

A MULTILATERAL MANIPULATION
SOFTWARE FRAMEWORK FOR HUMAN-ROBOT
COLLABORATION IN SURGICAL TASKS

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF MECHANICAL
ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Kirk Nichols
December 2015

© 2015 by Kirk Arthur Nichols. All Rights Reserved.
Re-distributed by Stanford University under license with the author.



This work is licensed under a Creative Commons Attribution-Noncommercial 3.0 United States License.
<http://creativecommons.org/licenses/by-nc/3.0/us/>

This dissertation is online at: <http://purl.stanford.edu/xw523zh6135>

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Allison Okamura, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Mark Cutkosky

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

J Gerdes

Approved for the Stanford University Committee on Graduate Studies.

Patricia J. Gumport, Vice Provost for Graduate Education

This signature page was generated electronically upon submission of this dissertation in electronic format. An original signed hard copy of the signature page is on file in University Archives.

Abstract

In robot-assisted surgery, exploration and manipulation tasks can be achieved through collaboration among robotic and human agents. Collaboration can potentially include multiple agents working towards a shared objective – a scenario referred to as multilateral manipulation. We present a flexible software framework, called the Multilateral Manipulation Software Framework (MMSF) to expedite development of various multilateral manipulation strategies. The goal of the MMSF is to facilitate rapid development of human-robot collaborative models in a thread-safe manner. We demonstrate the effectiveness of implementing the MMSF in three mock surgical tasks: inclusion segmentation, debridement, and electrocauterization. We built autonomous agents capable of completing these tasks, and developed human-robot collaboration models using these autonomous agents. Example human-robot collaboration models tested in this work include (1) fully autonomous task execution, (2) shared control between a human and robotic agent, (3) supervised control where the operator dictates commands to the robot, (4) traded control between the two agents, and (5) bilateral teleoperation. For each of our three implemented surgical tasks, we demonstrate the nature of results achievable through use of the Multilateral Manipulation Software Framework by comparing the performance of different human-robot collaboration models. For the inclusion segmentation task, we conducted a user study where we compared the performance of different human-robot collaboration models against each other. This dissertation also describes implementations that allow the Multilateral Manipulation Software Framework to be used with Phantom Premium Haptic Devices, the RAVEN-II Surgical Robot, and the da Vinci Research Kit.

Acknowledgements

There are so many people to thank and acknowledge that led me to the person I am today; the academic side of which is reflected in this dissertation. The few words I write here are a gross underrepresentation of the impact these people had in my life, and I am ever-thankful for the love, support, and guidance that the people I am lucky enough to surround myself with provide.

Professionally, there have been a wealth of amazing mentors that channeled my academic curiosity into critical thinking and problem solving. Sara Marx, Doug Dietel, and Kevin Hammerly were instrumental in high school to channel my interests and develop my speaking and presenting abilities. My “unique” style of passionate and energetic presenting I owe to them. In undergrad I was fortunate enough to take classes from Dr. Anne Dougherty and Dr. James Curry, who not only encouraged me to continually “take more math”, but also funded me in a research position in Dr. Todd Murphey’s lab. Todd’s lab was my first exposure to research and robotics, and once I figured out I could get free travel by applying to academic conferences, I was hooked.

Dr. Allison Okamura took me into her lab as a rotating research student back when I first started at Johns Hopkins University, and in the beginning of 2011, accepted me into the CHARM lab at Stanford University. The unique thing about Allison is not only has she provided excellent guidance over the years throughout my PhD, but she has also become a cherished friend. Her family recognizes me from the past six Thanksgiving dinners and more than once Allison and I have had an impromptu run workout together after seeing each other on the Stevens Creek Trail. It is a rare occurrence to have such a talented PhD advisor and friend in one person.

This dissertation would not have happened without the generous grant from the National Science Foundation, and I am forever in their debt.

The rest of my dissertation and thesis defense committee deserve thanks as well: Dr. Mark Cutkosky, Dr. Chris Gerdes, Dr. David Camarillo, and Dr. Kenneth Salisbury. The contribution of this dissertation to the academic community has been greatly strengthened by their comments and support.

CHARM lab is great group of people, and I really know of no other lab that is so close knit. Over the past year three CHARMers got married, and I look forward to more reunions such as these in the future. Notable CHARMers that I have had extensive time interacting with are Michelle, Ann, Nick, Andrew, Sam, Tania, Sean, Joey, Darrel, and others. There are two rotators of mine the I would also like to thank: Lawrence Kim, who helped me debug hardware issues with the RAVEN-II, and Kirsten Kaplan, whose work over the past six months contributed to this dissertation, and is an endless supply of energy and baked goods.

There are an uncountable number of friends that have helped me balance academic life with living. Kristin Spencer and I seem to have spent as many hours working on classes together in high school as we did laughing, and her motivation and support in the time since has humbled me. ΘT also deserves mention, especially Chris, Billy, Jeff, and Peter.

My family has always been there for me and I was so fortunate to have them see my thesis defense either streaming over the Internet or in person. My mother has been endlessly supportive, although she was nervous when I first announced that I was moving to Baltimore. My step-mother has been a fantastic cheerleader as I progressed through school, and her excitement is contagious. My father sparked my passion in engineering, and I owe my career to the years of dinner conversations growing up where he would describe the latest cool engineering problem he was working on. The contributions I make as I progress in my career should always be credited to him first, and me second. My sisters were my foundation growing up, and continue to amaze me with the passion and dedication in their lives.

My family in California should also be mentioned, and by that I mean Dr. Troy Adebar. We have been officemates for over four years, roommates for over three years,

and interned twice together. His humor and advice throughout the years relieved countless obstacles in my PhD career, and one of the highest honors in my lifetime was being best man at his wedding. Germany the dog also deserves special mention in this category for her inexhaustible supply of indifference.

Finally, there's Miranda, a.k.a. face, a.k.a. puma. Who would have known that a chance meeting in a dark bar in Baltimore would have turned into the most supportive, playful, and energetic relationship I have ever had. I am ever thankful to her for my sanity, sense of wonder, and continuing interest in everything equestrian.

Contents

Abstract	v
Acknowledgements	vi
1 Introduction	1
1.1 Motivation	1
1.2 Background: Human-Robot Collaboration	3
1.2.1 Teleoperation	4
1.2.2 Shared Control	4
1.2.3 Traded Control	6
1.2.4 Supervisory Control	7
1.3 Background: Robot Software Frameworks	8
1.4 Contributions	8
1.5 Dissertation Overview	10
2 Autonomous Agent for Inclusion Segmentation	12
2.1 Introduction	13
2.1.1 Background	14
2.1.2 Summary of Contributions	16
2.2 Inclusion Segmentation Algorithm	17
2.2.1 Design Goals	17
2.2.2 Initialization	19
2.2.3 Exploration of Tissue	21
2.2.4 Expectation-Maximization Steps	21

2.2.5	Iteration Procedure	28
2.2.6	Exiting Procedure	29
2.3	Simulation of the Inclusion Segmentation Algorithm	30
2.3.1	Setup	30
2.3.2	Simulation Results	30
2.4	Experiment 1: Accuracy	33
2.4.1	Artificial Tissue	34
2.4.2	Generation of Classifier	35
2.4.3	Evaluation Metrics	39
2.4.4	Results	40
2.5	Experiment 2: Robustness with Respect to Initial Classifier Value . .	41
2.5.1	Artificial Tissue	41
2.5.2	Procedure	42
2.5.3	Results and Discussion	42
2.6	Conclusions	44
3	The Multilateral Manipulation Software Framework	45
3.1	Software Framework	46
3.1.1	Base Class Structure	46
3.1.2	Inter-class Communication	49
3.1.3	Threading	51
3.2	Collaboration Models for Inclusion Segmentation Task	51
3.2.1	Bilateral Teleoperation	52
3.2.2	Supervised Control	55
3.2.3	Fully Autonomous	55
3.2.4	Shared Control	56
3.2.5	Traded Control	57
3.3	Experiment	58
3.3.1	Artificial Tissue	58
3.3.2	Robotic Palpation	59
3.3.3	Graphical User Interface	59

3.3.4	Evaluation Metrics	60
3.3.5	Protocol	61
3.4	Results and Discussion	61
3.4.1	Software Framework	61
3.4.2	Inclusion Segmentation	63
3.5	Conclusions	64
4	Implementations and Extensions of the Multilateral Manipulation	
	Software Framework	66
4.1	Implementation of the MMSF on Surgical Robots	67
4.1.1	Implementation of the MMSF on the da Vinci Reserach Kit	67
4.1.2	Implementation of the MMSF on the RAVEN-II	69
4.2	Human-Subject Study of Human-Robot Collaboration Models for an Inclusion Segmentation Task	71
4.2.1	Implementation of the Inclusion Segmentation Task	72
4.2.2	Experiments	77
4.2.3	Results and Discussion	82
4.2.4	Conclusions	86
4.3	Extending the Multilateral Manipulation Software Framework to a De- bridement Task	87
4.3.1	Introduction	88
4.3.2	Methods	89
4.3.3	Experiment	93
4.3.4	Results and Discussion	95
4.3.5	Conclusions	98
4.4	Extending the MMSF to Combined Stiff Region Exploration and Elec- trocauterization	98
4.4.1	Introduction	98
4.4.2	Extension of the MMSF to the Electrocautery task	100
4.4.3	Experiment	104
4.4.4	Results and Discussion	108

4.4.5	Conclusion	110
5	Conclusion	112
5.1	Synopsis of Contributions	112
5.2	Future Work	115
	Bibliography	119

List of Tables

2.1	Notation	22
2.2	Robustness testing for the inclusion segmentation algorithm	42
3.1	Class instantiation table for inclusion segmentation task	54
3.2	Experimental results across collaboration models for the inclusion segmentation task	65
4.1	Data Packet from dVRK to MMSF	69
4.2	Data Packet from MMSF to dVRK	70
4.3	Class instantiation table for debridement task	91
4.4	Experimental results for debridement task	97
4.5	Experimental results for electrocautery task	111

List of Figures

2.1	Inclusion segmentation algorithm flow chart	18
2.2	Initializing grid for inclusion segmentation algorithm	20
2.3	Neighborhoods and cliques around point i	24
2.4	Illustration of the boundary point finding algorithm	29
2.5	Simulated inclusion segmentation results across several iterations . . .	31
2.6	Simulated inclusion segmentation results for four environments	33
2.7	Elastography data acquisition	36
2.8	Experimental setup for inclusion segmentation	37
2.9	Example force-displacement curve from palpation	38
2.10	Training the GDA classifier	39
2.11	Evaluation metrics	40
2.12	Experimental results for the inclusion segmentation task	41
3.1	UML diagram of select MMSF base classes	47
3.2	UML diagram of remaining base classes	49
3.3	Sequence diagram of notifier/notifyee communication	50
3.4	Collaboration models tested for inclusion segmentation task	53
3.5	Experimental setup for MMSF in inclusion segmentation task	58
3.6	Evaluation metrics	60
3.7	Experimental results for all collaboration models	62
4.1	The da Vinci Research Kit	68
4.2	The RAVEN-II Surgical Robot	71
4.3	Collaboration models for human-subject study	73

4.4	Illustration of boundary identification algorithm	76
4.5	Experimental setup for human-subject study	78
4.6	Human-subject study results for inclusion segmentation experiment .	83
4.7	Human-subject study results for boundary identification experiment .	85
4.8	Sensitivity and specificity for boundary identification experiment . . .	87
4.9	Class hierarchy diagram for debridement task	89
4.10	Collaboration models for debridement task	90
4.11	Debridement algorithm flow chart	92
4.12	Experimental setup for debridement task	94
4.13	Graphical user interface and <i>Cut Robot Action</i> for debridement task .	96
4.14	The male reproductive organ	99
4.15	Collaboration models tested for electrocautery task	102
4.16	Class hierarchy diagrams for the stiff region exploration and electro- cauterization subtasks	104
4.17	Experimental setup for the stiff region exploration and electrocauteri- zation subtasks	106
4.18	Sensitivity and specificity results for the surgical procedure	109

Chapter 1

Introduction

1.1 Motivation

Human-robot collaboration is becoming increasingly common in applications such as robotic surgery [1], search and rescue [2], and space [3]. In each of these applications, an operator sits at a master console and interacts with a robotic device (the master robot) while a slave robot follows the operator's motions in real time. As historically teleoperated tasks become either too demanding for a human user to completely control, or too repetitive to justify full human attention, there is a need to formalize the ways in which humans and robotic agents can work together to achieve a shared goal. However, it is unlikely that a one-size-fits-all combination of a human operator and a robotic agent input would improve all human-robot collaborative procedures, which motivates the need for a framework to describe the cooperation between the two agents and enable a spectrum of collaboration models for a specific task. This dissertation presents a tool, the Multilateral Manipulation Software Framework (MMSF), which is used to structure a collaborative task to facilitate rapid development of human-robot collaboration models. The MMSF is created and used in this work as a development tool in the early stages of framing a human-robot collaborative task, to allow the system designer to better understand which types of collaboration models result in the best performance. We take an object-oriented approach towards the MMSF to minimize redundant code and enable rapid development. The success of

the MMSF presented in this work is dependent on the following, to demonstrate wide applicability of the MMSF across tasks, robotic platforms, and collaboration models:

- Extending the base structure of the MMSF to several human-robot collaborative tasks.
- For each task, application of the class structure to realize a variety of different human-robot collaboration models.
- Implementing the MMSF on a surgical procedure consisting of multiple sub-tasks, possibly with switching human-robot collaboration models between sub-tasks.
- Demonstrating the MMSF on several different robotic platforms.
- Ensuring thread-safe communication between processes in the MMSF.

We choose mock surgical tasks to demonstrate the use of the MMSF because the field of robot-assisted minimally invasive surgery (RMIS) is opportune for exploring human-robot collaboration. RMIS is becoming increasingly popular in part because it can provide increased dexterity and control to the surgeon compared to open or laparoscopic surgery through the use motion scaling, stereoscopic visualization, and articulated wrists [1]. Established RMIS procedures include radical prostatectomy [4], pancreatectomy [5], hysterectomy [6], and throidectomy [7]. In these clinical applications of RMIS, the surgery was teleoperated under complete control of the human surgeon. Teleoperation has become the industry-standard control scheme for the majority of clinical surgical robotic platforms, including the da Vinci Surgical Robot (Intuitive Surgical, Sunnyvale, CA, USA) and the Magellan Robotic System (Hansen Medical, Mountain View, CA, USA). While advancements in machine learning, artificial intelligence, and computer vision continue to provide richer quantitative analysis of the surgical field and robotic control, most commercial medical robots do not yet automate surgical procedures. In contrast, the academic community has characterized and automated a number of surgical sub-procedures, such as debridement [8], drilling bone during chochleostomy [9], and surgical knot tying [10, 11]. It is clear

from advancements in academia and the increasing usage of robotics in the operating room that more advanced forms of human-robot collaboration are on the horizon – and we must identify safe, effective methods to leverage the inputs of both a surgeon and artificial intelligence. The goal of these human-robot collaborative systems is to combine the advantages of human and artificial agents to increase success in surgical procedures, which is ultimately defined by better patient outcomes. In this dissertation, we demonstrate the MMSF by extending it towards multiple surgical subtasks, including inclusion segmentation, debridement, and electrocauterization. We frame these tasks in the context of the MMSF and demonstrate how this formalization of our collaborative tasks leads to rapid prototyping of new models of human-robot collaboration. We furthermore show examples of experimental results and conclusions one can draw by comparing different models of human-robot collaboration in these experimental tasks.

1.2 Background: Human-Robot Collaboration

In our review of prior art in the field of human-robot collaboration models, we use the taxonomy introduced by Conway *et al.* [12]. The spectrum of human-robot collaboration models has endpoints of teleoperation and full autonomy. These collaboration models define the extremes of this spectrum because neither collaboration model combines input from both the human and autonomous agents. All other human-robot collaboration models in some way combine inputs from both agents, whether via linear summation (commonly used in shared control), switching inputs (traded control), or supervised control, where a human agent provides high-level input to be interpreted and carried out by a robotic agent. For the works cited in our review of prior art, and in this dissertation, the human agents communicate their intent by interacting with either a master manipulator in a teleoperation scheme, or by interaction with a graphic user interface. However, human-robot collaboration can extend beyond these two constraints dictating the nature of human agent interaction, and an excellent review of different methodologies of human-robot collaboration beyond the scope of this work, including collaboration using visual or auditory cues, has been conducted

by Greena *et al.* [13].

1.2.1 Teleoperation

In teleoperation, a slave robot mimics the actions of a master robot controlled by an operator. This collaboration model was used in the previously mentioned works of [4, 5, 6, 7]. All these works used the da Vinci surgical robot as the robotic platform, where both master and slave robots were located in the same room. In another example, Marescaux *et al.* published a study where clinicians performed the first transatlantic gallbladder removal [14]. This surgery was conducted through the ZEUS robot (Computer Motion, Sunnyvale, CA). Teleoperation in the above studies was conducted unilaterally, such that master manipulator position dictates joint torques commanded to the robot. In bilateral teleoperation, robot end-effector position errors between the master and slave robots are transmitted as forces feedback to the master device. A recent work using bilateral teleoperation control to reduce hand tremor while providing haptic feedback to the user during robotic surgery is presented by Beretta *et al.* [15]. They designed a torque controller with non-linear force feedback to augment haptic perception to the operator and stabilize operator commands to the robot. Another similar example of force feedback being used to augment operator was conducted by De Lorenzo *et al.* [16]. In their work, force-feedback was used to convey forces on a needle tip during insertion back to the operator. A similar methodology was used by Yu *et al.*, but in their case the force-feedback was used to provide cues of robotic agent intent during bilateral teleoperated control of a mobile robot [17].

1.2.2 Shared Control

In shared control, the motions of the slave robot are governed by a summation of human and robot agent input. An application of shared control for commanding a surgical robot is the steady-hand robot, developed by Taylor *et al.* [18]. Their system assists in microsurgery manipulation tasks by filtering out natural surgeon tremor and thereby assisting in surgical tasks which require extreme precision. Another example of shared control is virtual fixtures; Bowyer *et al.* provides a detailed survey [19].

Virtual fixtures are often used in navigation tasks. Both Marayong *et al.* [20] and Abbott *et al.* [21] demonstrated improved performance during path following tasks with human input augmented by virtual fixtures. Marayong *et al.* concluded that results might improve for path following tasks with a dynamic control authority ratio, instead of a static control authority ratio. Dynamic control authority ratios have been explored in the works of Oguz *et al.* [22], Passenberg *et al.* [23], and Dragan *et al.* [24]. All three of these works optimize over an objective function to choose the ideal control authority. However, each of these works mapped their optimization result to ideal control authority ratio in different ways. Oguz *et al.* partitioned control authority ratios into three states [22], while Passenberg *et al.* instead chose a continuous mapping [23]. Dragan *et al.* had the robotic agent infer the user’s intent and adjust the control authority ratio depending on the level of confidence in its prediction [24].

Stefanov *et al.* changed the control authority based on the identified subtask within a larger robotic procedure, where each subtask would use a different assistance function to determine the degree to which the robot agents assist the human agent [25]. A recent implementation of shared control is in the work of Shamaei *et al.* [26]. They developed an autonomous agent to support a human or robotic agent conducting a tension-and-cut task. In their control formulation, the human set a parameter that varies the linear summation of human and robotic agent input for a single manipulator performing the cutting action, while another manipulator in the same workspace autonomously tensions a sample.

In the above examples of shared control, the ratio of human to autonomous input when controlling a slave device, or control authority ratio, was modified to weigh the inputs of the human and robotic agent. Dimension reduction is a different manifestation of shared control, in which human control of the robot is made more intuitive by reducing the dimensionality of the human agent’s control space. Tuna *et al.* [27] and Moustris *et al.* [28] studied the use of shared control in beating heart surgery when they investigated ways to make the surgery easier by negating the motion effects of heartbeat. Tuna *et al.* published multiple adaptive algorithms to predict heart motion, which can be used in a study like the one conducted by Moustris *et al.* In their

work, a surgeon operated on a motion-compensated image of the heart instead of the live video feed. Surgeon actions on the still image were then transformed back into operation space by the robotic agent.

In other examples of dimension reduction in shared control, Debus *et al.* demonstrated that for a pick-and-place task moving pegs into holes, where a robotic agent oriented the pegs while the human agent teleoperated the robot’s position, use of dimension reduction resulted in improved performance [29]. This is similar to the work done by Kofman *et al.* where they tasked a human to teleoperatively command a robot to grasp objects [30]. In close proximity to the desired object, the robot would autonomously reorient itself to the best grasping position. Work in dimension reduction continues to be an active field of publication, for example Wang *et al.* recently used dimension reduction so that a teleoperated system could avoid obstacles; the human only controls the end effector position while the other joints of the robot autonomously avoid collision with the surrounding environment [31].

Finally, Crandall *et al.*, when investigating human-robot collaboration in a navigation task, defined a “neglect factor”, which mathematically describes the degree of operator distraction when controlling multiple robotic agents [32]. They proposed and implemented a mathematical framework to determine the ideal amount of autonomous control as a function of the neglect factor and task difficulty.

1.2.3 Traded Control

In traded control, the robotic procedure is at times automated and at other times teleoperated. In the literature, this collaboration strategy has limited exposure, with both Parasuraman *et al.* [33] and Kortenkamp *et al.* [34] developing software architectures to facilitate the tradeoff between fully teleoperated and autonomous modes. Parasuraman *et al.* decomposed a given human-robot collaborative task into several top-level states comprising of several sub-states. From this decomposition, the authors created criteria that when satisfied warrant transfer of control authority from the human agent to the robotic agent.

Traded control has been implemented on the da Vinci Surgical Robot by Padoy

and Hager [35]. They decomposed a suture task into six different subtasks, which were derived from a series of expert demonstrations. With their human-machine collaborative approach, they were able to seamlessly transition from teleoperative to autonomous control of the slave manipulator between subtasks. Subtasks which required tissue manipulation, such as inserting the needle into the tissue simulant, were under teleoperative control, while subtasks which did not manipulate the tissue, such as needle passing from one robotic arm to the other, were under automated control.

1.2.4 Supervisory Control

In supervisory control, high-level tasks are specified by human input and low-level tasks are completed under autonomous control. This type of collaboration has been studied in the swarm robotics community, most notably in the works of Shirkhodaie *et al.* [36] and Parasuraman *et al.* [37]. Shirkhodaie *et al.* proposed a software framework that used a supervisory model to control several robots as they coordinated with each other. Parasuraman *et al.* enacted supervisory control as well; an operator gave waypoint commands to a swarm of robots while the robots were individually responsible for obstacle avoidance. Another multi-agent coordination control work is the research conducted by Chen *et al.* [38]. In their work, a human agent would supervise a swarm of UAVs as they navigated a course. Individual UAVs would communicate their autonomous intent to the human operator and prompt for supervisory input when deemed necessary. Chen *et al.* noted that when automation intent was communicated, the human agent had increased situational awareness while reducing cognitive workload. Another recent result by Sieber *et al.* formulated a mathematical model used in an experiment where a human operator issued commands to a team of robots that were cooperatively manipulating a deformable environment [39]. Through their implementation of shared control, a single human agent was able to accomplish a task which otherwise would have taken a team of operators to accomplish.

1.3 Background: Robot Software Frameworks

There are two prominent open-source software frameworks used widely in the research and industrial communities. Researchers at Johns Hopkins University, Massachusetts Institute of Technology, and Carnegie Mellon University developed and continue to maintain CISST-SAW [40], a component-based software architecture implemented on a variety of surgical robotic systems. CISST-SAW was developed to simplify the execution of a robotic surgical suite to a single codebase. CISST-SAW has seen widespread adoption with the da Vinci Research Kit (dVRK) [41], a robotic hardware system deployed at over a dozen universities. Willow Garage developed the Robot Operating System, (ROS) [42]. ROS is a collection of tools, libraries, and conventions that aim to simplify robotic tasks in a way that is robust and widely applicable across different robotic systems. ROS is the default coding environment for the PR2 robot [43], and has a large active community contributing hundreds of toolboxes to advance the robotics community (see www.ros.org). Both CISST-SAW and ROS specialize in sensing and actuation, simplifying interfaces among a large variety of robots and sensors to consistent interfaces that ease the development cycle.

1.4 Contributions

The previous human-robot collaborative works introduced above demonstrate and test collaboration models mostly in isolation, with only a few examples comparing two collaboration models. In this previous research, a software framework with the capability of implementing a variety of collaboration models was not developed or used likely because researchers focused on the design and testing of individual collaboration models. In contrast, this dissertation facilitates the development and testing of many collaboration models, for a variety of tasks. This approach is necessary because the capabilities of modern autonomous agents enables a wider variety of difficult tasks to be performed with some level of autonomy, resulting in the need to formalize the development of human-robot collaboration models. With the MMSF, a developer can test collaboration models not in isolation as is commonly done, but rather against a

host of other types of human-robot collaboration models. We present in this work the first such software framework to our knowledge designed to be used in the development stages of a human-robot collaborative task. We furthermore demonstrate the first instance of switching collaboration models between subtasks within a larger surgical procedure, acknowledging that a one-size-fits-all collaboration model between a human and autonomous agent may not work across all applications. The Multilateral Manipulation Software Framework is in part motivated by the successes in ROS and CISST-SAW, which have been widely accepted in the robotics community in part because they abstract complex behaviors and sensing into well-specified software classes. We seek to demonstrate that same abstraction and flexibility in the application area of human-robot collaboration models. The Multilateral Manipulation Software Framework, and its extensions used throughout this dissertation, is open-source and can be accessed from <http://charm.stanford.edu/Main/Resources>.

We show the contribution of the MMSF by extending it to three different mock surgical tasks and implementing it on three different robotic platforms. For each surgical task, we must develop an autonomous agent fully capable of completing the task without human input. For the inclusion segmentation task described in the next chapter, this agent is a contribution of this dissertation in its own right.

The major contributions of the research described in this dissertation can be summarized as follows:

- Developed an autonomous agent capable of segmenting stiff inclusions from surrounding soft tissue using data acquired via autonomous palpation. This is the first instance of a robot exploring and segmenting a stiff inclusion from surrounding soft tissue. This autonomous agent was developed so that it could be within the human-robot collaboration models implemented in the other contributions.
- Developed the Multilateral Manipulation Software Framework, a flexible software framework designed to expedite development of various human-robot collaboration models.
- Extended the Multilateral Manipulation Software Framework to three mock

surgical tasks: inclusion segmentation, debridement, and electrocauterization. This contribution includes development of autonomous agents to perform these tasks in addition to forming collaboration models that combine inputs from both the human and robotic agents.

- Implemented the Multilateral Manipulation Software Framework on three different platforms: Phantom Premium devices, the RAVEN-II Surgical Robot, and the da Vinci Research Kit.
- Demonstrated the performance of various human-robot collaboration models in an inclusion segmentation task by conducting a user study. Results indicate a trade-off between robotic agent precision and accuracy compared to human agent speed.

1.5 Dissertation Overview

This dissertation is composed of five chapters. Chapter 1, this chapter, motivates the development of the Multilateral Manipulation Software Framework and provides background on implementations of human-robot collaboration. Chapter 2 develops and demonstrates a fully autonomous robotic agent capable of segmenting stiff inclusions from surrounding soft tissue. Chapter 3 describes the Multilateral Manipulation Software Framework in detail, and demonstrates a usage of the framework by combining a human agent with the autonomous agent developed in Chapter 2 to form five different human-robot collaboration models capable of performing the inclusion segmentation task. Chapter 4 is divided into four sections. Section 4.1 describes implementation details necessary to configure the MMSF to interface with the da Vinci Surgical Robot, used for the experiments in Sections 4.2 and 4.4, and the RAVEN-II surgical Robot, used for the experiment in Section 4.3. Section 4.2 describes a user study across collaboration models for the inclusion segmentation task where we compare different collaboration models against each other. Section 4.3 extends the framework to a debridement task implemented on the RAVEN-II Surgical Robot. We demonstrate four different collaboration models and present results that demonstrate

the type of comparisons one can make by using the MMSF in a debridement task. The Section 4.4 describes the extension of the Multilateral Manipulation Software Framework to a surgical procedure consisting of two subtasks done in series: stiff region exploration and electrocauterization along the boundary. We compare results of different pairings of collaboration models between subtasks in a pilot study. Finally, Chapter 5 summarizes the results of the research, reviews the contributions of this dissertation, and posits future directions for continued work into human-robot collaboration.

Chapter 2

Autonomous Agent for Inclusion Segmentation

To test human-robot collaboration models implemented by the Multilateral Manipulation Software Framework, we must first develop autonomous agents to serve as the agent with which the human will collaborate. We are interested in surgical tasks, and wanted to start our investigation into human-robot collaboration by choosing a relatively simple task that can benefit from human and robotic agents working together.

In this chapter we develop an autonomous agent capable of segmenting a stiff inclusion from surrounding soft tissue. The autonomous agent is capable of tissue exploration and implements machine learning algorithms to develop a stiffness classifier used to label autonomously palpated points as part of the stiff inclusion or part of the surrounding soft tissue. To generate training data for the learning algorithms, we use Ultrasound Elastography, which generates an inclusion-specific characterization of tissue mechanics. Once we have identified the embedded hard inclusion in the elastographic image, Gaussian discriminant analysis generated a classifier to threshold stiffness values acquired from autonomous robotic palpation. This classifier was later used to classify newly acquired points as either part of the inclusion or surrounding soft tissue. An expectation-maximization algorithm with underlying Markov random fields improved this initial classifier over successive iterations to better approximate

the boundary of the inclusion. Results of the autonomous agent demonstrate robustness with respect to inclusion shape, size, and the initial classifier value. The work in this section is summarized in a paper published in the IEEE Transactions on Robotics [44].

2.1 Introduction

In open surgery, tactile sensation enables surgeons to perform a wide array of diagnostic operations. Such sensation is especially important when profiling cancerous regions in soft tissue structures such as prostate and breast. Cancerous tissues typically form lumps that are significantly stiffer than the surrounding tissues [45], and exhibit different stiffness depending on which organ is diseased [46]. During open surgery, surgeons can directly contact and palpate the tissue, and thus can use tactile sensation to localize tumors.

Minimally invasive laparoscopic surgery has become increasingly popular in the excision of a variety of cancers, including ovarian, colorectal, and prostate cancer [47, 48, 49]. Tactile cues in this setting are not directly felt by the surgeon, but instead are filtered via the laparoscopic tool to the surgeon’s hand. In current clinical robot-assisted minimally invasive surgery (RMIS), the surgeon is not in contact with the environment, either directly or indirectly. In this setting, haptic cues become almost non-existent, although a number of methods have been reviewed to explore the range of force-feedback and sensory substitution in RMIS [50].

RMIS is becoming an increasingly popular form of minimally invasive surgery because it provides increased dexterity and control for the surgeon, in comparison to manual minimally invasive surgery, while reducing patient trauma in comparison to open surgery. However, the lack of haptic feedback in RMIS limits surgeons’ capabilities to identify tissue mechanics. A study has demonstrated that RMIS can increase the likelihood of leaving behind cancerous cells upon extrication of a diseased region compared to open surgery [51]. A method to convey the shape, location, and mechanical properties of a suspected tumor would restore some of the information lost with the use of RMIS.

To this end, computer vision and machine learning methods can be applied to classify irregularities in soft tissue. After characterization of the mechanical properties of the tissue, machine learning methods can provide segmentation estimates of any hard tumors present. In addition to automatic inclusion segmentation, the locations of any tumors can be derived in the robot's coordinate system, which is important for planning and robotic assistance. In this chapter, we seek to automate acquisition of tissue mechanics data and develop a tumor segmentation strategy capable of overcoming the limitations of RMIS and accurately segment hard inclusions in soft tissue.

2.1.1 Background

Several groups have developed methods that allow detection, and in some cases segmentation, of lumps in soft tissue. Egorov and Sarvazyan developed a device that uses a pressure-sensitive sensor array to take mechanical images of the breast [52]. Held and operated similar to an ultrasound transducer, the operator maneuvers the probe across the surface of the breast and a software interface displays stiffness information and segments the inclusion. Egorov and Sarvazyan validated their design in a differentiation study between benign and malignant breast tumor samples [53], where they achieved an average sensitivity of 91.4% and specificity of 86.8%. The sensitivity and specificity metrics reported in [53] were generated by comparing the accuracy of labeling an actual tumor as benign or malignant. (In contrast, our results presented in this chapter use the metrics of specificity and sensitivity, which compare the areas of a hard inclusion to the surrounding artificial soft tissue.) A minimally invasive technique for stiffness distribution mapping was developed by Beccani *et al.* [54]. They designed a wireless probe that when grasped by a laparoscopic tool can be used to palpate a tissue sample and create a stiffness map. Their volumetric stiffness measurements posted an 8% local stiffness error compared to measurements gathered with an *ex vivo* sample of porcine liver.

Other groups used robotics to enable the mapping and segmentation of lumps in soft tissue. Howe *et al.* used a teleoperation scheme with haptic feedback to convey

the mechanics of a palpated artificial tissue. Use of their tactile display increased success of tumor identification and localization by a human user [55]. Trejos *et al.* showed that autonomous robotic tumor palpation resulted in increased tumor detection accuracy and a significant decrease in the maximum force applied compared to a teleoperated palpation scheme [56]. Ahn and Kim teleoperated a robot that mimicked the geometry and motions of a doctor’s hand. A robotic finger swept across the prostate and used the resultant force profile to assess the likelihood of tumors [57]. Sangpradit *et al.* implemented a model-based approach [58]. By comparing observed discrepancies between a finite element model approximating a wheel-tissue interaction and force data taken during rolling-contact experimentation, they were able to identify simulated tumors of diverse shapes and depths. Liu *et al.* used a force-sensitive wheeled probe to gather a “rolling mechanical image” and found their continuous measurement approach to be more sensitive to differences in force profiles caused by simulated tumors than single-site data acquisition [59]. While any of these methods may be used to acquire the tissue mechanical properties used in this chapter, here we use vertical palpation to acquire stiffness data. This builds on the methods of [60], where the authors used vertical palpation data to map the stiffness profile of a simulated calcified artery.

Segmenting a hard inclusion using any of the above approaches requires a method to analyze and interpret a stiffness image. Several research groups have accomplished this task by applying machine learning methods to identify hard inclusions in several types of medical images. Zacharaki *et al.* used support vector machines to identify the shape and grade of tumors in MRI images [61]. Li *et al.* explored the use of machine learning algorithms to identify the boundaries of tumors in CT images [62]. Both these studies required intervention from experts to train the learning algorithms by either identifying regions of interest [61], or by marking the rough location of the tumor [62]. In earlier work [63], we approximated the centroid of a suspected tumor from a stiffness mapping of tissue mechanics generated from palpation data across the surface of an artificial tissue. We compared the centroid locations generated from classified points across different machine learning algorithms. Our study showed promise in identifying the centroid of the suspected tumor. However, to identify the

boundary of the inclusion, our goal in this study, the classifier would need to be refined in real-time using acquired data to increase the sensitivity of the classifier and thereby reduce the number of false negative classifications.

Supervised machine learning algorithms (used in [62, 61, 63]) require an initial feature and labeling set, called the training set, to train the algorithm’s classifier. In the case of segmenting a hard inclusion from soft tissue, elastography can provide a non-invasive method of developing a relative stiffness image of soft tissues. In this study, we perform elastography analysis on ultrasound images. Ultrasound elastography compares radio-frequency ultrasound data captured during compression of the sample of interest and generates a relative stiffness map of soft tissue [64]. During a time-varying compression, points within an image can be tracked for their relative movements and subsequently be assigned a relative stiffness value. Elastography has been used to characterize tumors in a variety of tissues, including breast [65], liver [66], and prostate [67]. A classifier developed from an elastographic image can be useful in initially labeling points as part of a hard inclusion or part of the surrounding soft tissue. However, these classifiers are not robust to noise in the image. Furthermore, using elastography to classify and segment hard inclusions requires registration with the surgical robot, which is particularly difficult in non-rigid environments such as soft tissue [68].

2.1.2 Summary of Contributions

This work is the first to segment hard inclusions from tissue by applying machine learning methods to data acquired via autonomous robotic palpation. Furthermore, this is the first work that drives autonomous exploration of tissue mechanics in addition to autonomous segmentation. We present a tumor segmentation algorithm using machine learning methods to improve a stiffness classifier developed from elastography over successive iterations while converging to the boundary of the inclusion. We demonstrate the proposed algorithm with computer simulations of virtual tissues and through repeatable experiments with artificial tissues. The ratio of stiffness between

hard inclusion and soft tissue that we can detect is between 2.0 – 3.0, which is motivated from the stiffness ratio consistent with prostate tumors [45]. Results indicate that the tumor segmentation algorithm presented in this chapter consistently identifies the boundary of the hard inclusion embedded in soft tissue with high sensitivity and specificity.

This chapter is organized as follows. Section 2.2 introduces the structure and theory of the tumor segmentation algorithm. Section 2.3 presents simulated results. In Sections 2.4 and 2.5, two experiments are conducted. In Section 2.4, we test the ability of the tumor segmentation to repeatedly segment a hard inclusion from its surrounding soft tissue. In Section 2.5, we test the robustness of the initial classifier used in the tumor segmentation algorithm. Sensitivity and specificity metrics measure performance in both these experiments. In Section 2.6, we make concluding remarks and posit future directions.

2.2 Inclusion Segmentation Algorithm

We present in this section the structure and theory of the inclusion segmentation algorithm whose flowchart is shown in Figure 2.1. After introducing the design goals of the algorithm in the first subsection, each of the states in Figure 2.1 are explained in subsequent subsections.

2.2.1 Design Goals

2.2.1.1 Develop a strategy to label points as part of the inclusion or part of the surrounding soft tissue

In our labeling strategy, we seek to use not only an individual point's stiffness for its classification, but also the stiffnesses around this point. We use conditional random fields to generate the probability distribution of a point's possible classifications, in our case points classified as part of the hard inclusion or part of the surrounding soft tissue. Conditional random fields are a statistical modeling method that enable machine learning algorithms to take advantage of interactions between neighboring points,

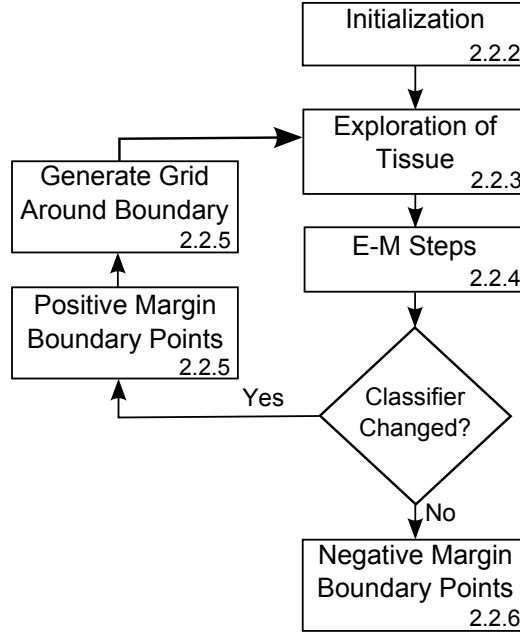


Figure 2.1: Inclusion segmentation algorithm flow chart. Each process block contains the section number describing its execution steps. For example, the “Exploration of Tissue” process is described in detail in Section 2.2.3.

such that one point’s classification depends on the properties of its neighbors. Li *et al.* in [69] applied Markov random fields, an example of conditional random fields, to segment digital mammograms and achieved a sensitivity of 90% and a specificity of 94%. Conditional random fields have also been used extensively to segment tumors in the liver [70], brain [71], and prostate [72]. We apply Markov random fields to segment a stiffness image in this chapter as part of our tumor segmentation algorithm.

2.2.1.2 Develop a classifier that is robust to noise in the training set

Stiffness classifiers developed from elastography are sensitive with respect to noise in the elastographic image, and only provide a rough estimate of which points are part of stiff inclusions and which are part of surrounding soft tissue [63]. Parameterizing a conditional random field algorithm, in our case with the optimal stiffness classifier, is difficult and is sometimes posed as a machine learning algorithm all to its own. We

use an expectation-maximization (EM) scheme to apply a conditional random field machine learning algorithm as well as improve its parameterization. This approach has been used in works to classify tumors in the breast [73] and brain [74]. Our approach improves a classifier in an EM algorithm using underlying Markov random fields to model stiffness data acquired via autonomous palpation data.

2.2.1.3 Develop an efficient tissue exploration procedure

We define an efficient tissue exploration procedure to be one that in detail maps regions theorized to belong to the hard inclusion and in low resolution maps the surrounding soft tissue. The structure of the inclusion segmentation algorithm, whose flowchart is shown in Figure 2.1, satisfies our third design goal. This algorithm was tested in both simulated (Section 2.3) and experimental (Sections 2.4, 2.5) settings. The algorithm begins with a seeding classifier and an initial grid of points to palpate. Then, the inclusion segmentation algorithm enters an iterative phase, which includes first mapping tissue mechanics with autonomous robotic palpation (investigating around points theorized to belong to the hard inclusion in additional detail) and second updating the stiffness classifier through an EM scheme using Markov random fields. This stiffness classifier is designed to label points as either belonging to the hard inclusion or surrounding soft tissue. If the new stiffness classifier developed using the EM scheme differs from the stiffness classifier seeded into the EM scheme, boundary points are identified, and a new grid of points to explore is calculated around the supposed boundary of the inclusion. However, if the stiffness classifiers have converged, the algorithm transitions to its final stages. The convex hull or moving average of the final boundary points are interpolated using Bezier curves, resulting in a smooth approximation of the boundary of the inclusion.

2.2.2 Initialization

Initialization of the inclusion segmentation algorithm requires two parameters: a classifier and a grid to palpate. In simulation, the initial classifier used to sort palpated points as part of the hard inclusion or part of the surrounding soft tissue is an input

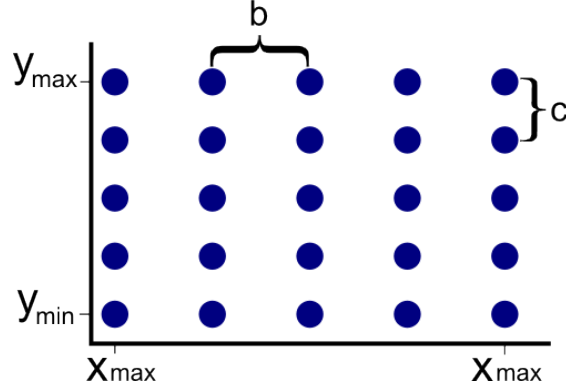


Figure 2.2: Initial grid of points to palpate with the autonomous inclusion segmentation algorithm. The parameters include $x_{min}, x_{max}, y_{min}, y_{max}$ define the boundary of our workspace, b the distance between points in the x direction, and c , the distance between points in the y direction.

variable set beforehand. In experiment, the initial classifier is derived using elastography data, described later in Section 2.4.2.1. In both experiment and simulation, the initial grid is a sparse linear grid over the workspace (See Figure 2.2). The initial grid of points to explore, G , is defined as follows:

$$G = \{(x, y) | x \in [x_{min}, x_{max}], y \in [y_{min}, y_{max}], \\ (x - x_{min}) \% b = 0, (y - y_{min}) \% c = 0\} \quad (2.1)$$

Where $x_{min}, x_{max}, y_{min}, y_{max}$ define the boundary of our workspace, b, c are the distances between points in the x and y directions respectively, and $\%$ is the modulus operator. A general guideline is that b and c should be chosen such that they are both less than the expected diameter of the hard inclusion, to ensure that during the autonomous agent's initial exploration of the tissue, at least one palpated point is sufficiently stiff to be labeled as part of the hard inclusion. (For simulation of the inclusion segmentation algorithm presented in Section 2.3, we chose b and c such that the initial grid was 10×10 . For our experiments, shown in Sections 2.4 and 2.5, we

chose b and c were chosen such that the initial grid was 7×7 .)

2.2.3 Exploration of Tissue

In Figure 2.1, tissue mechanics are mapped in the process “Exploration of Tissue”. As the robot autonomously palpates a tissue, if a mapped point is sufficiently stiff to be labeled as part of the hard inclusion, exploration and mapping of tissue mechanics continues in a local neighborhood around the stiff point before returning to the original exploration grid. If a newly mapped point in this neighborhood around the original point is also sufficiently stiff to be classified as belonging to the hard inclusion, the neighborhood around this newly mapped point will be explored. This adaptive-resolution exploration algorithm is similar to the works of Goldman *et al.*, which among other contributions uses an adaptive-resolution scheme to map impedances of a tissue sample [75].

Before a prospective point is palpated, its location is compared to all other points already palpated and constituting our dataset. If one of these other palpated points is sufficiently close to our prospective point (less than 5mm in experiment), then our prospective point is not palpated. This regulation of the minimum distance between points is to avoid oversampling points in close proximity to each other, which would drastically increase the calculation time of the expectation-maximization steps described in the next section.

2.2.4 Expectation-Maximization Steps

Prior work established that machine learning methods can be used to approximate the centroid locations of hard inclusions in soft tissue [63]. However, the classifier developed in this prior work was too conservative and had a low sensitivity, resulting in many points falsely identified as part of the surrounding tissue rather than part of the hard inclusion. Here, we expand the classifier method used in [63] using an expectation-maximization scheme with underlying Markov random fields to improve the classifier over successive iterations. Notation used in this section is introduced in Table 2.1.

Table 2.1: Notation

Symbol	Description
f	Set of labelings for all observations
F	Set of all possible labelings for all observations
d	True classification of dataset
ϵ	Noise injected into dataset
$U(f)$	Potential energy function
C_j	Set of all cliques of size j
V_c	Clique potential function for clique c
ω_c	Weighting constant for clique c
k_i	Measured stiffness at point i
k_t	Threshold stiffness
m_i	Weighting factor for V_c
$x^{(i)}$	Input features (point stiffnesses)
$y^{(i)}$	Target variable (point labelings)
ϕ	Class prior parameter
μ_0	Mean for points labeled $x^{(i)} = 0$
μ_1	Mean for points labeled $x^{(i)} = 1$
Σ	Covariance matrix for $x^{(i)}$ labeling probabilities

2.2.4.1 Expectation Step (Markov Random Fields)

Markov random field theory can use spatial information to define probability distributions of data. To use this approach, we assume points in close proximity to each other are likely to be similarly classified.

We define $f \in F$ as the set of the labelings for all observations in the dataset, while the underlying true classification of the dataset is denoted d . For our case of inclusion segmentation, f is the classification of the dataset of palpated points, while the true nature of these points as part of the inclusion or part of the surrounding soft

tissue is d . The relation between these two distributions is:

$$f_i = d_i + \epsilon_i \quad (2.2)$$

where i represents the i th element (palpated point) of the dataset, and ϵ_i represents noise injected into our measurement of d_i . We seek to find the maximum *a posteriori* (MAP) solution, $p(f|d)$, by computing the prior energy, $p(f)$, and the likelihood energy $p(d|f)$, using the relation in:

$$p(f|d) \propto p(d|f)p(f) \quad (2.3)$$

To pose our problem as a Markov random fields problem, the following two properties must be satisfied:

$$\text{Positivity: } p(f) > 0 \quad \forall f \in F \quad (2.4)$$

$$\text{Markovianity: } p(f_i|f_{S-\{i\}}) = p(f_i|f_{N_i}) \quad (2.5)$$

where $S - \{i\}$ is the set difference, $f_{S-\{i\}}$ denotes the set of labels at the sites in $S - \{i\}$, and

$$f_{N_i} = \{f_{i'} | i' \in N_i\} \quad (2.6)$$

is the set of labels at the sites neighboring i (see Figure 2.3(a)).

Satisfying the positivity clause is trivial. Markovianity is satisfied as long as the probability of a labeling of a point is a function of the neighborhood around the point of interest in addition to its own stiffness information. In the most extreme case, Markovianity can be satisfied by extending the neighborhood around the point of interest to include the entire workspace.

Markov random fields have an equivalence property with Gibbs distributions, which enables calculation of the joint probability of all the point's possible classifications $p(f)$ [76]. Under this equivalence, $p(f)$ can be written as:

$$p(f) = \frac{e^{(-\frac{1}{T}U(f))}}{\sum_{f \in F} e^{(-\frac{1}{T}U(f))}}, \quad (2.7)$$

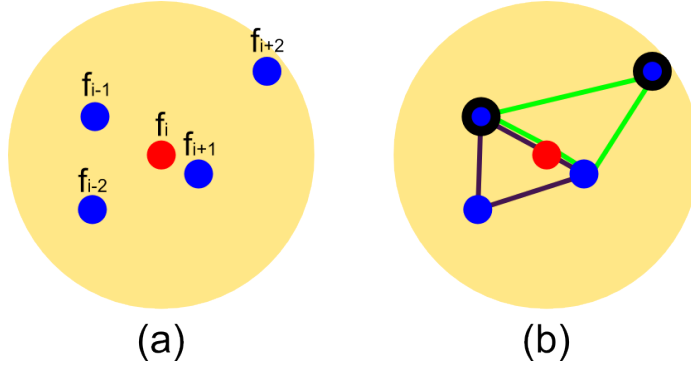


Figure 2.3: Illustration of neighborhoods and cliques around point i , pictured in red. Blue points are other data points in the workspace. (a) All points inside the yellow circle are in the neighborhood around point i . The labels of these points, f_j for $j \in [i-2, i+2]$, are contained in the set f_{N_i} . (b) Cliques containing four points, which include point i are the vertices of the polygon outlined in black. The vertices of the green polygon do not constitute a clique of size four which includes point i because the two points highlighted with a black outline, x^{i-1} and x^{i+2} , while contained in the neighborhood centered at point i , are a distance apart greater than the neighborhood radius. Hence, a neighborhood centered at one of the points highlighted in black would not contain the other point highlighted in black, which is in violation of the definition of a clique.

T is the “temperature constant”, which injects energy into the system to avoid solutions converging to local minima. Throughout this work, $\frac{1}{T} = \frac{1}{n}$, where n is the number of iterations of the current execution of the EM algorithm. $U(f)$ is the energy function, which is calculated by

$$U(f) = \sum_{c \in C} w_c V_c(f). \quad (2.8)$$

$V_c(f)$ is the individual clique potential function for clique $c \in C$. Cliques enforce Markovianity by using only the information within the clique around a point to influence its labeling (See Figure 2.3(b)). A clique of size k for point i is defined as

$$c_{i,k} = \{ j \mid d(i, j) < d_k \ \& \ d(j, j') < d_k, \ \forall j' \in c_{i,k} \} \quad (2.9)$$

where $d(i, j)$ computes the distances between points i, j and d_k is the maximum size allowed for the clique. The definition of a clique differs from the definition of the neighborhood around a point; it is necessary that the set of points forming the clique of a single point of interest are also included in that point's neighborhood, but the converse is not true. Equation 2.8 can be expanded to

$$U(f) = \quad (2.10)$$

$$w_1 \sum_{\{i\} \in C_1} V_1(f_i) + w_2 \sum_{\{i, i'\} \in C_2} V_2(f_i, f_{i'}) + \dots,$$

where w_c are weighting factors for each of the normalized clique potential functions V_c . To achieve a reasonable computation time on the order of tens of milliseconds, only cliques up to size four were calculated. In this work, $w = \{1.0, 0.8, 0.4, 0.2\}$. These values were chosen to emphasize smaller cliques, and were tuned in simulation (see Section 2.3).

A potential function is defined for each differently sized clique. In the following definitions of clique potentials, $f_i = 1$ implies that point i is labeled as part of the hard inclusion and $f_i = 0$ implies that point i is labeled as part of the surrounding tissue. k_t is the threshold stiffness, or classifier, input as a parameter to the algorithm.

$$V_1(f_i) = \quad (2.11)$$

$$\begin{cases} |k_i - k_t|, & \text{if } k_i < k_t \text{ and } f_i = 1 \\ m_1 + |k_i - k_t|, & \text{if } k_i \geq k_t \text{ and } f_i = 0 \\ 0, & \text{else} \end{cases}$$

$$V_2(f_i, f_{i'}) = \begin{cases} 0, & \text{if } f_i = f_{i'} \\ m_2 \frac{d_{\min}}{d(i, i')}, & \text{if } \frac{k_i + k_{i'}}{2} \geq k_t, f_i \neq f_{i'} \text{ and } f_i = 1 \\ \frac{d_{\min}}{d(i, i')}, & \text{else} \end{cases} \quad (2.12)$$

where d_{\min} is the minimum distance between any two members of C_2 , and in the subsequent equations represents the minimum average distance among all pairings of members in a clique.

$$V_3(f_i, f_{i'}, f_{i''}) = \begin{cases} 0, & \text{if } f_i = f_{i'} = f_{i''} \\ m_3 \frac{d_{\min}}{d(i, i', i'')}, & \text{if } \frac{k_i + k_{i'} + k_{i''}}{3} \geq k_t, f_i = 1 \text{ or} \\ & \frac{k_i + k_{i'} + k_{i''}}{3} < k_t, f_i = 0 \\ \frac{d_{\min}}{d(i, i', i'')} & \text{else} \end{cases} \quad (2.13)$$

$$V_4(f_i, f_{i'}, f_{i''}, f_{i'''}) = \begin{cases} 0, & \text{if } f_i = f_{i'} = f_{i''} = f_{i'''} \\ m_4 \frac{d_{\min}}{d(i, i', i'', i''')}, & \text{if } \frac{k_i + k_{i'} + k_{i''} + k_{i'''}}{4} \geq k_t, f_i = 1 \text{ or} \\ & \frac{k_i + k_{i'} + k_{i''} + k_{i'''}}{4} < k_t, f_i = 0 \\ \frac{d_{\min}}{d(i, i', i'', i''')} & \text{else} \end{cases} \quad (2.14)$$

Equation 2.12 heavily penalizes points labeled as soft tissue despite having a stiffness is greater than k_t , but incurs a smaller penalty for a point being labeled as part of the hard inclusion while having a stiffness less than k_t . For the remaining clique potential functions, the potential energy depends on combination of the average stiffness of the points, their average distance from each other, and the current labeling of the point incurring the potential penalty. The values m_i were tuned in simulation

(Section 2.3); $m = \{1.2, 1.0, 1.0, 1.0\}$. Like the weighting variables w_i , the difference of m_1 compared to the other members in m is to emphasize smaller cliques by weighting them with increased energy.

At this point, $p(f)$, first introduced in Equation 2.3, has been fully defined using the relation in Equation 2.7. Next, we model $p(d_i|f_i)$ with the following random replacement model:

$$p(d_i|f_i) = \begin{cases} 0.9, & \text{if } d_i = f_i \\ 0.1, & \text{else} \end{cases} \quad (2.15)$$

The final step of formulating our identification problem in the context of Markov random fields is to calculate the explicit posterior probability estimates $p(f|d)$ and choose the most probable classification. We expand $p(f|d)$ using the law of total probability.

$$\begin{aligned} p(f_i = 0) &= p(f_i = 0|d_i = 0)p(d_i = 0) + \\ &\quad p(f_i = 0|d_i = 1)p(d_i = 1) \\ p(f_i = 1) &= p(f_i = 1|d_i = 0)p(d_i = 0) + \\ &\quad p(f_i = 1|d_i = 1)p(d_i = 1) \end{aligned} \quad (2.16)$$

2.2.4.2 Maximization Step (Gaussian Discriminant Analysis)

Once points are classified using Markov random fields, the maximization step of the EM algorithm refines the classifier. Results of earlier work [63] suggest that among three one-dimensional classification algorithms: logistic regression, Gaussian discriminant analysis (GDA), and support vector machines, GDA resulted in the least-conservative classifier for elastography data. We reported that the distribution of stiffnesses between the modalities of elastography and autonomous robotic palpation differed, and the GDA classifier developed from elastography resulted in a classifier with higher sensitivity compared to logistic regression or support vector machines. Thus, in choosing GDA to generate the classifier we are choosing the algorithm best suited to label points correctly.

Symbols used in GDA include $x^{(i)}$, the input variables or features, in our case the

stiffnesses observed (indexed by point i), $y^{(i)}$ the output or target variable, in our case the classification, and θ_i , the variables parameterizing the space of functions mapping from the feature space to the target space. The classifier is defined as the value of the feature x such that x is equally likely to be part of the hard inclusion ($y^{(i)} = 1$), or belong to the surrounding soft tissue ($y^{(i)} = 0$). GDA fits each class of data to a Gaussian distribution with means μ_0 and μ_1 , covariance matrix Σ , and class prior parameter ϕ . The maximum likelihood estimates of these parameters are:

$$\begin{aligned}\phi &= \frac{1}{m} \sum_{i=1}^m 1\{y^{(i)} = 1\} \\ \mu_0 &= \frac{\sum_{i=1}^m 1\{y^{(i)} = 0\} x^{(i)}}{\sum_{i=1}^m 1\{y^{(i)} = 0\}} \\ \mu_1 &= \frac{\sum_{i=1}^m 1\{y^{(i)} = 1\} x^{(i)}}{\sum_{i=1}^m 1\{y^{(i)} = 1\}} \\ \Sigma &= \frac{1}{m} \sum_{i=1}^m (x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T\end{aligned}$$

Where $1\{arg\}$ returns 1 if arg evaluates to be true, and 0 otherwise.

2.2.5 Iteration Procedure

With the new classifier generated from GDA, the program execution will return to the “Exploration of Tissue” process (Figure 2.1) with the updated classifier and new palpation grid. To generate the new palpation grid we must first calculate the boundary points of the inclusion (Figure 2.4). The centroid of all points classified as belonging to the hard inclusion is the starting point for our search. Extending radially outward from the centroid location, a boundary point is defined as the last point classified as part of the hard inclusion along a “line of search” before leaving the inclusion. The collection of boundary points defined in this manner constitute a positive surgical margin. A negative surgical margin is not possible at this point during the inclusion segmentation algorithm since a dense boundary only exists for points classified as part of the hard inclusion. The palpation grid used to seed the next iteration of the inclusion segmentation algorithm is generated from the union of neighborhoods

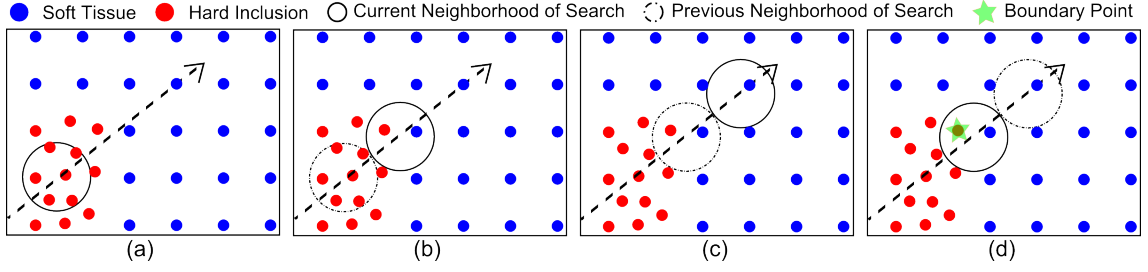


Figure 2.4: Illustration of the boundary point finding algorithm. Red points are classified as part of the hard inclusion, while blue points are classified as part of the surrounding tissue. The dashed arrow is the direction of search. The solid circles are the neighborhoods of search for this step of the algorithm, while the dashed circles are the neighborhoods of search from the previous step, included for reference. (a) All points in the neighborhood of interest are classified as part of the hard inclusion; the search continues outward along search line. (b) Points in this neighborhood belong to both classes, so the search continues. (c) All points are classified as belonging to soft tissue, thus the area of search moves one step back. (d) The point closest to the center of the neighborhood of search classified as belonging to the hard inclusion is identified as the boundary point for this search line, seen here as the green star.

around these calculated boundary points. As the inclusion segmentation algorithm iterates, an increasing number of points generated as part of the next grid to palpate fall within the minimum distance allowed between palpated points, to the extent that in the final iteration of the inclusion segmentation algorithm no new points are explored and the stiffness classifier converges. At this point, the inclusion segmentation algorithm proceeds to the exiting procedure.

2.2.6 Exiting Procedure

If classifiers developed from successive iterations of the EM algorithm converge, we exit the iterative process and transition to the final states in the inclusion segmentation algorithm. When the classifier has converged, all neighborhoods of points around the boundary of the inclusion have been explored, thus we have increased resolution

of the boundary consisting of points representative of the hard inclusion *and* the surrounding soft tissue. The first points classified as part of the surrounding tissue just beyond the boundary of the inclusion define a negative surgical margin. The convex hull or moving average of these points was used to provide additional smoothing. To provide a continuous boundary for the inclusion, boundary points were interpolated between using third-order Bezier splines.

2.3 Simulation of the Inclusion Segmentation Algorithm

To tune the parameters of the inclusion segmentation algorithm, such as m and w , results were simulated in MATLAB. We also tested the robustness of the inclusion segmentation algorithm subject to a wide variety of hard inclusion shapes, sizes, and stiffnesses. This section presents simulated results for a variety of programmed simulated tissues as well as shows several iterations of the inclusion segmentation algorithm for a single simulated tissue.

2.3.1 Setup

Simulated artificial tissues contained inclusions of various sizes, shapes, stiffnesses, and placement in the workspace. Each (x, y) coordinate pair of the 500×500 pixel simulated tissue image indexes the stiffness of the tissue at that point. The differences in stiffnesses between points representing the hard inclusion and points representing the surrounding tissue was designed to be consistent with observed prostate tumor data from [45]. Once the different stiffness level sets were simulated, a circular filter was applied to smooth the data.

2.3.2 Simulation Results

We demonstrate the evolution of the inclusion segmentation algorithm for one simulated tissue across several iterations in Section 2.3.2.1. We present simulated inclusion

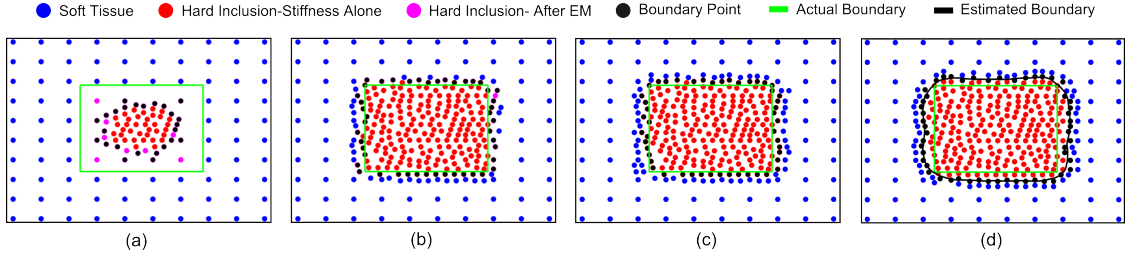


Figure 2.5: Simulated inclusion segmentation results across several iterations. Points in blue are classified as belonging to the soft tissue. Points in red are classified as belonging to the hard inclusion after being thresholded against the stiffness classifier. Points in magenta were not classified as part of the hard inclusion based solely on their stiffness alone, but were classified as part of the hard inclusion after taking into account the stiffnesses of their cliques. Points in black are the boundary points. The green outline is the actual location of the inclusion. (a) Simulated results after the first iteration of the algorithm. (b) Simulated results after the second iteration of the algorithm. (c) Simulated results after the third iteration of the algorithm. (d) Simulated results after the seventh and final iteration of the algorithm. The black line illustrates the approximated boundary of the inclusion.

segmentation results across four differently sized and positioned inclusions in Section 2.3.2.2.

2.3.2.1 Simulation Across Several Iterations

Several iterations of the inclusion segmentation algorithm for a rectangular inclusion are shown in Figure 2.5, whose underlying stiffness is pictured in Figure 2.6(a). The red points in Figure 2.5 were classified as belonging to the hard inclusion based solely on their individual stiffness, while blue points were not sufficiently stiff and were instead classified as belonging to the surrounding soft tissue. The points in magenta were not sufficiently stiff classified as part of the hard inclusion without taking into account the stiffness of their cliques. In the first iteration, (Figure 2.5(a)), relatively few neighborhoods around points classified as part of the hard inclusion are explored compared to subsequent iterations. This is because the classifier seeded into this

iteration of the algorithm was conservative, resulting in very few points (and their neighborhoods) explored with increased resolution. The initial classifier (set to 4.5 times the stiffness of the surrounding soft tissue) would classify all dark-red areas visualized in Figure 2.6(a) as part of the hard inclusion. Before the second iteration of the inclusion segmentation algorithm the classifier was improved to a value 2.8 times the stiffness of the surrounding soft tissue. Following the first iteration, the updated classifier relabeled some points as part of the hard inclusion; a majority of which were on the boundary. In Figure 2.6(a) the relabeled points were those in the orange region. Neighborhoods around the boundary points were explored in higher resolution during the next iteration of the EM algorithm. The right-most sub-plot of Figure 2.5 shows the boundary points ensuring a negative surgical margin. The inclusion segmentation algorithm for the rectangular inclusion took six iterations in total, with a final classifier value of 2.6 times the surrounding soft tissue stiffness. Of the points explored after the first iteration of the inclusion segmentation algorithm, 72% of those points were explored in the second iteration and 18% of those points were explored as part of securing a negative margin in our boundary identification.

2.3.2.2 Simulation Results Across Varied Samples

Segmentation boundaries for four different simulated tissues are shown in Figure 2.6. The stiffest part of the simulated inclusions are shown in red, while the surrounding soft tissue has stiffness visualized in blue. Figure 2.6(a) is the final result from the sequence in Figure 2.5. Some of the shapes were non-convex, so a moving-average filter among triads of consecutive points along the boundary was used to smooth the final set of boundary points to generate as close a fit to the boundary of the inclusion as possible.

Figures 2.6(b) and 2.6(c) show the algorithm's performance in identifying non-convex shapes. In Figure 2.6(b), the boundary of the inclusion is identified correctly, while in Figure 2.6(c) the boundary interpolation does not completely enclose the stiff region. In this case, the boundary point algorithm finds only the first transition of point labelings of hard inclusion to soft tissue. In 2.6(c), there are multiple such transitions, but only the first transition identifies a boundary point.

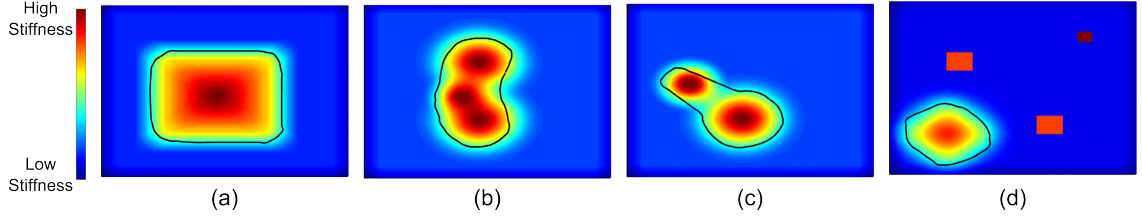


Figure 2.6: Simulated inclusion segmentation results for four environments. The black line represents the estimated boundary of the inclusion (a) The rectangular inclusion from Figure 2.5. (b) A kidney-bean shaped inclusion. (c) A modified kidney-bean inclusion. (d) An ellipsoid inclusion with simulated erroneous measurements. The simulation is robust with respect to these measurements.

Figure 2.6(d) shows the results for an artificial tissue with simulated data corruption in three regions. Tissue mechanical properties not accounted for in a simple approximation, like the linear spring model we use in this work, can result in erroneous stiffness measurements. Thus, the proposed algorithm must demonstrate robustness with respect to corrupted data points. Three tissue regions simulated corrupted stiffness measurements, where the measured stiffness is falsely stiffer than its actual stiffness. The corrupted regions are identified as part of the hard inclusion, however the centroid calculation of all points identified as part of the hard inclusion lies within the boundary of the hard inclusion, so the correct boundary points are identified.

2.4 Experiment 1: Accuracy

In the previous section we demonstrated the performance of the inclusion segmentation algorithm in simulation. Replicating these results in experiment requires a classifier to initialize the EM sub-algorithm, as opposed to the hard-coded value used in simulation. In experiment, we gathered elastography data from an artificial tissue. Elastography provides a non-invasive, inclusion-specific characterization of the tissue mechanics. From the multiple gray-scale images generated from elastography, we selected the image with the greatest contrast between the hard inclusion and

surrounding soft tissue and outlined the inclusion. This classification formed the training set from which GDA developed a classifier that sorted new data points as part of the hard inclusion or part of the surrounding soft tissue. Next, the classifier was mapped from sorting elastography data to sorting absolute stiffness data. This mapping was derived using full datasets of elastography and palpation data from a training-set tissue. The mapped classifier was then used to threshold data acquired via autonomous robotic palpation on a testing-set tissue. From force-position data acquired during a single palpation from the robot, the stiffness of the tissue at that point was approximated. The robot explored neighborhoods around points classified as part of the hard inclusion with higher resolution than the resolution prescribed by the initial palpation grid input into the inclusion segmentation algorithm. After completing the algorithm in Figure 2.1, the palpated points, their stiffnesses, and the boundary points were output to a MATLAB script which interpolated the boundary points and reported sensitivity and specificity metrics.

2.4.1 Artificial Tissue

Two artificial tissues were created to simulate the mechanics of a hard inclusion in soft tissue for experimentation: the training-set tissue and testing-set tissue. We used artificial tissues because they are low-cost and durable, with long shelf-life ideal for prolonged testing. Additionally, artificial hard inclusions can be made precisely to control for size, shape, and relative stiffness compared to the surrounding soft tissue. A variety of artificial tissue materials exist, including gelatin, silicone, and plastisol. For this experiment, the artificial tissue was created with gelatin, as gelatin samples are sufficiently firm to withstand palpation tests while being visible with elastography. Gelatin also has a sufficient degree of optical transparency, useful for validation of this method. Artificial tissues made from gelatin are not inherently echogenic, so fine fiber supplement grains were added as the speckle material for use with elastography. Both artificial tissue bases were created with a 10.2 g/250 ml ratio of gelatin powder to water, while hard inclusions were created using a ratio of 28.8 g/250 ml of gelatin powder to water. One pre-made, cubic hard inclusion with a side length of size 25

mm was positioned into the center each artificial tissue base, 4 mm below the surface of the tissue during the curing process of the softer tissue. The side length of 25 mm was inspired by the work of Perez *et al.* [77]. In their study, they found that across 93 different tumor samples of stage 1 adenocarcinoma of the lung, the median tumor size was 25 mm.

2.4.2 Generation of Classifier

The initial classifier used to seed the EM algorithm was developed from elastography. After gathering elastography data and autonomous robotic palpation data from the training-set artificial tissue, a mapping from our previous work converted the stiffness values between the two modalities, as commercial elastography does not generate absolute stiffness values [63]. This transformation was then applied to map a classifier developed from elastography to a value appropriate to sort stiffness values taken during autonomous robotic palpation of the test-set artificial tissue.

2.4.2.1 Elastography Data Acquisition

Elastography data was acquired using the elastography feature of a SonixMDP ultrasound machine (Ultrasonix Medical, Richmond, BC, Canada). Elastography compares radio-frequency pulse echoes during compressions in order to measure relative stiffness. For this study, the compressions were performed manually with the ultrasound transducer at a frequency of approximately 0.5 Hz. The ultrasound image was generated by placing the transducer on the top of the artificial tissue (see Figure 2.7(a)). Elastographic images store relative stiffness values for each pixel as an 8-bit datum. Although elastography data acquisition in general generates a video of several seconds, only one frame is necessary to supply the training set. We selected the frame with the largest contrast between the inclusion and the surrounding tissue. The setup used to gather the elastographic images is shown in Figure 2.7(a).

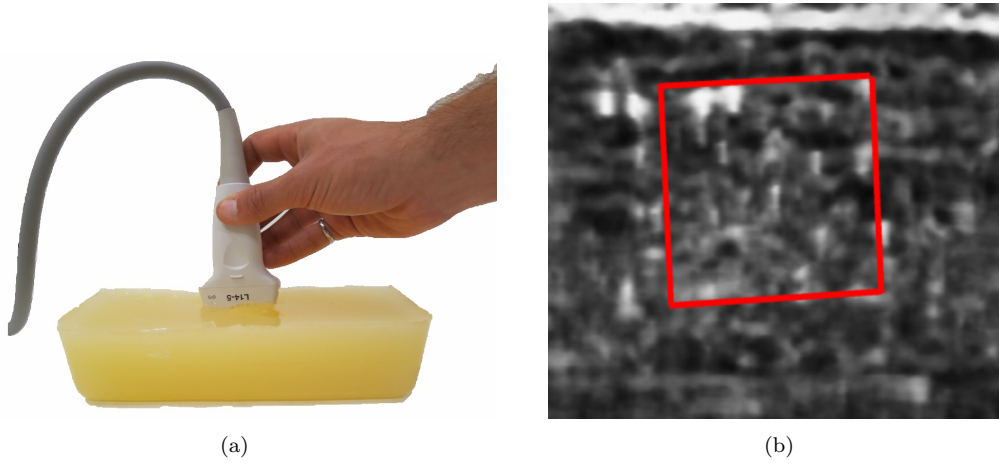


Figure 2.7: (a) Elastography setup with artificial tissue. The ultrasound transducer is positioned to image the artificial tissue from the top. (b) Elastographic image, with the hard inclusion outlined in red. This outline was obtained by visual inspection of the image and manual selection of the corners of the inclusion. From this user-created classification, Gaussian discriminant analysis was applied to develop a classifier that would sort new data points into one of the two classes. The classifier was then mapped from sorting elastography data to sorting absolute stiffness data, and provided the initializing classifier for the EM algorithm.

2.4.2.2 Robotic Palpation

A Phantom Premium 1.5 commercial haptic device (Geomagic, Morrisville, NC, USA) served as the robotic platform for autonomous palpation of the artificial tissue (Figure 2.8). Third-order polynomial trajectories of the robot end-effector were input to a proportional-derivative controller. The maximum force exerted by the robot on the tissue was limited to 1.8 N in order to prevent motor overheating in the Phantom Premium. Once the maximum force was reached, the path planner used to provide reference points for robot travel was updated to the next point in the grid pattern. The artificial tissue was palpated in a linearly spaced grid across the surface of the tissue.

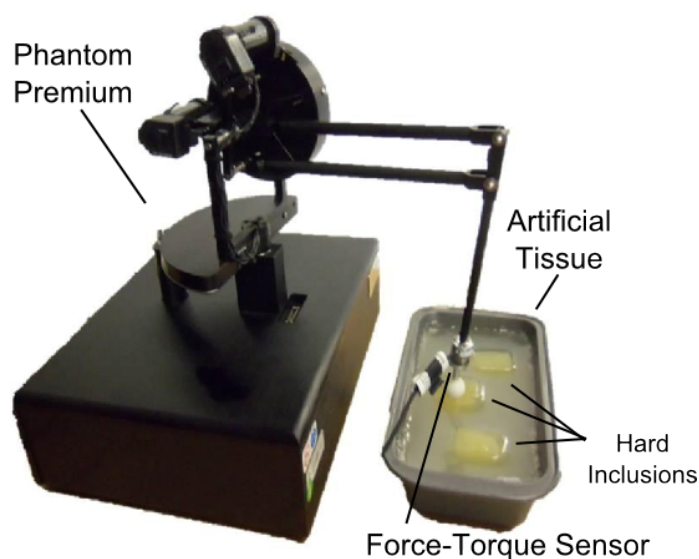


Figure 2.8: A Phantom Premium 1.5 positioned to autonomously palpate artificial tissue with embedded hard inclusions.

2.4.2.3 Stiffness Approximation

A Nano-17 force-torque sensor with a spherical ball (ATI Industrial Automation, Apex, NC, USA) attached to end point of the Phantom Premium acquired force data. A spherical ball was used to provide as much of a symmetric force profile in the plane orthogonal to the axis of palpation as possible. The raw force-displacement profile exhibits hysteresis, which is expected from a soft-tissue interaction. A linear spring model approximated the tissue mechanics using only the data from the downward stroke. Force-displacement data used to approximate stiffness was thresholded between 1.2 N and 1.8 N of force for autonomous palpation; teleoperated palpation did not have the upper threshold. The upper threshold limited stress on the tissue, while the lower threshold isolated a reliable linear response in the force profile during a palpation. Sample force-displacement curves, which show nearly linear dynamics in this force range, are seen in Figure 2.9.

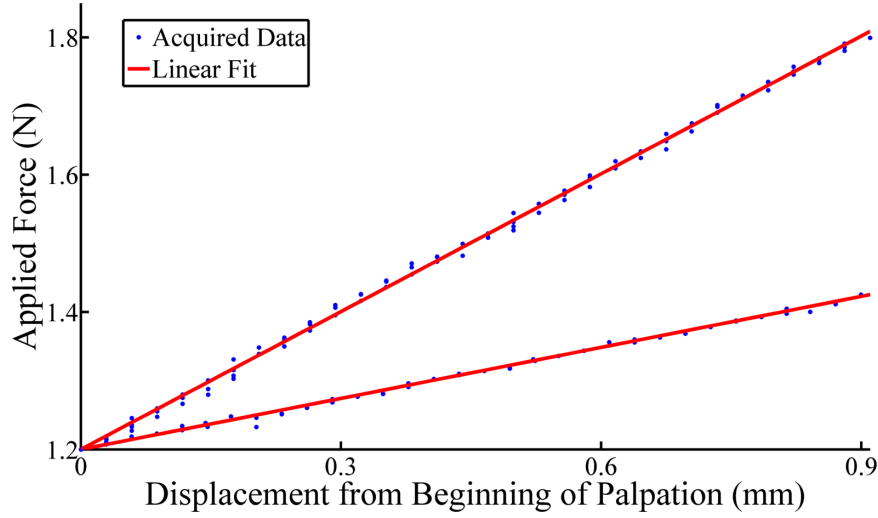


Figure 2.9: Example force-displacement curve generated from the downward stroke of a palpation. The blue line is the acquired data, and the red line is the least-squares approximation of the acquired data.

2.4.2.4 Mapping the Classifier to Sort Palpation Data

We presented the mapping from a classifier generated from elastography data to one that can be applied to autonomous palpation data in an earlier work [63]. The elastography data from the training-set tissue was scaled linearly to match the values observed from the autonomous robotic palpation data for the same sample tissue, with a relative stiffness of 0 mapping to the least-stiff palpated point, and a relative stiffness of 255 mapping to the most stiff palpated point. Since the elastography data maps relative stiffness between points linearly, a linear mapping between intensity values and absolute stiffnesses was appropriate.

2.4.2.5 Identifying the Inclusion in the Elastographic Image and Generating the Elastography Classifier

The mapping from relative to absolute stiffnesses established from the training-set tissue developed a classifier specific to the test-set tissue. We manually segmented the inclusion imaged from elastography of the test-set tissue by selecting the corners of the

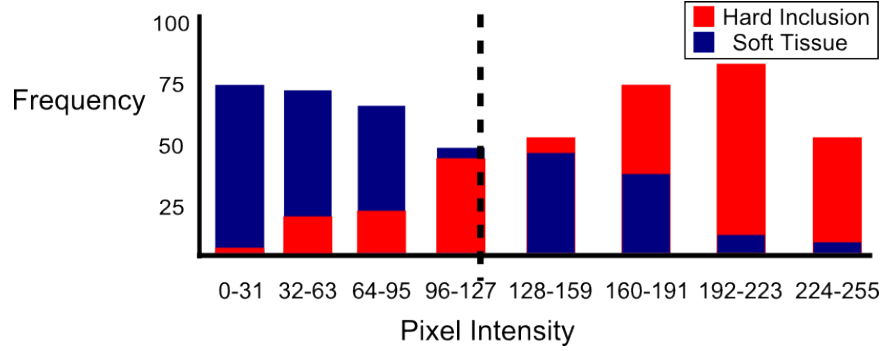


Figure 2.10: Training the GDA classifier. The blue bars are trained as not belonging to the inclusion, while the red bars are trained as being part of the inclusion. The dashed line represents the learned classifier. For the test-set tissue, all points to the right of the classifier would be labeled as part of the hard inclusion, while all points to the left of the classifier would be labeled as the surrounding soft tissue.

inclusion in the image to train the machine learning algorithm (Figure 2.7(b)). The stiffness data (greyscale values) contained within the outline of the identified inclusion were labeled as part of the inclusion while all other stiffness data were labeled as part of the surrounding soft tissue. From this labeling, GDA developed the initial classifier. Figure 2.10 shows the classifier developed from the data in Figure 2.7(b) using GDA. This classifier was then mapped to an absolute stiffness value using the mapping developed from the training-set tissue.

2.4.3 Evaluation Metrics

A series of metrics measured the effectiveness of the algorithm by comparing the predicted location of the hard inclusion to the actual location of the hard inclusion. Figure 2.11 provides a graphical explanation of these metrics. The intersection between the predicted and actual hard inclusion locations is correctly predicted to be part of the hard inclusion. All other predicted hard inclusion area is identified falsely as part of the hard inclusion. Likewise, all remaining area of the registered inclusion not intersecting the predicted location of the hard inclusion is identified falsely as part of the surrounding soft tissue. All other area in the workspace not predicted as

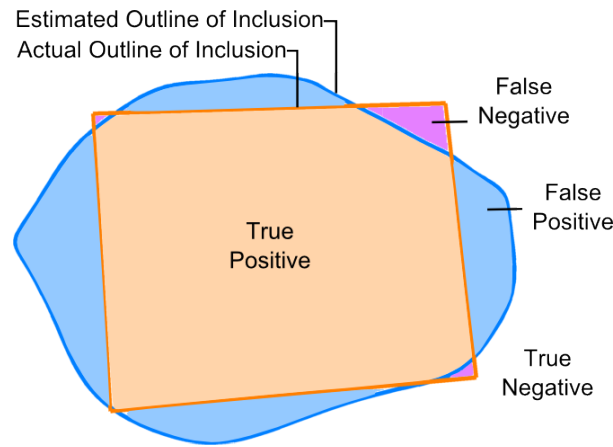


Figure 2.11: A graphical explanation of the evaluation metrics used in this study. The blue boundary is the estimated outline of the inclusion while the orange line encapsulates the actual outline of the inclusion. This figure is derived from data presented in Section 2.4.4.

part of the hard inclusion or registered as such are classified correctly as part of the surrounding tissue. Sensitivity and specificity summarize these four non-intersecting quadrants. Sensitivity measures the proportion of true positives which are correctly identified as such while specificity measures the proportion of true negatives which are correctly identified. A perfect prediction algorithm would have sensitivity of 1.0 and specificity of 1.0. Classification methods with sensitivity and specificity values greater than 0.8 have been generally accepted by the academic community as a success and used as benchmark performance against other classification algorithms [78, 79].

2.4.4 Results

The inclusion segmentation approximations are shown in Figure 2.12. All sets were initialized with the same initial classifier value and initial palpation grid. The metrics analyzing the results are included in Figure 2.12. In all three trials, the sensitivity and specificity were above 0.92. Among all three trials, the maximum ratio of stiffness of palpated points between hard inclusion and soft tissue was 2.3.

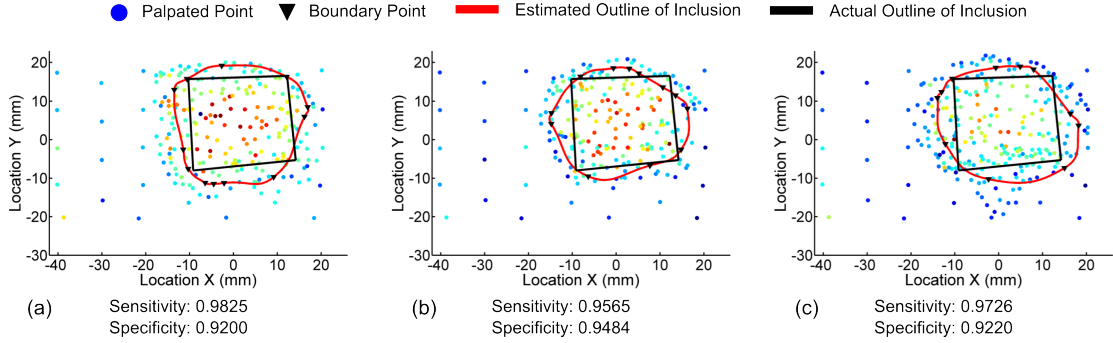


Figure 2.12: Experimental results for three trials. The red line interpolates through the boundary points in black and forms the derived boundary of the inclusion. The black box represents the actual location of the inclusion. Sensitivity and specificity metrics are included for each trail.

2.5 Experiment 2: Robustness with Respect to Initial Classifier Value

We conducted an experiment to test the robustness of the inclusion segmentation algorithm with respect to the initial classifier value. Since no two elastography images generate the same classifier due to noise in the image, our results from the previous experiment would be compromised if we could not also demonstrate the inclusion segmentation algorithm’s robustness with respect to its initial classifier value.

2.5.1 Artificial Tissue

For this experiment, an artificial tissue was created using PVC plastisol (M-F Manufacturing, Fort Worth, Texas). Artificial tissues made from PVC plastisol are more durable than those created from gelatin, which is necessary as this sample was subject to extensive experimentation. The artificial tissue base was created with a 1.75:1 ratio of liquid plastic to softener; while the hard inclusion was created using solely liquid plastic. As before, the hard inclusion was a 25 mm square cube and embedded 4 mm below the surface of the tissue.

Table 2.2: Sensitivity and Specificity for Robustness Testing Across Decreasing Seeding Classifier Values

$k_{\text{init}}(\text{N/mm})$	Sensitivity	Specificity
1.05	0.972	0.922
1.00	0.945	0.879
0.95	0.925	0.906
0.90	0.884	0.938
0.85	0.927	0.887
0.80	0.875	0.915
0.75	0.976	0.948
0.70	0.976	0.902
0.65	0.946	0.924
0.60	0.942	0.933

2.5.2 Procedure

Ten trials were performed varying the initial classifier value from 0.6 N/mm to 1.05 N/mm in increments of 0.05 N/mm. Below this range the robot would explore the entire tissue with fine resolution, losing the efficiency purported with the algorithm. No points had a measured stiffness greater than 1.10 N/mm. The stiffness difference between the hard inclusion and surrounding soft tissue in this test is on the lower end of observed difference in stiffness between prostate tumors and their surrounding tissue [45].

2.5.3 Results and Discussion

The sensitivity and specificity metrics subject to various initial starting stiffness classifier values are shown in Table 2.2. Note that for all initial classifier values, the specificity and sensitivity of the algorithm is at or above 0.875. This suggests that the algorithm is robust with respect to the seeding classifier value input to the expectation-maximization sub-algorithm. Among all trials, the maximum ratio of

stiffness of palpated points between hard inclusion and soft tissue was 2.4.

While we have demonstrated the robustness of the autonomous inclusion segmentation algorithm with respect to the initializing classifier, there is variance in the sensitivities and specificities for the segmented region among the initial classifier values. The autonomous inclusion segmentation algorithm is not deterministic, and differing values for initializing classifiers will yield a different number and distribution of points palpated, as neighborhoods explored during the inclusion segmentation algorithm, and in particular during the first iteration, are correlated with the initial classifier value.

The maximum force exerted on the tissue during autonomous palpation, nearly 1.8 N, is less than half of the maximum force exerted by Trejos *et al.* with their autonomous agent [56]. In their work the autonomous agent palpated with approximately 4 N of force, and furthermore did not segment the boundary autonomously. Trejos *et al.* did conduct trials where they limited the force of their palpations to 2 N, which resulted in reportedly poor success rate caused by image artifacts in their generated contour map. Lastly, their definition of success was achieved when the human-identified boundary of the inclusion overlapped with the actual location of the hard inclusion, and did not include sensitivity and specificity analysis like we do in this work. The other significant work in autonomous robotic palpation and segmentation mentioned in the background is the work of Sangpradit *et al.* [58]. While their method showed success in identifying embedded circular inclusions, our developed autonomous inclusion segmentation algorithm does not need *a priori* knowledge of the shape of the inclusion, evidenced in simulation in this chapter, and in experiment as shown later in Section 4.4.

The maximum resolution of the inclusion segmentation algorithm is limited by multiple factors. The precision of the robot to palpate a commanded point and the size of the palpation end effector are obvious factors, however a more subtle limitation is the Markov random fields algorithm. As the resolution of points increases, depending on the radius used to form a clique, an increasingly large search needs to be conducted to use Markov random fields. This search in the worst case is $O(n^m)$ for n palpated points and investigating cliques up to size m . In this work, only cliques up to size four

were used to limit computational complexity. Our resolution factor in this work, 5mm, was chosen to limit the autonomous inclusion segmentation algorithms’s execution time to less than 10 minutes, as reported in Section 4.2, which is within an order of magnitude from the amount of time it would take a human operator to perform the same task under bilateral teleoperation.

2.6 Conclusions

In this chapter, supervised machine learning methods successively classified stiffness data acquired via autonomous robotic palpation and segmented the boundary of a hard inclusion embedded in soft tissue. Elastography data provided an initial classifier to an expectation-maximization sub-algorithm with underlying Markov random fields. Initial data provided a mapping from a classifier derived from elastography data to one suitable for stiffness discrimination from autonomous palpation data. The results show that the boundary of the inclusion can be segmented from the surrounding soft tissue with high sensitivity and specificity. Furthermore, the inclusion segmentation algorithm is robust with respect to the initial classifier value seeded into the expectation-maximization sub-algorithm.

We have demonstrated the ability of the algorithm to differentiate a ratio of stiffness commonly seen in prostate tumors [45]. We theorize that the algorithm will continue to perform in instances where the stiffness ratio between surrounding tissue and inclusion is greater, like for tumors of the breast, where the stiffness ratio can be well over 8 [52].

We seek to expand this work by exploring ways to combine this autonomous agent with a human agent. To help expedite the development and testing of different human-robot collaboration models in an inclusion segmentation task, we use the Multilateral Manipulation Software Framework developed in the next chapter.

Chapter 3

The Multilateral Manipulation Software Framework

In this chapter we present a flexible software framework to expedite development of various multilateral manipulation strategies. We demonstrate the effectiveness of an implementation of the framework in a palpation task. Five different collaboration models were tested in which the goal of the multilateral manipulation system is to segment a stiff inclusion from its surrounding soft tissue: three of these collaboration models used machine learning methods for segmentation, and two required human operator segmentation. The collaboration models tested were: (1) fully autonomous exploration of the tissue developed in the previous chapter, (2) shared control between a human and robotic agent, (3) supervised control where the operator dictates commands to the robot, (4) traded control between the two agents, and (5) bilateral teleoperation. Results indicate tradeoffs in sensitivity, maximum force applied, safety implications, and duration of experiment among the five models.

This chapter is organized as follows. Section 3.1 introduces the software framework at a high level, and presents an extension of this framework for an example surgical task. Section 3.3 presents the experimental setup used to evaluate five exemplar different collaboration models including the metrics gathered to compare the collaboration models to each other. In Section 3.4, we present and discuss the results across the five different collaboration models. We also discuss how well the software

framework suited our needs to rapidly prototype and implement new collaboration models. Finally in Section 3.5, we make concluding remarks.

3.1 Software Framework

We present in this section an overview of our multilateral manipulation software framework. A preliminary version of this work appears in [80]. The design goal of this software framework is to facilitate rapid prototyping of collaboration models, each with a varying degree of autonomy. This is accomplished in an objected-oriented sense by a set of base classes which generalize human-robot collaborative tasks. These base classes are then extended into procedure-specific child classes which are further extended to encode implemented collaboration models. The design choices in the base classes and subsequent extensions seek to leverage as much of the same code as possible across collaboration models, to ease prototyping and development. Additionally, we require the Multilateral Manipulation Software Framework to use thread-safe communication, which avoids problems such as deadlock and allows synchronization among threads. This is the first instance to our knowledge of a software framework developed and tested to specifically enable a wide variety of human-robot collaboration models. The design of this framework was heavily influenced by ROS [42] and CISST [40], which are software frameworks used for sensing and actuation in tasks involving robotic systems. Indeed, what ROS and CISST contribute to actuation and sensing, we seek in this work to contribute to the design and implementation of human-robot collaboration models. The object-oriented design of the MMSF is heavily influenced by Cheriton [81], which presents a detailed methodology of object-oriented design in modeling and simulation.

3.1.1 Base Class Structure

Figure 3.1 and Figure 3.2 illustrate a subset of the class structure, and the extensions necessary for the framework to execute an inclusion segmentation task. Robotic tasks are implemented by extending six base classes: *Robot*, *Graphical Display*, *File I/O*,

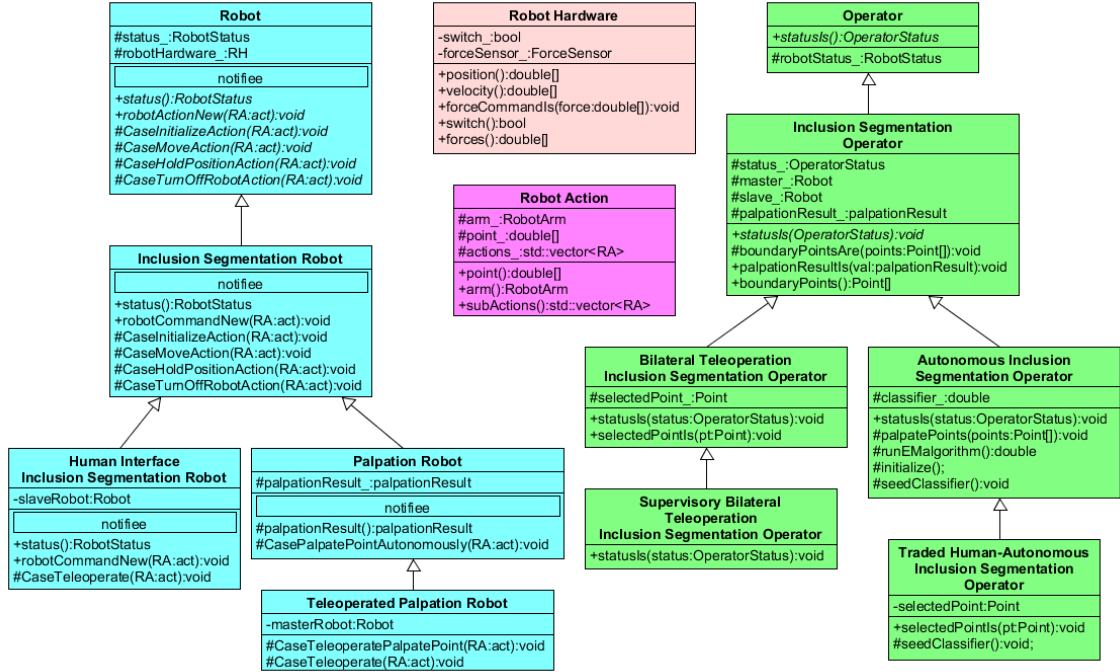


Figure 3.1: Unified Modeling Language (UML) diagram of the *Robot* (blue), *Robot Hardware* (RH, pictured pink), *Robot Action* (RA, pictured purple), and *Operator* (green) classes of the MMSF implementation for the inclusion segmentation task. The diagram also includes a selection of the extensions of the classes for the inclusion segmentation task used to implement the collaboration models described in Section 3.2. The *notifier* subclass is described in Section 3.1.2.

Operator, *Procedure*, and *Robot Action*, to make them task specific. The seventh base class, the *Manager* class, is instantiated once every time the framework is implemented, and not extended to each robotic task. We now introduce each base class in turn and discusses their intent, content, and any necessary dependencies.

Robot Action: A *Robot Action* is the fundamental command unit issued by the *Operator* to a *Robot*. The most basic commands extended from this class, for example, are *Move Robot Action* and *Hold Position Robot Action*. Other *Robot Action* extensions are *Initialize Robot Action* and *Turn Off Robot Action*. Parameters of a *Robot Action* vary widely depending on the specific extension; they may include

information like a desired position. *Robot Actions* can consist of series of basic *Robot Actions*, for example a series of *Move Robot Actions*.

Robot: All robotic agents in a collaborative model derive from the abstract *Robot* class. A *Robot* interfaces to the *Operator* by implementing commanded *Robot Actions*. This class may be extended to include acquisition and processing of information from attached sensors, as well as handle commands unique to the surgical task implemented. Figure 3.1 shows that the *Inclusion Segmentation Robot* defines methods for the basic *Robot Actions* while the *Palpation Robot* includes methods to handle palpation actions which are specific to our surgical task (described in Section 3.2). Additional extensions may enable the robot to be used as a human-input device. A *Robot* at no time creates *Robot Actions* to be implemented by itself.

Each *Robot* has a reference to an instance of *Robot Hardware*, which is used as the hardware-specific interface from the robotic platform to the MMSF. In this work, we structure our *Robot Hardware* class to interface with Phantom Premium haptic devices (Geomagic, Morrisville, NC, USA). In work presented later in this dissertation the *Robot Hardware* class was structured to take commands from the the *Robot* class and implement them on the RAVEN surgical robot and da Vinci Reserach Kit (Chapter 4).

Procedure: The *Procedure* class is an auxiliary class and is a member variable of the *Operator*. The singleton instance of the *Procedure* is responsible for keeping a list of all the *Robot Actions* to be completed. Data collected during execution of the *Robot Actions* is also stored in the *Procedure*.

Operator: Workflow of the procedure is facilitated by the *Operator* class. The *Operator* pops a *Robot Action* from the *Procedure* stack and routes the command to the appropriate *Robot* instances. Once the *Robot Action* completes, the *Operator* interprets the data collected in the *Procedure*, and creates new *Robot Actions* as necessary. All task-specific machine-learning techniques presented later in this work are coded into this class.

File I/O: This auxillary class performs data transfer to/from files.

Graphical Display: This class visually displays all information suitable for the user. Additionally, the graphical display can be extended to become a human input

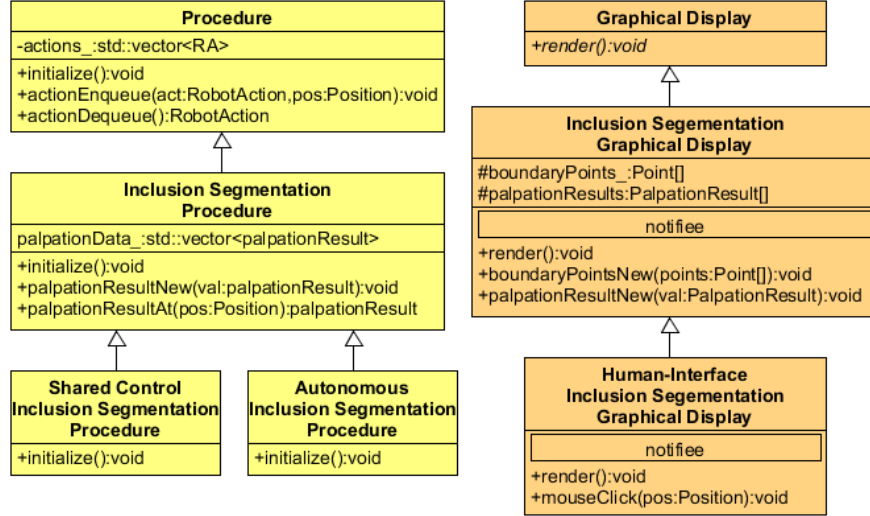


Figure 3.2: UML diagram of the *Procedure* (yellow) and *Graphical Display* (orange) classes, with extensions to realize the collaboration models for the inclusion segmentation task tested in Section 3.3. RA is an abbreviation for the *Robot Action* class.

device.

Manager: This class provides references through pointers of all base classes in the framework for query by other base classes. The *Manager* class is also responsible for instantiating the full set of classes necessary to conduct the surgical tasks assigned.

3.1.2 Inter-class Communication

Inter-class communication is facilitated through a notifier/notified interface [81]. For an example of inter-class communication, refer to Figure 3.3. Each notifying class has a subclass *notifier*. To illustrate the notifier/notified interface we will consider the *Robot::statusIs(status:RobotStatus)* method, which is called each time the *Robot* changes its state to indicate whether or not it is executing an action. The *Robot::notifier* subclass has an *onRobotStatusIs(void)* virtual method which is called every time the *Robot::statusIs(status:RobotStatus)* method updates the *Robot::status_* attribute. This virtual method is overwritten by a child class of the *Robot::notifier*, the *OperatorReactor::notifier*. The *OperatorReactor::notifier* accesses the updated

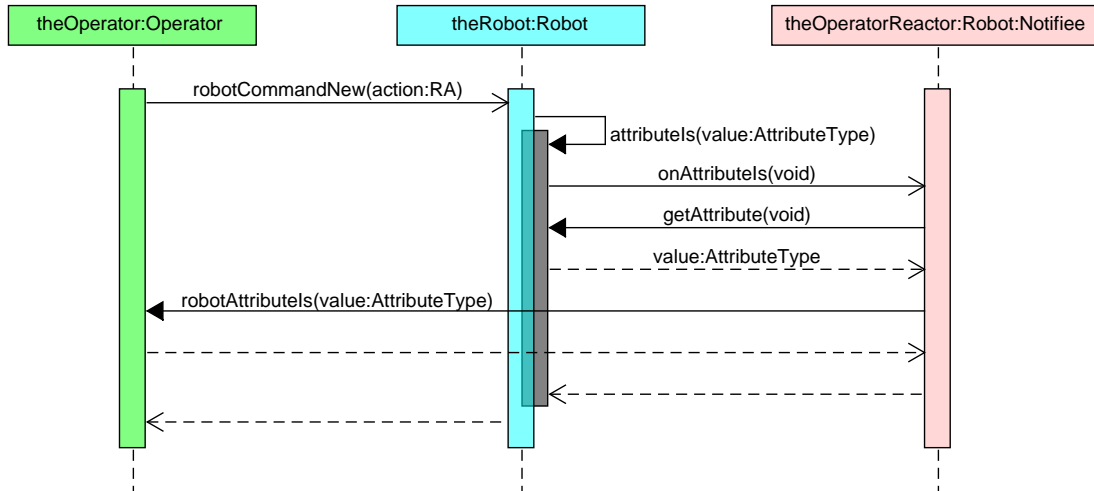


Figure 3.3: Sequence diagram showing the flow of information during the execution of a *Robot Action* command sent from the *Operator* (green) to a *Robot* (blue). During the execution of the *Robot Action*, an attribute of the *Robot* is changed, for example the `Robot::status_` attribute may be set to `COMMAND_COMPLETED`. This instigates a new sequence of actions, whose sub-process is colored grey. The changed attribute executes the `OnAttributeIs()` virtual method of the *theRobotNotifier* (pink). An instance of a *Robot::Notifier*, *theOperatorReactor*, an extension of the *theRobotNotifier* of which `OnAttributeIs()` has been defined, notifies the *Operator* of the attribute change. This is an example of the call-down, notify-up paradigm, where the *Operator* class called down the *Robot Action* to be implemented and the *Robot* class notified the *Operator* when the action was completed.

attribute value (the *RobotStatus*), and updates its own *robotStatus_* attribute. This interface is thread-safe, and formalizes communication between classes and processes to allow easy scalability.

3.1.3 Threading

Each *Robot* instance is assigned its own thread, with a typical execution rate of 1kHz, a frequency sufficient for haptic feedback. The *Graphical Display* is assigned an individual thread running sufficiently fast to render graphics, while main execution thread is reserved for the *Operator*. Synchronization among threads occurs through the notifier-notiffee structure introduced above.

3.2 Collaboration Models for Inclusion Segmentation Task

We demonstrate the MMSF introduced in the previous section by extending its base classes to support the task of inclusion segmentation in soft tissue. In this section we introduce five different collaboration models that can segment a stiff inclusion from its surrounding soft tissue. These collaboration models, pictured graphically in Figure 3.4, vary in how they combine the robotic and human agents' intents. A collaboration model for the inclusion segmentation task includes a method to map tissue mechanics and a method to segment the boundary of the inclusion.

Each of the described collaboration models instantiates at least one extension of each base class mentioned in Section 3.1. Table 3.1 lists the specific classes instantiated for each collaboration model.

Recall that the *Operator* class will query its *Procedure* member variable for each *Robot Action*. For the inclusion segmentation task three new *Robot Actions* are created: *Palpate Point Autonomously Robot Action*, *Palpate Point Teleoperatively Robot Action*, and *Palpate Tissue Teleoperatively Robot Action*. The *Palpate Point Autonomously Robot Action* consists a series of three *Move Robot Actions*: the first sub-action is to position the robot at the desired palpation location but just above

the tissue, while the remaining two sub-actions are the downward and upward strokes of palpation. For the *Palpate Point Autonomously Robot Action*, the downward stroke of the palpation command is force-controlled, to limit stress on the environment. The *Palpate Point Teleoperatively Robot Action* implements a bilateral teleoperation controller where the position and velocity are exchanged between the master and slave robots. However, only the palpating axis is teleoperated; the other axes are controlled by the robotic agent to fix the location of the point palpated. In the *Palpate Point Teleoperatively Robot Action* action, the *Robot* class is responsible for partitioning the force-displacement data into ranges that represent a single palpation, from which a stiffness can be approximated. This is in contrast to its autonomous counterpart that by design encodes the downward and upward strokes of the palpation and thereby knows how to partition the force-displacement data. The *Palpate Tissue Teleoperatively* command is similar to the *Palpate Point Teleoperatively Robot Action*, except that the human operator is in control of all degrees of freedom, and can palpate in any desired location.

Once the *Inclusion Segmentation Robot* has indicated that the palpation has completed, the *Inclusion Segmentation Operator* will examine the new palpation data. Depending on the new data acquired and the collaboration model, the *Inclusion Segmentation Operator* may create new *Robot Actions* in response.

We now describe each of the collaboration models prototyped.

3.2.1 Bilateral Teleoperation

In bilateral teleoperation, a master and slave robot are connected with a bilateral proportional-derivative controller. The human operator is free to explore an artificial tissue via palpation by the *Bilateral Teleoperation Inclusion Segmentation Operator* issuing a *Palpate Tissue Robot Action* command to both master and slave robots. Once the human operator is confident in their ability to define the boundary of the inclusion, the human operator will indicate the boundary of the inclusion through the *Human-Interface Inclusion Segmentation Graphical Display*.

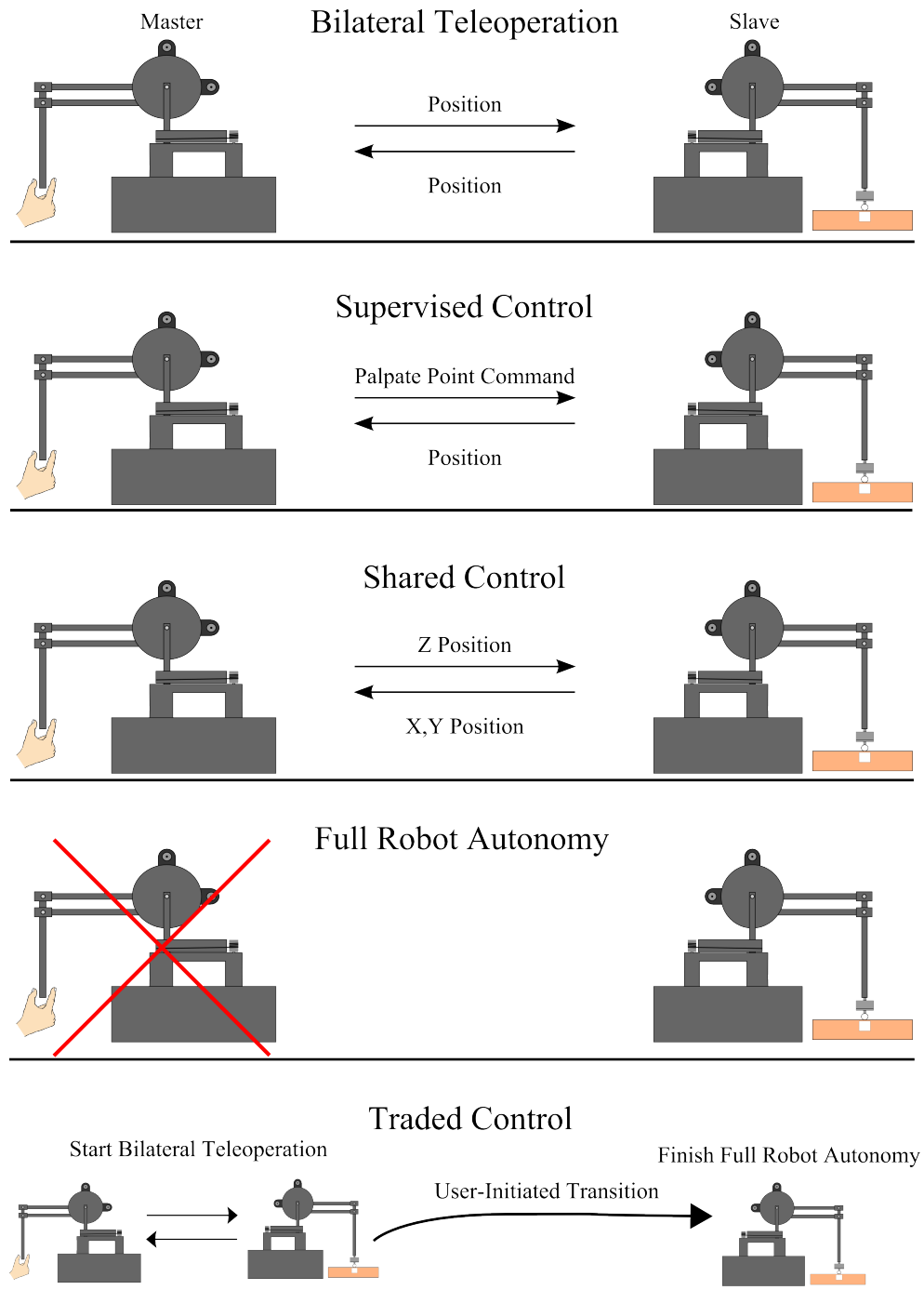


Figure 3.4: Collaboration models tested. Five different collaboration models are shown along with their information exchange between the master and slave robots.

Table 3.1: Class Instantiation Table

Collaboration Model	Procedure	Operator	Slave Robot	Master Robot	Graphical Display
Bilateral Teleoperation	*	Bilateral Teleoperation	Teleoperated Palpation	Human Interface	Human Interface
Supervised Control	*	Supervisory Bilateral Teleoperation	Palpation	Human Interface	Human Interface
Fully Autonomous	Autonomous	Autonomous	Palpation	N/A	*
Shared Control	Shared Control	Autonomous	Teleoperated Palpation	Human Interface	*
Traded Control	*	Traded Human-Autonomous	Teleoperated Palpation	Human Interface	Human Interface

Listed in each collaboration model's row are the extensions to the task-specific extensions of the base classes that are instantiated to form the collaboration model. A * indicates that the instantiated class is the inclusion segmentation extended base class

3.2.2 Supervised Control

Supervised control uses the same class instantiation set as bilateral teleoperation with the exception that its operator is the *Supervisory Bilateral Teleoperation Inclusion Segmentation Operator*, an extension from the *Bilateral Teleoperation Inclusion Segmentation Operator*. This operator differs from its parent by extending the `statusIs(status:OperatorStatus)` method such that the slave robot follows palpation commands indicated by the *Human-Interface Inclusion Segmentation Robot*. A palpation command is created when the human operator performs a palpation motion with the master robot (in free space). Once the user finishes the palpation motion, the location of the palpation parameterizes a *Palpate Point Autonomously Robot Action* sent to the slave robot. Like the bilateral teleoperation collaboration model, the boundary of the inclusion is identified by the user through the *Human-Interface Inclusion Segmentation Graphical Display*.

3.2.3 Fully Autonomous

The fully autonomous inclusion segmentation model was developed in the previous chapter and is briefly summarized here. The fully autonomous collaboration model, encoded in the *Autonomous Inclusion Segmentation Operator*, segments the hard inclusion from surrounding soft tissue by using an autonomous palpation algorithm that develops a classifier used to sort points based on their stiffness as part of the hard inclusion or part of the surrounding soft tissue. The initial stiffness classifier value is improved during successive iterations using an expectation-maximization (EM) algorithm. Improving the stiffness classifier using data acquired from palpation can map the classifier from its initial value to one specialized for the tissue sample as hand. The E-step of the EM algorithm is facilitated by a Markov random fields algorithm. Markov random fields classify a point using the information from the point's neighborhood in addition to the information stored in the point. This classification strategy makes intuitive sense for an inclusion segmentation task as we would expect the neighborhood around a point to have the same labeling as the point of interest.

Once all the points are relabeled the M-step of the algorithm, using Gaussian discriminant analysis, generates an updated stiffness classifier for the next iteration of the palpation algorithm.

Dense resolution around points theorized to belong to the hard inclusion as well as the supposed boundary of the inclusion is facilitated in two ways. First, if a palpated point is classified as part of the hard inclusion against the one-dimensional stiffness classifier, the neighborhood around this point is then palpated by the *Autonomous Inclusion Segmentation Operator* creating new *Palpate Point Autonomously* commands and sending these commands to the *Palpation Inclusion Segmentation Robot*. Second, between iterations of the EM algorithm the next set of points to be mapped is formed as the union of neighborhoods around points on the supposed boundary of the inclusion. After each iteration of the EM algorithm, the neighborhood radius gets progressively smaller, ensuring a higher resolution of the inclusion boundary. The algorithm exits once the classifier has converged and the neighborhoods around the boundary points of the inclusion have been explored. At this point, the final boundary of the inclusion is output.

The fully autonomous collaboration model instantiation set in Table 3.1 contains an instance of the *Autonomous Inclusion Segmentation Procedure*, which includes *Palpate Point Autonomously* actions, created by the *Autonomous Inclusion Segmentation Operator*, to generate the initial grid of points explored as the first part of the inclusion segmentation algorithm.

3.2.4 Shared Control

Shared control is a human-robot collaboration model in which we use the operator from Section 3.2.3 in conjunction with bilateral teleoperation. In this collaboration model, we sought to use the methodology of the fully autonomous palpation algorithm but give the human agent control over tissue interaction. The human agent controls the palpating degree of freedom during the downward stroke until the force threshold is reached, at which point the robotic agent controls the upward stroke of palpation. In addition the robotic agent decides where to palpate. In this sense, the human agent

is solely responsible for imparting forces on the tissue through palpation, adding a measure of safety with a human in the loop. The tissue exploration procedure is the same as in Section 3.2.3, and the autonomous agent is responsible for identifying the outline of the inclusion. The *Shared Control Inclusion Segmentation Procedure* differs from the *Autonomous Inclusion Segmentation Procedure* by replacing all *Palpate Point Autonomously Robot Actions* with *Palpate Point Teleoperatively Robot Actions*. The *Teleoperated Palpation Inclusion Segmentation Robot* class is extended from its parent class to handle this type of *Robot Action*.

3.2.5 Traded Control

In traded control, the control authority alternates between the human operator under bilateral teleoperation at the beginning of the procedure, and the robotic agent under full autonomy for the remainder of the procedure. The motivation for this collaboration model is to take advantage of the human operator’s ability to quickly recognize the difference in stiffness between the hard inclusion and surrounding soft tissue, as well as provide the initializing classifier for the fully autonomous machine learning methods. Once the human operator palpates enough points in the bilateral teleoperation setup to confidently seed an initial classifier, the machine learning algorithms developed in Section 3.2.3 guide the remainder of the tissue mapping and classification. This collaboration strategy was designed to provide the same performance as the fully autonomous collaboration model, but rely on the human operator to seed the algorithm.

To instantiate this collaboration model, the *Traded Human-Autonomous Inclusion Segmentation Operator* is extended from the *Autonomous Inclusion Segmentation Operator*. The extended *Operator* uses the inclusion segmentation algorithm from Section 3.2.3, but includes additional methods so the user can initialize the classifier used in the segmentation algorithm using the method described above.

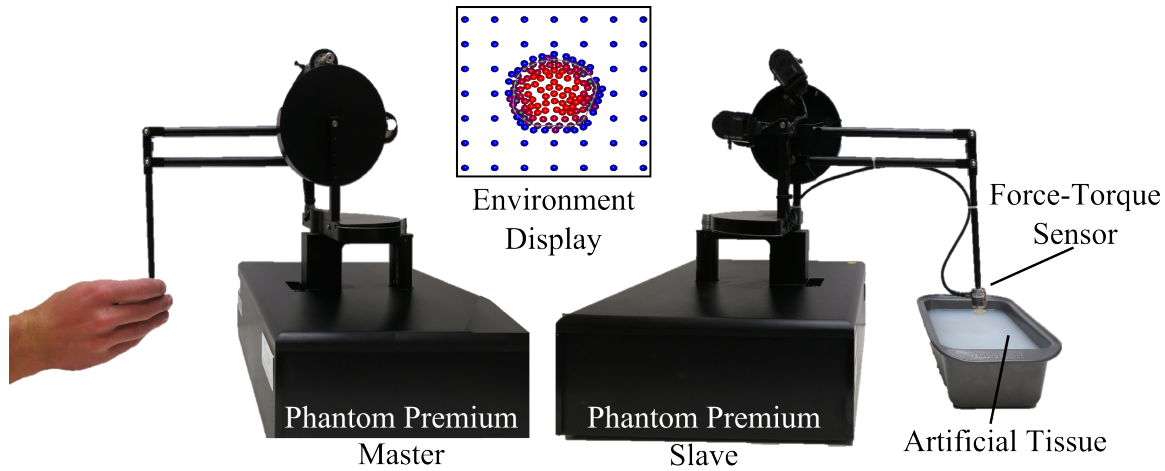


Figure 3.5: Experimental setup. A human operator (left) sends commands to the slave robot (right) that is positioned to palpate an artificial tissue with an embedded stiff inclusion. A graphical display shows the collected data (see Section 3.3.3).

3.3 Experiment

The five collaboration models, all derived from the MMSF introduced in Section 3.1, were tested with the exemplar task of segmenting a hard inclusion from surrounding soft tissue. One robot was used as a human input device, while another was positioned to palpate the artificial tissue. As tissue mechanics were mapped from force-displacement data, the palpated points and their stiffnesses were displayed graphically. The experiment concludes with identification of the boundary by the human or robotic agent. Evaluation metrics are generated by a comparison between the estimated location of the inclusion and the actual location. Other metrics included maximum force exerted on the tissue, as well as the number of palpations and experiment duration.

3.3.1 Artificial Tissue

An artificial tissue was created to simulate the mechanics of a hard inclusion in soft tissue for experimentation. The artificial tissue was constructed from silicone (Smooth-On, Inc. Easton, PN, USA). We chose silicone instead of plastisol (which

was used in Chapter 2) because silicone tissues take less time and effort to construct compared to plastisol tissues. A gelatin-based artificial tissue would also be insufficient due to the extensive testing the artificial tissue underwent. A pre-made, cylindrical hard inclusion with a diameter of 25 mm was made from pure silicone. After the hard inclusion cured, the artificial tissue base was poured using one part silicon-thinner to three parts silicone. During the curing process of the softer tissue, the hard inclusion was positioned into the center of the artificial tissue base, 4 mm below the surface.

3.3.2 Robotic Palpation

Two Phantom Premium 1.5A haptic devices were employed as the master and slave devices (Figure 3.5). The human operator grasped the end effector of the master robot while the slave device was positioned to palpate the artificial tissue. For autonomous motions, third-order polynomial desired trajectories of the robot end-effector were input to a proportional-derivative controller. These trajectories were generated at the onset of a new *Move Robot Action*, interpolating between the current location and the location parameterizing the *Move Robot Action*.

All collaboration models require an estimate of the stiffness of a palpated point. To acquire a stiffness estimate, a Nano-17 force-torque sensor (ATI Industrial Automation, Apex, NC, USA) attached to end point of the slave robot acquired force data while position measurements were taken from the Phantom Premium. A linear spring model approximated the tissue mechanics using only the data from the downward stroke. Force-displacement data used to approximate a stiffness was thresholded between 1.2 N and 1.8 N of force, where the force-displacement relationship was nearly linear [44].

3.3.3 Graphical User Interface

A graphical user interface was created to serve two purposes: (1) provide a means for a human operator to see and interpret the stiffness data acquired during palpation and (2) provide a means by which a human operator can select data points previously

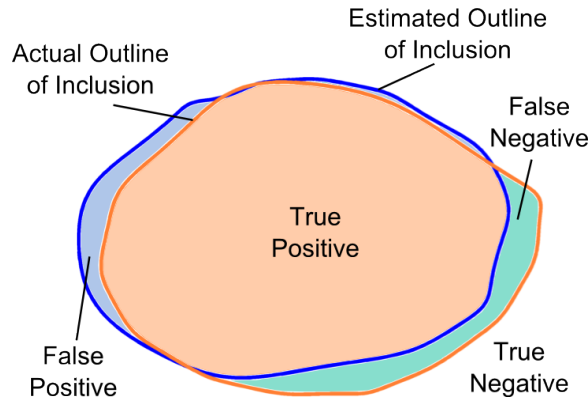


Figure 3.6: A graphical explanation of the evaluation metrics used in the study. The blue boundary is the estimated outline of the inclusion, and the orange boundary is the actual outline of the inclusion.

acquired, either for inclusion boundary segmentation, or to initialize the stiffness classifier, as is required by the traded control collaboration model (Section 3.2.5).

3.3.4 Evaluation Metrics

Several metrics measured the effectiveness of the algorithm by comparing the predicted and actual locations of the hard inclusion. Figure 3.6 provides a graphical explanation of these metrics. The intersection between the predicted and actual hard inclusion locations is correctly predicted to be part of the hard inclusion (true positive). All other area predicted as the hard inclusion is falsely identified (false positive). Likewise, all remaining area of the registered inclusion not intersecting the predicted location of the hard inclusion is falsely identified as part of the surrounding soft tissue (false negative). All other area in the workspace not predicted as part of the hard inclusion or registered as such are classified correctly as part of the surrounding tissue (true negative). Sensitivity and specificity summarize these four non-intersecting quadrants. Sensitivity measures the proportion of true positives which are correctly identified as such while specificity measures the proportion of true negatives which are correctly identified. A perfect prediction algorithm would have sensitivity of 1.0 and specificity of 1.0. For inclusion segmentation, a sensitivity close to 1.0 is more

important than a specificity close to 1.0; to ensure that no potentially cancerous tissue is left behind. The autonomous palpation algorithm developed in [44] used in full robot autonomy, shared control, and traded control was developed to maintain a minimum sensitivity of nearly 1.0, even at the sacrifice of some specificity.

Other metrics tabulated include the total number of palpations, the maximum force applied on the tissue by the slave robot, and the duration of the experiment. The experiment duration is defined as the time from when the slave robot receives its first command to when the boundary of the inclusion is identified.

3.3.5 Protocol

Each collaboration model was tested once. Every tested collaboration model used a different human operator naive to the properties of the hard inclusion, to avoid learning factors in the experiment. The shape and location of the hard inclusion were constant for all testing.

3.4 Results and Discussion

3.4.1 Software Framework

In Table 3.1 we showed the class instantiation set unique to each collaboration model. With the differences between these instantiated classes shown in Figures 3.1 and 3.2 and described in Section 3.1, we can see how a wide variety of different behaviors, evident in different forms of human-robot collaboration models, share much of the same code. We have shown in Section 3.2 that differences between collaboration models typically manifest in the *Operator* issuing different *Robot Actions* to *Robots* extended to interpret those commands. Indeed, some collaboration models only differ in their class instantiation set by one or two classes, putting the developer in a position to leverage as much code as possible between very different collaboration models. We have shown in Figure 3.3 and Section 3.1.2 how the notifier/notifyee interface standardizes inter-class communication to facilitate rapid prototyping.

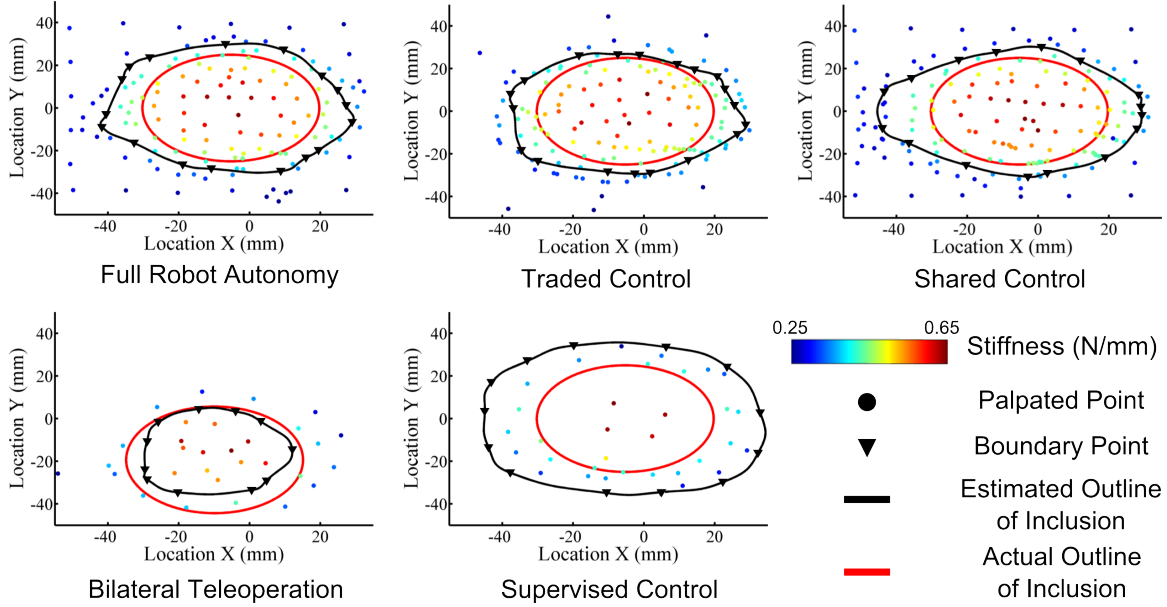


Figure 3.7: Experimental results showing the palpations, estimated and actual boundary of the inclusion for the five collaboration models. The black line interpolates through the boundary points in black and forms the derived boundary of the inclusion. The red line represents the actual location of the inclusion. The circles represent the stiffnesses of the points palpated, with red being most stiff and blue being the least stiff.

The call-down, notify-up structure was effective in being able to handle our surgical task and experimental setup. In Section 3.1.2, we introduced our manifestation of the call-down, notify-up structure when the *Operator* called down a *Robot Action* to the *Robot*. After data acquisition, the *Robot* notified the *Operator* of the new data (see Figure 3.3). In a future design, if we wanted to repeat this task with a different slave robot, the *Inclusion Segmentation Operator* would not have to be recoded to interface with the new device. Furthermore, the *Robot* used in the procedure need not change. Instead, a different *Robot Hardware* class would be written to interface between the MMSF and the particular hardware setup.

3.4.2 Inclusion Segmentation

Figure 3.7 shows the experimental results for each collaboration model, including the data points collected via palpation, the estimated boundary of the inclusion, and the actual boundary of the inclusion. Table 3.2 shows the metrics gathered from testing each collaboration model. For each collaboration model, sensitivity and specificity metrics were collected from the estimated boundary of the inclusion. In addition, maximum force applied, the number of palpations, and the experiment duration were recorded.

The results in Table 3.2 shows that collaboration models which used force-thresholding result in lower maximum force than collaboration models whose downward palpation motions were under human control. Supervised control, full robot autonomy, and shared control have roughly the same maximum force: approximately 1.8 N. However, among these three collaboration models, shared control has a slightly larger maximum force exerted on the tissue. This is because the downward stroke of the palpation for this collaboration model is controlled by the human operator, so when the force threshold was reached, the end effector of the robot had a larger velocity than if the downward stroke were under fully autonomous control. For the remaining collaboration models, recall that in the first phase of traded control, the robot was controlled through bilateral teleoperation. We would expect each bilateral teleoperation user to exert a different maximum force on the tissue. In this experiment, the user who tested the traded control model happened to exert a larger maximum force on the tissue compared to the user who tested bilateral teleoperation.

In both bilateral teleoperation and supervised control, the number of palpations as well as the experiment duration were less than the other collaboration models. However, for these same collaboration models, either the sensitivity or the specificity were worse compared to models that used the fully autonomous exploration and boundary identification methods. In the case of bilateral teleoperation, the user identified a boundary that was entirely included in the actual boundary of the inclusion. In the supervised control case, the user selected a boundary far in excess of the actual boundary, resulting in a low specificity.

The results in Table 3.2 are not conclusive in terms of evaluating the merits

of different collaboration models, rather they demonstrate the implementation of the MMSF and motivate the design of future user studies. In Section 4.2.1.1, we perform a multi-user study that involves more complicated shapes to segment and is implemented on a da Vinci Surgical System. The experimental results here are an illustration of the MMSF presented in this chapter, to show the nature of data that can be acquired quickly by formalizing our task with the MMSF.

3.5 Conclusions

In this work we introduced a software framework capable of supporting a variety of human-robot collaboration models with differing amounts of autonomous control. We implemented an inclusion segmentation task by properly extending the base classes forming the software framework, and programmed five different collaboration models. Results in an inclusion segmentation experiment indicated tradeoffs in sensitivity, specificity, duration, and safety implications for this task. The framework was demonstrated to be easily extensible, and have a well defined inter-class communication protocol capable of supporting rapid development while being thread-safe.

The work presented in this chapter shows promise that the Multilateral Manipulation Software Framework can successfully be used to develop human-robot collaboration models for surgical tasks. In the next chapter we extend and test the MMSF in a number of ways. First, we conduct a user study where the performance of different human-robot collaboration models are compared against each other. We perform this study so we can quantify the value of blending human and robotic agent input. Second, we extend the MMSF to other surgical tasks and robotic systems, to test its flexibility.

Table 3.2: Experimental results for all collaboration models

Collaboration Model	Sensitivity	Specificity	Max Force (N)	Number Palpations	Experiment Duration (s)
Bilateral Teleoperation	0.64	1.00	2.11	114	273.43
Supervised Control	1.00	0.62	1.81	45	210.48
Full Robot Autonomy	1.00	0.83	1.81	148	370.41
Shared Control	1.00	0.81	1.84	168	465.85
Traded Control	1.00	0.87	3.41	155	338.25

Chapter 4

Implementations and Extensions of the Multilateral Manipulation Software Framework

This chapter presents implementation and extensions of the Multilateral Manipulation Software Framework introduced in the previous chapter. In Section 4.1, we describe implementation details to use the MMSF to control two different surgical robots: the da Vinci Research Kit (dVRK) [41] and the RAVEN-II surgical robot (hereafter abbreviated as RAVEN)[82] (Applied Dexterity, Seattle, WA). In Section 4.2, we present and draw conclusions from a user study conducted to test four human-collaboration models from the inclusion segmentation task against each other. This work was implemented on the da Vinci Research Kit. In Section 4.3, we extend the MMSF to an entirely new surgical task: a debridement task conducted by the RAVEN. In Section 4.4, we extend the MMSF again to an electrocautery task performed with the dVRK, and use the MMSF described in Chapter 3 to conduct a pilot study where different pairings of collaboration models between stiff region exploration and electrocautery along the boundary exhibit different performances. Each of the sections in this chapter extends or validates the contributions from Chapter 3 by demonstrating the MMSF in a wider context.

4.1 Implementation of the MMSF on Surgical Robots

In this section we describe how to implement the MMSF on the dVRK and the RAVEN). The dVRK was used as the robotic platform in Sections 4.2 and 4.4, while the RAVEN-II Surgical Robot was used in Section 4.3. Both of these robotic systems require more complex software interfaces with the MMSF than previously developed for the Phantom Premium devices used in Chapter 3.

4.1.1 Implementation of the MMSF on the da Vinci Reserach Kit

In this subsection we describe the steps necessary to implement the MMSF to the dVRK (see Figure 4.1). The dVRK interfaces with the CISST-SAW framework, developed primarily at Johns Hopkins University [41]. The CISST-SAW software framework is a component-based architecture, which enables well-specified integration of robot hardware and stereovision with control and actuation. A master computer, running CISST-SAW, provides a single access point which can be used to prototype tasks for the dVRK. The MMSF is not designed to be run as a module of CISST-SAW; such a specification would detract from its applicability to systems not operating on the CISST-SAW framework, like the RAVEN robot we use in Section 4.3. For this chapter, we needed to build a communication link between the MMSF and the dVRK which allows us to implement a variety of human-robot collaboration models in a stable fashion. To implement the MMSF to the dVRK we used two different computers, one executing sensing and control of the dVRK, and another computer running the MMSF. Communication between the two platforms was implemented with TCP/IP sockets on both computers which were connected via an Ethernet cable. To facilitate effective communication, two new processes were created on the dVRK side, one for Ethernet communication (running at 100Hz), and another process which interprets any received Ethernet packages and responds as commanded. The MMSF implemented an additional process running the *Robot Hardware* class extended to

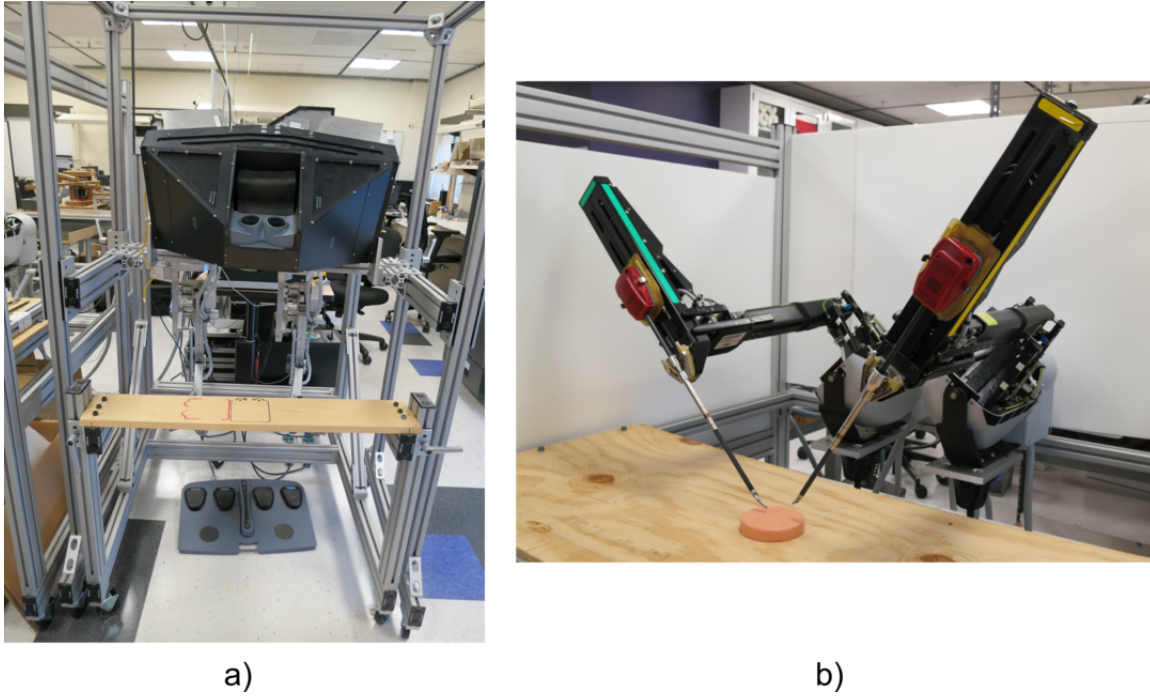


Figure 4.1: The da Vinci Research Kit. (a) Master console with master manipulators and stereovision hood. (b) Patient-side manipulator arms outfitted with large needle drivers.

support the dVRK. This process would execute sufficiently fast to receive all Ethernet packets from the dVRK. The Ethernet packet sent from the dVRK to the MMSF for this study is illustrated in Table 4.1. In addition to general state information, data specific to the stiff region segmentation subtask implemented in Sections 4.2 and 4.4 (in particular, the stiffness and location of palpated points) are also included.

The communication protocol between the computers running the dVRK and MMSF represents a significant design choice. The communication latency between the dVRK and MMSF is not stable enough with our given hardware setup to send out force/torque data as was used in Chapter 3. In response to this problem, the unit of information sent from the MMSF to the dVRK was abstracted to the *Robot Action*, which is stable despite latency (Table 4.2). The dVRK, when issued a new command, would interpret and implement the command. In this sense, the processing reserved for the *Robot* class in our previous work is now implemented on the dVRK.

Table 4.1: Data Packet from dVRK to MMSF

Symbol	Description
<i>PacketNumber</i>	Ethernet packet number
<i>Complete</i>	Boolean value indicating completion of issued <i>Robot Action</i>
<i>Stiffness</i>	Calculated stiffness of most recently palpated point
<i>P</i>	Location of most recently palpated point
<i>MSML</i>	Position and rotation of the left master-side manipulator
<i>MSMR</i>	Position and rotation of the right master-side manipulator
<i>PSML</i>	Position and rotation of the left patient-side manipulator
<i>PSMR</i>	Position and rotation of the right patient-side manipulator
<i>PSMLDesired</i>	Transform of <i>MSML</i> into slave robot workspace
<i>PSMRDesired</i>	Transform of <i>MSMR</i> into slave robot workspace
<i>GripperL</i>	State of left master-side gripper DOF
<i>GripperR</i>	State of right master-side gripper DOF
<i>F</i>	Force/torque data from the attached sensor

When a command is completed, the dVRK would set the appropriate flag in its outgoing Ethernet packet (Table 4.1), and the MMSF would progress through its state machine.

4.1.2 Implementation of the MMSF on the RAVEN-II

The slave robot used in Section 4.3 is the RAVEN robot initially developed at Washington University and University of California, Santa Cruz (see Figure 4.2). The RAVEN robot is an open-architecture surgical platform designed for the academic community to share findings and perform collaborative research. It is similar in function to the da Vinci robot, with two 7-degree-of-freedom arms (pose, position, and grasp). The RAVEN interfaces directly with da Vinci surgical tools but is only calibrated for one specific type of tool: the large needle driver.

Table 4.2: Data Packet from MMSF to dVRK

Symbol	Description
<i>PacketNumber</i>	Ethernet packet number
<i>RobotArm</i>	Enumerated value indicating the left or right robot arm
<i>RobotID</i>	Enumerated value indicating master or slave robot
<i>ActionID</i>	Enumerated value indicating commanded <i>Robot Action</i>
<i>P</i>	Vector indicating desired position of <i>Robot Action</i>

Like the dVRK, the RAVEN robot has a computer dedicated to running the surgical robot hardware. However, the default setup provided by Applied Dexterity includes a second computer, connected to two Geomagic Touch haptic devices (Geomagic, Morrisville, NC, USA). In the standard setup provided by Applied Dexterity, the human input device laptop, connected to the Geomagic Touch devices, senses positions, poses, and gripper angles. This state information is sent via a UDP socket to the computer running the RAVEN hardware, where it is interpreted and provides reference signals used to control the RAVEN arms. State information is not sent from the RAVEN computer to the computer connected to the Geomagic Touch devices, although bidirectional communication is a desired feature considered for future RAVEN software releases. Joint control for the RAVEN is performed locally on the computer running the RAVEN, resulting in closed-loop control locally with respect to the RAVEN and open-loop control with respect to the computer connected to the human input devices.

The standard configuration of the RAVEN provided by Applied Dexterity provides a simple access point for the MMSF: the computer connected to the Geomagic Touch devices. The MMSF executes on the same computer as the one connected to the human input devices, and reads state information. This state information is augmented according to the implemented collaboration model, and *Robot Actions* are formed as before in Chapter 3. However, instead of a proportional-derivative controller guiding the slave robot along third-order polynomial trajectories as in Section 3.3.2, the trajectory reference locations, used as inputs to the controller, are sent to the RAVEN

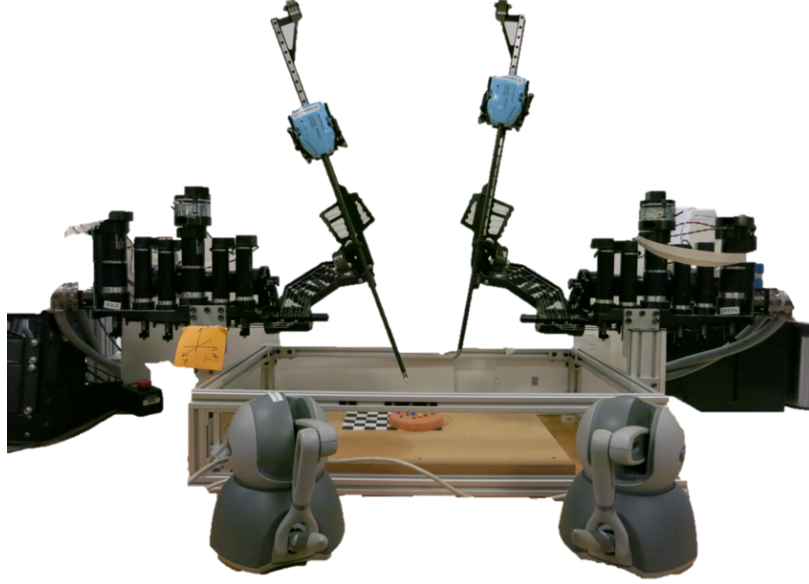


Figure 4.2: The RAVEN-II Surgical Robot. Two Geomatic Touch haptic devices (bottom) are used as the human input devices.

robot directly using the UDP socket, where local controllers on the computer connected to the RAVEN robot follow the trajectory. Since the MMSF is augmenting state information in an otherwise functional information pipeline between the human input device computer and the RAVEN hardware computer, the implementation details for the MMSF with the RAVEN need not be as involved as the details required to use the MMSF with the dVRK.

4.2 Human-Subject Study of Human-Robot Collaboration Models for an Inclusion Segmentation Task

In this section, we verify the Multilateral Manipulation Software Framework developed in the previous chapter by conducting a user study where we quantify how

variations in human-robot collaboration models affect task performance. In the *inclusion segmentation experiment*, twelve participants explored an artificial tissue and identified the inclusion boundary under the collaboration models of (1) teleoperation, (2) supervised control, (3) traded control, and (4) full autonomy. In the *boundary identification experiment*, we isolate the performance of human and robotic agents in the boundary identification sub-task; participants and a robotic agent independently identified the boundary of four virtually palpated tissues. Results from the inclusion segmentation experiment indicate that human agents complete the task faster; teleoperation had the fastest task times. Results of both experiments indicate that the robotic agent identifies boundaries with higher sensitivity and less variance than human agents. This suggests that task accuracy increases when a robotic agent segments the boundary, while including a human agent can decrease the overall task time.

This study was performed in collaboration with Kirsten E. Kaplan, as reported in Kaplan *et al.* [83]. Kaplan’s contributions included design of the graphical display rendered to the human subjects, the enhanced and extended boundary identification algorithm, design of tool fixtures, design of the experimental protocol, conduction of the human subject study, and analysis of the results. My contributions to this work include extending the Multilateral Manipulation Software Framework to support the dVRK (described in Section 4.1), experiment design, and software development necessary to conduct the user study. The conclusions drawn from this section are included as part of this dissertation since they go beyond the pilot study conducted as part of the previous chapter and rigorously compares different collaboration models against each other.

4.2.1 Implementation of the Inclusion Segmentation Task

In this study, our implementation of the inclusion segmentation task can be divided into two main sub-tasks: *tissue exploration*, in which an artificial tissue is palpated repeatedly to locate a stiff inclusion, and *boundary identification*, in which information gathered from these palpations is used to identify the boundary of the stiff inclusion.

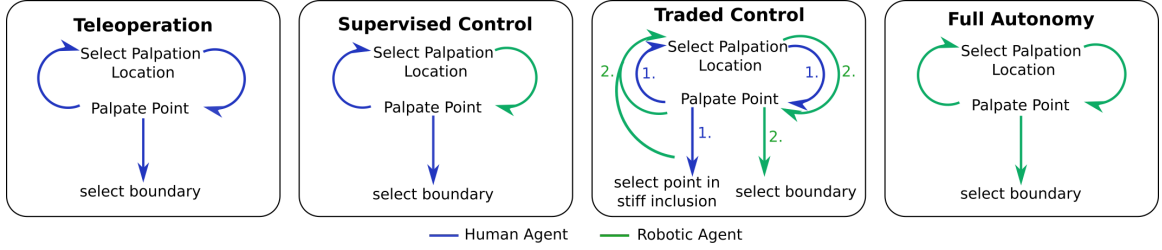


Figure 4.3: Human-robot collaboration models tested on an inclusion segmentation task. For each collaboration model, the figure indicates which agent, human or robot, is responsible for completing specific aspects of the task. The numbers 1 and 2 in Traded Control indicate the order in which the two agents have control; the human agent selects and palpates points until ready to identify a single point in the stiff inclusion, at which point the robotic agent gains control and completes the task.

The tissue-exploration sub-task is implemented differently for each of our collaboration models. The boundary identification sub-task is binary, with either the human or robotic agent responsible for identification, and is executed after completion of the tissue exploration sub-task. The following subsections describe the implementation and motivation of our human-robot collaboration models and our improved autonomous boundary identification algorithm. The human-robot collaboration model section focuses on implementation of the tissue exploration sub-task, though it also indicates which agent is responsible for boundary identification in each collaboration model.

4.2.1.1 Human-Robot Collaboration Models

Each collaboration model assigns different degrees of shared control between the human and robotic agents in order to assess how overall task performance varies. In this work, we use collaboration models developed in the previous chapter. In order of increasing robot independence, they are: (1) teleoperation, (2) supervised control, (3) traded control, and (4) full autonomy (Figure 4.3). To review these collaboration models in the context of this study, we briefly summarize them below. More detail can be found in Chapter 3.

Teleoperation In teleoperation, the human controls the master robot manipulators and these movements are mimicked by the slave robot. The human operator is free to explore the workspace and palpate the tissue at will. The human palpates enough points to feel confident in their ability to identify the boundary and then selects the boundary from amongst the palpated points. This collaboration model represents the control case, where a human commands all aspects of the experimental task.

Supervised Control Supervised control consists of the human issuing high-level commands and the robot performing repetitive, lower-level operations. The human moves the master manipulators to the location they desire to palpate and closes the gripper on the end of the master manipulator. The slave robot then proceeds to palpate the tissue at the commanded location. Robotic palpation consists of a downward movement which transitions to the upward stroke when the force of palpation exceeds 1.8 N. The human agent determines when enough points have been palpated, then selects the inclusion boundary. This collaboration model was included to investigate task performance when a human is responsible for the exploratory aspects of the procedure—choosing where to palpate in order to completely locate the boundary as well as segmenting this boundary—while the robot performs those aspects that are repetitive and benefit from repeatability and precision in applied force.

Traded Control In traded control, the procedure switches from teleoperation to full autonomy when indicated by the human. For this study, the human is responsible for palpating the tissue roughly 10 to 20 times and selecting one of these points as located in the stiff inclusion. This data is used to create an initial classifier for the machine learning algorithm, described in the following section, which provides input for the robotic agent to complete the palpation sub-task autonomously. The robotic agent is responsible for identifying the boundary in this collaboration model. This collaboration model was included to provide the human user with the task of initially locating the tumorous tissue within the larger workspace. The robot then takes the initial definition of inclusion from the human and performs additional palpations to fully locate and identify its boundary.

Full Autonomy The fully autonomous collaboration model used here is described in detail in Chapter 2. Briefly, an autonomous palpation algorithm is responsible for selecting palpation locations. It begins palpating a linearly-spaced grid of points on the surface of the tissue using a provided stiffness classifier to label points based on stiffness as either part of the stiff inclusion or part of the surrounding tissue. In Chapter 2 we demonstrated that the initializing classifier can come from elastography and that the results from autonomous exploration and segmentation are robust with respect to initial value. For the present study we set the initializing classifier to 0.4 N/mm. The classifier improves as the algorithm iterates with the potential for some points to be relabeled, allowing points previously considered too soft to be in the inclusion to be reclassified as stiff and vice-versa. Additional points in the vicinity of the inclusion boundary are palpated to increase resolution along the boundary. The approach used to identify the boundary once the algorithm has completed is described in detail in Section 4.2.1.2.

4.2.1.2 Boundary Identification Algorithm

For autonomous boundary identification, our algorithm begins by labeling all the palpated points as soft or stiff based on the final stiffness classifier from the autonomous palpation algorithm and selecting the stiff point with the largest x-value. This point becomes the start point for the search for the inclusion boundary. Each iteration of the initial search is performed by defining a search circle with a radius of 5 mm. All points within the bounds of the search circle are considered neighboring points, and are classified as stiff inclusion or soft tissue points. The circle is considered to be on the boundary if both inclusion and soft tissue points are located in its bounds. The search extends to the right from this point by translating the search circle until the boundary is located; the first boundary point is defined as the soft tissue point closest to the start point, similar to the approach used in Chapter 2.

Once the algorithm identifies an initial boundary point, each successive iteration of the boundary search proceeds by moving the search circle along the vector from the previous boundary point to the current one (Figure 4.4(a)). For the initial iteration, when only one boundary point has been identified, movement of the search proceeds

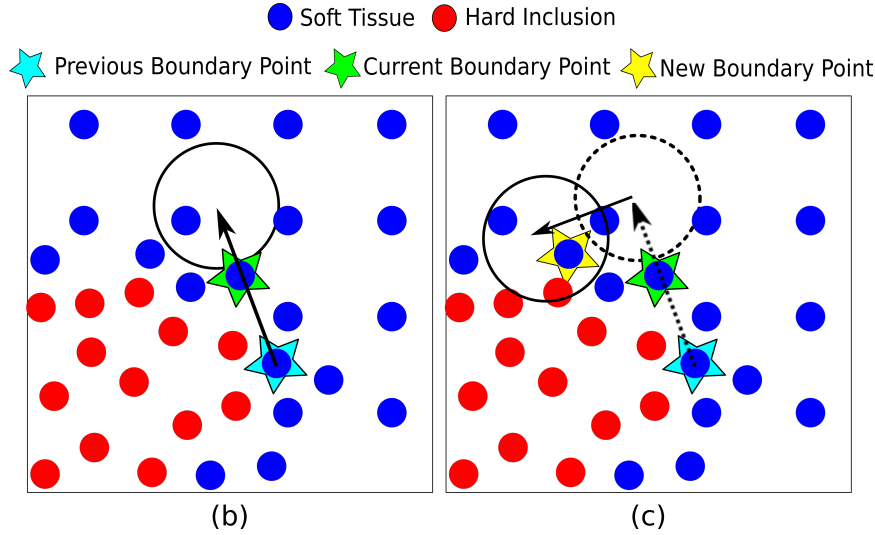


Figure 4.4: Illustration of boundary identification algorithm. Red and blue circles represent palpated points classified as stiff inclusion and soft tissue, respectively. (a) Initial step of search for each new boundary point moves a search circle (solid circle) along the vector from previous to current boundary point (solid line). Search is considered to be on the boundary if circle contains both soft and stiff points. (b) If the search circle from (a) was entirely outside the inclusion (dashed circle), the search (solid circle) is shifted to the left along a line perpendicular to the initial search line (solid line). If this circle is located on the boundary, the soft tissue point closest to the inclusion is selected as the next boundary point.

in the positive y-direction. This initial direction ensures that the search will move around the inclusion counterclockwise. If the circle is not located on the boundary, the algorithm determines if it is outside or inside the inclusion. In the former case the search circle will move to the left, as the counterclockwise directionality ensures that this will be the direction of the boundary (Figure 4.4(b)). If this circle is also outside the inclusion, the algorithm will proceed in a counterclockwise square search pattern around the last identified boundary point until the search circle is either on the boundary or in the inclusion. These events are reversed if the initial circle is inside the inclusion, with movement to the right and clockwise. The boundary searching algorithm concludes when the first boundary point returns within the bounds of the

search circle. A cubic spline interpolation is used to connect the boundary points and define the final inclusion boundary.

4.2.2 Experiments

Our study consists of two experiments. The first experiment is the inclusion segmentation experiment, in which participants perform the full inclusion segmentation task. Using a da Vinci Research Kit and a provided graphical interface, participants explored an artificial tissue by physically palpating it with the robot and then identified the boundary of a stiff inclusion located in the tissue (Figure 4.5). This experiment was performed under the four different human-robot collaboration models described in Section 4.2.1.1. Following completion of the first experiment, participants performed the second experiment, the boundary identification experiment. In this experiment, participants performed the sub-task of boundary identification for virtual tissues by selecting boundaries based on a series of points, palpated by our autonomous algorithm, that matched the type of data they collected in the inclusion segmentation experiment.

4.2.2.1 Participants

This study was conducted with twelve right-handed participants (6 males and 6 females, ages 24 to 35) who gave their informed consent to partake in the study. Seven of the participants had little to no previous experience with human-machine interface devices; the others had previously developed or used teleoperated and/or haptic devices. The experimental protocol was approved by the Stanford University Institutional Review Board.

4.2.2.2 Artificial and Virtual Tissues

Artificial tissues were created for the inclusion segmentation experiment. Each tissue consisted of a stiff inclusion surrounded by softer material. This approach mimics the mechanics of a stiff mass in soft tissue, with benefits over real tissue of extended shelf life and durability, lower cost, and the ability to design shapes and stiffnesses

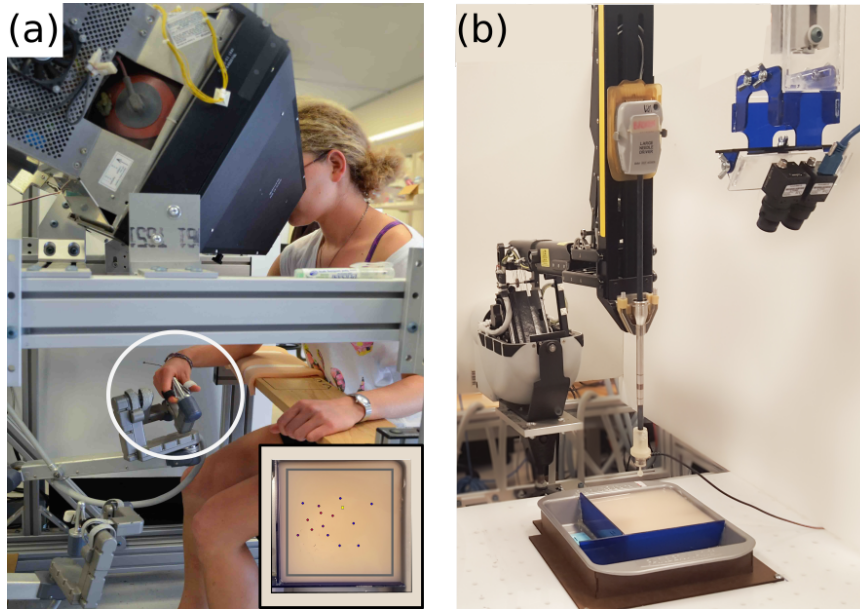


Figure 4.5: Experimental setup of inclusion segmentation experiment. (a) In collaboration models involving a human, the participant sits at the master console of a da Vinci Research Kit (dVRK) and controls the right master manipulator (circle). Inset is a portion of the user interface participants see during the tissue exploration sub-task; points on the image represent palpation locations, with color indicative of stiffness. (b) The dVRK patient-side surgical robot arm, with an attached force sensor, is controlled by a combination of the dVRK master manipulator in (a) and an autonomous robotic agent. Force-displacement data was collected using a force sensor attached to the slave arm. A stereoscopic camera system images the workspace. The video feed from the camera setup is provided to participants alongside the display shown in the inset in (a).

to desired specifications. Four artificial tissues were created, each with a differently shaped inclusion. Inclusions were made from pure silicone of Shore Hardness 00-30 (Smooth-On, Inc. Easton, PN, USA). The four shapes were roughly an oval, crescent, star, and heart, with maximum cross-sectional lengths ranging from 5 to 7 cm. Each inclusion was placed in a square container and surrounded by a soft tissue base made of three parts silicone with a Shore Hardness of 00-10 to one part silicone thinner. Additional soft base mixture was added so that the top surface of the inclusion was at a depth of 2 mm.

Virtual tissue models of a stiff inclusion embedded in soft tissue were created for the boundary identification experiment in MATLAB by defining and smoothing sets of stiffnesses in a two-dimensional array. Inclusion shapes for the virtual tissues were distinct from the stiff inclusions used for artificial tissues. Palpated points were generated for the virtual tissues by running the palpation algorithm described in Section 4.2.1.1 until completion.

4.2.2.3 Surgical Robotic Platform

Inclusion segmentation experiments were performed using a da Vinci Research Kit. Participants sat at the master console and manipulated the right master arm to maneuver a slave arm in the tissue workspace (Figure 4.5). A Nano-17 force-torque sensor (ATI Industrial Automation, Apex, NC, USA) attached to the slave arm was used to determine the force applied to the tissue perpendicular to the tissue surface. Force data combined with slave arm position data approximated the tissue stiffness for each palpated point using a linear spring model (Chapter 2).

During teleoperation, the position of the master manipulator of the dVRK provides the desired position for the slave arm. The desired position is the reference for a proportional-derivative (PD) controller that commands a force on the slave arm. The master manipulator was not used to command slave arm orientation. In autonomous control, the desired slave position was determined by the robotic agent and sent through the same PD controller to command slave position. In the teleoperation and traded collaboration models when the human physically palpated the artificial tissue, force feedback was provided to the user in the z-direction by scaling the force

from the force sensor by a factor of 0.7 on the slave arm and transforming the force to the master manipulator frame.

4.2.2.4 Graphical User Interface

Participants were provided with a graphical user interface for each experiment. The interface for the exploration phase of the inclusion segmentation experiment included a stereoscopic display of the workspace from two Flea3 cameras (Point Grey, Richmond, BC, Canada) as well as a static overhead image of the tissue with overlaid points (Figure 4.5). The static image indicated the location and stiffness of each point palpated to participants, as well as their current location in the workspace relative to the palpated points. The mapping of points from the workspace to the overhead image was based on scaling obtained from an initial registration of the slave robot coordinates in each corner of the artificial tissue. For the boundary identification aspect of both experiments, the interface consisted of a display of colored circles that represented palpated points. This displayed the same information as the static display in the exploration phase of the inclusion segmentation task. The relative stiffness of each circle was displayed on a color scale from blue (soft) to red (stiff).

4.2.2.5 Experimental Procedure

Each participant performed five trials in the inclusion segmentation experiment and four trials in the boundary identification experiment. The inclusion segmentation experiment was completed before participants performed the boundary identification experiment. Prior to the start of the experiments, participants were informed about the nature of the task and the surgical relevance of selecting a inclusion boundary for removal of a tumor.

Inclusion Segmentation Experiment Prior to training, participants watched a live demonstration on how to control the dVRK master manipulator for exploring and palpating the tissue. Participants then trained by performing the teleoperation and supervised control collaboration models on a training tissue with a circular stiff inclusion. Training rounds were conducted in the same manner as experimental trials

and familiarized participants with how the colored dots correlated to stiffness as well as which actions were required for the different collaboration models. Following completion of the two practice rounds, participants completed three experimental trials of palpating an artificial tissue to locate and then identify the boundary of a stiff inclusion. All participants received each collaboration model and inclusion shape once. The order in which participants received collaboration models and the pairing of inclusion shape with collaboration model were random. Across all participants, each inclusion shape was paired with each collaboration model three times. Participants were not present for execution of the fully autonomous collaboration model.

Participants were instructed to palpate the tissue to locate the stiff inclusion, and to identify the boundary of this inclusion. Force data from the force sensor attached to the slave arm was used to provide force feedback to the user. This was done to help the user apply the desired amount of force, as too little force prevents an adequate stiffness measurement and too much force can cause tissue trauma. For each collaboration model, participants were instructed which aspects of the task they were responsible for completing. Participants verbally communicated when they were satisfied with the palpation data acquired for each round. Participants then proceeded to a computer monitor to select the boundary from amongst the palpated points or, in the case of traded control, to select a point representing the stiff inclusion.

Boundary Identification Experiment After completing the inclusion segmentation experiment, participants performed the boundary identification experiment using a computer monitor and mouse. Four different sets of points sampled from the virtual tissues by the autonomous palpation algorithm were displayed one at a time to the user. Participants were instructed to select points to define the boundary of the stiff inclusion in the same manner as the first experiment. The order in which users received the four virtual tissues was random.

4.2.2.6 Metrics and Data Analysis

For each trial of the inclusion segmentation experiment we recorded the procedure time, force exerted on the tissue, number of palpated points, and the chosen boundary points. Participants also answered a post-experiment survey, indicating which collaboration model out of the ones with human input they preferred and how they decided when they were finished exploring the tissue. For the boundary identification experiment, we recorded the chosen boundary points.

The accuracy of the identified boundary was assessed using the specificity and sensitivity metrics defined in Chapter 2. Specificity measures the proportion of the area of the stiff inclusion correctly identified as stiff (true positive). Sensitivity measures the proportion of the area of the soft tissue correctly identified as soft (true negative). Both metrics range from 0.0 to 1.0, where a value of 1.0 for both sensitivity and specificity indicates a perfect boundary identification. A higher sensitivity is more important than a high specificity, as a high sensitivity indicates that more of the stiff inclusion was correctly identified. Clinically, a high sensitivity results in fewer tumorous cells remaining behind. Thus the autonomous boundary identification algorithm is designed to achieve a high specificity at the expense of a reduced sensitivity.

Due to the non-normality of our data, all metrics were analyzed using either Wilcoxon rank sum tests or Kruskal-Wallis with Dunn post-hoc tests, depending on the number of groups present. Variances were compared with the Brown-Forsythe test since the distribution was neither normal nor symmetric. Human boundary identification was tested against inclusion shape for the boundary identification experiment using a Wilcoxon signed rank test.

4.2.3 Results and Discussion

4.2.3.1 Inclusion Segmentation Experiment

Figure 4.6 shows the results of the inclusion segmentation experiment. The duration of the procedure was significantly shorter under teleoperation compared to all other

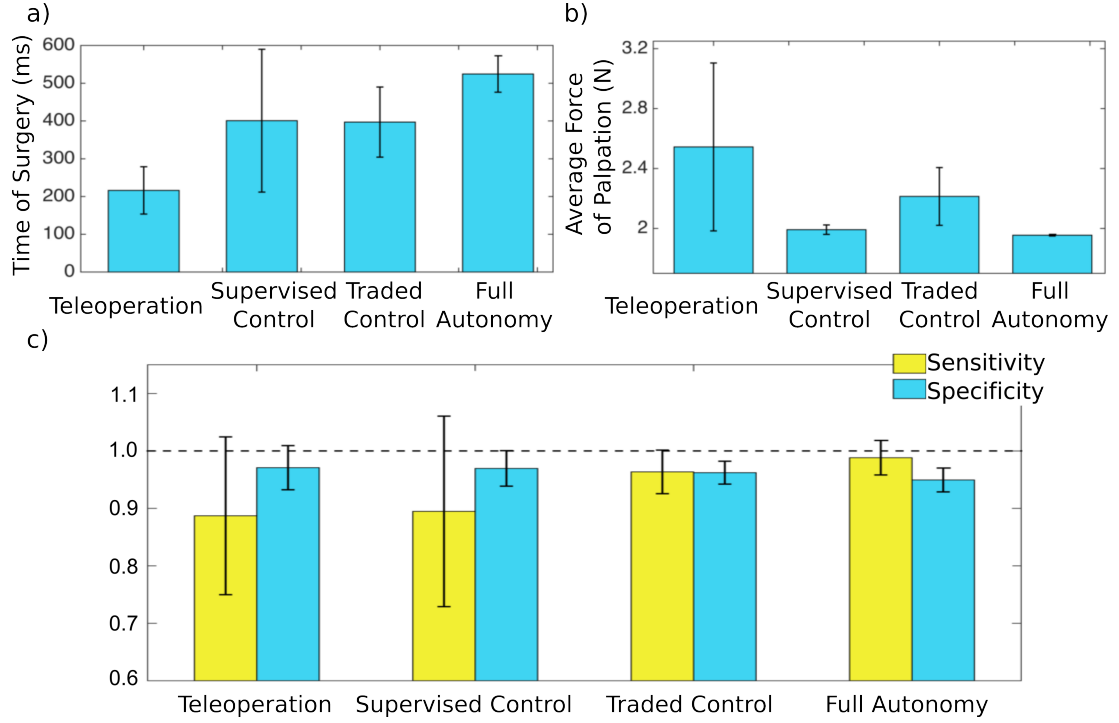


Figure 4.6: Inclusion segmentation experiment metrics under all collaboration models. (a) time of procedure, (b) average maximum force exerted on the tissue during palpation, and (c) sensitivity (dark grey) and specificity (white) of estimated inclusion boundary. (While the values of sensitivity and specificity cannot be greater than 1.0, the error bars representing ± 1 standard deviation are provided for visualization purposes even though they extend past the limit of one.)

collaboration models (Figure 4.6(a)). Supervised control was also faster than autonomous control ($p = 0.083$). Faster task times for complete human control relative to autonomous control are consistent with the literature [8]. The force exerted on the tissue varied between collaboration models, where the force for each trial was defined by taking the average across all palpations of the maximum force of each palpation (Figure 4.6(b)). Comparing the force exerted on the tissue between teleoperation, with the human directly responsible for performing the palpation, and supervised control, with the robot performing the palpation motion, we show that supervised control imparts less force ($p = 0.0009$). Trejos *et al.* [56] found a similar decrease in

force applied to the tissue for robotic versus human palpations. Supervised control also had significantly less variance in the applied force ($p = 3.47 \times 10^{-7}$). The results suggest that collaboration models which include both human and robotic agents can compromise between the extremes of teleoperation and autonomy. Shared control collaboration models can be significantly faster than that of autonomous control while limiting the large, variable forces present in teleoperation.

As shown in Figure 4.6(c), collaboration models with autonomous identification had a higher sensitivity and a lower specificity than collaboration models in which the human was responsible for selecting the boundary ($p = 0.053$ and $p = 0.022$ for sensitivity and specificity, respectively). Teleoperation and supervised control resulted in fewer palpated points compared to autonomous control. Figure 4.7 shows selected trials of boundaries identified for both experiments and across all collaboration models. Note the relative size and location of estimated boundaries to the true boundaries in Figure 4.7(a). Significantly less variance in sensitivity was also seen for collaboration models relying on autonomous boundary identification compared to human boundary identification ($p = 2.49 \times 10^{-9}$). As a high sensitivity is of key importance in this task, traded and autonomous control offer the best performance in this regard.

Only one subject had similar sensitivities and specificities compared to the fully autonomous collaboration model across all three trials. However, other subjects demonstrated repeatedly poor sensitivities, with two subjects demonstrating sensitivities of less than 0.65 across their combined teleoperation and supervised control collaboration model trials. Subjects who self-identified as having extensive experience in robotics (greater than one year) demonstrated less variance in sensitivity and specificity ($p < 0.05$) across the teleoperated and supervised control models but did not exhibit significantly higher mean sensitivity or specificity compared to subjects self-identified as having less than one year of experience with robotics. While some subjects were able to achieve performance similar to the autonomous agent, not all subjects achieved this level of performance and perhaps would have benefited from more or a different type of training. We would expect that more intensive training, possibly implemented by letting the subjects freely palpate a region while seeing a graphical overlay of the actual location of the stiff inclusion may do more to convey

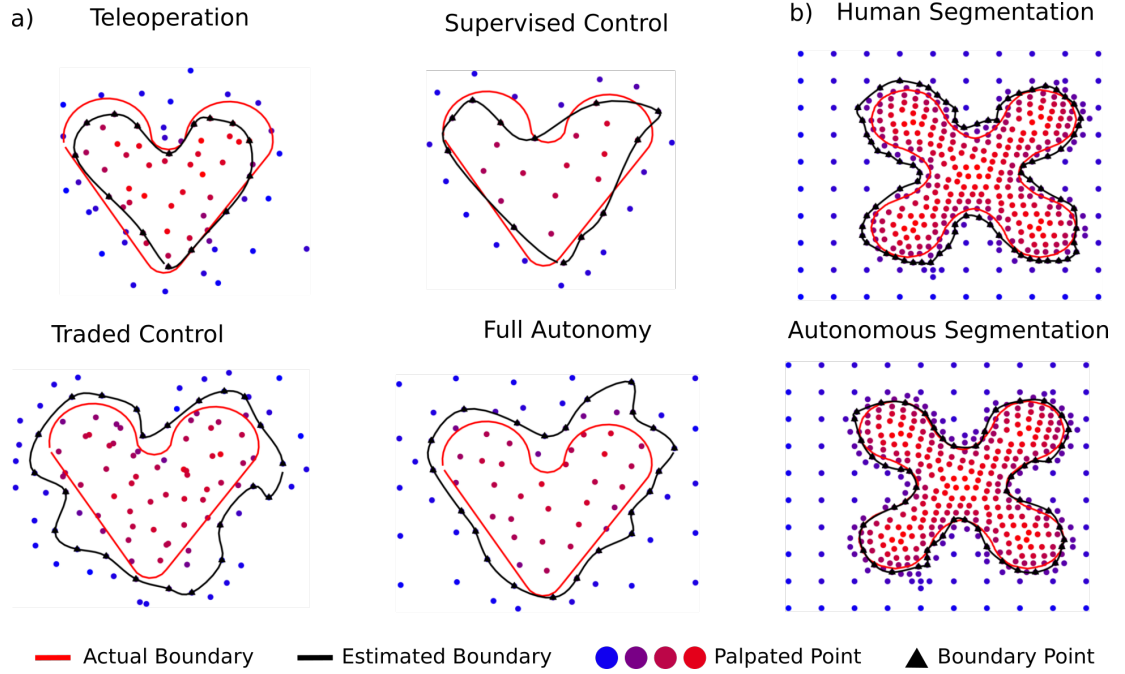


Figure 4.7: Examples of human and robot selected inclusion boundaries for the inclusion segmentation and boundary identification experiments, respectively. (a) Inclusion segmentation experiment. Human subjects chose palpation locations and the final boundary for teleoperation and supervised collaboration models. The robotic agent chose many or all palpation locations and the final boundary for traded control and autonomous collaboration models. (b) Boundary identification experiment. Boundaries identified by human (top) and robotic (bottom) agents for virtual tissues.

the difference in rendered stiffness between the hard inclusion and the surrounding soft tissue compared to subjects being verbally instructed where the boundary of the inclusion was located during training trials. It may also be the case that increased haptic feedback would assist in subjects identifying a more accurate boundary of the inclusion and reduce the maximum force exerted on the tissue, however the ideal amount of haptic feedback for this task is beyond the scope of this work.

Based on participant responses for collaboration models involving a human agent, teleoperation was desirable for the speed it offered as well as the high degree of

control available to the human operator. Six participants preferred traded control, citing the minimal effort on their part as an attractive feature of this collaboration model. Supervised control was considered to be easier by some, though two others felt robotic palpation was too slow and hindered their task performance. Several participants also indicated that one metric they used to determine when they had a sufficient number of points to segment the boundary was fatigue. This observation is reflected by the lower number of palpated points present for human-led palpation methods versus those led by a robotic agent.

4.2.3.2 Boundary Identification Experiment

Figure 4.8 shows the sensitivity and specificity results from the boundary identification experiment. The sensitivity of boundaries segmented by the robotic agent was higher and had less variation compared to the results obtained by human agents ($p = 0.0465$ and $p = 0.00043$ for median and variance, respectively). Human agent results also show higher variance for specificity ($p = 0.00013$). Specificity was not dependent on shape for human users. These results suggest that while specific participants such as the individual who identified the boundary in Figure 4.7(b) may perform well at this task, the overall variability in our population of participants limits the significant claims we can make about the performance of humans in our experiments.

4.2.4 Conclusions

In this work we assessed the performance of our previously defined human-robot collaboration models with the goal of determining which model resulted in superior task performance as defined by metrics of procedure time, force, and accuracy of boundary identification. The implementation of our autonomous boundary identification algorithm was improved to increase accuracy for non-convex shapes. Determining the best collaboration model depends on which performance metrics are deemed to be the most important. Teleoperation results in shorter procedure times. Collaboration models which use autonomous palpation minimize force impact on the tissue, and

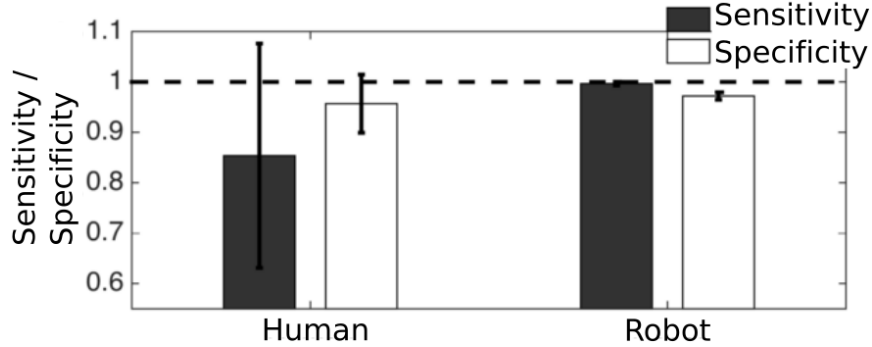


Figure 4.8: Sensitivity (dark grey) and specificity (white) for boundary identification experiment. Error bars indicate the standard deviation. (While the values of sensitivity and specificity cannot be greater than one, the error bars representing ± 1 standard deviation are provided for visualization purposes even though they extend past the limit of one.)

autonomous boundary identification has more consistent, higher specificity results. However, no collaboration model excels in all these metrics. The high variability present between human agents in these collaboration models is of particular importance, as this variation in applied force and sensitivity can have an unpredictable impact on a surgical task.

4.3 Extending the Multilateral Manipulation Software Framework to a Debridement Task

In this section, we examine multilateral manipulation in the task of simulated debridement: removing dead tissue or damaged tissue fragments to allow remaining healthy tissue to heal. We extended our previously developed multilateral manipulation software framework to the task of debridement and implemented four different collaboration models: (1) fully autonomous debridement, (2) shared control between a human and robotic agent, (3) supervised control where the operator identifies foreign bodies to be excised, and (4) teleoperation. We demonstrate these collaboration

models with the RAVEN-II robot, an open-architecture surgical robot with two cable-driven 7-DOF arms. Each collaboration model included methods to remove foreign bodies from the field of view of a stereoscopic camera setup. Results indicate tradeoffs in experiment duration, hardware requirements, and safety implications among the four collaboration models. The work in this section appears in a paper presented at the IEEE/RSJ International Conference on Intelligent Robots and Systems [84].

4.3.1 Introduction

4.3.1.1 Background: Robotic Debridement

Autonomous multilateral debridement has been investigated in earlier works [8, 85]. Kehoe *et al.* demonstrated autonomous multilateral debridement with the RAVEN-II surgical robot, hereafter abbreviated as RAVEN. In their work, a mounted camera detected foreign bodies in the surgical field while the RAVEN used both arms to remove the bodies [8]. In similar work, Murali *et al.* used a more clinically realistic debridement task conducted with a da Vinci surgical robot. Their surgical task included grasping foreign bodies and cutting them from the surrounding normal artificial tissue [85]. The robotic agent learned by observation from experts, and played back recorded trajectories in artificial tissue. In our study, we performed a debridement task involving identification, grasping, and cutting of foreign bodies.

4.3.1.2 Summary of Contribution

In this work we present a novel extension of our previously developed multilateral manipulation software framework from Chapter 3 as follows: (1) We extend the software framework to the new surgical task of debridement. (2) We implement four different collaboration models for the debridement task, including pure teleoperation, supervisory control, shared control, and full robot autonomy. (3) Using the RAVEN surgical robot, we demonstrate the flexibility of the framework to extend beyond one human input device and one robotic arm.

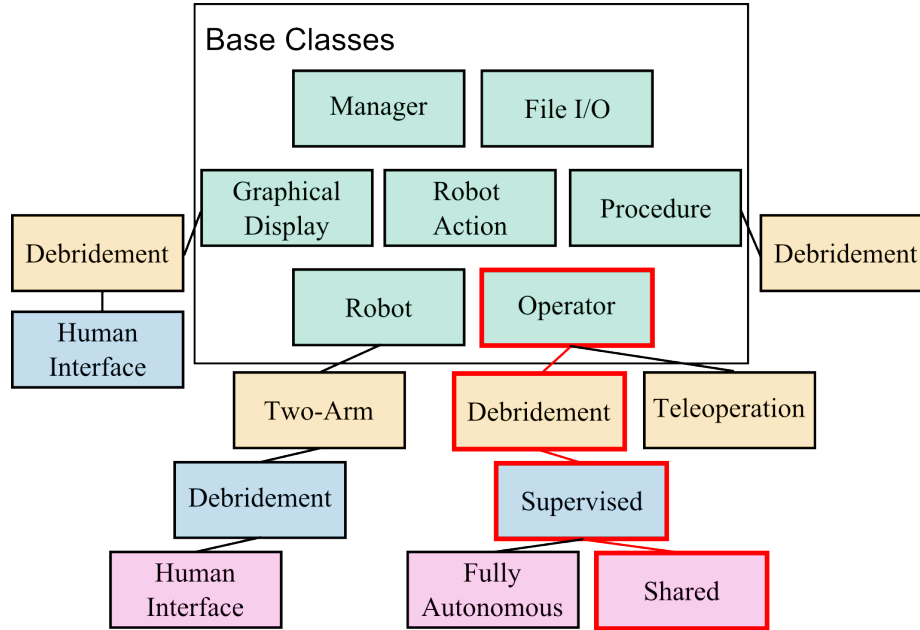


Figure 4.9: Class hierarchy diagram. All base-level classes (green) except the *Manager* are extended for the specific task (here, the debridement task). Four such extensions from base classes are pictured. Child class extensions from task extensions are in blue, further extensions of which are pictured in pink. Lines between classes indicate an extended class relationship. Class names are read from the bottom up i.e. Shared Debridement Operator (for this example the class names and extensions colored red).

4.3.2 Methods

In this section we describe the extension of the MMSF from Chapter 3 to the debridement task.

4.3.2.1 Software Framework

Base Class Structure Figure 4.9 illustrates the class structure and the extensions necessary for the framework to execute an debridement task. The classes used in this work are described in detail in Chapter 3. In Figure 4.9, base classes are green while the task-specific extensions are yellow. Further extensions of the task-specific classes, pictured in blue and pink, will be introduced in Section 4.3.2.2, when we describe the

collaboration models prototyped for a debridement task.

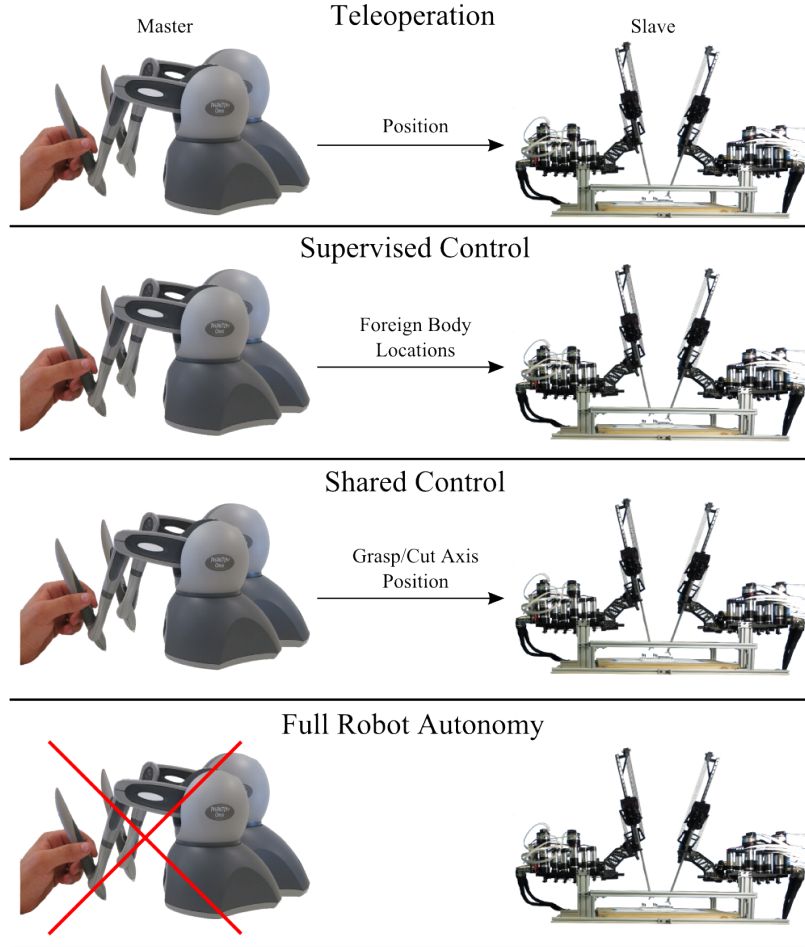


Figure 4.10: Four collaboration models tested. The different collaboration models are shown along with their information exchange between the master and slave robots.

4.3.2.2 Collaboration Models for Debridement Task

We use the software framework introduced in the previous section by extending its base classes to support the task of debridement. We implemented four different collaboration models. Each of these different collaboration models, pictured in Figure 4.10, differ in how they combine the intents of the human and robotic agents. Each collaboration model is tasked with identifying and removing multiple foreign bodies from

artificial tissue. While all collaboration models aim to accomplish the same task, different collaboration models emphasize performance, autonomy, or safety implications. The described collaboration models instantiate at least one extension of each base class picture in Figure 4.9. Table 4.3 lists the specific classes instantiated for each collaboration model.

Table 4.3: Class Instantiation Table

Collaboration Model	Operator	Slave Robot	Master Robot	Graphical Display
Teleoperation	Teleoperation	Two-Arm	Human Interface	Debridement
Supervised Control	Supervised	Debridement	N/A	Human Interface
Fully Autonomous	Autonomous	Debridement	N/A	Debridement
Shared Control	Shared	Debridement	Human Interface	Debridement

Each row lists the collaboration model with its instantiated extensions of the base classes (refer to Figure 4.9).

For the debridement task, the following new *Robot Actions* are created: *Grasp Robot Action*, *Ungrasp Robot Action*, *Cut Robot Action*, and *Move Under Guidance Robot Action*. While most of these actions' purposes are self-evident, the *Move Under Guidance Robot Action* is used exclusively in the supervisory control collaboration model and will be described in Section 4.3.2.2.

Three of the collaboration models extend the *Debridement Operator Class*. The *Debridement Operator Class* class contains the state machine seen in Figure 4.11, which encodes our debridement task. Each of the collaboration models extending this class follow the same state machine, but differ in how one or more of the states is handled, i.e. which *Robot Actions* are sent from the *Debridement Operator* to all *Robot* classes.

Supervised Control The supervised control collaboration model is the only class directly extending the *Debridement Operator*. In this collaboration model, the stereoscopic camera setup presents left and right images to the human operator at the beginning of the procedure. The user interacts with the graphical display by clicking on the locations of all foreign bodies in both images. The foreign bodies are then triangulated and registered in the robotic workspace, and removed autonomously. For supervised control, the human operator uses his or her judgment to interpret and

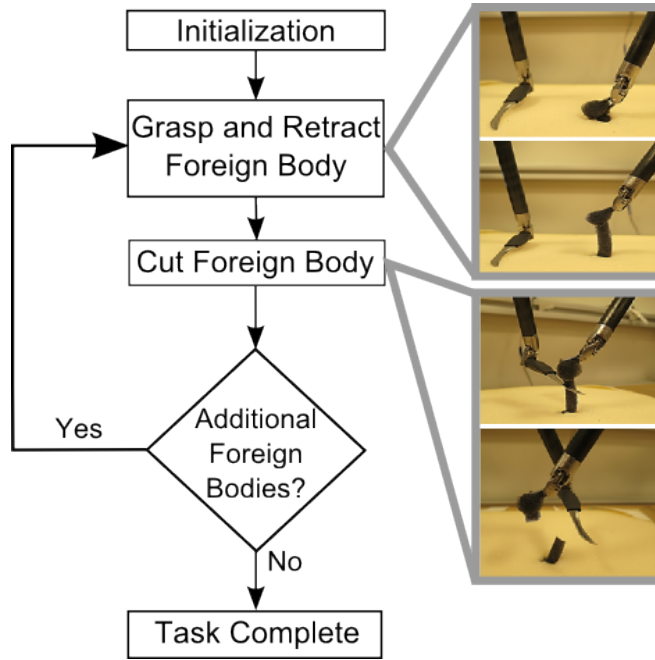


Figure 4.11: Debridement algorithm flow chart. Pictures on the right show the robot during execution of the Robot Actions associated with the states.

analyze the surgical field, and the robotic agent handles the surgical procedure.

Full Robot Autonomy The fully autonomous collaboration model is very similar to the supervised control collaboration model, hence its *Operator* is an extension of the *Supervised Debridement Operator*. The only difference between this collaboration model and the *Supervised Debridement Operator* is that in this collaboration model, all foreign bodies are automatically identified using the algorithms described in Section 4.3.3.1.

Shared Control The shared control collaboration model leverages many of the methods used in the supervised control collaboration model, with a few differences. The motivation for this collaboration model is to combine robotic precision and computer vision with teleoperated control for tissue manipulation actions to emphasize safety. In shared control, the inclusions are identified autonomously. After inclusion

identification, the robotic agent autonomously positions the grasping arm above the foreign body of interest. At this point, instead of the robotic agent grasping the inclusion autonomously, a *Move Under Guidance Robot Action* is issued by the *Operator* to both robots. With the new type of *Robot Action*, the slave robot remains positioned above the foreign body of interest. However, the degree of freedom perpendicular to the surface of the tissue is under teleoperated control. Teleoperated control also extends to the grasping degree of freedom. The operator can, under this collaboration model, move the robot to grasp the foreign body, grasp the body, and retract the body. Once the foreign body has been retracted above the tissue to a specified height, the *Operator* issues a *Hold Position Robot Action* to the grasping arm and issues a *Move Under Guidance Robot Action* to the cutting arm. The cutting arm has its teleoperated degree of freedom parallel to the surface of the tissue, such that the operator can cut the retracted tissue. Once the cutting arm has returned to the position from which the *Move Under Guidance Robot Action* was first issued, control of the cutting arm will return to the robotic agent. Then, the next foreign body will be excised.

Teleoperation In teleoperation, the master robots send position and orientation data to the slave robot. The human operator is free to explore the environment and remove any foreign bodies at will. There is no graphical display in this collaboration model. Table 4.3 shows that the Teleoperation collaboration model does not use a *Debridement Robot*, but the parent class *Two-Arm Robot*. Since teleoperation involves no decision-making on the part of the robotic agent, the more simple *Two-Arm Robot* class is instantiated. Additionally, because teleoperation does not require the state machine in Figure 4.11, the teleoperation collaboration model extends the *Operator* class directly instead of the *Debridement Operator*.

4.3.3 Experiment

The four different collaboration models developed in the previous section were tested using the experimental setup shown in Figure 4.12. Two Geomagic Touch devices (Geomagic, Morrisville, NC, USA) were used as the human input devices, while the

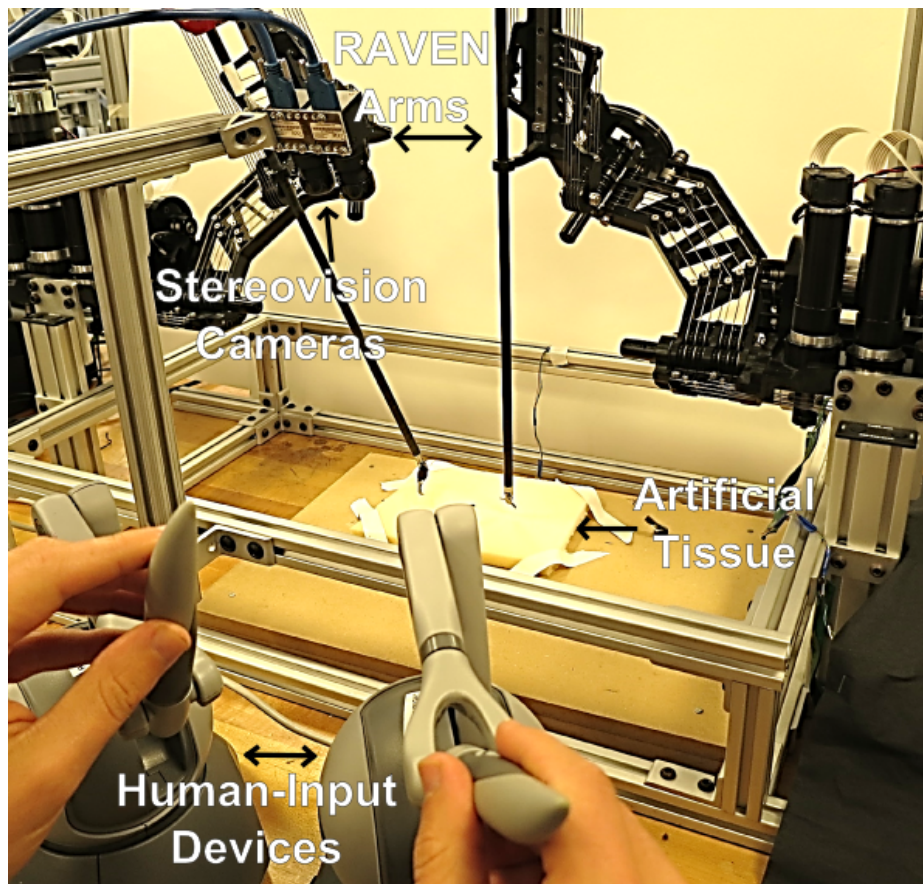


Figure 4.12: Experimental Setup. Two Geomagic Touch devices (bottom) control two RAVEN arms (center). A stereoscopic camera setup (top) captures and analyzes images of the artificial tissue (center).

slave robot, the RAVEN robot, was positioned to excise foreign bodies from the artificial tissue. A stereo camera setup positioned above the tissue analyzed captured images for foreign bodies. The experiment ended when the slave robot excised all foreign bodies. The metric gathered from these experiments is the amount of time from the onset of the experiment to the moment that all three foreign bodies have been removed. A single expert user tested all collaboration models prototyped.

4.3.3.1 Graphical User Interface

The graphical user interface captures and analyzes images to calculate positions of foreign bodies in the artificial tissue. Two Flea3 cameras (Point Grey, Richmond, BC, Canada) were used for stereo vision. Intrinsic and extrinsic camera parameters were calculated using the CalTech calibration toolbox [86]. Images captured were analyzed for foreign body locations using opencv [87]. Specifically, the opencv module takes the input images, converts it to intensity values on a greyscale and performs a Hough transform to search for circles in the image. With all foreign bodies identified in both the left and right camera images, the software module triangulates the positions of the bodies. A sample image of the artificial tissue analyzed for foreign bodies is shown in Figure 4.13a. The filled dark circles are the foreign bodies. Once the image was captured, a Hough transform would output detected circles, whose outlines in red and centers in green are overlaid on Figure 4.13a. The green center points, once transformed into the robot task space, parameterize the *Move Robot Actions* to position the RAVEN grasping arm above a foreign body of interest.

4.3.3.2 Artificial Tissue

The artificial tissue and foreign bodies were constructed from packaging foam (Figure 4.13). We chose foam material because foam can be easy to grasp and cut with the da Vinci large needle driver installed on the RAVEN. To enable cutting by the RAVEN, the end-effector of the left arm was outfitted with a sharp blade. The foreign bodies, constructed from black foam, were tethered to the base tissue with a thin strips of additional foam, so that the foreign bodies can be retracted (Figure 4.13b).

4.3.4 Results and Discussion

4.3.4.1 Robotic Debridement

The results for all four collaboration models excising three foreign bodies are seen in Table 4.4. In this work we used time as the metric, although in future work we intend to transition to a more robust experimental setup which in addition to being

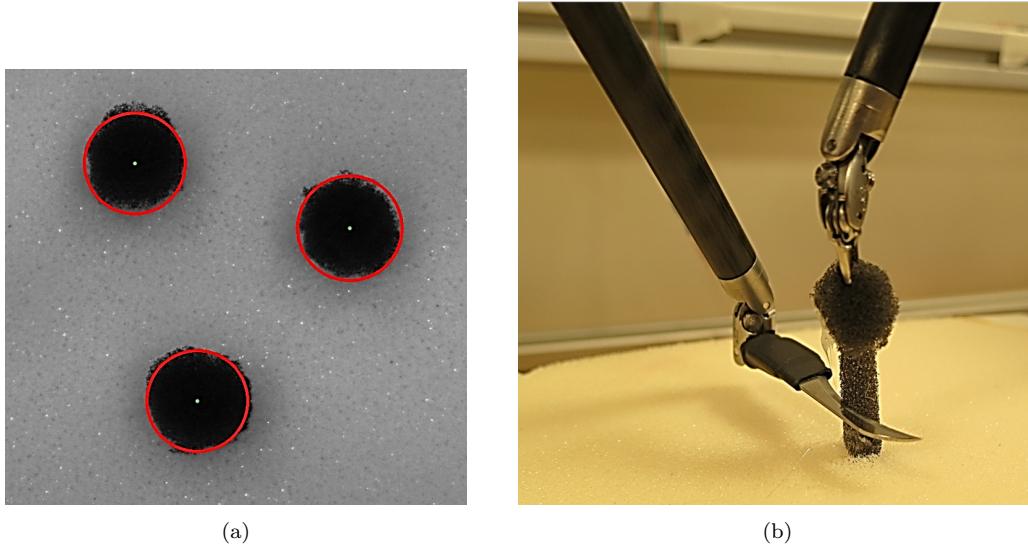


Figure 4.13: (a) Output from the computer vision algorithm. The green dots represent the estimated center of the foreign body while the red circle approximates the boundary. (b) The *Cut Robot Action*. The right grasper has retracted the detected foreign body while the left arm moves in a position to cut the tether from the body to the base tissue.

compatible with a wider array of tools, can also grasp smaller targets consistently. With this more robust setup we can remove inclusions embedded in the tissue instead of tethered, and compare sensitivity and specificity metrics of the removed foreign body. We could also compare tissue disturbance across collaboration models.

We first present the time pre-movement, which is the time spent capturing and analyzing images of the surgical field. The majority of the difference in execution time between the supervised control and full robot autonomy collaboration models is spent pre-movement. This is expected since for the supervised control collaboration model the user identifies the foreign bodies in both images before robotic movement.

Secondly, for collaboration models where the user teleoperated the grasping degree of freedom (shared control and teleoperation), the amount of time spent grasping was much greater than the time observed for autonomous grasping. This is because each grasp under teleoperated control took multiple attempts, as opposed to autonomous

Table 4.4: Experimental Results

Collaboration Model	Time Pre-Movement	Time Moving	Time Grasping	Total Time
Supervised Control	19.56	90.27	3.60	113.43
Full Robot Autonomy	1.24	86.54	3.60	91.38
Shared Control	1.27	98.18	8.04	107.49
Teleoperation	0.00	56.38	21.91	78.29

grasping. Grasping difficulty was increased when all degrees of freedom were controlled by the operator, as it took several attempts for the user to converge to the ideal grasping angle already encoded in the autonomous grasping action. Due to the multiple grasping attempts, collaboration models with *Grasp Robot Actions* under teleoperated control caused more tissue disturbance.

Finally, we note that the teleoperation collaboration model took less total time than any of the other collaboration models. However, for collaboration models which used *Move Robot Actions*, only one arm was moving at a time while the other arm was typically out of the surgical field. With greater emphasis on the path-planning algorithms used for *Move Robot Actions*, we believe that the experiment duration for autonomous or semi-autonomous collaboration models would significantly decrease.

4.3.4.2 Software Framework

Extending the framework to the debridement task using the RAVEN robot presented a significant challenge that may affect design choices in the future as we continue to extend the framework towards more complicated experimental setups and tasks. We were motivated to use the RAVEN robot because it is an open-source commercially available research platform. However, the RAVEN’s unilateral communication protocol is significantly different than the bilateral communication used with Phantom commercial devices. Bilateral communication was necessary for closed-loop control in our previous work. In the future, we may define communication protocols as part of the class structure, such that there is a *Two-arm open-loop Robot* and a *Two-arm closed-loop Robot*. Each of these robots would handle *Robot Actions* differently. Hardware-specific protocols could be isolated into a *Robot Hardware* class, included

as a member variable of the *Robot* class.

4.3.5 Conclusions

In this work we extended our previously developed multilateral manipulation setup to the task of debridement. We tested four collaboration models: Teleoperation, Shared Control, Supervised Control, and Full Robot Autonomy. Each of these collaboration models differed in how the inputs from the human and robotic agents are combined.

4.4 Extending the MMSF to Combined Stiff Region Exploration and Electrocauterization

In this section, we investigate multilateral manipulation in a surgical procedure consisting of two surgical subtasks: stiff region exploration and electrocautery along a boundary. The stiff region exploration subtask is a subset of the inclusion segmentation subtask presented in Chapter 3, and we use the same four collaboration models tested in Section 4.2 for the stiff region exploration subtask. For the electrocautery along the boundary subtask, four collaboration models were developed: (1) fully autonomous electrocautery along the boundary, (2) shared control between a human and robotic agent, (3) supervised control where the operator identifies the boundary to be electrocauterized, and (4) teleoperation. A pilot study was conducted in which an expert user was asked to complete the surgical procedure with different combinations of collaboration models between the two subtasks. Results indicate that use of a robotic agent for the electrocautery subtask can reduce task time, maximum force exerted on the tissue, and average force exerted on the tissue, while increasing specificity.

4.4.1 Introduction

One of the most difficult steps during robotic radical prostatectomy is dissecting the bladder from the prostate. Robotic radical prostatectomy is one of the most

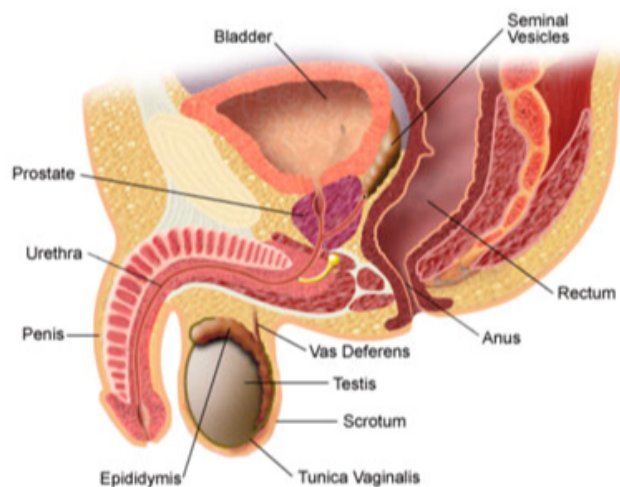


Figure 4.14: The male reproductive organ. Drawing by CFCF, distributed by a CC-ASA 3.0 unported license.

prominent forms of RMIS, and use of RMIS compared to open radical prostatectomy has been shown to reduce blood loss, hospital stay, and trauma to other neighboring organs [4]. Figure 4.14 shows the male reproductive anatomy from the lateral perspective, with the bladder adjacent to the prostate. In fact, the bladder and prostate are fused together by multiple layers of bladder muscle in addition to fatty tissue. The connection also includes urinary sphincter muscles that are responsible for urinary continence [88]. The typical procedure for separating the bladder from the prostate is to dissect into the bladder neck, exposing a Foley catheter inserted into the urethra. Surgeons grab and position the Foley catheter at the top of the surgical field, leaving the prostate at the bottom of the surgical field below the bladder, which is behind and above the catheter [89]. The boundary between the two organs must be cauterized during radical prostatectomy so that the organs can be separated. However, the boundary is not easily visible, and often the most prominent discriminating factor between the two organs are their stiffnesses, as determined via palpation [90]. The

lack of mapping mechanical properties of the two organs during RMIS has been cited as one of the strongest critiques of robotic prostatectomy in favor of open prostatectomy [90]. We seek in this section to simulate electrocauterization of the boundary between the prostate and the bladder by extending the MMSF to a procedure consisting of two subtasks: stiff region exploration and electrocauterization along the boundary. This would be the first instance of the MMSF being extended to a surgical procedure consisting of multiple subtasks, and in this section we seek to demonstrate results exploring how different pairs of collaboration models for the two subtasks vary performance in our procedure simulation.

In this section we present an extension of the MMSF first introduced in Chapter 3. This extension is novel in the following ways: (1) We implement an electrocautery task with the MMSF. (2) We implement four different collaboration models for the electrocautery along a boundary subtask: bilateral teleoperation, shared control, supervised control, and full robot autonomy. (3) We demonstrate a surgical procedure consisting of two different surgical subtasks, stiff region exploration and electrocauterization along a boundary. We test pairs of collaboration models, providing a first investigation of varying control authority between a human and robot agent dependent on surgical subtask.

4.4.2 Extension of the MMSF to the Electrocautery task

The electrocautery subtask as defined in this study takes a set of palpated points with their associated stiffnesses output from the stiff region exploration task and cauterizes along a boundary either determined autonomously or by the human operator as prescribed by the implemented collaboration model. This is an important departure from the inclusion segmentation task described in Chapter 3, where the boundary of the segmented inclusion is output from the surgical subtask. Instead, the first subtask is responsible for tissue exploration only. Each of the collaboration models for the electrocautery subtask, shown in Figure 4.15, vary both in how they identify the boundary of the stiff region and how the boundary of the region is cauterized. The

four different collaboration models used for the boundary identification and electrocautery subtask are bilateral teleoperated control, shared control, supervised control, and full robot autonomy. Two base classes, *FileIO* and *Graphical Display*, are not implemented as part of this subtask. *FileIO* is not extended for this subtask because there is no data unique to the electrocautery subtask which requires recording, while the *Graphical Display* class is omitted from the implementation of this task for reasons to be given in Section 4.4.3.4.

To implement the electrocautery subtask, one new *Robot Action* was created, the *Electrocauterize Shared Control Robot Action*, described in detail in the following paragraph. All four collaboration models instantiate the same *Electrocautery Procedure*, *Electrocautery Robot*, and *Human-Interface Electrocautery Robot* classes. The collaboration models only differ by which extension of the *Electrocautery Operator* is instantiated (Figure 4.16(b)).

4.4.2.1 Shared Control

In shared control, boundary points for the stiff region are never explicitly identified by the human agent during execution of the stiff region exploration and electrocautery task (see Figure 4.15). Instead, the user first indicates where he or she would like the cauterization to begin by squeezing the master-side gripper of the dVRK. At this point, a *Move Robot Action* positions the slave-side robot to the desired location on the surface of the tissue and at a pre-specified depth into the tissue. Following completion of this action, an *Electrocauterize Shared Control Robot Action* is issued to the slave robot. The depth of the electrocautery tool is held constant by the robotic agent while the user teleoperates the tool across the surface of the tissue. This division of labor between the robot and human agent is added as a safety measure, such that the human cannot cauterize to a dangerous depth into the tissue while teleoperating. The experiment concludes when the user indicates that he or she has finished cauterizing along the boundary.

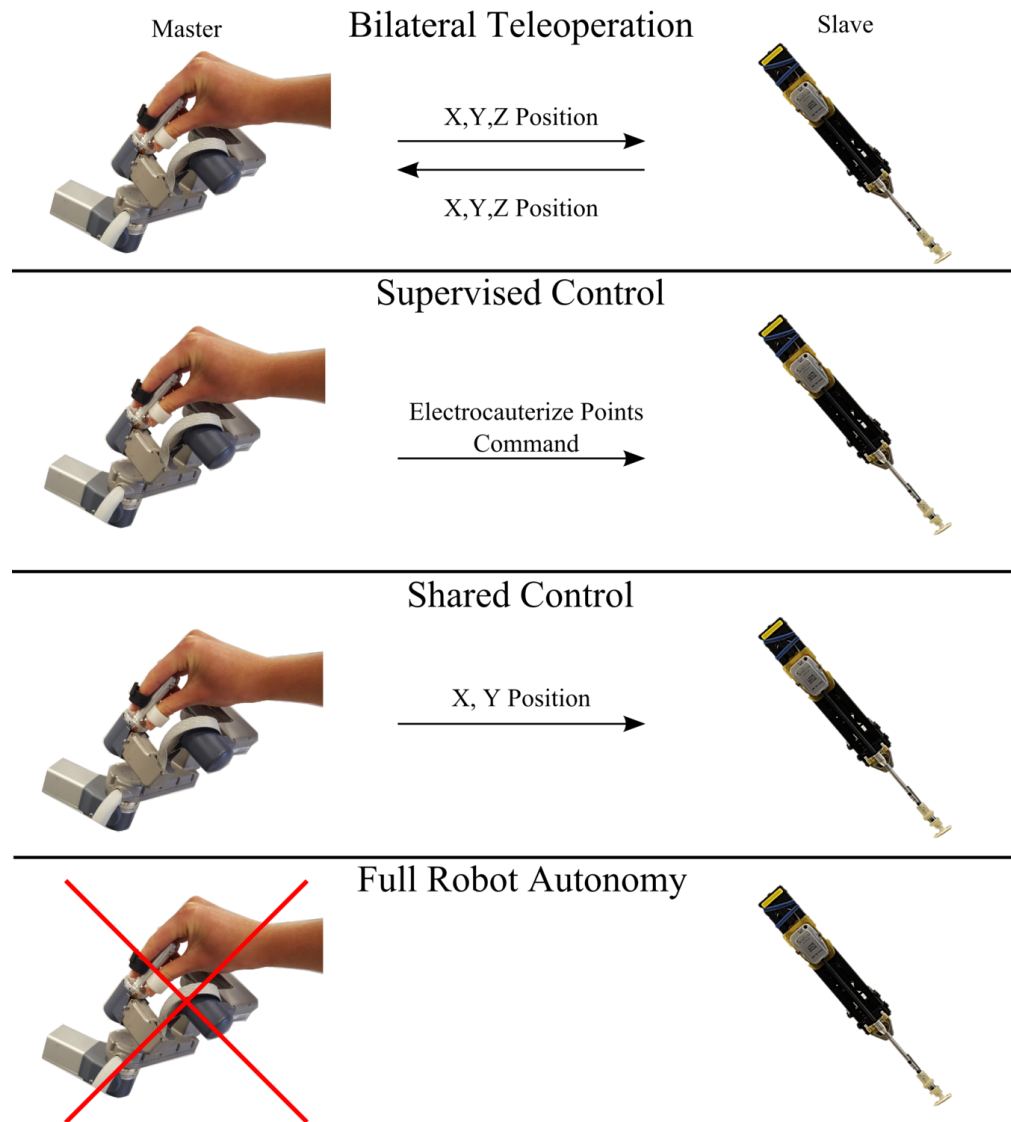


Figure 4.15: Collaboration models tested. Four different collaboration models are shown along with their information exchange between the master and slave robots.

4.4.2.2 Bilateral Teleoperated Control

In bilateral teleoperated control, boundary points for the stiff region are not indicated by the user, as in shared control. Following the stiff region exploration subtask, the user directly begins bilaterally teleoperating the robot to cauterize the tissue. The subtask is started when the *Teleoperation Electrocautery Operator* issues a *Teleoperate Robot Action* command to the robot. Once the user indicates that he or she has completed cauterization of the boundary of the stiff region, the experiment ends. The *Teleoperation Electrocautery Operator* extends the *Shared Electrocautery Operator*, in order to issue a *Teleoperate Robot Action* instead of an *Electrocauterize Shared Control Robot Action*. The remainder of the procedure is the same between the two collaboration models.

4.4.2.3 Supervised Control

In supervised control, the human agent indicates the boundary of the stiff region while the robot cauterizes along the user-specified boundary. Using the graphical environment displayed visually in the stereoscopic display from Section 4.2, the user sees a virtual proxy of the master manipulator alongside the palpated points and their representative stiffnesses. The user indicates points along the boundary by moving the virtual master manipulator proxy in the graphical environment to a desired position and squeezing the master manipulator grippers. Once the user indicates that they have completed their boundary point identification procedure, Bezier splines are used to interpolate among the set of points, and an equally spaced distribution of 10 points along the interpolated boundary parameterize a set of *Move Robot Actions* sent to the slave robot. The depth of the *Move Robot Action* is held constant by the same robotic agent as in shared control.

4.4.2.4 Fully Autonomous Control

In fully autonomous control, the boundary points are identified using the procedure in Chapter 2. Following the same procedure as Section 4.4.2.3, the boundary points are interpolated between using Bezier splines, and an evenly-distributed set of points

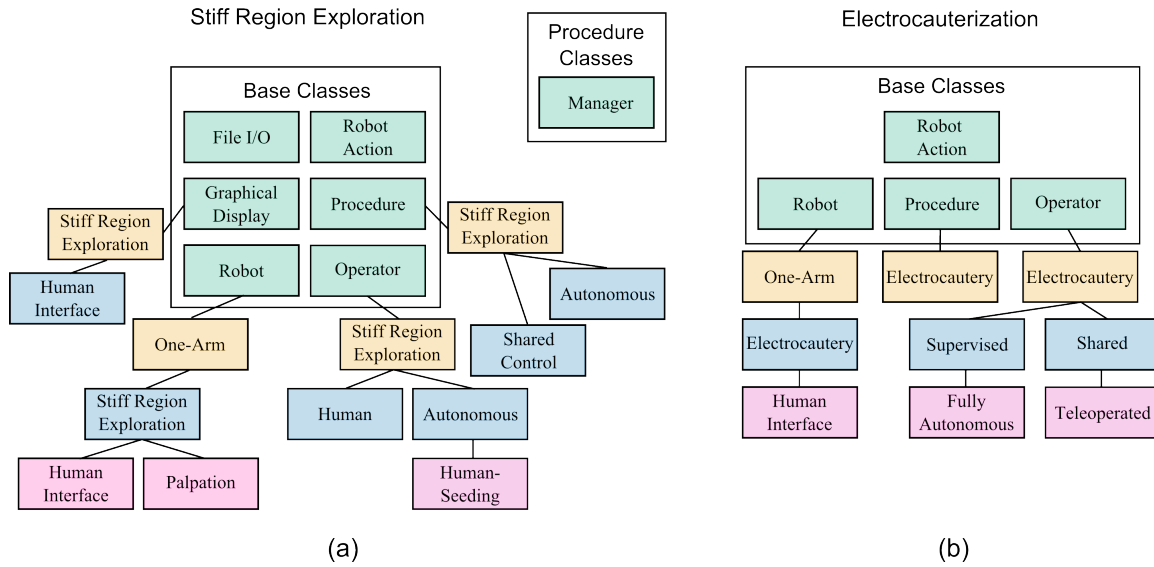


Figure 4.16: The class hierarchy diagrams for the stiff region exploration (a) and electrocauterization subtasks (b). All base-level classes (green) are extended for the specific subtask. Child class extensions from subtask extensions are in blue, with the exception of the *Robot* class. Further extensions are pictured in pink. Lines between classes indicate an extended class relationship. The *Manager* class is at the procedure context, not limited to the implementation of a single subtask, and is therefore pictured outside of the hierarchy trees.

parameterizes a set of *Move Robot Actions* subject to the depth constraint from Section 4.4.2.1. The *Fully Autonomous Electrocautery Operator* extends the *Supervised Electrocautery Operator* by redefining the method by which boundary points are specified. However, once the boundary points are identified, the two collaboration models behave similarly.

4.4.3 Experiment

An artificial tissue with an embedded stiff region was placed beneath a dVRK slave arm augmented with an attached Nano-17 force-torque sensor (ATI Industrial Automation, Apex, NC, USA) and configurable tool tip which was switched between a palpation end-effector and electrocauterization tool during the procedure. A human

user sits at the dVRK master console and manipulates the right arm of the master manipulators. Through the stereoscopic display, users visualize both the dVRK slave arm and the stiffness of the palpated points with the same graphical display developed in Section 4.2 (see Figure 4.17). Users were asked to identify and cauterize along the boundary of a stiff region. For the stiff region exploration subtask, users explored the tissues using one of the collaboration models described in Section 4.2. Once tissue exploration is completed, either by user indication, for supervised or tele-operated control, or by the autonomous palpation algorithm running to completion in the other cases, the procedure would transition to the electrocautery subtask. This differs from the inclusion segmentation subtask developed in Chapter 3, where the human or robotic agent was explicitly responsible for identifying the boundary of the stiff inclusion after exploration. In this work, once the stiff region exploration subtask has completed, the tool tip was changed from the palpation attachment to the electrocautery tool (see Figure 4.17). The surgical procedure then continues under one of the collaboration models described in Section 4.4.2. The performance of collaboration models are compared against each other with the following metrics: sensitivity, specificity, average force exerted by the electrocautery tool on the tissue, maximum force exerted by the electrocautery tool on the tissue, and time of procedure.

4.4.3.1 Artificial Tissue

Artificial tissues were created from gelatin. We used gelatin because it can be constructed to exhibit a sufficient range of stiffness to mimic hard inclusions in surrounding soft tissue, and it melts when exposed to temperatures of greater than 150° F, facilitating implementation of our electrocautery subtask. Different concentrations of gelatin powder mixed with water yield different stiffnesses, and in this respect we were able to create a stiff region (9.0 g gelatin/8 oz water) and surrounding soft tissue (5.6 g gelatin/8 oz water). Stiff oval inclusions with principal axis lengths of sizes 18 mm and 26 mm made from highly concentrated gelatin powder were placed into gelatin of a lower concentration of size 6×6 inches during the surrounding gelatin's curing process.

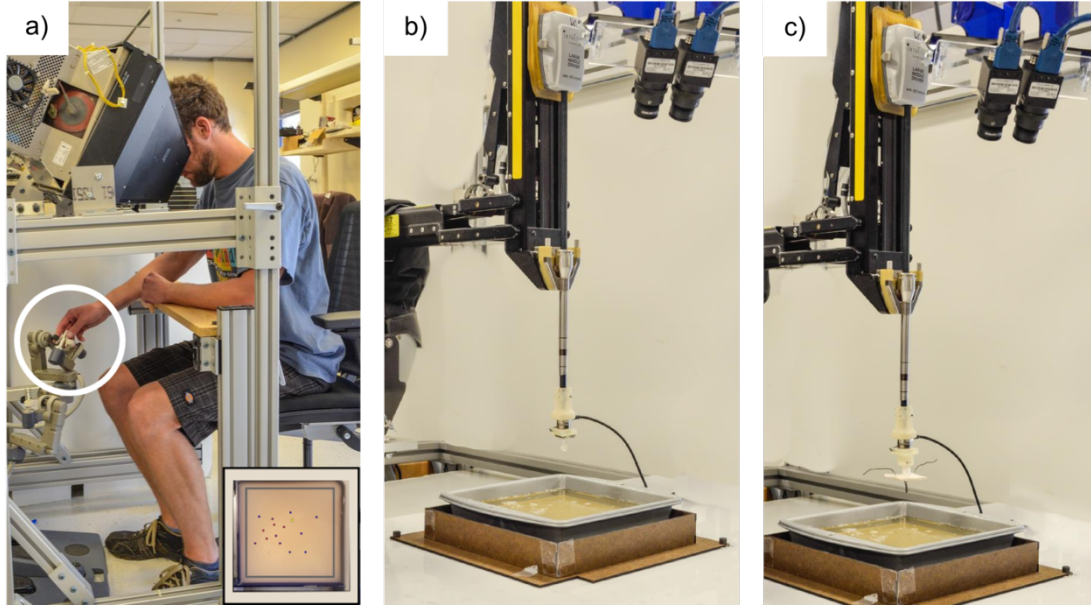


Figure 4.17: Experimental setup for the stiff region exploration and electrocauterization along the boundary subtask. A human user sits at the master console of the dVRK (a) and manipulates a single master manipulator (white circle). A visualization rendered through the stereoscopic display shows the slave manipulator arm and a top-down view of the sample tissue, with the stiffness of palpated points being represented by colored circles across the surface of the tissue. Red points represent more stiff points, while blue points less stiff points. (b) The slave-side manipulator was outfitted with an attachment that contained a force-torque sensor and a palpation probe for the stiff region exploration subtask. (c) The slave-side manipulator outfitted with an electrocautery tool for the electrocauterization subtask.

4.4.3.2 Electrocautery Tool

High current driven through nichrome resistive wire electrocauterized the gelatin phantom. The wire was fixed in place with a ceramic fixture, to prevent damaging other components in the tool. The ceramic part was held in place with a 3D-printed part which attached the electrocautery tool to the force-torque sensor.

4.4.3.3 Stiff Region Exploration

The first subtask, stiff region exploration, is the same as developed in Section 4.2 with one minor adjustment. Previously, collaboration models for the inclusion segmentation task developed in Section 4.2 concluded when the autonomous or human agent identified the boundary of the inclusion by selecting a set of palpated points. Here, the stiff region exploration subtask completes when the human agent indicates that they are comfortable identifying the boundary of the inclusion, not after they have already completed identifying the boundary. For inclusion segmentation collaboration models that generate a boundary autonomously, the final segmented boundary is no longer generated by the stiff region exploration subtask.

4.4.3.4 Graphical Interface

The graphical interface is the same as that in Section 4.2 for both the computer running the MMSF and the dVRK computer. The graphical display created by the MMSF is still used to seed the stiffness classifier for the stiff region exploration task implementing the traded control collaboration model. The dVRK graphical display shows the palpated point data in real time. The cauterization task requires the user at the dVRK master console to see the palpated data gathered during the first part of the procedure but does not require any new data to be displayed by the MMSF. For this reason the implementation of the electrocautery subtask does not include any instantiations of a *Graphical Display* class.

4.4.3.5 Collaboration Pairs Tested

We define a collaboration pair to be the combination of (1) a collaboration model used for the stiff region exploration subtask and (2) a collaboration model used for the electrocautery subtask. These collaboration models need not be the same for both subtasks, and the purpose of this study is to take advantage of being able to switch between collaboration models between subtasks. Figure 4.18 shows the tested collaboration pairs. The collaboration models available for the stiff region exploration subtask are shown horizontally, while the collaboration models available for the electrocautery subtask are shown vertically. Two pairings are not realizable in our definition of the surgical procedure: supervised or teleoperated control for stiff region segmentation followed by fully autonomous control for the electrocautery subtask. These pairings cannot be realized because they rely on human exploration for the stiff region exploration task, and do not provide a densely-populated sampling of points along the boundary of the stiff region. A dense sampling of points along the boundary of the stiff region is necessary for the fully autonomous collaboration model for the electrocautery subtask to generate a boundary along which to cauterize.

4.4.3.6 Evaluation Metrics

To compare the performance of different collaboration pairs against each other, we primarily rely on the sensitivity and specificity metrics used in Chapter 2. In this study we compare the actual boundary of the inclusion against the cauterization path of the robot, whether under human or autonomous control. Sensitivity and specificity metrics comparing the actual boundary of the inclusion against an identified (but not cauterized) boundary of the inclusion are explained in Section 4.2. We also record the time of the electrocauterization procedure, and the average and maximum force exerted on the tissue.

4.4.4 Results and Discussion

The sensitivity and specificity results for the stiff region exploration and electrocauterization along the boundary subtasks for all collaboration pairs are summarized in

		Stiff Region Exploration			
		Bilateral Teleoperation	Supervised Control	Traded Control	Full Robot Autonomy
Electrocautery	Bilateral Teleoperation	0.995 / 0.738	1.000 / 0.750	1.000 / 0.740	1.000 / 0.686
	Supervised Control	1.000 / 0.795	1.000 / 0.770	1.000 / 0.830	0.769 / 0.556
	Shared Control	1.000 / 0.738	1.000 / 0.709	1.000 / 0.824	1.000 / 0.787
	Full Robot Autonomy	✗	✗	1.000 / 0.845	1.000 / 0.879

Figure 4.18: Sensitivity and specificity results for the surgical procedure. Across columns are the collaboration models for the stiff region exploration subtask, while down rows are the collaboration models tested for the electrocauterization along the boundary subtask. For each pairing of collaboration models, the sensitivity is in the upper-left hand corner while the specificity in the lower-right hand corner. The two pairs of collaboration models which cannot be implemented for this procedure have the red crosses.

Figure 4.18. The time to complete the electrocautery subtask as well as the maximum and average force exerted on the tissue for the electrocauterization collaboration models averaged across all stiff region exploration collaboration models are shown in Table 4.5. Since these results are from a pilot study with one expert subject, the results indicate trends among the pairings of collaboration models and not statistically significant differences among pairings of collaboration models.

The results in Figure 4.18 show that while most collaboration model pairs resulted in a sensitivity of 1.0, collaboration pairs which used the autonomous agent had a specificity that was nearly 0.1 higher than any other electrocautery collaboration model. Thus for this individual human user, all collaboration models that relied on the human to control the shape of the electrocauterized boundary resulted in a

cauterized region that included 10% more healthy tissue than the autonomous agent.

Additionally, the autonomous agent spent less time electrocauterizing the tissue compared to the human (see Table 4.5). The autonomous agent took half the time cauterizing the tissue compared to the human under teleoperation, and 15 seconds less compared to the shared collaboration model. In the supervised collaboration model, the human would identify the boundary of the stiff region while the autonomous agent would cauterize along the boundary. The time metric reported includes both the boundary identification phase and the electrocautery phase. However, since the electrocautery phase for supervised control uses the same method as fully autonomous control, the actual time that the supervised control collaboration model spent cauterizing the tissue is the same as the amount of time the autonomous agent spent electrocauterizing, around 24 seconds.

Another result shown in Table 4.5 is that the teleoperation collaboration model exerted a higher maximum force and an higher average force on the tissue, which the autonomous agents used in the other collaboration models were designed to limit by controlling the position of the end effector perpendicular to the tissue surface.

4.4.5 Conclusion

In this section we extended the MMSF to a surgical procedure consisting of multiple subtasks: stiff region exploration and electrocauterization along a boundary. Initial results from the electrocautery task suggest that collaboration with an autonomous agent can reduce task time, force exerted on the tissue, and increase specificity. To make claims comparing different collaboration models would require more extensive testing via a formal user study.

Table 4.5: Time of procedure and force exerted on the tissue metrics gathered for the electrocauterization along the boundary collaboration models.

Collaboration Model	Time of Procedure (s)	Maximum Force (N)	Average Force (N)
Fully Autonomous Control	24.23	0.161	0.025
Shared Control	74.46	0.321	0.018
Supervised Control	40.85	0.424	0.141
Teleoperative Control	50.41	2.15	0.662

The results above are the average of the electrocautery collaboration models across all stiff region exploration collaboration models. For example, the results shown for the fully autonomous control collaboration model are the average of the collaboration pairs of: fully autonomous stiff region exploration with fully autonomous electrocauterization along the boundary and traded control stiff region exploration with fully autonomous electrocauterization along the boundary.

Chapter 5

Conclusion

5.1 Synopsis of Contributions

In this dissertation we developed the Multilateral Manipulation Software Framework, which is capable of enabling rapid prototyping and deployment of varied human-robot collaboration models. The MMSF was extended to support three different mock surgical subtasks, and implemented on three different hardware systems. To conclude my dissertation, here we summarize the main contributions of this work.

In Chapter 2, we developed the first autonomous agent in literature to both explore tissue mechanics autonomously and segment hard inclusions from surrounding soft tissue in the robot’s own coordinate system. The autonomous agent exercised two different capabilities: tissue exploration and classifying palpated points as part of the hard inclusion or part of the surrounding soft tissue. The autonomous palpation algorithm explored the tissue in an intelligent way by exploring neighborhoods around points theorized to belong to the hard inclusion, instead of generating a high-density grid of palpated points across the entire surface of the tissue. Classification of the points was facilitated by an expectation-maximization scheme, with Markov random fields constituting the E-step, and Gaussian Discriminant Analysis constituting the M-step. The use of Markov random fields enabled our classification algorithm to use stiffness information from the clique around a palpated point to influence its classification, such that points sufficiently close to each other were more likely to be

similarly classified. Use of the expectation-maximization algorithm helped generate a sample-specific final classifier for stiffness discrimination. Over three trials segmenting a rectangular stiff inclusion from its surrounding soft tissue, sensitivity was above 0.95 and specificity was above 0.92.

The expectation-maximization machine learning algorithms require an initializing classifier. In the first experiment in Chapter 2 we used elastography to generate the initializing classifier. To prove the robustness of our inclusion segmentation algorithm, we tested the performance of the inclusion segmentation algorithm subject to a wide range of initializing classifier values. For all values of the initializing classifier, the sensitivity and specificity were greater than or equal to 0.875.

With an autonomous agent in hand, we developed the Multilateral Manipulation Framework (MMSF), described in Chapter 3 and extended in Chapter 4. The MMSF was created as a developer’s tool to enable rapid prototyping and deployment of varied human-robot collaboration models. The MMSF decomposes a human-robot collaborative task into seven base classes, six of which can be extended to implement a particular surgical subtask, while the seventh, the *Manager* class, is procedure-wide and is only instantiated once, regardless of how many surgical subtasks are implemented. The implementation details of the MMSF were communicated through UML diagrams, and an in-depth explanation of the inter-thread communication protocol established the integrity of data communicated between the base classes of the MMSF. In Section 4.4, we described an implementation of the MMSF to a surgical task consisting of multiple subtasks.

The MMSF was extended to support five different human-robot collaboration models in an inclusion segmentation task. The autonomous agent developed in Chapter 2 established our notion of fully autonomous control, while this same robotic agent was combined with human agent input to form human-robot collaboration models of supervised control, shared control, and traded control. We also tested a bilateral teleoperation collaboration model, in which the human was in full control of the slave robot. Results from a pilot study suggested that the autonomous agent was better at segmenting a hard inclusion from soft tissue with a higher sensitivity compared a

human agent, while collaboration models which relied on human agent input completed the task faster and resulted in fewer palpations. Additional results suggested that methods which relied on the autonomous agent to limit palpation force on the tissue did so more effectively than collaboration models where the palpating degree of freedom was under complete human control. These results were expanded upon in a user study, where twelve subjects were tested in the inclusion segmentation task (Section 4.2). Results showed that for traded control and fully autonomous control, where the robotic agent at least in part explored and segmented the inclusion, the sensitivity was higher compared to supervised control and bilateral teleoperation, where exploration and segmentation was the responsibility of the human agent ($p = 0.53$). Additionally, the variance of the sensitivity under robotic boundary segmentation was significantly less than human segmentation ($p = 2.49 * 10^{-9}$). The final result from this study is that the force exerted on the tissue was significantly less for collaboration models which used the robotic agent for force-limited palpations compared to when the palpating degree of freedom was under complete human control ($p = 0.009$).

In Section 4.1 we described the implementation steps necessary to implement the MMSF on the RAVEN-II and dVRK robots, which run codebases ROS and CISST-SAW respectively. These implementations, used throughout Chapter 4, demonstrate the platform-agnostic feature of the MMSF, which underscores the MMSF's relevance to a wide variety of hardware platforms.

The final part of the work we presented as part of this dissertation, in Sections 4.3 and 4.4, extended the MMSF beyond the sole task of inclusion segmentation implemented exclusively on Phantom Premium devices. In Section 4.3, we extended the MMSF to a debridement task and implemented this task on the RAVEN-II Surgical Robot. Four different collaboration models are developed and tested, including teleoperation, supervised control, shared control, and full robot autonomy. Computer vision techniques informed the autonomous agent used in this study. Results showed that methods which relied on autonomous grasping of debridement targets resulted in less time interacting with the tissue than methods relying on human grasping. In Section 4.4 the MMSF was extended to support an electrocauterization task. The collaboration models we implemented for the electrocautery subtask were teleoperation,

shared control, supervised control, and full robot autonomy. As part of extending the MMSF to the electrocauterization task, the experimental procedure used this subtask was combined with the stiff region exploration task, a subset of the inclusion segmentation task from Chapter 3. Section 4.4 compared the performance of collaboration model pairings between the two subtasks and showed the autonomous agent was more successful in cauterizing a region with higher specificity while exerting lower forces on the tissue compared to the human agent.

5.2 Future Work

The inclusion segmentation algorithm introduced in this work could be extended in a variety of ways. To expand the algorithm to enable detection of multiple inclusions, a k-means algorithm could be implemented between the EM and the boundary point steps of the tumor segmentation algorithm to estimate how many different clusters of points identified as hard inclusions are in the image. Each cluster’s boundary points could then separately identified. When using the inclusion segmentation algorithm to segment smaller inclusions than the one in this work, it is necessary to change the parameterization of the clique potentials for smaller cliques. Additionally, neighborhood searches around stiff points would need to be smaller. However, the methods introduced in this work remain unchanged. Experimentally, the size of the force-torque sensor as well as precise control of the robot can create a lower bound for inclusion size. To adapt this work to a more realistic clinical setting, organ movement must be accounted for. Brock performed a comparative study of several approaches towards deformable organ registration [91]. While outside the scope of this dissertation, our presented inclusion segmentation algorithm could be used alongside such organ movement models. An additional requirement to implement the autonomous inclusion segmentation algorithm *in vivo* would be to use a smaller, cheaper, potentially disposable palpation probe, which is currently an area of research with a group of our collaborators [92].

The autonomous agent developed for inclusion segmentation demonstrated longer

task execution times compared to the other collaboration models in all our experiments. However, improvements could be made to lower the duration of the inclusion segmentation task under autonomy in a number of ways. Firstly, the *Palpate Point Autonomously Robot Action* command was constructed as a series of three *Move Robot Actions*, initially to position the end effector above the point of interest, and the downwards and upwards motion of the palpation. These *Move Robot Actions* were completed in two seconds each, making an autonomous palpation complete in six seconds in the longest case, although due to force thresholding the palpation duration was between four and six seconds. Regardless, four to six seconds is longer than the time that it took subjects to teleoperatively palpate a point, motivating the need to more quickly palpate autonomously. If we were to palpate faster, our assumption of a linear spring dynamical model for tissue mechanics under palpation would become increasingly invalid as the viscosity component of tissue viscoelasticity will become a stronger factor in the force data gathered from the force-torque sensor. Methods for generating robust tissue mechanics models from *in vivo* data which we could use in this case have been developed by Ottensmeyer [93, 94]. Another method for reducing the amount of time the fully autonomous robotic agent requires to segment the boundary of the inclusion is to reduce the number of palpations required before the robot is confident in its segmentation. While the fully autonomous agent presented in this work used a variety of machine learning methods for classification of palpated points, different machine learning methods could be used to drive the exploration phase of the inclusion segmentation algorithm. This new exploration methodology would be in contrast to the method presented in this work where we palpated in neighborhoods theorized to belong to the hard inclusion and between successive iterations of the inclusion segmentation algorithm, generated grids for further exploration as the union of neighborhoods around points along the theorized boundary of the inclusion. We could instead use *a priori* tissue models to inform which points, if palpated, would provide the most informative data about the boundary of the inclusion. These tissue models could be informed by solid mechanics models, or by various atlases of tumor cell lines such as those seen in [95]. The classification step of the inclusion segmentation algorithm could remain the same in this case; however, we could potentially

arrive at a more efficient and a possibly more deterministic sampling of points along the boundary of the inclusion.

There are many different directions to continue meaningful work in human-robot collaboration, but there are two in particular that have the most potential to form high-performing human-robot collaboration models and apply them in tasks where such collaboration models are necessary. The first is formulating collaboration models based on human intent. The introduction briefly mentions work by Dragan *et al.*, who conducted a human-robot collaborative task in which a robotic agent inferred and reacted to perceived human intent [24]. This approach towards autonomy could be powerful when combined with learning through demonstration techniques, such as those reviewed by Argall *et al.* [96]. A novel combination of these methods would be where a robotic agent learns a task through demonstration, and then by sensing human intent along a trajectory corresponding to the execution of a surgical task, more fluidly shares control with the human agent. A less-rigid division of labor between the robotic and human agents may enable a spectrum of human-robot collaboration models which could potentially be more intuitive for a human operator to understand and accept. However, collaboration models which use an autonomous agent trained by demonstration would be difficult to translate into the MMSF without framing the demonstration as a state machine. Zappella *et al.* have demonstrated their ability to segment different actions and create a state machine from a surgical demonstration [97]. These different gestures would constitute new *Robot Actions*, from which collaboration models can be developed.

Another area of research well-poised for the near future is implementing the MMSF on surgical tasks which require more than two arms to complete. The debridement task examined in Section 4.3 was limited by the experimental setup available to an implementation where foreign bodies were not embedded in the tissue. A more realistic task would be to find the stiff inclusion, cut an artificial skin to expose the inclusion, and then dissect it. This procedure would require more than two robotic arms. Now that the MMSF can interface with the RAVEN robot and the dVRK, such a setup could be constructed. Robotic agents could be developed to support exposing the inclusion by retracting the skin, and could also help with the dissection

task more directly with cutting and tensioning. The combinations of human-robot collaboration models are quite large, but with the MMSF we can enable one human operator to complete new types of tasks. These tasks would normally require the human agent switching control between the multiple sets of arms or a second human agent operating in concert with the first to complete.

Bibliography

- [1] K. Fuchs, “Minimally invasive surgery,” *Endoscopy*, vol. 34, no. 2, pp. 154–159, 2002.
- [2] W. Lee, S. Kang, M. Kim, and M. Park, “ROBHAZ-DT3: teleoperated mobile platform with passively adaptive double-track for hazardous environment applications,” in *IEEE International Conference on Intelligent Robots and Systems*, 2004, pp. 33–38.
- [3] L. F. Penin, K. Matsumoto, and S. Wakabayashi, “Force reflection for time-delayed teleoperation of space robots,” in *IEEE International Conference on Robotics and Automation*, 2000, pp. 3120–3125.
- [4] J. C. Hu, X. Gu, S. R. Lipsitz, M. J. Barry, A. V. DAmico, A. C. Weinberg, and N. L. Keating, “Comparative effectiveness of minimally invasive vs open radical prostatectomy,” *Journal of the American Medical Association*, vol. 302, no. 14, pp. 1557–1564, 2009.
- [5] Y. Chen, J. Yan, Z. Yuan, S. Yu, Z. Wang, and Q. Zheng, “A meta-analysis of robotic-assisted pancreatectomy versus laparoscopic and open pancreatectomy,” *Saudi Medical Journal*, vol. 34, no. 12, pp. 1229–1236, 2013.
- [6] T. N. Payne and F. R. Dauterive, “A comparison of total laparoscopic hysterectomy to robotically assisted hysterectomy: surgical outcomes in a community practice,” *Journal of Minimally Invasive Gynecology*, vol. 15, no. 3, pp. 286–291, 2008.

- [7] N. R. Jackson, L. Yao, R. P. Tufano, and E. H. Kandil, "Safety of robotic thyroidectomy approaches: Meta-analysis and systematic review," *Head & Neck*, vol. 36, no. 1, pp. 137–143, 2014.
- [8] B. Kehoe, G. Kahn, J. Mahler, J. Kim, A. Lee, A. Lee, K. Nakagawa, S. Patil, W. D. Boyd, P. Abbeel, and K. Goldberg, "Autonomous multilateral debridement with the raven surgical robot," in *IEEE International Conference on Robotics and Automation*, 2014, pp. 1432–1439.
- [9] C. Coulson, R. Taylor, A. Reid, M. Griffiths, D. Proops, and P. Brett, "An autonomous surgical robot for drilling a cochleostomy: preliminary porcine trial," *Clinical Otolaryngology*, vol. 33, no. 4, pp. 343–347, 2008.
- [10] H. Mayer, F. Gomez, D. Wierstra, I. Nagy, A. Knoll, and J. Schmidhuber, "A system for robotic heart surgery that learns to tie knots using recurrent neural networks," *Advanced Robotics*, vol. 22, no. 13-14, pp. 1521–1537, 2008.
- [11] J. Van Den Berg, S. Miller, D. Duckworth, H. Hu, A. Wan, X.-Y. Fu, K. Goldberg, and P. Abbeel, "Superhuman performance of surgical tasks by robots using iterative learning from human-guided demonstrations," in *IEEE International Conference on Robotics and Automation*, 2010, pp. 2074–2081.
- [12] L. Conway, R. A. Volz, and M. W. Walker, "Teleautonomous systems: Projecting and coordinating intelligent action at a distance," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 2, pp. 146–158, 1990.
- [13] S. A. Greena, M. Billinghurstb, X. Chena, and J. G. Chasea, "Human-robot collaboration: A literature review and augmented reality approach in design," *International Journal of Advanced Robotic Systems*, vol. 5, no. 1, pp. 1–18, 2008.
- [14] J. Marescaux, J. Leroy, M. Gagner, F. Rubino, D. Mutter, M. Vix, S. E. Butner, and M. K. Smith, "Transatlantic robot-assisted telesurgery," *Nature*, vol. 413, no. 6854, pp. 379–380, 2001.

- [15] E. Beretta, F. Nesi, G. Ferrigno, and E. De Momi, "Force feedback enhancement for soft tissue interaction tasks in cooperative robotic surgery," in *IEEE International Conference on Intelligent Robots and Systems*, 2015, pp. 2009–2015.
- [16] D. De Lorenzo, Y. Koseki, E. De Momi, K. Chinzei, and A. M. Okamura, "Coaxial needle insertion assistant with enhanced force feedback," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 2, pp. 379–389, 2013.
- [17] N. Yu, K. Wang, Y. Li, C. Xu, and J. Liu, "A haptic shared control algorithm for flexible human assistance to semi-autonomous robots," in *IEEE International Conference on Intelligent Robots and Systems*, 2015, pp. 5241–5246.
- [18] R. Taylor, P. Jensen, L. Whitcomb, A. Barnes, R. Kumar, D. Stoianovici, P. Gupta, Z. Wang, E. DeJuan, and L. Kavoussi, "A steady-hand robotic system for microsurgical augmentation," *The International Journal of Robotics Research*, vol. 18, no. 12, pp. 1201–1210, 1999.
- [19] S. A. Bowyer, B. L. Davies, and F. Rodriguez y Baena, "Active constraints/virtual fixtures: A survey," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 138–157, 2013.
- [20] P. Marayong and A. M. Okamura, "Speed-accuracy characteristics of human-machine cooperative manipulation using virtual fixtures with variable admittance," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 46, no. 3, pp. 518–532, 2004.
- [21] J. J. Abbott, P. Marayong, and A. M. Okamura, "Haptic virtual fixtures for robot-assisted manipulation," in *IEEE International Symposium on Robotics Research*. Springer, 2007, pp. 49–64.
- [22] S. O. Oguz, A. Kucukyilmaz, T. M. Sezgin, and C. Basdogan, "Haptic negotiation and role exchange for collaboration in virtual environments," in *IEEE Haptics Symposium*, 2010, pp. 371–378.

- [23] C. Passenberg, A. Glaser, and A. Peer, “Exploring the design space of haptic assistants: the assistance policy module,” *IEEE Transactions on Haptics*, vol. 6, no. 4, pp. 440–452, 2013.
- [24] A. D. Dragan and S. S. Srinivasa, “A policy-blending formalism for shared control,” *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 790–805, 2013.
- [25] N. Stefanov, C. Passenberg, A. Peer, and M. Buss, “Design and evaluation of a haptic computer-assistant for telemanipulation tasks,” *IEEE Transactions on Human-Machine Systems*, vol. 43, no. 4, pp. 385–397, 2013.
- [26] K. Shamaei, Y. Che, A. Murali, S. Sen, S. Patil, K. Goldberg, and A. M. Okamura, “A paced shared-control teleoperated architecture for supervised automation of multilateral surgical tasks,” in *IEEE International Conference on Intelligent Robots and Systems*, 2015, pp. 1434–1439.
- [27] E. E. Tuna, T. J. Franke, O. Bebek, A. Shiose, K. Fukamachi, and M. C. Cavusoglu, “Heart motion prediction based on adaptive estimation algorithms for robotic-assisted beating heart surgery,” *IEEE Transactions on Robotics*, vol. 29, no. 1, pp. 261–276, 2013.
- [28] G. P. Moustiris, A. I. Mantelos, and C. S. Tzafestas, “Shared control for motion compensation in robotic beating heart surgery,” in *IEEE International Conference on Robotics and Automation*, 2013, pp. 5819–5824.
- [29] T. Debus, J. Stoll, R. D. Howe, and P. Dupont, “Cooperative human and machine perception in teleoperated assembly,” in *Experimental Robotics VII*. Springer, 2001, pp. 51–60.
- [30] J. Kofman, X. Wu, T. J. Luu, and S. Verma, “Teleoperation of a robot manipulator using a vision-based human-robot interface,” *IEEE Transactions on Industrial Electronics*, vol. 52, no. 5, pp. 1206–1219, 2005.

- [31] X. Wang, C. Yang, H. Ma, and L. Cheng, “Shared control for teloperation enhanced by autonomous obstacle avoidance of robot manipulator,” in *IEEE International Conference on Intelligent Robots and Systems*, 2015, pp. 4575–4580.
- [32] J. W. Crandall, M. Goodrich, D. R. Olsen Jr, C. W. Nielsen *et al.*, “Validating human-robot interaction schemes in multitasking environments,” *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 35, no. 4, pp. 438–449, 2005.
- [33] R. Parasuraman, T. B. Sheridan, and C. D. Wickens, “A model for types and levels of human interaction with automation,” *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 30, no. 3, pp. 286–297, 2000.
- [34] D. Kortenkamp, R. P. Bonasso, D. Ryan, and D. Schreckenghost, “Traded control with autonomous robots as mixed initiative interaction,” in *AAAI Symposium on Mixed Initiative Interaction*, 1997, pp. 95–97.
- [35] N. Padoy and G. D. Hager, “Human-machine collaborative surgery using learned models,” in *IEEE International Conference on Robotics and Automation*, 2011, pp. 5285–5292.
- [36] A. Shirkhodaie, “Supervised control of cooperative multi-agent robotic vehicles,” in *IEEE Southeastern Symposium on System Theory*, 2002, pp. 386–390.
- [37] R. Parasuraman, S. Galster, P. Squire, H. Furukawa, and C. Miller, “A flexible delegation-type interface enhances system performance in human supervision of multiple robots: Empirical studies with roboflag,” *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 35, no. 4, pp. 481–493, 2005.
- [38] T. Chen, D. Campbell, L. F. Gonzalez, and G. Coppin, “Increasing autonomy transparency through capability communication in multiple heterogeneous uav management,” in *IEEE International Conference on Intelligent Robots and Systems*, 2015, pp. 2434–2439.

- [39] D. Sieber, S. Music, and S. Hirche, “Multi-robot manipulation controlled by a human with haptic feedback,” in *IEEE International Conference on Intelligent Robots and Systems*, 2015, pp. 2440–2446.
- [40] The CISST libraries. Available: <http://www.cisst.org/cisst>.
- [41] P. Kazanzidesf, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio, “An open-source research kit for the da Vinci® Surgical System,” in *IEEE International Conference on Robotics and Automation*, 2014, pp. 6434–6439.
- [42] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “ROS: an open-source robot operating system,” in *ICRA Workshop on Open Source Software*, 2009.
- [43] S. Cousins, “Ros on the PR2,” *IEEE Robotics & Automation Magazine*, vol. 17, no. 3, pp. 23–25, 2010.
- [44] K. A. Nichols and A. M. Okamura, “Methods to segment hard inclusions in soft tissues,” *IEEE Transactions on Robotics and Automation*, vol. 31, no. 2, pp. 344–354, 2015.
- [45] K. Hoyt, B. Bastaneda, M. Zhang, P. Nigwekar, P. A. di Sant’Agnese, J. V. Joseph, J. Strang, D. J. Rubens, and K. J. Parker, “Tissue Elasticity Properties as Biomarkers for Prostate Cancer,” *Cancer Biomarkers*, vol. 4, no. 4-5, pp. 213–225, 2011.
- [46] S. Venkatesh, M. Yin, J. Glockner, N. Takahashi, P. Araoz, J. Talwalkar, and R. Ehman, “MR Elastography of Liver Tumors: Preliminary Results,” *American Journal of Roentgenology*, vol. 190, no. 6, pp. 1534–1540, 2008.
- [47] R. Tozzi, C. Köhler, A. Ferrara, and A. Schneider, “Laparoscopic Treatment of early Ovarian Cancer: Surgical and Survival Outcomes,” *Gynecologic oncology*, vol. 93, no. 1, pp. 199–203, 2004.

- [48] P. Kienle, J. Weitz, M. Koch, and M. Büchler, “Laparoscopic Surgery for Colorectal Cancer,” *Colorectal Disease*, vol. 8, no. 3, pp. 33–36, 2006.
- [49] B. Guillonnet, H. El-Fettouh, H. Baumert, X. Cathelineau, J. Doublet, G. Fromont, and G. Vallancien, “Laparoscopic Radical Prostatectomy: Oncological Evaluation After 1,000 Cases at Montsouris Institute,” *The Journal of urology*, vol. 169, no. 4, pp. 1261–1266, 2003.
- [50] A. M. Okamura, L. N. Verner, T. Yamamoto, J. C. Gwilliam, and P. G. Griffiths, “Force feedback and sensory substitution for robot-assisted surgery,” in *Surgical Robotics*. Springer, 2011, pp. 419–448.
- [51] S. B. Williams, M.-H. Chen, A. V. D’Amico, A. C. Weinberg, R. Kacker, M. S. Hirsch, J. P. Richie, and J. C. Hu, “Radical Retropubic Prostatectomy and Robotic-Assisted Laparoscopic Prostatectomy: Likelihood of Positive Surgical Margin(s).” *Urology*, vol. 76, no. 5, pp. 1097–1101, 2010.
- [52] V. Egorov and A. P. Sarvazyan, “Mechanical imaging of the breast,” *IEEE Transactions on Medical Imaging*, vol. 27, no. 9, pp. 1275–1287, 2008.
- [53] V. Egorov, T. Kearney, S. B. Pollak, C. Rohatgi, N. Sarvazyan, S. Airapetian, S. Browning, and A. Sarvazyan, “Differentiation of benign and malignant breast lesions by mechanical imaging,” *Breast cancer research and treatment*, vol. 118, no. 1, pp. 67–80, 2009.
- [54] M. Beccani, C. Di Natali, L. Sliker, J. Schoen, M. Rentschler, and P. Valdastri, “Wireless tissue palpation for intraoperative detection of lumps in soft tissue,” *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 2, pp. 353–361, 2014.
- [55] R. D. Howe, W. J. Peine, D. A. Kontarinis, and J. S. Son, “Remote Palpation Technology for Surgical Applications,” *IEEE Engineering in Medicine and Biology Magazine*, vol. 14, pp. 318–323, 1995.
- [56] A. Trejos, J. Jayender, M. Perri, M. Naish, R. Patel, and R. Malthaner, “Robot-Assisted Tactile Sensing for Minimally Invasive Tumor Localization,” *The International Journal of Robotics Research*, vol. 28, no. 9, pp. 1118–1133, 2009.

- [57] B. Ahn, Y. Kim, C. K. Oh, and J. Kim, “Robotic Palpation and Mechanical Property Characterization for Abnormal Tissue Localization.” *Medical & Biological Engineering & Computing*, pp. 1–11, 2012.
- [58] K. Sangpradit, H. Liu, P. Dasgupta, K. Althoefer, and L. D. Seneviratne, “Finite-Element Modeling of Soft Tissue Rolling Indentation.” *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 12, pp. 3319–3327, 2011.
- [59] H. Liu, D. P. Noonan, B. J. Challacombe, P. Dasgupta, L. D. Seneviratne, and K. Althoefer, “Rolling Mechanical Imaging for Tissue Abnormality Localization During Minimally Invasive Surgery.” *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 2, pp. 404–414, 2010.
- [60] T. Yamamoto, B. Vagvolgyi, K. Balaji, L. L. Whitcomb, and A. M. Okamura, “Tissue Property Estimation and Graphical Display for Teleoperated Robot-Assisted Surgery,” in *IEEE International Conference on Robotics and Automation*, 2009, pp. 4239–4245.
- [61] E. I. Zacharaki, S. Wang, S. Chawla, and D. Soo, “Classification of Brain Tumor Type and Grade Using MRI Texture and Shape in a Machine Learning Scheme,” *Magnetic Resonance in Medicine*, vol. 62, no. 6, pp. 1609–1618, 2010.
- [62] Y. Li, S. Hara, and K. Shimura, “A Machine Learning Approach for Locating Boundaries of Liver Tumors in CT Images,” in *International Conference on Pattern Recognition*, 2006, pp. 400–403.
- [63] K. Nichols and A. Okamura, “Autonomous Robotic Palpation: Machine Learning Techniques to Identify Hard Inclusions in Soft Tissues,” in *IEEE International Conference on Robotics and Automation*, 2013, pp. 4369–4374.
- [64] J. Ophir, I. Cespedes, H. Ponnekanti, Y. Yazdi, and X. Li, “Elastography: A Quantitative Method for Imaging the Elasticity of Biological Tissues,” *Ultrasonic Imaging*, vol. 13, no. 2, pp. 111–134, 1991.

- [65] B. S. Garra, I. E. Cespedes, J. Ophir, S. R. Spratt, R. A. Zuurbier, C. M. Magnant, and M. F. Pennanen, "Elastography of Breast Lesions: Initial Clinical Results," *Radiology*, vol. 201, no. 1, pp. 79–86, 1997.
- [66] L. Curiel, R. Souchon, O. Rouviere, A. Gelet, and C. J. Y., "Elastography for the Follow-up of High-Intensity Focused Ultrasound Prostate Cancer Treatment: Initial Comparison with MRI," *Ultrasound in Medicine & Biology*, vol. 31, no. 11, pp. 1461–1468, 2005.
- [67] R. Righetti, F. Kallel, J. R. Stafford, R. E. Price, T. A. Krouskop, J. D. Hazle, and J. Ophir, "Elastographic Characterization of HIFU-Induced Lesions in Canine Livers," *Ultrasound in Medicine & Biology*, vol. 25, no. 7, pp. 1099–1113, 1999.
- [68] T. Adebar, M. Yip, S. Salcudean, R. Rohling, C. Nguan, and L. Goldenberg, "Registration of 3D Ultrasound Through an Air-Tissue Boundary," *IEEE Transactions on Medical Imaging*, vol. 31, no. 11, pp. 2133–2142, 2012.
- [69] H. Li, M. Kallergi, L. Clarke, V. Jain, and R. Clark, "Markov Random Field for Tumor Detection in Digital Mammography," *IEEE Transactions on Medical Imaging*, vol. 14, no. 3, pp. 565–576, 1995.
- [70] Y. Masuda, T. Tateyama, W. Xiong, J. Zhou, M. Wakamiya, S. Kanasaki, A. Furukawa, and Y. W. Chen, "Liver Tumor Detection in CT Images by Adaptive Contrast Enhancement and the EM/MPM Algorithm," in *IEEE International Conference on Image Processing*, 2011, pp. 1421–1424.
- [71] J. Nie, Z. Xue, T. Liu, G. S. Young, K. Setayesh, L. Guo, and S. T. Wong, "Automated Brain Tumor Segmentation Using Spatial Accuracy-weighted Hidden Markov Random Field," *Computerized Medical Imaging and Graphics*, vol. 33, no. 6, pp. 431–441, 2009.

- [72] X. Liu, D. L. Langer, M. A. Haider, Y. Yang, M. N. Wernick, and I. S. Yetik, "Prostate Cancer Segmentation with Simultaneous Estimation of Markov Random Field Parameters and Class," *IEEE Transactions on Medical Imaging*, vol. 28, no. 6, pp. 906–915, 2009.
- [73] S. C. Agner, J. Xu, H. Fatakdawala, S. Ganesan, A. Madabhushi, S. Englander, M. Rosen, K. Thomas, M. Schnall, M. Feldman *et al.*, "Segmentation and Classification of Triple Negative Breast Cancers using DCE-MRI," in *IEEE International Symposium on Biomedical Imaging*, 2009, pp. 1227–1230.
- [74] M. Prastawa, E. Bullitt, N. Moon, K. Van Leemput, and G. Gerig, "Automatic Brain Tumor Segmentation by Subject Specific Modification of Atlas Priors," *Academic Radiology*, vol. 10, no. 12, pp. 1341–1348, 2003.
- [75] R. E. Goldman, A. Bajo, and N. Simaan, "Algorithms for autonomous exploration and estimation in compliant environments," *Robotica*, vol. 31, no. 01, pp. 71–87, 2013.
- [76] S. Z. Li, *Markov Random Field Modeling in Image Analysis*. Springer, 2009.
- [77] C. A. Perez, P. W. Grigsby, S. M. Nene, H. M. Camel, A. Galakatos, M.-S. Kao, and M. A. Lockett, "Effect of tumor size on the prognosis of carcinoma of the uterine cervix treated with irradiation alone," *Cancer*, vol. 69, no. 11, pp. 2796–2806, 1992.
- [78] L. Massoptier and S. Casciaro, "A new fully automatic and robust algorithm for fast segmentation of liver tissue and tumors from ct scans," *European radiology*, vol. 18, no. 8, pp. 1658–1665, 2008.
- [79] D.-R. Chen, R.-F. Chang, W.-J. Kuo, M.-C. Chen, and Y.-L. Huang, "Diagnosis of breast tumors with sonographic texture analysis using wavelet transform and neural networks," *Ultrasound in medicine & biology*, vol. 28, no. 10, pp. 1301–1310, 2002.

- [80] K. A. Nichols and A. M. Okamura, "A framework for multilateral manipulation in surgical tasks," in *The Hamlyn Symposium on Medical Robotics*, 2014, pp. 5–6.
- [81] D. Cheriton. Object-Oriented Programming from a Modeling and Simulation Perspective. Online: <http://web.stanford.edu/class/cs249a/>.
- [82] B. Hannaford, J. Rosen, D. W. Friedman, H. King, P. Roan, L. Cheng, D. Glozman, J. Ma, S. N. Kosari, and L. White, "Raven-II: an open platform for surgical robotics research," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 4, pp. 954–959, 2013.
- [83] K. E. Kaplan, K. A. Nichols, and A. M. Okamura, "Toward human-robot collaboration in surgery: Performance assessment of human and robotic agents in an inclusion segmentation task," in *IEEE International Conference on Intelligent Robots and Systems*, submitted.
- [84] K. A. Nichols, A. Murali, S. Sen, K. Goldberg, and A. M. Okamura, "Models of Human-Centered Automation in a Debridement Task," in *IEEE International Conference on Intelligent Robots and Systems*, 2005, pp. 5784–5789.
- [85] A. Murali, S. Sen, B. Kehoe, A. Garg, S. McFarland, S. Patil, W. D. Boyd, S. Lim, P. Abbeel, and K. Goldberg, "Learning by Observation for Surgical Subtasks: Multilateral Cutting of 3D Viscoelastic and 2D Orthotropic Tissue Phantoms," in *IEEE International Conference on Robotics and Automation*, 2015, in press.
- [86] J.-Y. Bouguet, "Matlab camera calibration toolbox," 2000.
- [87] G. Bradski and A. Kaehler, *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008.
- [88] A. Shrivastava and M. Menon, "Robotic radical prostatectomy: A step-by-step approach," in *Robotic Urologic Surgery*. Springer, 2007, pp. 81–90.

- [89] D. F. Friedlander, M. Alemozaffar, N. D. Hevelone, S. R. Lipsitz, and J. C. Hu, “Stepwise description and outcomes of bladder neck sparing during robot-assisted laparoscopic radical prostatectomy,” *The Journal of Urology*, vol. 188, no. 5, pp. 1754–1760, 2012.
- [90] M. P. Freire, A. C. Weinberg, Y. Lei, J. R. Soukup, S. R. Lipsitz, S. M. Prasad, F. Korkes, T. Lin, and J. C. Hu, “Anatomic bladder neck preservation during robotic-assisted laparoscopic radical prostatectomy: description of technique and outcomes,” *European Urology*, vol. 56, no. 6, pp. 972–980, 2009.
- [91] K. K. Brock, “Results of a multi-institution deformable registration accuracy study (midras),” *International Journal of Radiation Oncology* Biology* Physics*, vol. 76, no. 2, pp. 583–596, 2010.
- [92] S. McKinley, A. Garg, S. Sen, R. Kapadia, A. Murali, K. Nichols, S. Lim, S. Patil, P. Abbeel, A. M. Okamura *et al.*, “A disposable haptic palpation probe for locating subcutaneous blood vessels in robot-assisted minimally invasive surgery,” *IEEE CASE*, 2015.
- [93] M. P. Ottensmeyer, “In vivo measurement of solid organ visco-elastic properties,” *Studies in health technology and informatics*, pp. 328–333, 2002.
- [94] —, “Tempest i-d: An instrument for measuring solid organ soft tissue properties,” *Experimental Techniques*, vol. 26, no. 3, pp. 48–50, 2002.
- [95] R. K. Hay, J.-G. Park, and A. Gazdar, *Atlas of human tumor cell lines*. Academic Press, 2013.
- [96] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, “A survey of robot learning from demonstration,” *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.
- [97] L. Zappella, B. Béjar, G. Hager, and R. Vidal, “Surgical gesture classification from video and kinematic data,” *Medical image analysis*, vol. 17, no. 7, pp. 732–745, 2013.