are characterized by the virtual, normally three dimensional (3D), scene they create and allow users to interact with. In a medical application, this scene would generally be a patient's body and/or internal organs, and users would interact with it through the use of a virtual tool, typically a haptic robotic device that can provide force feedback in various DOFs. The tip of the virtual

tool (called the end-effector of a haptic device) would correspond to the location of the virtual tool tip in the scene. A VR system would display the scene and accompanying information on a computer monitor. These types of systems can suffer from a lack of spatial orientation, as the visual feedback the user receives is not directly in the operating space, or where the user is manipulating the haptic robot or other device. A mannequin-based system, on the other hand, would rely on a mannequin representing a patient to provide a physical scene that could be interacted with directly. Systems using mannequins ([9], [8], [10], [22], [23]) either use them as simply a dummy for providing orientation and visual clues, or as part of a more involved platform, such as one which warps patient data to its shape. Some mannequin-based systems incorporate other devices as well, such as haptic devices, but only use the mannequin as a dummy; the only connection the mannequin would have with the system is its physical location in space. In some mannequin-based systems, the only form of force feedback comes from the internal structure of the mannequin itself, as no external devices are used to provide forces to the user. Both types of medical training systems use medical images to provide visual feedback to the user.

In this chapter, a review of current medical training systems in literature will be presented, as well as a discussion and background on some of the technical details presented in the thesis. The systems review will focus on the two main archetypes mentioned, VR- and mannequin-based systems. Several topics of interest will be mentioned in the technical background, ranging from a brief overview of some of the medical imaging modalities and display types to a more thorough review on needle insertion and AR applications.

2.1 Technical Background

2.1.1 Medical Imaging

In the systems discussed, there are two main types of medical images that are used: computed tomography (CT), and ultrasound (US). Since these are the two types of modalities used in this thesis, they will be the focus of discussion.

Computed Tomography

CT imaging is a system of imaging based on X-rays. A patient will lie on a mechanical table which can be moved in controlled increments into and out of a ring containing a rotating X-ray source and receiver, as shown in Figure 2.1. In an axial CT scan, which is the method used for recording the volumes used in this thesis, the table is stopped at each increment for the ring to perform an entire scan. The resulting data from the rotation forms one slice of the volume, and is computed via computer; once the entire body (or area of interest) has been scanned, the slices are combined to form the entire 3D CT volume. An issue with CT imaging is the existing delay in obtaining an image slice due to the imaging process which does not allow for easy use of the images for guidance during actual procedures. As a result, CT imaging is generally performed prior to, or after, operations to either initially plan procedures or view the effects of the procedures. In addition, there is an expressed concern with CT imaging becoming an increasing source of radiation exposure [24].

Ultrasound

US imaging is a very common form of imaging used for real-time diagnostics and operations. It uses high frequency sound waves to penetrate deep into tissue, and measures the amount of reflectance and scattering of the wave to determine the

Figure 2.1: *Left:* An example of CT scanning showing rotating X-Ray source and receivers, and a mechanical table. *Right:* An sample CT scan from the abdomen of a patient [25].

intensity of the returned images [26]. Imaging is performed through the use of a probe connected to an US machine. The probe can either be rectilinear or rounded; a rounded probe will create a fan-shaped image while a rectilinear probe creates a rectangle. There are many benefits to using US over other imaging modalities: It is low cost, low risk, and easy to use. However navigating through bodies using US images is more difficult than other types of medical images, especially in real-time as only 2D slices can be viewed, and navigation from a 2D viewpoint adds difficulty. The fact that US is quick to generate images and low cost has made it the most common form of real-time imaging, such as in surgical guidance.

2.1.2 Hardware and Software

In the following, a brief literature on the hardware and software components of the system will be discussed.

Haptic Devices

The most common form of hardware device in medical training simulators is a force feedback device, commonly referred to as a haptic device. A haptic device is an input/output robotic device with which a user interacts with a virtual environment. Most haptic devices have the same general shape - multiple linkages forming an arm shape that allows for free motion in the X,Y and Z planes, commonly equipped with a form of gimbal at the tip of the device (called the end-effector) to provide additional rotational degrees of freedom. Each joint is driven by a motor and contains a position measurement system, an encoder in most cases. At reset of the device, the linkages are placed in a neutral, 'home' position and all encoders are set to 0, so all further movements are relative to this chosen calibration origin. Based on the device structure and the angles obtained from the encoders, the 3D position of the end effector in the haptic device frame of reference can be calculated. The direct drive devices also provide the amount of force their end-effector applies to the user hand. Studies of the mechanical, electrical, kinematic and dynamic properties of the PHANToM haptic device are presented by Cavusoglu *et al.* [27] and Tahmasebi *et al.* [28].

Figure 2.2 displays a Sensable Technologies PHANToM Premium 1.5A device that has been used for this work. The device provides three active DOFs (force and movement) in the X, Y, Z planar directions, and additional three DOFs of passive rotation in pitch, roll, yaw of the gimbal attached to the end-effector. The device is able to provide up to 1.5N and 8.5N of continuous and maximum short-term exertable force respectively. The position of the end point of the stylus that has been attached to the gimbal can be calculated with an accuracy of 0.03mm.

Figure 2.2: A PHANTOM Haptic Device. *Left:* Schematic showing the linkages and angles of the device [28]. *Right:* PHANTOM Premium 1.5A.

2.1.3 Needle Insertion and Bending Methods

Simulation of needle insertion into soft tissue is an ongoing area researched by many groups. Much of the work on needle insertion and force generation will be discussed in the context of medical simulators in Section 2.2. This section provides an overview of models used for flexible needles. Although this thesis does not incorporate needle bending into the simulations for spinal anesthesia, some background information is presented for comprehensiveness.

A 2D Finite Element Model (FEM)-based approach for needle insertion is explored by DiMaio [29]. DiMaio constructs a finite element (FE) mesh using triangular elements which is deformed based on the position of a needle tip. Several methods of mesh deformation are examined, including mesh adaptation, re-meshing and simple mesh deformation. Needle FE models have been constructed; however they are deemed too computationally expensive for real-time application with haptic devices.

Denhghan and Salcudean present a brief overview of three needle bending methods [30] using a FE model with static triangular elements, FE model with non-linear beam elements, and an angular spring model where the needle is broken up into

segments connected by springs. The FEMs are observed to be computationally expensive. Moreover, the non-linear beam models cannot preserve needle length during deformation, whereas the angular spring model can be solved quickly using an iterative method. The angular spring model is also the most accurate of the 3 methods, when evaluated experimentally.

A more thorough review of needle insertion and bending methods is discussed by Abolhassani *et al.* [31]. They explore various kinematic and static models used to predict the motion of the tip of the needle for various needle types, including bevel-tip needles. An experimental setup used to collect data during *in-vivo* needle insertions is discussed [32]. A model based on the principles of infinitesimal force per unit length [33], and a nonholomic “bicycle-like” model [34] are examined in detail.

A mechanical epidural anesthesia insertion simulator is presented by clinicians at the Beth Israel-Deaconess Medical Center [35], which simulates tissue properties by applying forces to a needle via cables with tension driven by brushless motors.

Sitzman *et al.* [36] examine the effect of various needle parameters (gauge, type, tip bend) on needle deflection. Three different types of needles are used in the study: the Quincke (bevel tip), Sprotte (pencil point) and Whitacre (pencil point), with gauges from 18 to 29 and tip bends of 0, 5 and 10 degrees. The needles are inserted into porcine tissue and the deflection is checked at insertion distances of 20, 40 and 60mm. Higher gauge needles exhibit higher deflections, and all deflections increase as the needles are inserted deeper inside the tissue. As would be expected, the Quincke type needle deflects the most because of its bevel tip. For spinal anesthesia procedures, a pencil tip needle is generally used, with a gauge of around 23-25. For insertion depths of 20-40mm, these needles deflect by a maximum of 2.3mm at 40mm

(0.8mm max at 20mm depth).

2.1.4 Augmented Reality

Augmented Reality (AR) is a visual technique in which what a person sees is enhanced by overlaying another form of graphical data on a specific area in the field of view. This is accomplished by placing markers on the surface that is desired to be augmented and tracking them to be able to recreate the surface in a virtual 3D space. The desired overlay image, a volume or a plane or other form of data, is then aligned with the body in the virtual space. Then, some form of volume rendering/projection and blending needs to be run on the image data to integrate it back with the original image.

An example of a simple AR application is demonstrated by Paolis *et al.* [37], where the authors use large markers with different patterns to select a certain organ to be overlaid on a mannequin. The markers also control the location of the organ projection. This is a very simple form of AR, as the organ is simply inserted directly onto the image at the desired location, with no form of blending.

Simply inserting a second image on top of a picture/video feed such as done by Paolis *et al.* [37], creates obvious depth perception issues, as the overlaid image will always appear to be in front of the background image. Figure 2.3 shows an example of this depth perception issue. Here a spine model is overlaid onto a box containing the printed version of the model. As can be seen, the overlaid image appears to be in front of the background, not inside the box as it should be. This issue is discussed in more detail by Sielhorst *et al.* [38], where the authors evaluate seven different visualization techniques to reduce this mismatch via user study on 20 surgeons.

The depth perception issue is further discussed by Bichlmeier *et al.* [39], where

Figure 2.3: An example of depth perception issues created by simply overlaying on image on another. The overlaid image appears to be in front of the background.

the authors present a three component approach in volume ray casting to reduce mismatches in image overlays due to depth perception. In this case, the background image is the body/mannequin being overlaid by the anatomical volume image. The three components are:

- *Curvature*: The curvature of the body being overlaid on is measured by calculating the difference between the normal vectors at neighbouring locations on the skin surface. High curvature areas are made more opaque, to show the underlying background image through the overlay.
- *Angle of Incidence*: The angle of incidence between the incoming ray from the eye and the normal vector on the surface of the body is measured. Areas of the body with a low angle of incidence, implying the eye is looking directly at that location, would be more transparent to show the overlaid image.

- *Distance Fall-off*: The further away from the line of sight an area on the body is, the more opaque that area becomes, obscuring the overlaid image.

Finally, Kutter *et al.* [40] present a method for rendering high quality visualizations in real time for use in AR systems. The interesting technique used in the paper is a method for properly obscuring the overlaid image if a hand is placed over it. Normally AR systems do not account for this, and if a hand were to be placed over the body being overlaid onto, the hand would be obscured by the overlaid image. Kutter uses the same technique that is used for special effects in television and movies - the “green screen” effect, wherein a specific colour is used when an area is desired to be replaced with a different image or transparency. In this paper, the user wears a light blue surgical glove (light blue being a very uncommon colour for the background images used) and the areas of the background image containing the light blue colour are given full opacity, even if the overlaid image would have been there. This allows the hand to properly obscure the overlaid image as if it were really passing over the volume.

2.2 Medical Simulators

2.2.1 Virtual Reality

The review of VR training systems presented here will begin by discussing systems with limited functionality, and will continue through to systems with increased functionality and which include elements similar to the system proposed by this thesis.

A lumbar puncture training system presented by M. Farber *et al.* uses a PHANTOM Premium 1.5A haptic device with six active DOFs (providing force) [6]. This

(a) An US scanning system [41].

(b) Insertion with breathing model[42]

(c) A vertebroplasty simulator [5]

(d) US-guided needle puncture. Courtesy of Franck Vidal and Nigel W. John, Bangor University, UK [3]

Figure 2.4: Virtual Reality training simulators.

allows the system to completely constrain the virtual needle (represented by the haptic device) when it enters tissue. Force feedback during needle insertion is provided by Hooke's law for linear spring models and a proxy-based algorithm. The spring stiffness value is modified based on the stiffness of the area being interacted with, and the displacement Δx is calculated as the distance from the current needle position back to the location of the first entry into the current tissue area. The only form

of visual feedback in this system is the 3D CT volume shown on a computer monitor, along with the position of the virtual needle.

An US diagnostic simulator is presented by Tahmasebi *et al.* [41], where a single PHANToM Premium 1.5A haptic device is used for force feedback. The system can be seen in Figure 2.4(a) The system can load a variety of patient specific datasets, but the US scanning is simply a re-slice of a 3D US volume. Other imaging modalities that are loaded allow the user to view the patient being scanned from different poses, and can adjust the transparency of a virtual 3D view to see varying levels of details (skin, bone, fat, etc). The virtual model loaded is haptically rendered as a solid surface and applies a constant force to the user.

An US-guided needle puncture simulator for interventional radiology is presented by Vidal *et al.* [3], and is shown in Figure 2.4(d). The system is used for training for biopsy operations. Two PHANToM Omni haptic devices are used - one representing an US probe and the other the biopsy needle. The user manipulates the haptic devices to interact with a virtual environment specified by a patient CT volume. As in [6], the force feedback is provided by a proxy-based approach. The PHANToM devices used in the system have three active DOFs, therefore the force is only applied at the end-effector (tip) of the device. This would allow the user to rotate the device freely even when movement is constrained to an input path. US images are simulated from the CT volume using a multi-stage image pipeline which combines a simple shadow mask from the CT volume and an off-line computed speckle pattern to simulate an US image. Stereoscopic vision goggles were used to provide additional depth to the image displayed on a computer monitor.

F.P. Villard *et al.* [42] present a simulator for percutaneous transhepatic cholangiography, a procedure which involves inserting a needle into the bile ducts under fluoroscopic guidance. The system, shown in Figure 2.4(b), uses one PHANToM Omni haptic device to provide force feedback, and a combination of PC monitor, mirrored surface and stereoscopic goggles for visual feedback. The work focuses on simulating the motion imparted on organs by a patient breathing, using a 3D ChainMail algorithm which connects small regions of tissues together like a chain. The rib motion is performed assuming ribs are rigid objects with two rotational DOFs. The angles of rotation vary sinusoidally between their minimum and maximum. To provide the fluoroscopic guidance, the system simulates fluoro images through a GPU-based ray tracing algorithm. While the work in this thesis does not incorporate breathing, the model presented by Villard is a form of deformation, which this thesis incorporates.

A group from the Chinese University of Hong Kong [7] presents an example of a training system for US-guided needle insertion through a game environment. The system is split up into 3 parts: seeking a target with an ultrasound probe and then needle, and finally using real data to perform an insertion. It is not said what type of haptic device is used, or how many, but from the paper it is assumed only one device is used, and the scanning/needle insertion is completed separately. To simulate US images, the group uses two approaches: for simple target tracking (a sphere in a homogeneous volume) images are simulated from the slice of the volume, while during the final stage of the system when real data is used, a 3D US volume is used and re-sliced. The US images are deformed using a mass-spring model, where the needle is modeled as a rigid object represented by multiple nodes. Each of the needle nodes has a radius within which it can affect nodes of the tissue model. If a node in the tissue

mass-spring structure is within this radius, a force is imparted on it which propagates through to adjacent nodes. This deformation is only applied directly to the re-sliced US images from the volume. This system incorporates both a deformation model and US slicing, however the US images come from a re-slice of a 3D US volume for patient-specific cases.

A group from the National University of Singapore [5] discusses what is probably the most intricate VR simulation system of those reviewed here, and the components can be seen in Figure 2.4(c). The group presents a vertebroplasty simulator in the form of a computer game-like environment, and uses a multitude of haptic devices to provide force feedback. To provide force feedback for needle insertion, a customized Delta Haptic device with six active DOFs and high stiffness (to simulate the forces needed for insertion into bone) is used, along with an Immersion CyberGlove and CyberGrasp to provide accurate tracking of the hand and force feedback to the fingers during injection of Poly(methyl methacrylate) into the bone. A workbench with a monitor display being projected onto a semitransparent mirror is used as the method of display, with the virtual scene being simply an outline of a body with a spine model inside it. The force feedback comes solely from the interaction between the needle tip and the bone. Bones are modeled as finite element beams with linear elastic material properties.

In work specifically related to the procedure being simulated in this thesis, Kulcsar *et al.* [43] discuss the different feelings that a clinician would experience during the spinal anesthesia procedure. They use an “expert perception-based approach” to select the stiffness values for the haptic rendering of the different tissues. This method relies on multiple experts setting a slider for a specific feeling during the

spinal anesthesia procedure, such as cutting through skin or the loss of resistance felt after pushing through the ligamentum flavum, which would affect the the stiffness of the feedback. The continuous scale was simply from “too hard” to “too soft”.

From a review of VR systems, it is obvious the main features that they provide. Various virtual patients can be loaded into the systems and explored by the users. This patient data, commonly in the form of a CT volume, can also be used as the basis for different force models which generate output to haptic devices which provide force feedback to the users. Users can view the patient data being explored on a computer monitor, which typically provides additional sources of visual feedback, such as auxiliary views of the patient.

However, there are also some disadvantages with purely VR simulators, particularly when it comes to the lack of a physical surface near the operation site for users to spatially orient themselves with. Some systems attempt to address this by using stereoscopic goggles to provide depth to a computer-generated image, but the image is still purely virtual and the depth can be unrealistic. In the next section a review of mannequin-based training systems that provide a more physical target to trainees will be discussed.

2.2.2 Mannequin Based

Hishikawa *et al.* [44] study the effects of mannequin training for tube thoracostomy on the confidence of medical students. A mannequin and porcine tissue are used, the porcine tissue representing prior training methods students used. The performance is measured based on completion of tasks, total elapsed time and a score assigned from reviewers. Two groups of students are compared, one group using the mannequin

simulator and the other using the porcine tissue. They reported that the group using the simulator felt significantly more confident to perform the procedure on a patient, compared to the group not trained on a simulator, while the actual performance during training was not statistically different.

An example of a mannequin-based simulator for US-guided needle insertion procedures is presented by Magee *et al.* [9]. The group uses a gel-based mannequin for their work, and a dummy US probe and a biopsy needle. The mannequin is scanned using a 3D scanner and its surface is manually segmented to obtain an outline in 3D. A patient 3D CT volume is also taken and manually segmented. The two outlines are then registered using the Iterative Closest Point Algorithm [45] to rigidly align the two volumes, and then non-rigidly deformed using a method based on the RANSAC algorithm [46]. The dummy US probe and biopsy needle are tracked using 3D motion sensors, which allow their positions to be known relative to the mannequin/CT volume. The group simulates an US image from the CT volume of the patient currently loaded. The needle is displayed in the US image by inserting a high Hounsfield value object into the CT volume at the calculated needle position. The only form of force feedback provided to the user during needle insertion is the material properties of the mannequin.

An example of a system that attempts to incorporate a mannequin into a VR training environment is discussed by Gorman *et al.* [10]. They present a simple Lumbar puncture simulator which uses a PHANTom Premium haptic device with an attached needle that runs through the length of a mannequin. The force is calculated using multiple haptic bounding boxes for each tissue type. A VRML model of the spine is loaded into the system as a very stiff object. Besides the mannequin, a

PC displays the internal scene albeit showing only the spine model and approaching needle.

A slightly more detailed simulator that combines VR and mannequins is presented by Ra *et al.* [8]. This system uses one haptic device to simulate the biopsy needle insertion. The mannequin is used to overcome the shortcomings of a haptic device that only has three active DOFs by physically constraining the needle to move only along the path created by a hole in the mannequin. A patient's CT volume is used as the input to the system, and also the basis for the force model. A lookup table is created based on the ranges of Hounsfield units for varying tissue types in the CT model, and the value of the stiffness term k in Hooke's Law is modified based on the CT value at the needle tip voxel. The system displays on a computer monitor a CT slice of the area on which the procedure is being performed, and a rigid model of a needle shown. An "after action" report is generated upon completion of a procedure, with information such as time to target, the trajectory followed, and the error distance from the target.

From the review of mannequin-based training systems, it is clear that most systems do not use a mannequin to its fullest potential. Most systems use the mannequin as a stand-in, or to physically restrict a needle to a certain insertion direction. The only work that reports a different use of a mannequin is the one by Magee *et al.* [9], where they warp a CT volume to fit the shape of the mannequin. A mannequin's usefulness can be extended even further. In addition to providing a physical object for trainees to orient themselves with, a mannequin provides a solid object that can be tracked in space and integrated more fully into a training system, something that is not explored often, and something that this work attempts to address.

2.3 Summary

This chapter set out to provide a solid background on the topics that will be discussed in this thesis, including information on medical imaging modalities, needle insertion models, and augmented reality. A review of current medical training systems in the literature was presented, broken up into those systems that were purely VR, and those that involved a mannequin. From the discussion on the two types of systems came the conclusion that mannequins have not been used to their fullest potential in training simulators, and combining the two types of systems would enable a mannequin to be more involved in the training environment, and help offset the drawbacks of a purely virtual system.

Chapter 3

System Overview and Description

3.1 Overview

The basic premise of the proposed system was presented in Chapter 1: we aim to create a system which will simulate the spinal anesthesia procedure and allow a trainee to perform a virtual operation on a mannequin that has been registered to patient CT data. The mannequin will provide the user with spatial orientation while navigating the virtual patient, and the registered CT data set can be used to overlay a cutaway of the volume over the video feed of the mannequin.

The discussion on training simulators in Chapter 2 outlined some drawbacks to both VR and mannequin based systems, which the proposed system attempts to address. To provide a user with more spatial orientation and physical visual clues, a mannequin is used in addition to the functionality generally found in VR systems. By registering the mannequin to a patient specific data set and tracking it in 3D space, the system also solves a major issue with mannequin-based systems: integrating the mannequin more thoroughly into the system. For the proposed system, the tracking

and registering is used to create an augmented reality overlay on the computer monitor. This system attempts to counteract the drawbacks of both types of simulation systems by utilizing the benefits of both to their fullest.

An overview of the system can be seen in Figure 3.1(b). The MicronTracker2 camera is on the left looking towards the mannequin in the centre. The mannequin has been registered to a patient specific CT volume. The camera tracks the markers on the mannequin, as well as the dummy US probe, and sends the tracking information back to the PC to compute the augmented reality overlay. In front of the mannequin is the PHANToM haptic device with attached needle guide. The haptic device interacts with an FEM based on the patient CT volume to generate forces to the user, and deformations to the volume. To the right of the mannequin is the computer monitor which displays the GUI. Figure 3.1(a) shows the system being operated by a user. The user can be seen holding the dummy US probe and haptic device. A schematic of the system, and the data flow between components, can be seen in Figure 3.2. The MicronTracker2 camera sends the video feed, and the tracking information from the mannequin and dummy probe, to the PC. The PHANToM haptic device sends the position and orientation info of the end-effector to the PC, and receives force data calculated from the tissue model.

3.2 Components

The components of the system will be discussed briefly here, with more details provided in the following chapters.

(a) The developed system in use.

(b) The system components: MicronTracker2 Camera, mannequin, PHANToM, monitor.

Figure 3.1: Overview of the developed system.

3.2.1 Camera System

Since we desire to overlay a CT volume on a live video, we need a camera system to provide a video feed. For other functions of the program, we also need to be able

Figure 3.2: System schematics showing the data flow between components. MicronTracker2 sends video feed and tracking info to PC, which receives position info from PHANTOM and returns force data.

to track markers on a dummy US probe and mannequin. To satisfy both of these requirements, we use an optical tracking camera system - a Claron Technologies MicronTracker2 system (Figure 3.3, left). The MicronTracker2 provides tracking capabilities by tracking black and white markers optically, as well as the ability to capture images recorded by either of its 2 cameras in order to display a video feed. The tracking system has an accuracy of 0.153mm RMS and provides various functions useful for the augmented reality overlay, such as 3D to 2D projection.

3.2.2 Mannequin

In order to provide a trainee with better spatial orientation at the operating site we use a mannequin of a human torso, sourced through Pacific Labs Sawbones Inc. (Figure 3.3, right). The mannequin has a foam interior and plastic exterior representing skin. It has a cutout section that may be removed to place a model of the lumbar spine into the body. A transparent cover is available in addition to an opaque cover. The mannequin also provides a physical object onto which markers can be placed, and then tracked via the MicronTracker2 camera system. This is used in order to register

Figure 3.3: Two components of the system. *Left:* The MicronTracker2 camera system
Right: The torso mannequin from Sawbones, Inc.

the mannequin with a 3D CT volume of a patient, thereby allowing the mannequin to act as a replacement for any patient. The combination of registration with a patient CT and tracking of the mannequin in 3D allows the patient CT volume to be overlaid onto a live video containing the mannequin, which the work in this thesis takes advantage of.

3.2.3 Haptic Device

The force feedback necessary to create the feel of a needle insertion is provided through the use of a Sensable Technologies PHANTOM Premium 1.5A haptic device with three degrees of active feedback (force in the x, y, z translational directions), and additional three degrees of passive feedback via yaw, pitch and roll rotations. The PHANTOM has positional accuracy of 0.03mm and can provide maximum constant force of 1.5N with peak limit force of 8.5N. The Premium 1.5A is controlled through Sensable's OpenHaptics 2.0 architecture, which provides means to create the high priority haptics loop and necessary access functions to send and receive data to/from the haptic

Figure 3.4: The retractable needle guide attachment. *Left:* The needle is at full length just prior to insertion. *Right:* The guide retracts up the stylus as the virtual needle is inserted into body.

device.

The haptic device interacts with the tissue FEM to generate the deformations provided to the CT volume, as well as to generate the forces sent back to the user. Based on recommendations from specialists, a retractable needle and guide is attached to the end of the haptic device in order to provide a physical reference point for the virtual needle. As shown in Figure 3.4, the attached needle is the same length as the virtual needle, and when the guide is pressed up against the mannequin, it will retract along the stylus of the haptic device, allowing all the force generation to come from the haptic device and not the physical needle entering the mannequin. The needle

guide is not currently calibrated to have its tip aligned perfectly with the virtual tip; the guide tip is manually aligned to be as close to the virtual tip as possible.

3.2.4 Finite Element Model

In order to create image deformations as well as force feedback, a 3D, static linear elastic FEM is registered to the mannequin by bringing the haptic device tip point in contact with the markers placed on the mannequin. An FEM allows for easy calculation of forces and displacements, and a static linear model provides satisfactory realism while maintaining low computation requirements.

In order to appropriately displace nodes in the FEM mesh, when the needle enters a new mesh element, the closest node to the needle is found and displaced. The details of the FEM design and an explanation of other methods of mesh deformation are provided in Section 4.3.

3.2.5 Ultrasound Simulation

Since the system is simulating an US guided needle insertion, having US images is a necessity. However, the patient specific data used as an input to the system may only contain a CT volume and no US volume. In this case, it is necessary to create images that simulate those of a real US scan. An example of a simulated US image can be seen in Figure 3.5. The CT image is a slice of the lumbar region of the spine. As is discussed by Shams *et al.* [25], it is possible to correlate the Hounsfield values of a CT image with the intensities of an US image. This fact allows an US image to be simulated from a CT image, following an algorithm based on the properties of acoustic impedance, as described in detail in Section 5.5.

Figure 3.5: A CT image of lumbar spine *left* with simulated US image *right*.

Figure 3.6: The data flow in program. Turquoise boxes are program functions that input/output data. The type of data is described on the arrow.

3.3 Program Flow and Description

The flow of data between functions in the program is visualized in Figure 3.6. The MicronTracker2 camera sends the video feed frames, as well as the positions of the markers, to the visual tracking routine which calculates the position and orientation

Figure 3.7: The program loops with data flow between steps.

of the US probe and mannequin. Meanwhile, the PHANTOM device sends the position and orientation of the end effector to the FEM routine, which returns the forces calculated by the tissue model. The US simulation routine takes the US probe information as well as the virtual needle tip and mesh deformations calculated from the FEM routine, and simulates an US image from the CT volume data. The video feed frames, as well as the mannequin position is send from the visual tracking routine to the CT overlay routine. The augmented overlay is computed in the CT overlay routine and sent to the display, along with the simulated US image and CT image. Black arrows show data flow, with the type of data being passed written beside them.

Algorithmically, the program contains two main loops, one for the graphical display and one for the haptic feedback. Each loop has a vastly different rate requirement, and is the reason the functions of the system must be split up. The graphical loop should run at a rate above 20 FPS, which is the rate required for the eye to not notice lag in visual updates. The haptic loop, on the other hand, must run at a rate 2 magnitudes faster - at least 1000 Hz. I will refer to this rate as the haptic constraint in the future. At rates much lower than this, the system may become unstable or the force feedback update can become coarsely quantized which can be felt by the user and is not desired. This is because the update rate of forces to the haptic device must be high enough to maintain a relation between the damping of the system and the stiffness that is being created [47]. It is therefore necessary to limit calculations in the haptic loop to essential haptic functions as opposed computationally expensive image related functions. The details of the haptic and GUI loops, including the access between the loops and outside devices are shown in Figure 3.6.

The system software has been written in C++ under Microsoft Windows using the Microsoft Visual Studio 2005 compiler. Software libraries used included the Visual Toolkit (VTK) [13], the Insight Toolkit (ITK) [14], Sensable's OpenHaptics 2.0 architecture for driving input/output to the haptic device, the MicronTracker2 SDK from Claron Technologies to receive information from the MicronTracker2 camera, and QT [15] to create the graphical user interface (GUI). MATLAB was also used for some pre-analysis, such as to calculate the stiffness matrix for the FEM.

Using the object oriented approach of C++, different functionalities of the system are controlled by individual classes which communicate with each other. The main classes in the program can be seen in Figure 3.8, and are explained here:

Figure 3.8: The classes in the program.

- *mainwindow*: Controls the main GUI features such as menus, display widgets and starting and stopping of the system. There are three main widgets in the main window, one large display for the video feed from the camera and overlay image, and two smaller displays to show the current CT slice and simulated US image. The *mainwindow* class controls the main graphical loop.
- *displayWidget*: Controls access to the MicronTracker2 camera, receiving the image data from the camera as well as tracking information (3D positional and rotational data) for the markers. The widget calculates all necessary transforms to display the wire-frame outline of the slice through the CT volume from the US probe, and the virtual needle position. The CT overlay is created in this class, and blended with the background image from the MicronTracker2 camera.

- *USsliceWidget*: Simulates an US image from a CT image. Takes the x, y, z 3D position and $\theta_x, \theta_y, \theta_z$ angles of rotation of the probe as inputs and re-slices the CT volume to extract a slice which is used in the simulation.
- *PHANTOM*: Controls input and output to the haptic device, reads positional/orientation data into the program and writes force data back to the device. All the calculations necessary for the FEM and calculations for transverse needle forces are contained in this class. The high priority haptic loop runs within this class.
- *SliceDispWidg*: Displays the auxiliary view of the CT slice containing crosshairs at needle tip and needle vector back to base. Also displays the CT slice from the current probe position, and the simulated US image.
- *USsim.cu* and *DispWidg.cu*: Contains the CUDA code for accelerating calculations on the GPU.

3.4 Summary

This chapter has briefly outlined the system, and its components. It further provided details on the interaction among the components, the operation of the program and the data flow between program functions.

Chapter 4

Tissue Deformation and Force Feedback

4.1 Overview

There are multiple models available that can be used to model deformations in tissue: mass-spring based, surface, and finite element models are a few of these options. A mass-spring model connects many different mass nodes with springs that sometimes cross each other, leading to complicated calculations for the node parameters. A surface model would be useful if deformations were only required to be viewed from outside the volume (looking at the surface), which is not the case for our system, as the deformations we are interested in occur in the interior of the volume. In light of this, the best choice for our system is an FEM.

The second decision that must be made when deciding on a deformation model is the characteristics of said model. Do we wish to use a dynamic model, or a static one? Linear or non-linear? A dynamic model includes mass and damping terms to model

Figure 4.1: Physical needle guide for modeled needle.

the effects of inertia and energy dissipation, while a static model assumes that these effects are negligible. Since skin and soft tissue are dominated by elastic responses over small deformations, the small loss in realism caused by a static model is worth the trade-off in place of a more computationally expensive dynamic model [48]. The choice between a linear or non-linear model again comes down to how realistic the desired deformations are, and if the model can handle the computations required. Soft tissue is, by nature, non-linear so the most realistic deformations would need a non-linear model. Over small strains and deformations, however, this non-linearity can be approximated well by a linear model [49]. Moreover, since the deformations involved with a spinal needle insertion tend to be very small (as discussed in Section 2.1.3 and [36]), a linear model would be satisfactory for estimating deformations. Therefore, the chosen deformation model is a static linear elastic FEM.

The virtual needle modeled in the system is a rigid, symmetrical, 24-gauge needle commonly used in spinal anesthesia procedures. Since the needle is rigid and non-bevel tipped, deflection within the tissue during spinal insertion is deemed negligible (0.5mm-1mm at 60mm insertion depth [36]) and a needle bending model was deemed

Figure 4.2: *Left:* Force $\vec{f}$ acting on the node at $\vec{x}$ of the mesh, prior to calculation of displacements. *Right:* The mesh after displacement, with displacement $\vec{u}$ resulting from $\vec{f}$ being applied to the node at $\vec{x}$.

too computationally expensive [29]. The physical needle used as a guide can be seen in Figure 4.1.

4.2 Finite Element Construction

An FEM is a discretization of an elastic object, broken down into a mesh. The mesh is a 2 or 3 dimensional grid (depending on the type of FEM) of nodes which define the object. The nodes are interconnected to form a mesh of polyhedra - commonly triangular in a 2D mesh, and tetrahedral in a 3D mesh [50]. The continuous deformations through the object can be broken down into the discrete deformations through each element. With a sufficiently fine mesh, these discrete deformations can approximate the continuous deformations experienced in a real elastic object. An example of the mesh deformations $\vec{u}$ resulting from a force $\vec{f}$ can be seen in Figure 4.2.

Let us define the 3D elastic object as $\mathbf{S}$, containing the nodes $\vec{x} = (x, y, z)^T$, which are deformed by the displacement vector $\vec{u}(\vec{x}) = (u_x, u_y, u_z)^T$ when under external force on its boundary δ . The strain in the object is written [51] as

$$E_{\text{strain}} = \frac{1}{2} \int_{\mathbf{S}} \vec{\epsilon}^T \vec{\sigma} dx$$

where $\vec{\epsilon}$ and $\vec{\sigma}$ are the engineering strain and stress vectors, respectively. The strain vector $\vec{\epsilon} = (\epsilon_x, \epsilon_y, \epsilon_z, \gamma_{xy}, \gamma_{xz}, \gamma_{yz})^T$ and the the displacement vectors are related through $\vec{\epsilon} = \mathbf{B}\vec{u}$, where

$$\mathbf{B} = \begin{pmatrix} \frac{\delta}{\delta x} & 0 & 0 \\ 0 & \frac{\delta}{\delta y} & 0 \\ 0 & 0 & \frac{\delta}{\delta z} \\ \frac{\delta}{\delta y} & \frac{\delta}{\delta x} & 0 \\ \frac{\delta}{\delta z} & 0 & \frac{\delta}{\delta x} \\ 0 & \frac{\delta}{\delta z} & \frac{\delta}{\delta y} \end{pmatrix}$$

The engineering stress is related to strain through $\vec{\sigma} = \mathbf{C}\vec{\epsilon}$, where $\mathbf{C}$ is a square “material matrix” [51] obtained as

$$\mathbf{C} = \begin{pmatrix} \lambda + 2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda + 2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda + 2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{pmatrix}$$

and λ and μ are material constants that depend on the Young’s Modulus (E) and

poisson ratio (ν) of the material according to:

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

$$\mu = \frac{E}{2(1+\nu)}$$

The properties of the different tissues in the model were selected from values found in the literature [52], [53], [54]. Soft tissue was set with $E = 20\text{kPa}$, $\mu = .3$, and ligament with $E = 170\text{kPa}$, $\mu = .3$.

The total energy stored in the model E , is a function of displacement $\vec{u}$, according to:

$$E(\vec{u}) = E_{\text{strain}}(\vec{u}) - E_{\text{int}}(\vec{u}) - E_{\text{ext}}(\vec{u}) \quad (4.1)$$

$$= \frac{1}{2} \int_{\mathbf{S}} \vec{u}^T \mathbf{B}^T \mathbf{C} \mathbf{B} \vec{u} dx - \int_{\mathbf{S}} \vec{f}^T \vec{u} dx - \int_{\delta} \vec{g}^T \vec{u} da \quad (4.2)$$

where E_{int} and E_{ext} represent the work done over the displacement $\vec{u}$ by the internal and external forces $\vec{f}$ and $\vec{g}$, respectively.

Now that we have set up the energy involved in the elastic deformation of the object, we must apply the discretization needed in order to perform finite element analysis. As discussed earlier, we discretize the 3D object into tetrahedral elements ($\mathbf{S}^e$), cornered by the nodes N_i , where $\vec{x}_i = (x_i, y_i, z_i)^T \in N$. The deformation vector $\vec{u}$ is also discretized in this matter. For each element $\mathbf{S}^e$ of the discretized object, the deformation acting on the four nodes comprising the element is

$$\vec{u}^e = (\vec{u}_1^{eT}, \vec{u}_2^{eT}, \vec{u}_3^{eT}, \vec{u}_4^{eT})^T$$

This gives us the deformation at any of the 4 nodal points of an element, but in order to determine the displacement at any point x within the element, we must use linear interpolation. This is done by taking the displacement at each $\vec{u}_i \in \vec{u}^e$ and weighting it based on the distance from the associated node to the location $\vec{x}$. In other words:

$$\vec{u}(\vec{x}) = \sum_{i=1}^4 L_i \vec{u}_i^e$$

where L_i is the natural coordinate of the element, and is related to the global coordinates through

$$\begin{pmatrix} 1 \\ x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{pmatrix} \begin{pmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \end{pmatrix}$$

where x_i, y_i, z_i are the coordinates of each of the nodes of the element. Inverting the above equality solves for L_i .

This allows the matrix $\mathbf{B}$ to be discretized to the elemental matrix $\mathbf{B}^e$, and the strain energy function becomes

$$E_{\text{strain}} = \frac{1}{2} \sum_e \int_{\mathbf{S}^e} \vec{u}^{eT} \mathbf{B}^{eT} \mathbf{C} \mathbf{B}^e \vec{u}^e dx$$

A deformed body is at equilibrium when its total potential energy is at a minimum, or when the first derivative of the energy function vanishes. Therefore, wish to find the displacement $\vec{u}$ which causes $\frac{\delta E(\vec{u})}{\delta t} = 0$. This results in the following condition for

all elements, when the first derivative is applied to (4.2) :

$$0 = \int_{S^e} \mathbf{B}^{eT} \mathbf{C} \mathbf{B}^e \vec{u}^e dx - \vec{f}^e$$

where $\vec{f}^e$ is a 12×1 element force vector (4 nodes x 3 dimensions). All the matrices within the integral are constant, and can be pulled out to yield

$$\begin{aligned} 0 &= \mathbf{B}^{eT} \mathbf{C} \mathbf{B}^e \vec{u}^e \int_{S^e} dx - \vec{f}^e \\ \vec{f}^e &= \mathbf{B}^{eT} \mathbf{C} \mathbf{B}^e \vec{u}^e V^e \\ \vec{f}^e &= \mathbf{K}^e \vec{u}^e \end{aligned}$$

Here, V^e is the volume of the element, and $\mathbf{K}^e$ is the stiffness of the element. These element stiffness matrices and force vectors must be combined into a global stiffness matrix for the entire object, and global force vector. This is done through a transfer function ($global(\cdot)$) that takes the element nodes and turns them into global nodes:

$$\begin{aligned} \mathbf{K} &= \sum_e global(\mathbf{K}^e) \\ \vec{f} &= \sum_e global(\vec{f}^e) \end{aligned}$$

This allows us to write the final static, linear elastic FEM equation as

$$\vec{f} = \mathbf{K} \vec{u} \tag{4.3}$$

which can be solved for the displacement vector u through a matrix inversion calculation. The force and displacement vectors are of length $3N \times 1$, where N is the

number of nodes in the system. For any node i , the elements $3i - 2, 3i - 1$ and $3i$ correspond to x, y, z components of the node, respectively.

The size of the volume being meshed is $87.6 \times 96 \times 150.6\text{mm}$. The volume is discretized using a 3D Delaunay tessellation function provided by MATLAB into 11708 elements.

4.3 Mesh deformations

Once we have the determining equation (4.3), we can begin to create deformations on our object by applying external forces. Solving for the displacement vector $\vec{u}$ requires inverting the stiffness matrix $\mathbf{K}$, which requires it to be non-singular. However, the matrix, $\mathbf{K}$ is singular, and any force acting on any node can cause infinite number of displacements. This is because the object is free to move in all directions, and as a result does not have a unique solution when external forces are applied. To remedy this issue, we apply boundary conditions to a number of the nodes - the minimum number of nodes needing to be fixed is equal to the dimension of the object. This is done by setting the columns and rows of $\mathbf{K}$ corresponding to the fixed nodes to 0 except for $\mathbf{K}_{3i,3i}$ where i is the node number being fixed [50]. In order to create these boundary conditions, the nodes of the mesh corresponding to the bottom of the CT volume and those corresponding to bone are set as fixed. This results in a matrix that is not singular, and can be inverted to solve for the displacement vector $\vec{u}$.

Creating an algorithm for deforming the mesh based on the position of the needle tip is another challenge, as the needle will not always travel along a path that exactly strikes nodes. There are a few methods described in [29] that will be discussed in order of increasing complexity.

Figure 4.3: *Left:* Mesh prior to adaptation. *Right:* Mesh post adaptation. Mesh elements i , j , k , l share the node n being displaced

The first method of deformation is to snap nodes to the needle as it passes, or deform the closest node to the needle tip. In order to snap a node to the needle tip, the closest node to the needle tip is found, and then the displacements for that node are set so that it is always attached to the needle, forcibly moving it from its initial location to the needle. This can create the undesirable effect of force/deformation discontinuities when a new node is selected, as each time a node is forced onto the needle shaft it exerts forces on the nodes around it. This can be remedied by only applying the needle tip forces to the closest node instead of forcing the node onto the needle. This can result in slightly less accurate deformation results, but if the mesh is sufficiently fine and the deformations are sufficiently small, the discrepancy may be acceptable.

The second method reduces the force and deformation discontinuities that occur from forcing nodes onto the needle shaft by adapting the FEM mesh around this forced displacement to preserve the forces on the surrounding nodes. To perform this

adaptation, the elements that have the node n_i being displaced as a vertex must be updated with the new position and volume information, which requires changes to the element stiffness matrix $\mathbf{K}^e$ for all affected elements, and by extension the global stiffness matrix. For each affected element, we can let $\mathbf{K}^{e'}$ be the new element stiffness matrix calculated after the node is displaced, and $\Delta\mathbf{K}^e = \mathbf{K}^{e'} - \mathbf{K}^e$ to be the change from prior to displacement to after. According to [29], the new global stiffness matrix $\mathbf{K}'$ can be calculated as

$$\mathbf{K}' = \mathbf{K} + \mathbf{V}\Delta\mathbf{K}^e\mathbf{V}^T$$

where

$$\mathbf{V} = [\mathbf{V}_{kl}], k = 1, \dots, 2N, l = 1, \dots, 6$$

is a $2N \times 6$ selection matrix defined as

$$\mathbf{V}_{kl} = \begin{cases} 1 & : k = L_l \\ 0 & : k \neq L_l \end{cases}$$

where L contains the row numbers of the global stiffness matrix $\mathbf{K}$ that correspond to those in $\mathbf{K}^e$. Since re-inverting the entire global matrix every time would be extremely computationally intensive, the Sherman-Morrison-Woodbury formula [55] can be used to solve for the inversion with only having to invert a 6×6 matrix instead of the entire $3N \times 3N$ matrix:

$$(\mathbf{K} + \mathbf{V}\Delta\mathbf{K}^e\mathbf{V}^T)^{-1} = \mathbf{K}^{-1} - \mathbf{K}^{-1}\mathbf{V}\Delta\mathbf{K}^e(\mathbf{I} + \mathbf{V}^T\mathbf{K}^{-1}\mathbf{V}\Delta\mathbf{K}^e)^{-1}\mathbf{V}^T\mathbf{K}^{-1}$$

with $(\mathbf{I} + \mathbf{V}^T\mathbf{K}^{-1}\mathbf{V}\Delta\mathbf{K}^e)$ being the matrix that needs to be inverted instead of $\mathbf{K}'$. The drawback to this method is that the calculation of the inverse for every element

during adaption is not fast enough to meet the requirements of the haptic constraints. An example of this can be seen in Figure 4.3, where the node n is displaced. The four elements i, j, k, l all share n as a vertex, and therefore the element stiffness matrices $\mathbf{K}^e, e = i, j, k, l$ must be updated to reflect the change in the position of n and their new volumes.

The final mesh deformation method is to subdivide the FEM mesh into finer sections closer to the needle as it moves through the existing mesh. This involves inserting new nodes into the mesh and recalculating the global stiffness matrix for every re-mesh, which is very computationally expensive and impractical for a system with haptic force feedback. However, mesh subdivision can create the most accurate deformations, as the space that is most desirable to look at during deformation, the volume around the needle, is constantly being discretized to a finer degree.

Based on our haptic constraints, the simplest method, i.e. deforming the closest node to the needle tip, was chosen for the simulator. In order to determine the closest node to the needle tip, the following algorithm is followed. A matrix is formed prior to running the system which stores the tetrahedral element in the FEM that each voxel in the CT volume belongs to. This allows the element that the tip is currently in to be easily determined. Once the element is found, the four vertices of the tetrahedron are found from a list of elements and their associated nodes. The node closest to the location of the needle tip is then calculated based on weighting of the tetrahedral vertices (as done in Section 4.2). The appropriate indices of the displacement vector $\vec{u}$ for this node are then updated with the amount of displacement imparted by the needle tip. In order to eliminate force discontinuities, the needle displacement applied to the node is calculated as the displacement from the location of the needle where

Figure 4.4: Needle inserted into mesh at element e , with closest vertex n being displaced

the closest node was found. The forces from this displacement are then calculated and sent back to the haptic device, based on equation (4.3), with the tip force $\vec{f}_{tip}$ being equal to $\vec{f}$ from (4.3). If a force discontinuity is observed, the difference between the next force and current force is smoothed by applying the difference over multiple haptic iterations. This is shown in Figure 4.4, where the needle has entered element e of the mesh, with the node n determined as the closest vertex of the element being displaced.

Once a certain distance threshold is reached or another element is entered, the node being displaced is updated following the same procedure as before. All previously displaced nodes retain their displacements, and “slip” along the shaft of the

needle, providing friction to the needle based on the tissue type. The amount of friction depends on how far into the tissue the needle has traveled, and can be calculated in magnitude as

$$f_{fric} = \sum_t d_t \mu_t$$

where t are the tissue types, d is the length of needle inside the tissue, and μ_t is the force distribution per metre for the tissue type. Friction forces for tissues were based on values discussed in the literature ([49]). The total force experienced at the base of the needle by the user inserting along an insertion vector $\vec{v}_i$ is

$$\vec{f}_n = \vec{f}_{tip} - f_{fric} \vec{v}_i \quad (4.4)$$

An important consideration when calculating forces is how to “switch” the direction in which the forces are applied when the user begins retracting the needle. The static model is satisfactory for producing forces in one displacement direction, however if the magnitude of the displacement becomes 0 or negative, the forces will still be applied in the original direction, only smaller than previously, as if a finger pushing on a solid elastic material is being withdrawn. In order to properly account for the fact that the forces will be felt in the opposite direction once the trajectory of the needle is reversed, we need to detect this change and negate the forces applied appropriately. In a perfect world, this would be easily done by detecting when the velocity of the virtual needle reaches 0 and becomes negative. The reliability of this detection in reality depends on the accuracy of the velocity measurement from the haptic device. The PHANToM provides positional measurement with accuracy 0.03 mm. However considering the sampling rate of 1KHz, the velocity measurement

Figure 4.5: State machine for latching between force directions. Machine begins at state A when needle is inserted into the body

accuracy reduces to:

$$\Delta v = \frac{\Delta x}{\Delta t} = \frac{0.03\text{mm}}{0.001\text{sec}} = 30\text{mm/sec}$$

Since typical needle insertions are performed at very low velocities ($< 10\text{mm/sec}$), this large resolution in measurement is not acceptable if the velocity is used directly to detect a certain value. There are several ways to compensate for this lack of resolution, the first being to use a simple average over a large enough number of samples. This can still create issues around the switching point, as the average can still change quickly between the two sides, creating instability in the system. In order to eliminate this, there needs to be a latching effect involved which will hold the

direction as a state, and not allow it to switch until the velocity stays in the other direction for a certain period of time.

The proposed state machine can be seen in Figure 4.5. The state machine begins at State A, for the needle being inserted into a body. If the direction of insertion continues to be positive (needle going into the body), the force remains opposing the insertion vector. If the velocity becomes negative (needle retracting from body) at any time, the state machine moves to State B, where the force direction remains the same. The machine remains in State B as long as the velocity is still negative, and will return to State A if the velocity becomes positive again. If a latch timer is exceeded in State B (latch time of 100ms was used), the machine moves to State C, where the force from the movement of the needle during the latch time is calculated and applied by adding a small amount of the total force differential every haptic loop iteration until the needle force reaches the proper force after the latching. The machine then move to State D, which is the opposite of State A - the force is applied in the direction of the insertion vector as long as the needle velocity remains negative. The second half of the state machine is the same as the first half, except with forces and velocities reversed.

4.4 Transverse Needle Resistance

If the base of a needle inserted into a body is displaced such that the base moves off the insertion vector, the needle will exert a force opposing the motion to pull it back towards the proper vector, as it acts like a bending rod, with fixed point being the point of insertion. This can be modeled by assuming the needle undergoes bending as a thin cylinder, and using a standard beam deflection model to calculate the forces

from bending.

Modeling the needle as a long, thin cylinder fixed at one end has the following formula [56] for deflection δ_{end} at the end of the beam:

$$\delta_{end} = \frac{f_{end} l_{out}^3}{3EI_n}$$

where f_{end} is the force applied to the end of the beam, l_{out} is the length of the beam (the portion of the needle outside the body), and E_n and I_n are the Young's modulus and moment of inertia for the beam. For the needle as a thin cylinder, the moment of inertia is

$$I_n = m_n l_n / 3$$

where $m_n = \rho l_n r_n^2$ is the mass of the needle, l_n is the entire length of the needle, ρ is the density of the needle, and r_n is the radius of the needle. This can be rearranged to give

$$f_{end} = \frac{3EI_n \delta_{end}}{l_{out}^3} \quad (4.5)$$

the force as a function of the displacement of the needle base (δ_{end}) and length of the needle outside the body (l_{out}). This gives a force function which provides higher force for larger needle deflections when the needle is further inside the body.

The magnitude of the needle base deflection δ_{end} is found by calculating the shortest distance from the needle base b_n , as reported by the haptic device, to the insertion vector $\vec{v}_i$, ensuring that the distance is the length of the vector perpendicular between b_n and $\vec{v}_i$. The direction of this shortest distance is $\vec{v}_b$, and the direction in which the force f_{end} is applied is $\vec{v}_{end} = -\vec{v}_b$.

The total force applied to the user will be a combination of the force $\vec{f}_n$ created

Figure 4.6: Needle inserted into body and bent. Original insertion vector $\vec{v}_i$, current needle base b_n , closest point along original insertion vector to current base p_b direction of bend $\vec{v}_b$ with end deflection δ_{end} , end force from bending f_{end} in direction $\vec{v}_{end}$, forces from friction and tip $\vec{f}_{tip}$, f_{fric}

by the needle (4.4) and the force $\vec{f}_{end}$ created by transverse displacement of the base (4.5):

$$\vec{f}_{tot} = \vec{f}_n + f_{end}\vec{v}_{end} \quad (4.6)$$

Figure 4.6 shows an example of the forces on a needle in a body under deflection. The needle with base b_n is being deflected off of its original insertion vector v_i by the amount v_b , resulting in the force f_{end} in direction $\vec{v}_{end}$. The forces from internal friction f_{fric} along the needle and the force from the tip $\vec{f}_{tip}$ are also shown.

4.5 Summary

In this chapter, the thought process behind the selection of the deformation model used in the system was outlined, starting with the choice of model and moving onto the specifics of whether the FEM would be static or dynamic, linear or non-linear. Among the options, a static, linear elastic FEM was chosen as the best option as it allows for the deformations and forces throughout the body to be easily calculated, and provides a balance between speed and accuracy.

The creation procedure for a three dimensional FEM was then detailed, showing how the determining equation (4.3) is reached. A description of the process followed during needle insertion was also explained, with the axial forces generated along the needle from insertion and the transverse forces at the needle base calculated.

Chapter 5

Visualization

5.1 Overview

In order to provide the users of the system with maximum information to assist them in guidance, we track a mannequin in 3D with a MicronTracker camera (hereafter referred to as a camera) and overlay a registered patient CT volume onto the video feed of the camera. The mannequin can be used as a replacement for any patient, where the patient CT data is warped to fit its shape. The work in this chapter is an effort to integrate the mannequin more thoroughly into the training system, by tracking it in 3D space and using it as a medium to create an augmented reality visualization by overlaying CT data onto its location in the video feed.

A dummy US probe is also tracked and its position and orientation relative to the mannequin is determined for slicing the CT volume. The process by which these US images are simulated from the CT volume is discussed in this chapter. Simulated US is used in order to allow any kind of patient data to act as input to the system, even in the cases where there is no US volume available.

Figure 5.1: Facet tracking information displayed by MicronTracker2. *Left:* An example of a cross shaped face with four points. *Right:* An example of two L-shaped facets with three points each.

In this thesis, a printed spine model and a torso mannequin are used as substitutions for a real patient in order to provide a solid object to overlay CT data onto. The procedure to create this overlay for each of the substitute objects is discussed in detail. Following the discussion on the augmented reality overlay, the other form of visualization present in the system, US imaging via simulation of US from CT data, is outlined.

5.2 Visual Tracking

The camera system tracks visual markers which contain patterns of 3D points called facets. Each facet is made up of at least three points which must create two unique vectors in length and direction. Common patterns include crosses and L shapes. The longer the vectors of a facet, the more accurately the tracker can locate it. Each visual marker is composed of at least one facet, but may contain multiple facets if it

Figure 5.2: A MicronTracker2 Camera with coordinate frame attached. Looking in the direction of the camera viewpoint, the x direction is to the right, the y direction down, and z direction outwards in the direction the camera is facing

is to be tracked from multiple directions (Figure 5.1).

During tracking, the camera can return the position of the marker, its three angles of rotation in the camera coordinate system in radians or as a 3×3 rotation matrix, and the locations of all the 3D points associated with the facets. All locations returned are in the camera coordinate system, which can be seen in Figure 5.2.

Along with tracking capabilities, the camera system allows the images recorded by either of its cameras to be captured and displayed onscreen, creating a video feedback. Each pixel of the recorded image is calibrated with the 3D camera space, and the MicronTracker SDK contains functions which can be used to project a location in 3D space onto the 2D image. This is useful for performing volume ray-casting as is discussed in the Section 5.4.2. The video feedback is also necessary to create the augmented reality overlay, which is discussed next.

5.3 Printed Spine Model Substitute

A printed spine model, from a 3D printer, based on patient CT was used as a substitute for a patient. The model was attached to the inside of a plastic box with multiple tracking markers placed around the outside, larger cross shaped facets on the sides, and smaller L-shaped facets on the front and back. A CT volume of the spine model was also acquired. The CT dimensions were $58.5\text{mm} \times 52.5\text{mm} \times 179\text{mm}$ with voxel spacing $.314 \times .314 \times .6\text{mm}$. The box and spine model, along with attached markers, can be seen in Figure 5.1. Since the printed spine model contained only the solid printed material and gel, the CT scan is not very detailed and only contains the bone structures.

5.3.1 Overlay Method

We used the Visualization Toolkit (VTK) to perform the image overlay as it contains functions for viewing medical data, and creating virtual scenes. In order to create the appropriate scene, one requires the use of a VTK camera function, whose properties must be matched to those of the camera. It was determined that the only major properties that had to be adjusted were the viewing angle of the camera, as well as the coordinate frame. The camera's viewing angle was computed by measuring its field of view in the horizontal direction. The viewing angle turned out to be 53 degrees. Once the camera properties were set, the CT volume was loaded into the VTK scene, with origin of the volume set at the distance of the box as measured by the camera, and rotation of the volume set as the rotation of the box as measured by the camera. Once this scene was created, a 2D snapshot of it was taken from the VTK camera's viewpoint and passed to the main program where the areas containing

Figure 5.3: A composition of a camera image and a VTK scene snapshot to create an image overlay.

the CT volume were given slight transparency and the areas not containing the volume were given full transparency. Further blending was accomplished by finding the edges of the projected CT volume and making the pixels become more transparent as they approached the boundary between projected image and empty space. Once all calculations were completed, each pixel in the VTK scene snapshot contained a transparency value that allowed the snapshot to be alpha blended with the current image from the camera using the formula

$$C_o = \alpha_a c_a + (1 - \alpha_a) c_b$$

where C_o is the final colour, c_a is the colour in the image to be overlaid (VTK scene snapshot), α_a is the transparency of the overlaid image and c_b is the colour of the underlying image from the camera.

The results of this first overlay attempt can be seen in Figure 5.4. The blending at the edges, image on bottom, slightly reduces the feeling of the image being “in front” of the background, which the top image demonstrates.

Figure 5.4: A comparison of a direct overlay (*Top*) with an overlay using edge blending (*Bottom*).

5.4 Mannequin Based Substitute

The initial printed spine model substitute and VTK scene snapshot composite created a basic and passable image overlay display. It was lacking in certain areas, however. The first issue was the “mannequin” in use. A plastic box with a model attached was not a realistic object to perform training on. Since realism is one of the objectives of the design, this box mannequin would have to be upgraded to an actual representation of a human in the form of a torso mannequin. In addition, the tracking methods used for the box needed to be updated, as the box was a perfect rectangular prism while a mannequin would not be. Finally, the VTK scene compositing technique was not as accurate as desired, creating growing offsets with movement away from the centre of the camera field of view, in addition to edge blending not being an appropriate

Figure 5.5: *Top:* Torso mannequin with spine model shown in the cutout region. CT beads are visible on the right, with MicronTracker markers on left. *Bottom:* A slice of the mannequin CT volume, with bead visible at top left.

illusion of the CT volume being “inside” the target mannequin. These offsets were likely a result of the radial and tangential distortion of the camera which couldn’t be properly accounted for with the VTK camera.

5.4.1 Mannequin

To upgrade the mannequin, a torso mannequin was purchased from Sawbones Inc. The mannequin's interior is foam, with a thin plastic coating to simulate skin. A cutout of the lumbar spine region allows a detailed spine model to be inserted into the mannequin and covered up with an opaque or clear cover. This mannequin was chosen for two reasons: (1) it is an accurate representation of a human torso, and (2) the inserted spine model would show up on a CT image which would be important for creating the augmented reality overlay function. In preparation for the overlay, the mannequin had to be CT scanned. Small (radius 2mm), round CT visible beads were attached to the surface of the mannequin as well as to the spinous processes and transverse processes of the spine model. The mannequin, with spine model inserted into cutout, can be seen in the top image Figure 5.5. The CT beads can be seen on the right side of the mannequin, and markers on the left.

The beads were manually annotated from the CT scan of the mannequin, with the voxel coordinate of each bead recorded. If a bead was on the spine model, the voxel within the bead closest to the bone was chosen, to obtain the location of the bone. For beads on the "skin" of the mannequin, the voxel within the bead furthest from the "skin" was chosen, as tracking markers would be attached to the bead at this location. A sample CT slice of the mannequin showing the beads can be seen on the bottom in Figure 5.5. In preparation for visual tracking, markers were attached to the side of the mannequin that would be in view of the camera. L-shaped markers were used, with the centre of the middle 3D point being placed directly over the CT visible bead.

5.4.2 Overlay Method

The new method of creating a CT overlay on the camera video feed involves two steps. The first is to register the CT volume of the patient to the CT volume of the mannequin, and the second is to perform the volume ray casting necessary for creating the overlaid image. These two steps will be discussed below.

Registration

In order to project the patient CT volume onto the image from the camera, we need to bring the patient CT volume into the space of the camera. To do this we perform two registrations: one simple affine transformation (rigid plus scaling) between patient CT and mannequin CT, and one rigid registration between the mannequin CT and camera coordinate space.

The annotated spine model beads from the mannequin CT volume are recorded, and appropriate landmarks in the patient CT are chosen to align with. These landmarks are chosen to be in the same location as their counterparts on the mannequin spine model - the same spinous process or transverse process.

To solve for the registration between mannequin CT and patient CT we use the following procedure to solve for the translation, rotation and scaling. First, we calculate the mean (centroid) of each of the data sets, and subtract the mean value from each set so that they are centered at the origin. We now have two sets of points, $\mathbf{M}$ for the mannequin and $\mathbf{P}$ for the patient. Therefore, the translation portion of the transformation is simply the difference between the two means, or $\mathbf{T} = \mathbf{M}_o - \mathbf{P}_o$. We wish to find the rotational matrix $\mathbf{R}$ and scaling factor s that minimize in the least squares sense the difference between the mannequin data and patient data after

the patient data is rotated and scaled. Or, in equation form, we wish to find the minimum of

$$||s\mathbf{P}\mathbf{R} - \mathbf{M}||^2$$

Since $\mathbf{R}$ is a rotation matrix it will be unitary, that is to say its transpose equals its inverse, and $\mathbf{R}\mathbf{R}^T = \mathbf{I}$, the 3×3 identity matrix. Expanding the least squares problem and using laws of matrix traces we get the following:

$$\begin{aligned} ||s\mathbf{P}\mathbf{R} - \mathbf{M}||^2 &= \text{trace}((s\mathbf{P}\mathbf{R} - \mathbf{M})^T(s\mathbf{P}\mathbf{R} - \mathbf{M})) \\ &= \text{trace}(s^2\mathbf{R}^T\mathbf{P}^T\mathbf{P}\mathbf{R}) + \text{trace}(\mathbf{M}^T\mathbf{M}) - \text{trace}((s\mathbf{P}\mathbf{R})^T\mathbf{M}) - \text{trace}(\mathbf{M}^T(s\mathbf{P}\mathbf{R})) \\ &= s^2\text{trace}(\mathbf{R}\mathbf{R}^T\mathbf{P}^T\mathbf{P}) + \text{trace}(\mathbf{M}^T\mathbf{M}) - s\text{trace}(\mathbf{R}^T\mathbf{P}^T\mathbf{M}) - s\text{trace}(\mathbf{M}^T\mathbf{P}\mathbf{R}) \\ &= s^2\text{trace}(\mathbf{P}^T\mathbf{P}) + \text{trace}(\mathbf{M}^T\mathbf{M}) - 2s\text{trace}(\mathbf{M}^T\mathbf{P}\mathbf{R}) \end{aligned}$$

Ignoring the scaling factor s for the moment, in order to minimize $||s\mathbf{P}\mathbf{R} - \mathbf{M}||^2$ we need to maximize the function $\text{trace}(\mathbf{M}^T\mathbf{P}\mathbf{R})$. We can determine the best $\mathbf{R}$ that will solve this by performing a Singular Value Decomposition (SVD) on $\mathbf{M}^T\mathbf{P}$:

$$\mathbf{M}^T\mathbf{P} = \mathbf{U}\mathbf{S}\mathbf{V}^T$$

where $\mathbf{U}$ and $\mathbf{V}$ are unitary and $\mathbf{S}$ is a diagonal matrix of singular values. Replacing $\mathbf{M}^T\mathbf{P}$ in the previous equation gives us

$$\begin{aligned}\text{trace}(\mathbf{M}^T \mathbf{P} \mathbf{R}) &= \text{trace}(\mathbf{U} \mathbf{S} \mathbf{V}^T \mathbf{R}) \\ &= \text{trace}(\mathbf{S} \mathbf{V}^T \mathbf{R} \mathbf{U})\end{aligned}$$

Since all of $\mathbf{U}$, $\mathbf{V}$, $\mathbf{R}$ are unitary, the matrix $\mathbf{V}^T \mathbf{R} \mathbf{U}$ will similarly be unitary, and its trace will have maximal value when its values are all contained on the diagonal. For the case of a unitary matrix, this means its maximum trace will occur when it equals the 3×3 identity matrix $\mathbf{I}$. Therefore,

$$\max(\text{trace}(\mathbf{M}^T \mathbf{P} \mathbf{R})) = \text{trace}(\mathbf{S})$$

is achieved by

$$\begin{aligned}\mathbf{V}^T \mathbf{R} \mathbf{U} &= \mathbf{I} \\ \mathbf{R} &= \mathbf{V} \mathbf{U}^T\end{aligned}$$

So we can calculate the desired rotation $\mathbf{R}$ from the SVD of $\mathbf{M}^T \mathbf{P}$. There is one caveat to this process that must be noted carefully. During the SVD process, it is possible for $\mathbf{V}$ and $\mathbf{U}$ to be calculated so that the determinant of $\mathbf{R}$ is -1, which would be a reflection and not a valid rotation (since the determinant of a rotation is always 1). Therefore, a check must be made after the SVD is calculated. If the determinant of $\mathbf{R}$ is -1, we negate the rightmost column of $\mathbf{U}$, and recalculate $\mathbf{R}$. This will guarantee us that the determinant of $\mathbf{R}$ is 1.

Figure 5.6: A registration example for the mannequin CT (red bounding box and blue dots) with the MicronTracker markers (green dots).

Finally, to solve for the scaling factor that minimizes the equation, we return to $s^2 \text{trace}(\mathbf{P}^T \mathbf{P}) + \text{trace}(\mathbf{M}^T \mathbf{M}) - 2s \text{trace}(\mathbf{S})$. To find the minimum of this, we take the derivative with respect to s and set it to 0, resulting in

$$\begin{aligned}
 0 &= 2s \text{trace}(\mathbf{P}^T \mathbf{P}) - 2 \text{trace}(\mathbf{S}) \\
 2s \text{trace}(\mathbf{P}^T \mathbf{P}) &= 2 \text{trace}(\mathbf{S}) \\
 s &= \frac{\text{trace}(\mathbf{S})}{\text{trace}(\mathbf{P}^T \mathbf{P})}
 \end{aligned}$$

This gives a translation $\mathbf{T}$, rotation $\mathbf{R}$, and scaling factor s that best solve the

rigid registration between the mannequin CT and patient CT. This process can also be used to find the registration between the mannequin and camera coordinate space, using the mannequin surface beads $\mathbf{M}_s$ and tracking markers $\mathbf{F}$, solving

$$||\mathbf{M}_s\mathbf{R} - \mathbf{F}||$$

Since the two data sets in this case are of the same scale, we do not need to include a scaling factor. When these two registrations are applied to the bounding box of the patient CT volume, it is properly aligned in camera coordinates with the position and orientation of the mannequin. An example registration between mannequin and camera coordinates can be seen in Figure 5.6. Green points represent the locations of the markers, dark blue points represent the CT visible bead locations on the mannequin surface and the light blue points are the CT beads on the spine model. The bounding box of the patient CT volume is shown in red.

Volume Ray Casting

Once the two CT volumes have been registered with each other and the camera coordinate space, we can utilize the functions of the MicronTracker2 SDK and a volume ray casting method to create the overlay effect.

As was previously mentioned, the SDK contains functions for projecting 3D points onto the 2D image plane seen by the camera. To use the functions, the point must be within the calibrated area of the camera, which generally means it must have a positive z value (be in front of the camera), and be in a position that is viewable by the camera. Since we are projecting points around detected markers, the 3D points to be projected will be within this calibrated area. Passing the 3D point in camera

coordinates will return the (x, y) pixel location on the screen, from top left across and down, representing where the ray from the camera viewpoint to the 3D location intersects the image plane.

For the ray casting method, we calculate the direction of the current ray as

$$V_d = \frac{P_{3D}}{||P_{3D}||}$$

P_{3D} being the 3D location. We begin our ray casting at P_{3D} , and continue along the direction specified by V_d until the current location is no longer inside the bounding box. The ray casting method is a fairly simple one. For each ray, we set the beam colour to be initially $(0,0,0,0)$, representing the RGB colour values and alpha (opacity) value respectively. At each step along the ray direction V_d , the current CT volume value is checked to determine what tissue type is at the voxel. Depending on the type of tissue, a certain colour value $C_{r,g,b}$ and alpha value C_α are selected. The beam colour b is then modified by the formula

$$\begin{pmatrix} b_R \\ b_G \\ b_B \\ b_\alpha \end{pmatrix} = (1 - b_\alpha) * \begin{pmatrix} C_r \\ C_g \\ C_b \\ C_\alpha \end{pmatrix} + b_\alpha b$$

We continue stepping along the ray until either the bounding box is exited, or the opaqueness of the beam becomes too high. In this case a value of b_α greater than 0.95 is used as an early ending flag. Once an ending condition has been met, the image pixel determined from the projection is set to the RGB beam colour values of the beam.

Figure 5.7: An example of the final output of CT overlay procedure.

The final stage of the overlay procedure is to blend the projected image with the background image from the camera. As mentioned in Section 2.1.4, Bichlmeier *et al.* [38] present a three component method for improving depth perception of overlaid images [39]. We currently only use one function, and it is similar to the distance falloff function they discuss. We do not have a full 3D scan of the mannequin with which to determine its curvature, or the angle of incidence. These are two factors that could be implemented for improved image perception.

An oval-shaped blending function is used to blend the edges of the overlaid image with the background. The oval is centered near the centre of the projected 3D boundary box, with a long major axis and narrow minor axis. The falloff function is calculated by finding the length of the vector from oval origin to current pixel, and then finding the ratio between this length and the radius of the oval at the angle of the current pixel (r). This gives a percentage of the radius, and for percentages over

a certain threshold r_t , the function

$$\alpha = \frac{1 - r}{1 - r_t}$$

is used to find the transparency of the pixel. Any ratio > 1 would be outside the oval and be completely transparent, while any value of $r < r_t$ would only be partially transparent.

The final product of the augmented reality overlay procedure can be seen in Figure 5.8 and the pseudo-code for the volume ray casting procedure can be seen below:

```
for p = 1 to end
    [Px,Py] = project3Dto2D(px,py,pz);
    rayDirection=||(px,py,pz)||;
    pnew=(px,py,pz);
    while pnew within bounding box
        calculate pixel intensity
    end while
    set colour(Px,Py)=returnedRayColour;
end
```

In addition to the augmented reality visualization, the system also provides users with CT and US images.

5.5 Ultrasound Simulation

In an US guided procedure, it is necessary to have an US image for the clinician to view. In a simulation system, the US volume of a patient is not always present with

Figure 5.8: Overview of US simulation procedure. *Left*: Slicing the CT volume at the probe position. *Centre*: Ray tracing down a fan shape region of the CT slice. *Right*: Final simulated US image.

other imaging volumes (for example CT or MRI). In these cases, it is necessary to simulate the US image from data of another imaging modality.

The process of creating an ultrasound image from a patient CT data set based on the position of a probe has three main steps: tracking the probe to get position/rotational information, preparing and re-slicing the CT volume, and finally simulating the acoustic reflectances and US properties from the extracted CT slice.

5.5.1 Probe Tracking

The dummy US probe is tracked in 3D by the camera in much the same way that the mannequin is tracked. An L-shaped marker pattern is placed on the side of the dummy probe that will face the camera. The three returned angles $\theta_x, \theta_y, \theta_z$ are the rotation of the probe about the camera x, y, z axis respectively. This may not be the same for the rotations on the CT volume, as it is dependent on the orientation of the CT volume.

In order to calibrate the probe in camera coordinates, a common calibration procedure is used. A small hole in the end of the probe being pointed onto the mannequin

is placed on a raised point that is fixed a table and rotated in 3D, the multiple poses being recorded by the camera. For each recording, the three points that make up the marker are stored. The set of all recorded points creates three concentric spheres, one for each point, all three centred at the point of rotation. All that needs to be done to solve for the centre point is to fit the data to a sphere, which becomes a least-squares problem. In order to properly solve this least-squares problem, at least four non co-planar points are needed, or else a sphere cannot be properly constructed. For each sphere we have the following equation (where $t = (t_x, t_y, t_z)$ is the centre of the sphere):

$$(x - t_x)^2 + (y - t_y)^2 + (z - t_z)^2 = r^2$$

Rearranging and letting $C = r^2 - t_x^2 - t_y^2 - t_z^2$ yields

$$2xt_x + 2yt_y + 2zt_z + C = x^2 + y^2 + z^2$$

which for n poses becomes

$$2x_it_x + 2y_it_y + 2z_it_z + C = x_i^2 + y_i^2 + z_i^2, i = 1 \dots n$$

Stacking the multiple poses into the matrix form $Ac = b$ results in

$$\begin{pmatrix} 2x_1 & 2y_1 & 2z_1 & 1 \\ 2x_2 & 2y_2 & 2z_2 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 2x_n & 2y_n & 2z_n & 1 \end{pmatrix} \begin{pmatrix} t_x \\ t_y \\ t_z \\ C \end{pmatrix} = \begin{pmatrix} x_1^2 + y_1^2 + z_1^2 \\ x_2^2 + y_2^2 + z_2^2 \\ \vdots \\ x_n^2 + y_n^2 + z_n^2 \end{pmatrix}$$

The least-squares solution to the cost function $\min_c ||Ac - b||^2$, is the pseudo-inverse

solution $c = (A^T A)^{-1} A^T b$. Since $b = (t_x, t_y, t_z, C)^T$, the coordinates of the rotation point, and the radius $r = \sqrt{C + t_x^2 + t_y^2 + t_z^2}$. The three spheres give three centres, the average of which can be found to compute the best fit for the true centre. Once the centre is known, the vector between it and the marker location can be calculated and stored so the offset will always be known.

A similar procedure is done to calculate the major axis of the probe. Placing the probe bottom edge flat on a plane and moving it around on that plane will create three sets of data which can again be fit with a least-squares solution - this time a plane. The equation of a plane is $z = Ax + By + C$ and we wish to minimize the sum of the squared errors between z_i and $Ax_i + By_i + C$ where $i = 1 \dots n$ is the number of poses [57]. Let $E(A, B, C) = \sum_{i=1}^n ((Ax_i + By_i + C) - z_i)^2$ be the error function we wish to minimize. E is a non-negative hyper-parabola, and it will have a minimum when its gradient is equal to 0. Therefore

$$\nabla E = (0, 0, 0) = 2 \sum_{i=1}^n (Ax_i + By_i + C - z_i)(x_i, y_i, 1)$$

which in matrix form results in

$$\begin{pmatrix} \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i y_i & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i y_i & \sum_{i=1}^n y_i^2 & \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n y_i & \sum_{i=1}^n 1 \end{pmatrix} \begin{pmatrix} A \\ B \\ C \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^n x_i z_i \\ \sum_{i=1}^n y_i z_i \\ \sum_{i=1}^n z_i \end{pmatrix}$$

Finding the normal vector to this plane and finding the relationship between it and the vectors of the marker allow the axis of the probe to be determined from measurements of the marker. Now that both the centre of the probe and its axis have been computed, the correct position/orientation of it can be determined to re-slice

the CT volume.

5.5.2 Volume Preparation

Once the desired position and orientation for the re-slice is obtained, the CT volume must be set up appropriately based on the position of the needle and the deformations from the FEM. There are two ways that both of these aspects can be performed: fully across the entire 3D volume, or only on the re-slice plane. Since we are not concerned with viewing any deformations off this plane, and are not viewing the needle anywhere besides in-plane, and also because computing everything for the full volume is much more computationally expensive, we can ignore calculations for any voxels of the volume not on the plane.

Deformations from the FEM are calculated using tri-linear interpolation. The element of the voxel is determined from the 3D element matrix, and the four vertices from the list of elements and their nodes. As discussed in Section 4.2, the deformations from the four node vertices are weighted based on their distance from the voxel, the calculated average deformation applied to the voxel.

The needle is treated as a thin cylinder centered on the insertion vector of the virtual needle. Determining whether the voxels of the re-slice plane are inside the needle or not is fairly simple. The vector between the voxel and the insertion point of the needle is calculated, and projected onto the insertion vector. For a voxel to be flagged as inside the needle two conditions must be met. First, the magnitude of the projection must be less than the total length of the needle in the body and secondly, the perpendicular distance from the insertion vector to the voxel must be less than the radius of the cylinder representing the needle. These voxels are set to a very high

Hounsfield value, to simulate the reflection from a stainless steel material [58].

5.5.3 Simulation

The algorithm for the simulation of US from CT is based on a simple acoustic model presented by Shams *et al.* [25]. The algorithm can deal with both rectilinear and curved probes, and operates on the assumption that the acoustic impedances of tissue can be calculated from the Hounsfield values in a CT image. If a curved probe is desired, the CT slice must first be warped into a fan shape based on the properties of the probe - its field of view (θ), near and far radii (r_{min}, r_{max}), and origin (f_{0x}, f_{0y}). This is done by taking the fan-shaped section of the CT image and transforming it back into a rectilinear image. The rectilinear transformation would have y-axis corresponding to the radius of the probe fan (from r_{min} to r_{max}), and x-axis corresponding to the length of the arc of the fan at that specific y (radius) location. This creates an image that is distorted at the top (at radius values closer to r_{min}) and becomes less distorted further along the y-axis (towards r_{max}).

In order to accurately simulate the effects of an US beam, the reflection of the sound waves at boundaries between tissue types must be modeled. At these boundaries there is a change in the acoustic impedance, which creates a barrier of sorts which only allows some of the sound energy to continue, reflecting the rest back in a scattering pattern. For a homogeneous material with density ρ , the characteristic impedance is defined as

$$Z = \rho c$$

where c is the wave speed or in this case, the speed of sound. At a boundary between

two materials with acoustic impedance Z_1 and Z_2 , the ratio of the sound energy which is reflected is called the coefficient of reflection:

$$R = \left(\frac{Z_2 - Z_1}{Z_2 + Z_1} \right)^2$$

To simulate the US image, the US beam is treated as a single ray being cast down each column of the image, starting with full intensity $I_0^t = 1$. The transmitted portion of the beam at voxel i is $I_i^t = (1 - R_i)I_{i-1}^t$ where I_{i-1}^t is the transmitted intensity of the beam at the previous voxel, and R_i is the coefficient of reflection between the previous medium and next medium. At each voxel along the beam path, the reflected intensity to the transducer is calculated as

$$I_i^r = I_{i-1}^t R_i \cos(a)_i \quad (5.1)$$

where

$$\cos(a)_i = \frac{\vec{r} \cdot \nabla u_i}{\|\vec{r}\| \|\nabla u_i\|}$$

is the angle of incidence between the beam direction and the surface normal, with $\vec{r}$ the direction of the ray and u_i the intensity of the CT image.

Since an US transducer generally has more than one active element sending and receiving US beams simultaneously, the beam from one element may be received by an adjacent element upon reflection, creating a blurring effect which becomes more pronounced the further it is traveled along the beam length. This is modeled by Shams [25] through the use of a Hanning window function which is applied over the image in the x-axis at each y-interval. This can be shown by modifying (5.1) to integrate over a small length $l = n_a(w_e + s_e)$ of the x -axis, where n_a is the number

Table 5.1: US Simulation comparison for a 300×300 pixel image

Method	Time
MATLAB	1.523 sec
Non-CUDA C++	0.022 sec
CUDA C++	0.00105 sec

of active transducer elements on the probe, w_e is the width of an element and s_e the spacing:

$$I_r(x, y) = \int_{x-l}^{x+l} I_t(x, y-1) R(u, y) \cos_a(u, y) w(u) du \quad (5.2)$$

where $w(u)$ is the Hanning window function, and the variable y represents the column voxel.

Once the US intensities are calculated for each voxel in the image, the entire image is log-compressed for better visibility [25].

The algorithm is implemented in CUDA in order to run on the GPU for increased simulation speed. This allows the simulation to be run at interactive rates. A comparison of the various speeds of the algorithm can be seen in Table 5.1, with implementations in MATLAB (original method), non-CUDA and CUDA C++ compared for a 300×300 pixel image. As can be seen, CUDA implementation improves the simulation speed by over a magnitude, reducing the strain on the graphics loop. A sum of the squared difference between the intensities of US images generated on CPU and GPU was conducted to determine the quality of the GPU image. The total sum of the squared differences was 8.86 over 44344 pixels. The total intensity of the GPU image was 931.95; therefore the total sum of the differences is less than 1% of the total image intensity. The differences in intensity came mainly from a few pixels.

5.6 Summary

In this chapter, a method to produce an augmented reality cutaway of a mannequin using an overlaid CT volume image was discussed. The initial procedure using a VTK scene and printed spine mannequin proved to be somewhat unrealistic and inaccurate, and a change to a real torso mannequin and volume ray casting method was used. In order to align the CT volume to be overlaid properly in camera coordinates, it was registered to a CT scan of the mannequin through hand annotated radio-opaque beads, and then to the camera space through tracking markers. Ray casting and MicronTracker2 SDK functions were then used to perform the image projection and overlaying.

As can be seen in Figure 5.8, the overlay is aligned with the mannequin, and appears as if the mannequin has been “cut open” to show the internal structure. The depth of the spine can be seen, as it appears further “inside” when compared to the hip bone (bottom left of the “cut away” section of the image).

There are still additional work that could be explored to improve the look of the overlay further such as incorporating the two other types of depth perception assistance discussed in [39], curvature of the body, and the angle of incidence from the eye.

The procedure for the simulation of ultrasound from CT was also explained; simulated images are needed since an US scan of a patient is not always possible when a CT scan is done, leaving only one imaging modality to work with.

Chapter 6

System Performance and Evaluation

6.1 Overview

The created prototype system must be evaluated to gauge the effectiveness of its functions. This can be done in one of two main ways: 1) objectively, through quantitatively measuring certain parameters of the system and comparing them to other systems or published results; or 2) subjectively, through user studies with trials and questionnaires. The evaluation of the system proposed in this thesis will include both objective and subjective analysis, by comparing the forces generated by the FEM with those recorded in real trials, and by performing a user impact study where two groups of subjects were asked to try the system and respond to a questionnaire.

Figure 6.1: A screenshot of the Graphical User Interface. Main video feed is on the left, the auxiliary view, simulated US image and CT slice are on the right.

6.2 Hardware and Software Performance

A screenshot of the system in action, shown through the GUI, can be seen in Figure 6.1. The main display shows the augmented video feed, with CT overlay visible near the centre of the display. The wireframe outline of the dummy US probe can be seen around the position of the probe, with the virtual needle displayed on screen as a red line. This line will change colour depending on the anatomy the needle is touching. The needle turns yellow when the ligamentum flavum is entered, and will turn green when the needle is in soft tissue or not in the volume. The auxiliary view in the top right shows the needle insertion vector and crosshairs located at the needle base. Once the needle is inside the tissue during training, it can be seen in both the CT slice (bottom right) and the corresponding simulated US slice (middle right).

The system is able to maintain the graphics loop at a rate of up to 25 frames per

Figure 6.2: A plot of the forces during simulated needle insertion versus time for an insertion piercing the ligamentum flavum (LF).

second, which is satisfactory for visual feedback. The haptic loop is maintained at a rate of at least 1000 Hz for an FEM model with 2000 nodes.

In order to objectively analyze the performance of the system, the forces generated by the model and sent to the PHANToM were recorded, as well as the type of tissue being traversed. The plot of these forces over time can be seen in Figure 6.2. These forces can be compared to those reported in the literature, to determine if the values measured during insertion into different tissue types (skin puncture, soft tissue, ligament) correspond. Skin piercing occurred at an average force of 1.5N, and peak force at exit of the ligamentum flavum was 4.7N. The force required to pierce skin was reported by Ra *et al.* [8] as approximately 1.5N. Naemura *et al.* reported peak forces of 5 ± 1 N for exiting the ligamentum flavum of porcine tissue with an 18-gauge needle. Brett *et al.* [59] reported a peak of 9N for the ligamentum flavum in porcine

Figure 6.3: A plot of the output from the finite state machine. Force (red) switches direction when velocity is over or under the threshold ($\pm 0.03 \text{ mm/s}$) for the latch time of 100ms. The binary switch is shown in green.

tissue with 16-gauge needles. The needle gauge used is responsible for the increased force reported. Compared to these results, the forces generated by the system are appropriate. In order to create an even more accurate model, force measurements during actual spinal anesthesia procedures could be performed as future work.

An example output from the Finite State Machine (Section 4.3) used to switch the direction of applied force can be seen in Figure 6.3. The friction force being applied is shown in red and the current velocity in blue. The switching signal from the state machine is shown in green. Once the velocity has been below or above the velocity threshold ($\pm 0.03 \text{ mm/s}$) for the latch time (100ms), the direction of the applied force is switched. For retracting the needle, a constant force of 0.75N was used.

The mannequin was positioned so that the area of interest, the lumbar region of the spine, would be completely within the workspace of the haptic device. Spinal anesthesia procedures are performed with patients sitting with the spine curved away

from the clinician, or lying on their side with back facing the clinician. The mannequin is in an erect position, and can not realistically be curved. It was reported that the position of the mannequin seemed reasonable for performing the procedure, however it could be placed on its side if an apparatus to hold it was constructed.

6.3 User Impact Study

An important part of system evaluation involves feedback from potential users. In this section, the user study performed is described and comments are discussed. The study was approved by Queen's Human Ethics Board.

Two groups of users were asked to try the system: three students and three technicians with experience in viewing ultrasound images and basic understanding of needle insertion procedures, and two medical residents with experience with needle insertions. The subjects consisted of one female and seven males with the range of age of 22 to 35. Subjects were healthy with no neurological or visual deficiencies. The users were given a short introduction to the system which included a 1-2 minute overview of its purpose, features and how to interact with it. Then, they were given 5-10 minutes to explore the system functionality and familiarize themselves with using the system to reach a certain location to feel the loss of resistance characterizing a successful spinal anesthesia procedure. The subjects were then asked to perform a trial and fill out a questionnaire on their experience with the system. The results of the questionnaire are presented below. Three experienced clinicians were also consulted to evaluate the functionality of the system and provide feedback; however they did not fill out the questionnaire.

6.3.1 Questionnaire Design

The questionnaire answered was composed of three sections: System Functionality, Graphical User Interface (GUI) and Haptic Feedback. Questions are presented in Tables 6.1, 6.2 and 6.3. The sections were designed to address the functionality of the proposed system and its potential to be used as a training device. Questions were created to obtain specific information about each of the program functions. For example in the GUI section, specific questions were asked about the CT overlay to determine if it was useful for initial placement of the needle as well as for general navigation. In the Haptic Feedback section, subjects were asked if they could feel the different tissues and they were asked if the mannequin was useful for spatial orientation. The users were asked to respond to the questions on a scale of 1-7 with 1 being strongly disagree, 7 being strongly agree, 4 being neither agree nor disagree. A copy of the questionnaire is provided in Appendix A. In the following tables showing the results of the questionnaire, an asterisk beside a user number indicates that user was one of the experienced residents. Some feedback received in the first round of trials with the inexperienced group was implemented in the system prior to the experienced group using the system.

6.3.2 Evaluation of the Questionnaire

As can be seen in Table 6.1, respondents were able to clearly understand the purpose of the system, and were able to easily interact with all of its components. The respondents found the mannequin to be useful for orientating themselves and a good aid for navigating the patient data, which was a goal that was hoped to be achieved at the start of the project.

Table 6.1: Questions on System Functionality. S1-6 correspond to inexperienced users, S7*-8* to experienced users.

No.	Question	S1	S2	S3	S4	S5	S6	S7*	S8*	Median
1.1	Purpose understood	7	7	6	7	7	7	6	6	7.0
1.2	Interaction straight forward	7	6	6	6	7	7	6	6	6.0
1.3	PHANToM easily manipulated	7	7	7	6	6	6	6	4	6.0
1.4	Mannequin spatial awareness	7	7	5	7	7	6	4	7	7.0

Table 6.2 shows the results of the replies to the GUI section of the survey. The answers were mostly satisfactory, the main difficulty the users encountered was being able to navigate the patient data using solely US data. This is to be expected, as navigating based on US alone is challenging. However, the users did report that the simulated US images were sufficiently similar to real US images. One piece of feedback received regarding US images was to modify the US simulation algorithm to show more clearly the boundaries between soft tissue and muscles or ligaments. Users found the combination of frontal CT view and oblique camera view with CT overlay to be useful for navigating and initially placing the needle. A comment was to have the needle inside the US slice more easily visible, as the reflected intensity was similar to the bone edge, and users sometimes could not differentiate between the bone edge and the needle when viewed in US. Possible solutions are to either modify the simulation algorithm to display less of a bright edge for bones, or to have a slightly different colour for the needle in the US image, although this would not be as realistic as a real needle in US. The wireframe outline of the US slice was not deemed very useful for orienting the US image within the volume, however the respondents did use it when attempting to guide the needle in US.

Table 6.2: Questions on the Graphical User Interface (GUI). S1-6 correspond to in-experienced users, S7*-8* to experienced users.

No.	Question	S1	S2	S3	S4	S5	S6	S7*	S8*	Median
2.1	Video feed easily seen	7	7	7	6	7	7	7	7	7.0
2.2	Overlay helpful for ini. placement	5	7	5	4	7	6	4	5	5.0
2.3	Front view helpful for ini. placement	5	7	6	7	7	7	6	6	7.0
2.4	Overlay helpful for navigating	7	5	5	4	6	6	4	7	5.5
2.5	US wireframe on screen helps orient	7	3	5	5	6	6	4	4	5.0
2.6	4 displays provide relevant info	7	6	6	4	7	7	4	7	7.0
2.7	Easy to navigate patient using US	4	2	3	3	7	6	6	4	4.0
2.8	Simulate US similar to real US	6	3	4	7	7	5	5	5	5.0
2.9	Enhanced bone edge in US useful	6	3	6	6	7	6	7	5	6.0
2.10	Virtual needle on screen useful	5	5	4	7	6	6	7	7	6.0

As seen in Table 6.3, when asked to manipulate the virtual needle and perform test insertions, users responded favourably to the haptic feedback. Most could easily tell when skin was punctured, and could feel when the needle struck bone. It was reported by one non-expert user that pressing on bone provided a feeling of pressing against rubber as opposed to a hard surface; the stiffness of the bone could be increased, however this could have the drawback of making the system unstable if the stiffness is too high for the sampling rate and system damping. The users also appreciated the changing colour of the display of the virtual needle when bone was struck (Q2.10).

When asked to insert the needle between two spinous processes and puncture through the ligamentum flavum, the users were able to successfully perform the procedure and feel the increased resistance upon entering the ligament, followed by the “pop through” from the loss of resistance when the ligament is pierced. Three additional questions were asked of the experienced residents for this section, to determine

Table 6.3: Questions for Haptic Feedback. S1-6 correspond to inexperienced users, S7*-8* to experienced users.

No.	Question	S1	S2	S3	S4	S5	S6	S7*	S8*	Median
3.1	Feel when pierced skin	7	7	7	7	7	7	7	7	7.0
3.1b	Skin puncture realistic	-	-	-	-	-	-	6	6	6.0
3.2	Feel needle in soft tissue	7	7	6	5	7	6	6	7	6.5
3.2b	Soft tissue realistic	-	-	-	-	-	-	5	5	5.0
3.3	Feel loss of resistance	7	6	7	7	6	6	7	6	6.5
3.3b	Loss of res. realistic	-	-	-	-	-	-	6	6	6.0
3.4	Feel bone	7	5	7	7	7	7	7	7	7.0
3.5	Feel vertebral wall	7	7	6	3	7	n/a	6	7	7.0

how they felt the system performed when compared to the real life sensation of piercing skin, traveling through soft tissue and feeling the loss of resistance upon piercing the ligamentum flavum. The experienced users reported that the forces felt realistic.

A summary of the suggested improvements by the respondents to the questionnaire is provided below. Several were implemented following the user studies with the inexperienced group, therefore the experienced residents evaluated the system with the new improvements.

- Add additional facets to the dummy US probe so it could be rotated completely without the marker disappearing from the view of the camera.
- Modify the US simulation code to show a brighter border between soft tissue and muscle/ligaments.
- Differentiate more between the needle and bone edge in the US simulation, as currently it is somewhat difficult to tell where the needle is when it is close to bone.
- Add a trajectory guide to the frontal view to give the user a better idea of where

the tip will be traveling upon entry. This was incorporated after the first round of trials.

- Increase the stiffness of the bone. This was incorporated after the first round of trials.
- Add a physically retractable needle to the haptic device end effector, which would give users a sense of where the virtual needle is in space. This was incorporated after the first round of trials.

Three clinicians were also consulted during the evaluation of the system, specifically an ER specialist, an anesthesiologist, and an orthopedic surgeon. All three clinicians were also instructors in medical school, who could comment on and identify the particular aspects of the system that would be useful for teaching/training purposes, and the areas that need to be improved. The clinicians did not take part in the questionnaire, and are therefore not represented in Tables 6.1, 6.2 and 6.3. Suggestions for improvements include:

- Finer segmentation of the CT scan, in order to differentiate between different ligaments, and the dura space, allowing users to detect additional failures in needle placement.
- Adding measurement labels with 1cm spacing to the physical needle so that the user can tell how far in to the body the needle is.
- Having trainees use the system without certain aspects of the visual feedback (e.g. without CT overlay), and then perform the procedure again with them active.

- Allow rotations of the needle within the first few centimeters of the insertion. This would be require additional active DOFs which the current haptic device could not provide.

One common discussion point that came out of the consultations was to create a “training plan” for trainees using the system. The instructors were in agreement that students should use the graphical aids as a source of “second information” and focus on the physical mannequin/needle. The additional visual feedback would be used to determine deviations or mistakes during insertion. Since the patient CT volume is registered to the mannequin based on the position of the physical spine model, users can palpate the physical mannequin to feel the spinous processes and the gap between them. The virtual needle can be inserted into this gap and the user can use the visual feedback in addition to the force to determine where the needle is inside the anatomy.

6.4 Summary

The user study discussed in this chapter had favourable reviews from its respondents. Most were satisfied with the way the system operated and displayed information, and they were able to navigate the patient volume successfully using all the aids provided, as well as the mannequin. The US simulation was satisfactory to the users, although some had difficulty navigating through the data solely with the US images. Some of the issues brought up in feedback were implemented, while others are future work.

Chapter 7

Summary and Conclusions

7.1 Summary of Contributions

This thesis outlines a prototype for a simulation system for spinal needle insertions, specifically spinal anesthesia, which has potential to be used as a training device. The proposed system allows a user to interact with a mannequin using a haptic device. The mannequin is tracked with an optical camera system and is registered to a patient-specific CT volume. Forces expected during needle insertion are modeled by a deformable FEM and “transferred” to the user via the haptic device. US images are simulated from the CT volume and presented through a GUI along with an augmented reality video feed. The system would allow training for spinal anesthesia to be performed at any time, without the need for live patients or access to a cadaver, and without the direct supervision of a trained clinician.

7.2 Contributions

The following are the specific contributions of this thesis.

7.2.1 System as a simulation environment

The designed systems' primary purpose is to be used as a simulation environment which trainees would be able to use whenever they desired to try an insertion on a specific patient. To this end, the system can accept any patient CT volume which has been registered with the mannequin. The mannequin and physical needle guide provide physical visual clues to the user; when this is combined with the force feedback from the haptic device it creates a much more realistic simulation than simply a virtual needle would. The results from trainee trials could be saved and viewed later by the trainee or a supervisor to gauge progress.

7.2.2 Tracking system

A MicronTracker2 optical tracking system is used to provide a video feed of the procedure, as well as track both the mannequin and dummy US probe. Markers placed on the mannequin align the mannequin in camera coordinates, as well as in haptic coordinates. The dummy US probe marker allows the position and orientation of the probe to be determined.

7.2.3 Graphical User Interface

A GUI was designed in QT in C++ to provide the user with additional visual feedback. The main display on screen shows the video feed from the MicronTracker

camera system, with augmented CT overlay. Additional displays show (1) an auxiliary view of the CT volume, with the needle tip position and needle insertion vector, (2) the CT slice from the position/orientation of the dummy probe and (3) simulated US image corresponding to the CT slice.

7.2.4 Overlay of CT volume

An augmented reality display of the mannequin is shown in the GUI. By registering the mannequin with a CT volume, and the mannequin to the camera space, the CT volume can be projected onto the image plane of the MicronTracker2 camera using a combination of MicronTracker2 SDK functions and a volume rendering algorithm. The overlaid image is centered over the spine to create a cut-away type view to provide users with more information and to help position the needle for insertion. A simple blending function is used to meld the two images to alleviate the depth perception problems that can occur when overlaying images.

7.2.5 Finite Element Model and Force Feedback

A PHANToM Premium 1.5A haptic device is used to generate force feedback to the user. The forces are generated through a 3D static, linear elastic finite element model. The finite element model was chosen over other options such as mesh-spring deformations and surface models because it provided the best compromise between speed, accuracy and ease of use.

The force to the user is calculated from the mesh deformations as well as the friction from the anatomy. A latching effect is used to determine when the needle is being retracted from the body, and apply the necessary forces to compensate for the

switch in direction of the force. Transverse motions of the needle base off the original insertion vector are calculated by modeling the needle as a thin, bending cylinder. The needle inside the body is treated as a rigid object.

7.2.6 CUDA-enabled Ultrasound Simulation

Since a complete 3D ultrasound scan of every patient may not be available, a simulated US image from the CT data is created. This simulation follows the algorithm proposed in [25], where a ray simulating an US beam is cast down each column of the CT image, losing intensity at each voxel based on the acoustic impedance of the boundary between the tissues.

The algorithm is implemented in CUDA on an NVidia GeForce GTX 470 graphics card to speed up the simulation to run in real time. This produces a substantial increase in speed, with a 300×300 pixel image taking on average 0.00105 seconds to simulate.

7.2.7 User Evaluation Study

In order to evaluate the effectiveness of the system to simulate a real needle insertion procedure, subjects were recruited to trial the system and answer a questionnaire designed to rate the important aspects of the design. The subjects were students, technicians and medical residents. The students and technicians had knowledge of needle puncture procedures but little practical experience, while medical residents had experience performing needle insertions. The survey had three sections: one on the functionality and ease of use of the system, one on the graphical feedback received from the GUI, and one for the haptic feedback. Subjects responded positively to the

system and its functionality, and provided useful feedback on the system.

Clinicians and instructors with active practice in percutaneous spinal interventions were also consulted. A “training plan” for trainees was discussed, in which users would be able to palpate the physical mannequin in order to find the gap between spinous processes; the needle is inserted based on this information, and the addition visual feedback is used to determine the accuracy or path of the needle.

The results of the user study showed that the system could be used as a training tool with some adjustments. The possibilities surrounding physically palpating the mannequin to feel the underlying spine model and then inserting a virtual needle were very promising for training applications, as this most accurately mimics the spinal anesthesia procedure.

Currently, the system would be best suited for performing para-median US guided needle insertions. In the mid-line approach, the dummy US probe could interfere with the needle position. To better simulate this procedure, a frozen US image could be taken from probe prior to removing it from the body. The needle insertion could then be conducted using this virtual US image.

7.3 Future Work

Throughout the thesis, there have been areas highlighted which could be improved upon. The improvements are broken down by area and include:

Hardware Improvements

1. Simulate the effect of US probe force on the US images by attaching a force

sensor at the contact point between probe and the body in order to determine the force being applied. This force could be applied to the FEM to further deform the CT volume prior to US simulation.

2. Improve the marker detection for the dummy ultrasound probe so that it can be rotated completely in all directions. Currently, the probe set up is fine for viewing US slices for needle insertions where the gap between spinous processes is desired, but cannot be rotated to show slices perpendicular to this. If the system was desired to be used for finding other anatomical targets in a body, this would have to be changed.
3. Currently the system is assuming the mannequin and camera are fixed in position once the initial calibrations are made. This is fine for the type of procedure being performed, but it may be useful to be able to have the mannequin and camera movable.
4. Construct an apparatus to hold the mannequin stable during physical palpations, or to hold the mannequin if it is placed on its side.
5. Create a more permanent needle guide which would allow the physical needle to slide out along the stylus when the virtual needle is removed from the body. For example, a spring could be added to provide this effect, and the force from spring compression would be compensated for in the tissue model.
6. For an improved simulation of needle forces felt during insertion, a device with 6 active DOFs and higher force output would be very useful, as it could provide torques to the user based on the haptic stylus rotations. Additional active DOFs adds substantially to the cost of the haptic device.

Software

1. Improvements to the ultrasound simulation algorithm to better show the difference between soft tissue and muscle/ligaments. This would improve the quality of the US simulation even more, and allow users to determine where they are in the patient volume more easily.
2. To create a more realistic augmented reality view through the overlaying of the CT volume on the video feed of the mannequin, some of the methods described by Bichlmeier et al. [39] could be implemented to create even better blending of the overlay and the background image. In order to properly do this, a complete 3D scan of the surface of the mannequin would have to be performed. Knowledge of the US probe position and dimensions could allow for a function that would make the pixels of the overlay covering the probe transparent.
3. The needle is currently modeled as a rigid structure when inside the body. If larger needle deflections are desired (such as for biopsy procedures, or prostate brachytherapy or contact with the bone), a needle bending model would have to be introduced to the FEM/force calculations.

Studies and patient anatomy

1. In order to obtain realistic measurements of forces, a study on the forces experienced during actual needle insertion should be performed, with tests on varying tissue types to determine the proper elastic properties for each one.
2. There is currently no quantitative comparison between the proposed system and one with US alone for guidance. A study on the differences between training environments could be conducted.

3. Finer segmentation of the CT scan, in order to differentiate between ligaments and other tissues around the spine, to be able to determine different types of needle insertion failure. This would require a radiologist familiar with the anatomy of the spine to segment the CT volume.

Bibliography

- [1] D. M. Gaba. The future vision of simulation in health care. *Quality and Safety in Health Care*, 13(suppl 1):i2–i10, 2004.
- [2] D. Gould, N. Chalmers, S. Johnson, C. Kilkenny, M. White, B. Bech, L. Lonn, and F. Bello. Simulation: Moving from technology challenge to human factors success. *CardioVascular and Interventional Radiology*, pages 1–9.
- [3] F. P. Vidal, N. W. John, A. E. Healey, and D. A. Gould. Simulation of ultrasound guided needle puncture using patient specific data with 3d textures and volume haptics. *Computer Animation and Virtual Worlds*, 19(2):111–127, 2008.
- [4] O. Goksel and S. Salcudean. Haptic simulator for prostate brachytherapy with simulated ultrasound. *Biomedical Simulation*, 5958:150–159, 2010.
- [5] C. Chui, J. S. K. Ong, Z.i Lian, Z. Wang, J. Teo, J. Zhang, C. Yan, S. Ong, S. Wang, H. Wong, C. Teo, and S. Teoh. Haptics in computer-mediated simulation: Training in vertebroplasty surgery. *Simulation and Gaming*, 37(4):438–451, 2006.

-
- [6] M. Farber, J. Heller, F. Hummel, C. Gerloff, and H. Handels. Virtual reality based training of lumbar punctures using a 6 degree of freedom haptic device. *Advances in Medical Engineering*, 114:236–240, 2007.
- [7] W. Chan, D. Ni, W. Pang, J. Qin, Y. Chui, S. Yu, and P. Heng. Learning ultrasound-guided needle insertion skills through an edutainment game. *Transactions on Edutainment IV*, 6250:200–214, 2010.
- [8] J. B. Ra, S. M. Kwon, J. K. Kim, J. Yi, K. H. Kim, H. W. Park, K.-U. Kyung, D.-S. Kwon, H. S. Kang, S. T. Kwon, L. Jiang, J. Zeng, K. Geary, and S. K. Mun. Spine needle biopsy simulator using visual and force feedback. *Computer Aided Surgery*, 7(6):353–363, 2002.
- [9] D. Magee, Y. Zhu, R. Ratnalingam, P. Gardner, and D. Kessel. An augmented reality simulator for ultrasound guided needle placement training. *Medical and Biological Engineering and Computing*, 45:957–967, 2007.
- [10] P. Gorman, T. Krummel, R. Webster, M. Smith, and D. Hutchens. A prototype haptic lumbar puncture simulator. *Studies in Health Technology and Informatics*, 70:106–109, 2000.
- [11] W.F. Casey. Spinal anaesthesia - a practical guide. *Update in Anaesthesia*, 12(8):1–7, 2000.
- [12] Claron Technologies. *MicronTracker2 Development Guide*. 2002.
- [13] W. Schroeder, K. Martin, and W. Lorensen. *The visualization toolkit: an object-oriented approach to 3D graphics*. Prent-Hall, 1998.

- [14] L. Ibanez, W. Schroeder, L. Ng, and J. Cates. *The ITK software guide*. Kitware, Inc., 2nd edition, 2005.
- [15] Nokia Systems. *Using QT*. 2009.
- [16] A. Nyssen, R. Larbuisson, M. Janssens, R. Pendeville, and A. Mayn. A comparison of the training value of two types of anesthesia simulators: Computer screen-based and mannequin-based simulators. *Anesthesia and Analgesia*, 94(6):1560–1565, 2002.
- [17] S. Shin, W. Park, H. Cho, S. Park, and L. Kim. Needle insertion simulator with haptic feedback. *Human-Computer Interaction. Interaction Techniques and Environments*, 6762:119–124, 2011.
- [18] B. Gonenc and H. Gurocak. Force control with hybrid actuator for virtual needle insertion. *IEEE World Haptics Conference*, pages 173 –177, 2011.
- [19] O. Goksel and S. Salcudean. Haptic simulator for prostate brachytherapy with simulated ultrasound. *Biomedical Simulation*, 5958:150–159, 2010.
- [20] P. Villard, P. Boshier, F. Bello, and D. Gould. Virtual reality simulation of liver biopsy with a respiratory component. In *Liver Biopsy*. 2011.
- [21] T. Horeman, S. Rodrigues, J. van den Dobbelsteen, F. Jansen, and J. Dankelman. Visual force feedback in laparoscopic training. *Surgical Endoscopy*, 1:1–7.
- [22] J. B. Cooper and V. R. Taqueti. A brief history of the development of mannequin simulators for clinical education and training. *Postgraduate Medical Journal*, 84(997):563–570, 2008.

- [23] T. Tsai, P.H. Harasym, C. Nijssen-Jordan, and P. Jennett. Learning gains derived from a high-fidelity mannequin-based simulation in the pediatric emergency department. *Journal of the Formosan Medical Association*, 105(1):94 – 98, 2006.
- [24] D. J. Brenner and E. J. Hall. Computed tomography – an increasing source of radiation exposure. *New England Journal of Medicine*, 357(22):2277–2284, 2007.
- [25] R. Shams, R. Hartley, and N. Navab. Real-time simulation of medical ultrasound from ct images. *Medical Image Computing and Computer-Assisted Intervention*, 5242:734–741, 2008.
- [26] S. Gill, P. Abolmaesumi, G. Fichtinger, J. Boisvert, D. Pichora, D. Borshneck, and P. Mousavi. Biomechanically constrained groupwise ultrasound to ct registration of the lumbar spine. *Medical Image Analysis*, 5761:803–810, 2010.
- [27] M.C. Cavusoglu, D. Feygin, and F. Tendick. A critical study of the mechanical and electrical properties of the phantom haptic interface and improvements for highperformance control. *Presence*, 11(6):555–568, 2002.
- [28] A. Tahmasebi, B. Taati, F. Mobasser, and K. Hashtrudi-Zaad. Dynamic parameter identification and analysis of a phantom haptic device. in *Proceedings of the IEEE Conference on Control Applications*, pages 1251–1256, August 2005.
- [29] S.P. DiMaio and S.E. Salcudean. Interactive simulation of needle insertion models. *IEEE Transactions on Biomedical Engineering*, 52(7):1167 –1179, July 2005.
- [30] E. Dehghan and S. Salcudean. Needle insertion study using ultrasound-based 2d motion tracking. *Medical Image Computing and Computer-Assisted Intervention*, 5242:660–667, 2008.

- [31] N. Abolhassani, R. Patel, and M. Moallem. Needle insertion into soft tissue: A survey. *Medical Engineering and Physics*, 29(4):413 – 431, 2007.
- [32] T.K. Podder, J. Sherman, D.P. Clark, E.M. Messing, D.J. Rubens, J.G. Strang, L. Liao, R.A. Brasacchio, Y. Zhang, W.S. Ng, and Y. Yu. Evaluation of robotic needle insertion in conjunction with in vivo manual insertion in the operating room. *Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication*, pages 66 – 72, Aug. 2005.
- [33] H. Kataoka, T. Washio, M. Audette, and K. Mizuhara. A model for relations between needle deflection, force, and thickness on needle penetration. *Medical Image Computing and Computer-Assisted Intervention*, 2208:966–974, 2001.
- [34] 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*, 25(5-6):509–525, 2006.
- [35] J. Magill, B. Anderson, G. Anderson, P. Hess, and S. Pratt. Multi-axis mechanical simulator for epidural needle insertion. *Medical Simulation*, 3078:267–276, 2004.
- [36] B.T. Sitzman and D.R. Uncles. The effects of needle type gauge and tip bend on spinal needle deflection. *Anesthesia and Analgesia*, 82(2):297–301, 1996.
- [37] L. T. Paolis, M. Pulimeno, and G. Aloisio. An augmented reality application for minimally invasive surgery. *Proceedings of the 14th Nordic-Baltic Conference on Biomedical Engineering and Medical Physics*, 20:489–492, 2008.

- [38] T. Sielhorst, C. Bichlmeier, S. Heining, and N. Navab. Depth perception – a major issue in medical ar: Evaluation study by twenty surgeons, 2006.
- [39] C. Bichlmeier, F. Wimmer, S. M. Heining, and N. Navab. Contextual anatomic mimesis hybrid in-situ visualization method for improving multi-sensory depth perception in medical augmented reality. *Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 1–10, 2007.
- [40] O. Kutter, A. Aichert, C. Bichlmeier, J. Traub, S. M. Heining, B. Ockert, E. Euler, and N. Navab. Real-time volume rendering for high quality visualization in augmented reality. In *International Workshop on Augmented environments for Medical Imaging including Augmented Reality in Computer-aided Surgery*. MICCAI Society, Sept. 2008.
- [41] A.M. Tahmasebi, K. Hashtrudi-Zaad, D. Thompson, and P. Abolmaesumi. A framework for the design of a novel haptic-based medical training simulator. *IEEE Transactions on Information Technology in Biomedicine*, 12(5):658 –666, sept. 2008.
- [42] P. Villard, F. Vidal, C. Hunt, F. Bello, N. John, S. Johnson, and D. Gould. A prototype percutaneous transhepatic cholangiography training simulator with real-time breathing motion. *International Journal of Computer Assisted Radiology and Surgery*, 4:571–578, 2009.
- [43] Z.M. Kulcsar, E. Lovquist, Fitzgerald A.P., A Aboulafia, and Shorten G.D. Testing haptic sensations for spinal anesthesia. *Regional Anesthesia and Pain Medicine*, 36(1):12–16, 2011.

-
- [44] S. Hishikawa, M. Kawano, H. Tanaka, K. Konno, Y. Yasuda, R. Kawano, E. Kobayashi, and A. T. Lefor. Mannequin simulation improves the confidence of medical students performing tube thoracostomy: A prospective, controlled trial. *The American Surgeon*, 76(1):73–78, 2010.
- [45] P.J. Besl and N.D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:239–256, 1992.
- [46] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24:381–395, 1981.
- [47] J.E. Colgate, P.E. Grafing, M.C. Stanley, and G. Schenkel. Implementation of stiff virtual walls in force-reflecting interfaces. *Virtual Reality Annual International Symposium*, pages 202 –208, Sep 1993.
- [48] S. Cotin, H. Delingette, and N. Ayache. A hybrid elastic model for real-time cutting, deformations, and force feedback for surgery training and simulation. *The Visual Computer*, 16:437–452, 2000.
- [49] S.P. DiMaio and S.E. Salcudean. Needle insertion modeling and simulation. *IEEE Transactions on Robotics and Automation*, 19(5):864 – 875, 2003.
- [50] M. Bro-Nielsen. Finite element modeling in surgery simulation. *Proceedings of the IEEE*, 86(3):490 –503, 1998.
- [51] K. H. Huebner. *The Finite Element Method for Engineers*. Wiley, 1975.
- [52] E.J. Chen, J. Novakofski, W.K. Jenkins, and Jr. O’Brien, W.D. Young’s modulus measurements of soft tissues with application to elasticity imaging. *IEEE*

- Transactions on Ultrasonics Ferroelectrics and Frequency Control*, 43(1):191 – 194, 1996.
- [53] A.L. Nachemson, Evans, and J.E. Some mechanical properties of the third human lumbar interlaminar ligament (ligamentum flavum). *Journal of Biomechanics*, 1(3):211 – 220, 1968.
- [54] R.L. Waters and J.M. Morris. An in vitro study of normal and scoliotic interspinous ligaments. *Journal of Biomechanics*, 6(4):343 – 348, 1973.
- [55] G. H. Golub and C. F. V. Loan. *Matrix Computations*. Johns Hopkins Univ. Press, 3rd edition, 1996.
- [56] L.G. Brazier. On the flexure of thin cylindrical shells and other thin structures. *Proceedings of the Royal Society London*, 116, 1927.
- [57] D. Eberly. Least squares fitting of data. 1998.
- [58] C. Coolens and P.J. Childs. Calibration of ct hounseld units for radiotherapy treatment planning of patients with metallic hip prostheses: the use of the extended ct-scale. *Physics in Medicine and Biology*, 42:1591–1603, 2003.
- [59] P.N. Brett, T.J. Parker, A.J. Harrison, T.A Thomas, and A. Carr. Simulation of resistance forces acting on surgical needles. *Proceedings of the Institute of Mechanical Engineers*, 211:225–347, 1997.

Appendix A

Evaluation Questionnaire

System Functionality

1. The system's purpose was easily understood
(strongly agree) 7 6 5 4 3 2 1 (strongly disagree)
2. Interaction with the system software by a user is straight forward
(strongly agree) 7 6 5 4 3 2 1 (strongly disagree)
3. The haptic device can be easily manipulated over the needed workspace on the mannequin
(strongly agree) 7 6 5 4 3 2 1 (strongly disagree)
4. The mannequin is useful for spatial awareness and finding your way around the patient
(strongly agree) 7 6 5 4 3 2 1 (strongly disagree)

GUI

1. The video feed from the camera can be easily seen while performing the procedure
(strongly agree) 7 6 5 4 3 2 1 (strongly disagree)
2. The overlaid CT image is helpful for initial needle placement during the procedure
(strongly agree) 7 6 5 4 3 2 1 (strongly disagree)
3. The frontal CT view is helpful for initial needle placement during the procedure
(strongly agree) 7 6 5 4 3 2 1 (strongly disagree)
4. The overlaid CT image is helpful for navigating the patient anatomy during insertion
(strongly agree) 7 6 5 4 3 2 1 (strongly disagree)
5. The ultrasound slice outline in addition to the CT volume helps orient the viewed slice within the volume
(strongly agree) 7 6 5 4 3 2 1 (strongly disagree)
6. Having 4 displays (one for video feed and overlay, one for frontal view of CT, one for CT slice and one for US slice) provides enough relevant, and helpful, information
(strongly agree) 7 6 5 4 3 2 1 (strongly disagree)
7. The patient anatomy can be navigated using the US images
(strongly agree) 7 6 5 4 3 2 1 (strongly disagree)

8. The simulated US slice is similar to real US images
(strongly agree) 7 6 5 4 3 2 1 (strongly disagree)
9. The enhanced bone edge in the US slice is useful for discerning bone in the image
(strongly agree) 7 6 5 4 3 2 1 (strongly disagree)
10. Having the virtual needle displayed on screen was useful
(strongly agree) 7 6 5 4 3 2 1 (strongly disagree)

Haptic Feedback

1. Could you feel when you pierced skin?
(strongly agree) 7 6 5 4 3 2 1 (strongly disagree)
2. Did the sensation feel similar to real skin puncture?
(strongly agree) 7 6 5 4 3 2 1 (strongly disagree)
3. Could you feel when the needle was in soft tissue?
(strongly agree) 7 6 5 4 3 2 1 (strongly disagree)
4. Was this similar to the real life feeling?
(strongly agree) 7 6 5 4 3 2 1 (strongly disagree)
5. Could you feel the loss of resistance from exiting the ligamentum flavum?
(strongly agree) 7 6 5 4 3 2 1 (strongly disagree)
6. Was this similar to the real life feeling?
(strongly agree) 7 6 5 4 3 2 1 (strongly disagree)

7. Were you able to feel the bone?

(strongly agree) 7 6 5 4 3 2 1 (strongly disagree)

8. If you inserted too far, were you able to feel the vertebral body on the opposite side?

(strongly agree) 7 6 5 4 3 2 1 (strongly disagree)