UNIVERSITY OF CALIFORNIA

Los Angeles

Vision-Guided Autonomous Surgical Subtasks via Surgical Robots with Artificial Intelligence

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Mechanical Engineering

by

Changyeob Shin

2020

ABSTRACT OF THE DISSERTATION

Vision-Guided Autonomous Surgical Subtasks via Surgical Robots with Artificial Intelligence

by

Changyeob Shin
Doctor of Philosophy in Mechanical Engineering
University of California, Los Angeles, 2020
Professor Jacob Rosen, Chair

The introduction of automation into surgery may redefine the role of surgeons in operating rooms. While the majority of the manipulation will be performed autonomously by surgical robots, the surgeons may focus on decision-making procedures. This will drastically reduce the burden to surgeons by allowing them to instead interpret the abundant and intelligent information from the system, and will enhance the surgical outcome. To introduce the automation into surgery, the surgical robots are required to have: 1) high precision, 2) motion planning capabilities, and 3) scene understanding. Currently, surgical robots are commonly designed as cable-driven due to safety and several benefits such as low inertia. However, the cable-driven system has low precision because of cable stretch and long chains of cables. Therefore, a new control scheme of cable-driven surgical robots should be developed to overcome these limitations. Surgery is a complicated task consisting of multiple subtasks. To achieve the intermediate steps, motion planner should be developed. In surgery, the manipulation target objects are mostly soft tissue which introduces challenges in modeling the dynamics between the tool and the soft tissue. The motion planner should deal with the unknown dynamics while accomplishing each task. The surgical environment is further complicated by the many blood-covered anatomical structures. Surgeons use the visual feedback through an endoscope camera or other imaging devices, which provide rich information. Although the imaging devices are useful in understanding the surrounding anatomy, images from the devices are high-dimensional and it is difficult to process using algorithms to get high-level information. Therefore, vision-based perception algorithms to understand the relevant anatomy should be developed.

This dissertation addresses the three problems above. In chapter two, a hybrid control scheme which utilizes both model-based and data-driven methods is introduced to improve the precision of the cable-driven surgical robots and robustness to hand-eye calibration errors. The convergence of the controller is shown theoretically and experimentally with the Raven IV. Additionally, the efficacy of the controller to clinical tasks is shown by demonstrating the autonomous operations of needle transfer and tissue debridement tasks. In chapter three, learning-based path planning algorithms are proposed for autonomous soft tissue manipulation. The planning algorithms learn the dynamics between the motion of a surgical tool and soft tissue, and the internal controller uses the learned dynamics to manipulate the soft tissue. The performance of developed algorithms is verified on a designed simulation and a robot experiment with the Raven IV. In chapter four, the semantic segmentation algorithm of the optical coherence tomography images for the automated lens extraction is presented. The algorithm uses the deep learning method and provides the capability of understanding the cross-sectional view of the eye anatomy. Furthermore, this segmentation algorithm is incorporated into the Intraocular Robotic Interventional and Surgical System (IRISS) to realize the semi-autonomous lens removal. The experimental results on 7 *ex vivo* pig eyes verified the efficacy of the developed framework.

The dissertation of Changyeob Shin is approved.

Fabien Scalzo

Jean-Pierre Hubschman

Tsu-Chin Tsao

Jacob Rosen, Committee Chair

University of California, Los Angeles

2020

To my parents . . .

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to first thank Prof. Jacob Rosen for our discussions on selecting interesting research topics and encouraging me to try new and compelling approaches to solve the research problems.

I am very grateful to my lab (Bionics) colleagues and friends. I could successfully go through the doctoral program at UCLA because of your friendship. You always made the lab fun and intellectual place. I would like to thank Sahba Aghajani Pedram. We performed research projects together, spent time outside the lab many times, and also did an internship at Intuitive together during Summer. I learned many things from you, and enjoyed talking with you about various topics in our life. Hao Lee has been my friend who did many things together: taking classes from the first year of the Ph.D program, preparing and taking the preliminary exam, and research in the lab. It was fortunate to have you in this long journey. I am grateful to Peter Walker Ferguson who is always nice to people and is willing to help people. I will miss our conversation which was always full of joy. Jianwei Sun, who joined the lab in the last year of my Ph.D. study, also deserves my appreciation. I learned a lot from him even in this short time. I would like to thank Dr. Ji Ma who is a bright person and always helped me in many research projects.

I was fortunate to work in the Advanced Robotic Eye Surgery Lab with Dr. Jean-Pierre Hubschman, Prof. Tsu-Chin Tsao, Dr. Matt Gerber, and Dr. Yu-Hsui (Martin) Lee. I enjoyed our weekly discussion on the projects and collaboration.

I appreciate my friends met at UCLA, Henry Huh, Dong-hyun Won, Hyobin Won, Soo Lee, Duck-ha Hwang, Ga Young Lee, Sangmin Lim, and Won Ho Chung for having good times and conversations together outside the school.

Lastly, I am thanking my parents who have given me unconditional love and supports. I could get this achievement because of your love. I love you.

VITA

2013	B.E. in Electrical Engineering, Korea University, Seoul, South Korea
2015	M.E. in Mechanical Engineering, Korea University, Seoul, South Korea
2015-Current	Ph.D. student in Mechanical Engineering, University of California, Los Angeles, Los Angeles, USA
2016-2020	Teaching Assistant, Associate, and Fellow in Mechanical Engineering, University of California, Los Angeles, Los Angeles, USA
2019	Applied Research Intern, Intuitive Inc., Sunnyvale, USA

PUBLICATIONS

* denotes equal contribution

Journal and Conference papers

[1] Sahba Aghajani Pedram*, **Changyeob Shin***, Peter W. Ferguson, Ji Ma, Erik P. Dutson, Jacob Rosen, Autonomous Suturing Framework and Quantification Using a Cable-driven Surgical Robot, *IEEE Transactions on Robotics*, Accepted, 2020.

[2] **Changyeob Shin**, Peter Walker Ferguson, Sahba Aghajani Pedram, Ji Ma, Erik P. Dutson, Jacob Rosen, Autonomous Tissue Manipulation via Surgical Robot Using Learning Based Model Predictive Control, IEEE International Conference on Robotics and Automation (ICRA), 2019.

[3] **Changyeob Shin**, Matthew J. Gerber, Yu-Hsiu Lee, Mercedes Rodriguez, Sahba Aghajani Pedram, Jean-Pierre Hubschman, Tsu-Chin Tsao, Jacob Rosen, Semi-Automated Extraction of Lens

Pieces in *ex vivo* Pig Eyes using Semantic Segmentation of OCT Images with Deep Learning, Submitted to Journal.

[4] **Changyeob Shin***, Sahba Aghajani Pedram*, Jacob Rosen, Autonomous Control of Cable-driven Surgical Robots With Online Residual Learning, Submitted to Journal.

[5] Sahba Aghajani Pedram, Peter Ferguson, **Changyeob Shin**, Ankur Mehta, Erik Dutson, Farshid Alambeigi, Jacob Rosen, Toward Synergic Learning for Autonomous Manipulation of Deformable Tissues Via Surgical Robots: An Approximate Q-Learning Approach, IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics, 2020, Accepted.

[6] Sahba Aghajani Pedram, Peter Ferguson, Matthew J. Gerber, **Changyeob Shin**, Tsu-Chin Tsao, Jean-Pierre Hubschman, Jacob Rosen, A Novel Tool Localization Framework in Cataract Surgery using an Integrated Probe and Machine Learning Algorithms, Submitted to Journal.

CHAPTER 1

Introduction

Surgical robots are widely used in the operating rooms because of their high precision, dexterity, and capability of augmenting a surgeon's ability. Among many surgical robots, da Vinci Surgical system in particular [1] has been successfully commercialized and deployed in many hospitals. Intuitive Surgical's da Vinci Surgery system had been used for over 563,000 surgeries in 2016 [2]. The Raven open-platform surgical system [3] on the other hand was developed for research purposes and has resulted in a community to encourage research collaboration. Surgical robots are typically master-slave systems in which a surgeon operates a master console and a slave system follows the motion of surgeon's hand. This teleoperation system has shown great success through the integration of 3D vision and ergonomic designs to provide comfort to the surgeon. Compared to the Laparoscopic surgery, which is also a type of minimally invasive surgery, the master-slave system in surgical robots provides intuitive interface to control the robotic tools in the camera view. This intuitive interface can drastically reduce the time for learning how to operate the system and allow surgeons to better perform the surgery. As it is a minimally invasive method, patients recover faster than in the case of conventional open surgery [4].

As the surgical robot technology has advanced, there has been efforts from researchers to introduce automation into the surgery. We expect that, with the automated operations, surgical robots perform physical efforts for surgeons while the surgeons focus on decision making processes. The incorporation of autonomy into surgery has resulted in ethical and legal issues being discussed for different levels of autonomy [5]. In this new approach, surgeons are not required to consistently perform physical labors which could last several hours for some surgeries. It would drastically save the energy of surgeons in the operating room and potentially improve the patient outcome by allowing the surgeons to focus on diagnosis or treatments. Furthermore, as imaging and diagnosis technologies advance, surgeons can be overwhelmed by the abundance of information provided by the surgical systems. However, with the autonomous operation, surgeons would instead be able to fully utilize the abundant information from the intelligent systems as it is no longer necessary to perform manipulations.

The successful introduction of autonomy into the robotic surgery requires robots to be able to 1) move a tool precisely, 2) plan the motion of the tool for accomplishing a task, and 3) understand the surgical environment well. Precise manipulation of the surgical tool relies on the control and estimation of the surgical robot system. Planning the motion of robot to achieve a long-term goal is managed by a path planner of the robot. Understanding of the surgical scene could be realized in many ways including vision sensing, texture sensing, etc. In this dissertation, each of these problems is addressed and the efficacy of each proposed solutions is evaluated with experiments.

1.1 Control of Cable-Driven Surgical Robots

Conventional surgical robots such as the da Vinci [1] and the Raven surgical system [3] are cable-driven. In cable-driven systems, the motors are installed remotely from the joints of the robot, which are actuated through cables. This mechanism has low inertia and weight, which results in more safety to the patient, and also is easier to sanitize. Despite their advantages, cable-driven systems have low kinematic accuracy due to cable-tensioning and coupling. These two factors are nonlinear and difficult to model. In teleoperation, surgeons manually compensate for the kinematic inaccuracies through the vision system. However, in autonomous operation, the robot should compensate for inaccuracies using external sources of measurement such as vision by itself. Even though industrial robots with higher accuracy could be used instead, it is more desirable to deploy existing cable-driven surgical robots which are specifically designed for safety and ease of sanitation. To consider these factors, this dissertation explores the method for improving the accuracy of existing cable-driven systems. There have been approaches by researchers to identify the uncertain parameters offline or to adapt/learn the system parameters online for achieving the control stability. However, the offline calibration has disadvantages such as simplified model and lack of generalization ability. On the other hand, the learning approaches suffer from the sensitivity to the initialization of parameters. To address these problems, in this study, a hybrid approach which utilizes both the model-based and learning schemes to improve the precision of a cable-driven surgical robot and robustness to hand-eye calibration errors was developed.

1.2 Motion Planning for Surgical Subtasks

Surgery is a complex procedure which consists of sequences of multiple subtasks. There have been numerous research efforts to analyze the sequence of the surgical tasks. Unsupervised learning was used to learn the trajectory of surgical tasks [6]. A finite state machine of the cutting task was learned from examples and the da Vinci Research Kit (DVRK) [7] was controlled to perform the task autonomously [8]. An optimization-based method for planning the suture needle with clinical suture parameters was proposed [9]. The motion of the surgical tool should be planned strategically to accomplish the task. In planning, optimization frameworks which minimize a defined cost function while satisfying constraints are used. In a robot manipulation task, dynamics between the motion of the surgical tool and the change of the target object's states are included in the set of constraints. However, in soft tissue surgeries, the dynamics between the tool and interest points on the tissue are unknown, prohibiting robot motion planning. To tackle this problem, learning based motion planning algorithms for soft tissue manipulation are introduced in this work.

1.3 Visual Understanding of Surgical Environment

A vision system provides rich information, but it is challenging to intelligently process images. It is especially the case in surgery because of the complex environment and limited view. Recently, the advance of deep learning has contributed to solve many challenging problems in computer vision and image analysis [10]. Deep learning utilizes a neural network which consists of many layers of different computation units. Generation of big dataset such as COCO [11] and ImageNet [12] and non-linear optimization methods (e.g. Adam [13]) are two of the most important factors for the success of the deep learning. Among the many different structures of deep neural networks, the convolutional neural network (CNN) works particularly well for image processing problems by exploiting a convolution operator for extracting visual features around the target pixel. As the layer becomes deeper, higher level information is extracted and used for image classification or segmentation problems. Examples of CNNs include AlexNet [14], VGG [15] and GoogLeNet [16]. In this research, we propose a deep learning-based image analysis method for the task of autonomously removing a lens piece. Surgeons have limited visual cues during cataract surgery, so the optical coherence tomography (OCT) is used to provide a cross-sectional view of the relevant anatomy. It is a non-invasive imaging method which can provide the preoperative, intraoperative, and postoperative scans for understanding the anatomy in different stages of the surgery. However, the OCT system suffers from low signal-to-noise ratio and Speckle noise. The noise in the OCT images is difficult to filter using conventional filters. To tackle this problem, the deep neural network is developed to semantically segment the OCT images and localize the intraocular structures. Furthermore, this segmentation algorithm is incorporated into the intraocular surgical system to demonstrate the semi-automated lens extraction.

1.4 Contributions of Each Chapter

Contributions of each chapter are summarized here.

- Chapter 2

  A visual servo kinematic controller which is a hybrid of model-based and data-driven approaches is proposed to compensate, in real-time, for uncertainty in system parameters such as cable tensioning and coupling.

It is shown that the proposed controller improves the positioning accuracy of Raven IV by factors of 50 for position and 10 for orientation compared to the kinematic controller.

It is reported that the developed controller exhibits the robustness to hand-eye calibration errors up to 30 degrees in all axes.

Efficacy of the controller is shown by demonstrating the needle transfer and tissue debridement tasks which require high precision.

- Chapter 3

  Two learning-based path planning algorithms (Reinforcement Learning and Learning from Demonstration) which utilize a fully connected neural network and a model predictive controller are proposed for autonomous tissue manipulation task.

  A simulation for the soft tissue manipulation was designed and the proposed two algorithms were tested. From the result, it is shown that both algorithms successfully perform the manipulation task. Furthermore, it is demonstrated that the policy from the learning from demonstration algorithm finished the task without further exploration.

  The learning from demonstration algorithm was applied on a robot experiment with the Raven IV and it is shown that the surgical robot successfully manipulated a target object made of highly elastic latex.

- Chapter 4

  A deep-learning framework to segment intraocular anatomy (cornea, iris, lens, and capsule) in OCT images was developed.

  OCT images from 10 pig eyes were collected to train the neural network and its performance was evaluated using the images from 8 different pig eyes.

  The integrated solution with the segmentation algorithm and an intraocular surgical system for the semi-automated lens extraction was developed and its effectiveness was demonstrated on 7 *ex vivo* pig eyes.

CHAPTER 2

Control of Cable-Driven Surgical Robots Using Online Residual Learning

2.1 Introduction

Autonomous motion control schemes in robot-assisted surgery (RAS) have been investigated by many researchers in recent years. Examples include automation of suturing [9, 17, 18, 19], tissue manipulation [20, 21, 22], tissue dissection [8] as well as autonomous motion control of endoscope [23], and ultrasound [24]. While considerable legal, regulatory, and ethical issues need to be addressed, introduction of autonomy into RAS could potentially improve accuracy, repeatability, and time to perform for certain subtasks, leading to an improved surgical outcome. Modern surgical robotic systems such as da Vinci [1] and Raven IV [25] (see Fig. 2.1) are cable driven which allows placement of motors on the robot base and reduces mass and inertia of the arms. This results in a compact and light-weight design of the manipulators which mitigates safety concerns in surgical robotics [26]. Despite the great advantages, one of the main difficulties of these systems is that the elasticity of the cables introduces a grand challenge for modeling and precise control. For example, the stiffness of cables is lower compared to rigid body links which may result in undesirable vibrations and relative position error between the motors and the links [27]. This position error along with the long kinematic chains and instrument deflections lead to relatively large position regulation errors ($> 10\,mm$) [28, 29]. For teleoperated systems, surgeons compensate for such errors using direct visual feedback of the surgical scenes [30]. Incorporating autonomous motions into RAS without the surgeons in the loop, however, requires high positioning accuracy

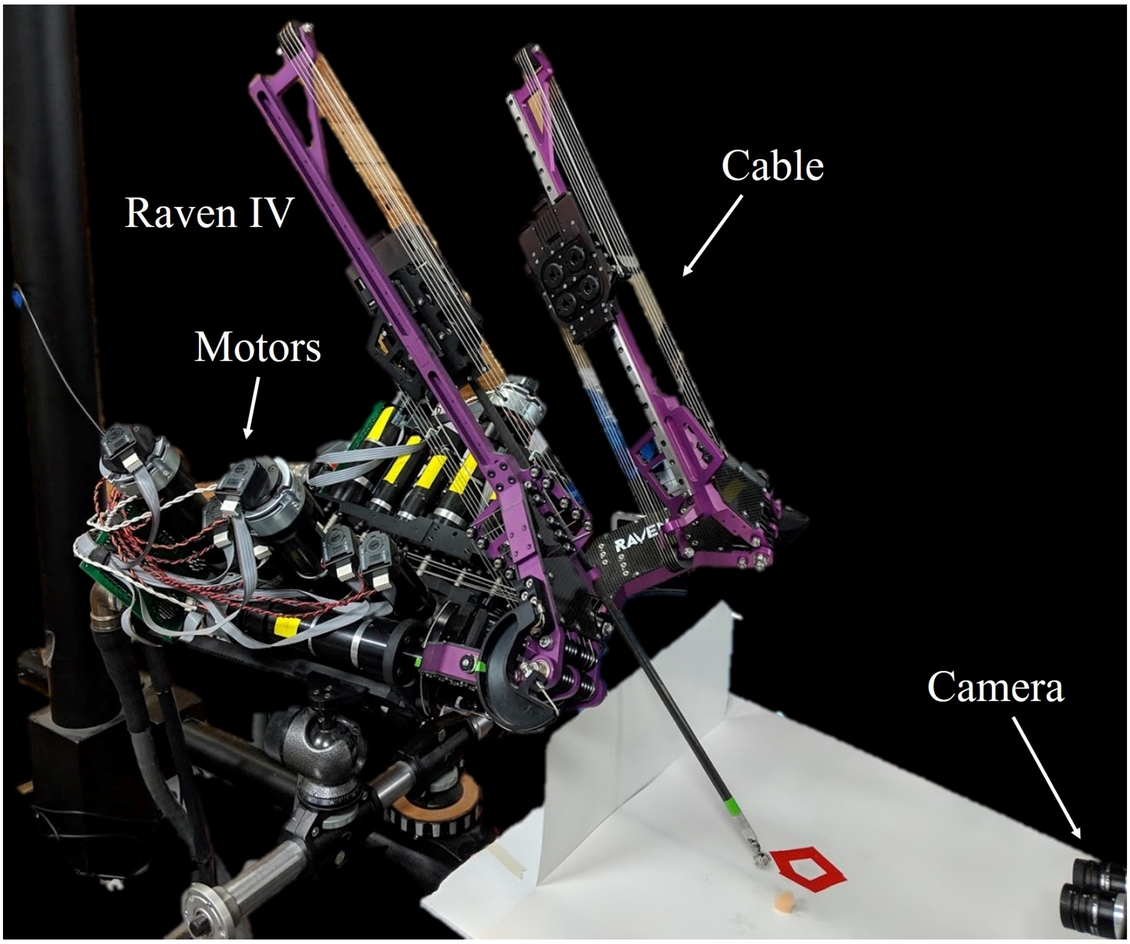


Figure 2.1: Raven IV cable-driven surgical robotic system.

for patient safety concerns [31]. Hence, enhanced solutions are required for the current surgical robots.

Accurate modeling of serial robotic manipulators is crucial for successful completion of grasping and manipulation tasks. In [32], the authors provide a comprehensive overview of kinematic calibration methods (i.e., to find D-H parameters). Many serial manipulators, however, include nonlinearities in kinematic chains such as friction and compliance which are not captured by such calibration approaches. Most of the previous studies either disregard such nonlinearities or estimate them through carefully engineered models and offline calibrations [33], [34], [35]. Although including such models and offline calibration results into improved controller performance, the main disadvantages are (i) the models are system specific and do not generalize well to other robots, (ii) the models are simple and do not fully capture the physics of the problem, and (iii) the offline calibration results are not robust to environment disturbances and changes.

In contrast, a body of research has recently focused on data-driven approaches. For example, Levine et al. [36] trained a large convolutional neural network to estimate the probability that gripper's motion results in successful grasps and used this information to servo the gripper. In [37], the authors gathered large number of expert demonstrations to train a visual servoing controller for needle insertion and picking tasks. Sturm et al. [38] proposed to learn the entire robot forward and inverse kinematic models from scratch through self-perception. All of these studies propose to learn a very complicated mapping while discarding readily available system models such as kinematics, registration, and sensor/actuator models. This adversely may harm accuracy and require large data set for proper generalization which is not sample efficient [39].

Hence, other studies have focused on a hybrid approach in which the correction terms such as cable stretch nonlinearities, which cannot be easily modeled, are learnt offline from input-output data. These residual terms, when deployed with a nominal system model (i.e., kinematics), lead to an improved performance. For example, Pastor et al. [39] examined the use of a linear function along with Gaussian Process Regression (GPR) to learn the pose-dependent correction terms for improving the state estimation of the ARM-S robot. Similarly, Mahler et al. [28] used the same idea but augmented the states with velocities to account for cable compliance nonlinearities. This improved the position accuracy and speed of performing debridement task with the Raven II surgical robot. In [40], the authors trained a deep neural network to learn the mapping between camera and robot base frames (coarse calibration) and used a random forest learning method to estimate the remaining error between the two frames (fine calibration). Both calibration steps were performed offline and improved the performance of the da Vinci Research Kit (dVRK) in terms of the success rate and position accuracy of the surgical debridement task. Aoyagi et al. [41] used the calibrated kinematic model of the PA10 Robot and trained an additional fully connected neural network to compensate for the residual errors of the kinematic chain. While the results of these studies have been encouraging, the main drawback is that such correction terms were learnt for particular tasks and/or specific regions of the robot state space. Since the nonlinearities of cable-driven systems are pose dependent, these approaches might be applicable. Moreover, if such nonlinearities (i.e. cables tensions) and/or system characteristics change with time, the training process needs to be performed again which is not efficient. As such, although learning the correction terms might be more feasible compared to the learning the entire model [40], generalization to other tasks (or robot configurations) and adaptability to system changes remain unsolved issues.

To generalize the learned models to a larger state space and to adapt the models for time dependent changes, online learning of such models or residues is crucial [42]. However, for most real-time applications online model learning poses two major difficulties. First, both the learning and prediction phases must be performed at high frequency (i.e. 20-200 Hz for learning and 200-5k Hz for prediction) [43]. Second, the model has to be adapted to the continuous stream of data. For example, to solve issues with high computational cost of GPR method at estimation step, one group of studies [43, 44, 45] deployed local GPR in which the training data is partitioned into local regions and independent Gaussian processes are learnt for each region. The prediction for a new point is performed by weighted average estimate from each model. While such method enabled real-time control of SARCOS robot arm and a soft robot, one major issue is that the performance heavily depends on how the data is partitioned [42]. Also, the effective way of partitioning high-dimensional data, especially in robotics, is still an open question.

As mentioned above, extensive offline and online algorithms have been developed to account for the nonlinearities in robotic systems and to improve modeling and control accuracy. On the other hand, many manipulation and/or positioning tasks can be successfully accomplished with proper feedback (i.e., vision), despite the unmodeled nonlinearities. Canonical examples are calibrated [46, 47] and uncalibrated visual servoing [48] methods. Calibrated visual servoing (CVS) is broadly categorized into position-based and image-based, depending on the space in which the error is calculated. In CVS, offline camera calibration (intrinsic and/or extrinsic) and robot kinematics along with a real-time visual feedback are deployed to servo the robot. The main disadvantage of CVS is that the performance (i.e. error convergence or transient/steady state response) is highly dependent on the accuracy of calibration and kinematics model [46]. Moreover, CVS framework does not encompass any notion of real-time adaption/learning and is not robust to environment disturbances. Uncalibrated visual servoing (UVS), on the other hand, relies on the real-time learning of the locally linearized mapping between the image space and actuation inputs. As a result, UVS is more robust to parameters changes or unknown environments. However, major drawbacks are (i) the performance sensitivity to hand-crafted initialization and (ii) no system model is included in the algorithm which might adversely affect the accuracy [39].

Contributions: In this chapter, we present a novel online residual learning (ORL) algorithm to address the shortcomings of the methods discussed above. ORL is a data-driven approach in which a linearized mapping between the robot joint angles and the robot end-effector pose is learned in real-time. The ORL utilizes readily available data solely from motor encoders and camera measurements and deploys the mapping to calculate the control inputs. The main contributions of this chapter are as follows:

- The ORL algorithm compensates, in real-time, for the system nonlinearities which are challenging to model (i.e., cable tensioning).
- The ORL algorithm combines the model-based approach (i.e., initialization with the robot Jacobian) with the learning-based approach (i.e., data-driven residual estimation in real-time) constituting a hybrid method which is sample efficient and robust to modeling errors.
- The ORL algorithm improves the control accuracy of the Raven IV open-platform surgical system by factors of 50 and 10 for position and orientation respectively.

2.2 Method

In this section, the general framework of the proposed method is explained. Notations and definitions used throughout the chapter are defined in Section 2.2.1. The kinematics model, vision-based controller, and state estimation of cable-driven robotic systems are discussed in Section 2.2.2. The proposed online residual learning algorithm is presented in Section 2.2.3. The end-effector grasper design and tracking are explained in Section 2.2.4. Hand-eye calibration algorithm is described in Section 2.2.5.

2.2.1 Notations and Definitions

$\mathbf{x} = [x, y, z, \phi, \theta, \psi]^\intercal \in \mathbb{R}^6$ is a pose state vector where the first three elements are positions and the last three are roll, pitch, yaw angles expressing rotation along x, y, and z axes respectively. $\hat{\bullet}$ refers estimated values of scalars, vectors, or matrices. We denote frames as following: *end-effector* (E), *robot-base* (B), and *camera* (C). ${}^{a}\mathbf{T}_{b} \in \mathbb{R}^{4\times4}$ represents a homogeneous transformation of frame b in frame a.

2.2.2 Cable-Driven Robotic Systems

1) Kinematics: In a typical cable-driven surgical robotic system such as Raven IV [3], motors are installed on the robot base and remotely actuate the joints through cable transmissions. For these systems, two mappings are defined as shown in Fig. 2.2: (i) a mapping from the motor space to the joint space, called $Transmission\ Kinematics$ ($\mathcal{T}$), and (ii) a mapping from the joint space to the end-effector space, denoted by $Forward\ Kinematics$ ($\mathcal{F}$). The Transmission Kinematics can be expressed as:

$$\mathbf{q} = \mathcal{T}(\mathbf{s}) \tag{2.1}$$

where $\mathcal{T} : \mathbb{R}^m \mapsto \mathbb{R}^q$, $\mathbf{s} \in \mathbb{R}^m$ is a position vector of m motors and $\mathbf{q} \in \mathbb{R}^q$ is a position vector of q joints. For the Raven system, m, and q are 7 and 6 respectively. The Forward Kinematics can be expressed as:

$${}^{\mathrm{B}}\mathbf{T}_{\mathrm{E}} = \mathcal{F}(\mathbf{q}) \tag{2.2}$$

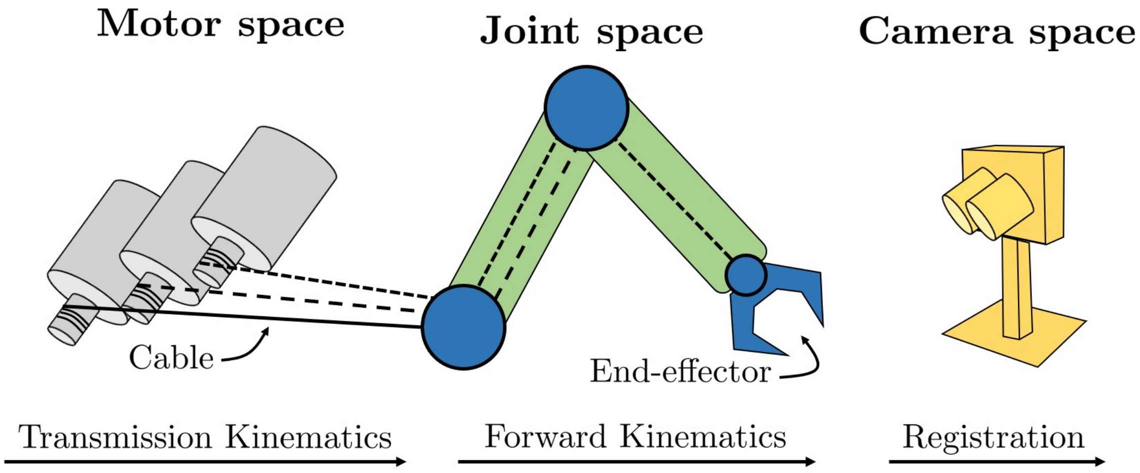


Figure 2.2: Motor space, joint space and camera space of cable-driven system with visual feedback.

where $\mathcal{F} : \mathbb{R}^q \mapsto SE(3)$. The $SE(3)$ denotes Special Euclidean group. In addition, the transformation ${}^{\mathrm{B}}\mathbf{T}_{\mathrm{E}}$ can be expressed as a 6 degrees of freedom (DoF) pose vector ($\mathbf{x}$) as:

$$\mathbf{x} = \zeta(\mathbf{T}) = \zeta\left(\begin{bmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}\right) = \begin{bmatrix} \mathbf{p} \\ \tan^{-1}(\frac{r_{32}}{r_{33}}) \\ \tan^{-1}(\frac{-r_{31}}{\sqrt{r_{32}^2+r_{33}^2}}) \\ \tan^{-1}(\frac{r_{21}}{r_{11}}) \end{bmatrix} \tag{2.3}$$

where ζ: $SE(3) \mapsto \mathbb{R}^6$ is the mapping between the homogeneous transformation matrix and the pose vector. r_{ij} refers to an element of $\mathbf{R}$ at i^{th} row and j^{th} column.

2) Vision-based Planning and Control: In this work, vision feedback from the stereo camera system is used for planning, control, and real-time residual learning. As a result, we argue that the ORL framework deploys the vision feedback stream in a novel way across different modules of a closed-loop cable-driven system.

For planning, we assume that the desired pose of the robot end-effector is determined by a path planning algorithm in camera space as ${}^{\mathrm{C}}\mathbf{T}_{\mathrm{E}}^{des}$. Similarly, the current pose of end-effector is measured and expressed as ${}^{\mathrm{C}}\mathbf{T}_{\mathrm{E}}$ in the camera space. To transfer this information to the robot base

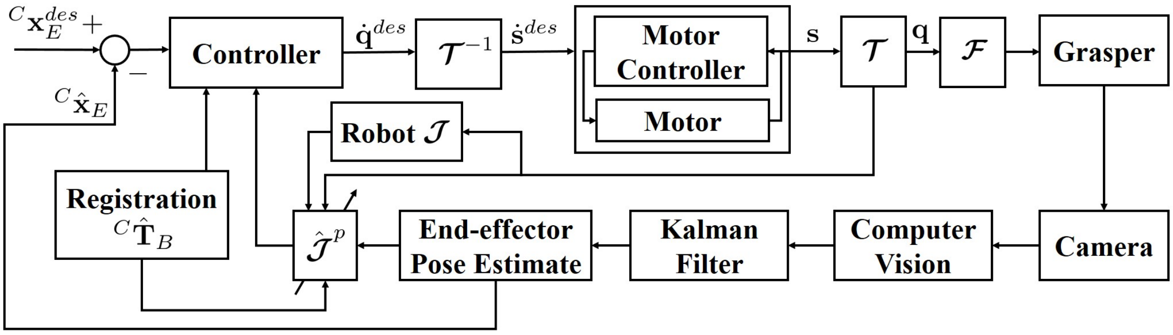


Figure 2.3: The proposed online residual learning control scheme for cable-driven robots with visual feedback.

frame, the relation between the camera frame and the robot base frame should be known. We refer this relation as *hand-eye calibration* and denote it as ${}^{\mathrm{C}}\mathbf{T}_{\mathrm{B}}$.

With this information, we can express the current and desired pose of the end-effector in the robot base frame as:

$$ {}^{\mathrm{B}}\mathbf{T}_{\mathrm{E}}^{des} = {}^{\mathrm{B}}\mathbf{T}_{\mathrm{C}}{}^{\mathrm{C}}\mathbf{T}_{\mathrm{E}}^{des} \tag{2.4} $$

$$ {}^{\mathrm{B}}\mathbf{T}_{\mathrm{E}} = {}^{\mathrm{B}}\mathbf{T}_{\mathrm{C}}{}^{\mathrm{C}}\mathbf{T}_{\mathrm{E}} \tag{2.5} $$

With the current and desired pose of end-effector, the controller loop is closed and the pose error vector is calculated as:

$$ {}^{\mathrm{B}}\mathbf{e}_{\mathrm{E}} = \zeta({}^{\mathrm{B}}\mathbf{T}_{\mathrm{E}}^{des}) - \zeta({}^{\mathrm{B}}\mathbf{T}_{\mathrm{E}}) \tag{2.6} $$

Using this pose error, desired joint velocities are calculated using the differential kinematics:

$$ \begin{gathered} \min_{\dot{\mathbf{q}}} \left\| {}^{\mathrm{B}}\mathbf{e}_{\mathrm{E}} - \hat{\mathcal{J}}\dot{\mathbf{q}} \right\|_2 \\ s.t.\ \mathbf{q}_{min} < \hat{\mathbf{q}} + \dot{\mathbf{q}}\delta t < \mathbf{q}_{max} \end{gathered} \tag{2.7} $$

In this equation, $\dot{\mathbf{q}} \in \mathbb{R}^6$ is the joints velocity vector, and δt is a control time step. $\mathbf{q}_{min}$ and $\mathbf{q}_{max}$ are minimum and maximum mechanical joint limits, and the $\hat{\mathcal{J}} \in \mathbb{R}^{6\times6}$ is the estimated robot Jacobian matrix defined as $\hat{\mathcal{J}} = \mathcal{J}(\hat{q})$. We refer to this matrix as *Robot Jacobian (RJ)*. The $\hat{\mathbf{q}}$

is the estimated joint position vector and $\mathcal{J}$ is the analytical robot Jacobian which is obtained by differentiating the end-effector pose vector with respect to the robot joint vector:

$$\mathcal{J} = \begin{bmatrix} \frac{\partial({}^b x_e)}{\partial q_1} & \frac{\partial({}^b x_e)}{\partial q_2} & \cdots & \frac{\partial({}^b x_e)}{\partial q_6} \\ \frac{\partial({}^b y_e)}{\partial q_1} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \frac{\partial({}^b \psi_e)}{\partial q_1} & \cdots & \cdots & \frac{\partial({}^b \psi_e)}{\partial q_6} \end{bmatrix} \tag{2.8}$$

3) Estimation Errors: In the previously discussed cable-driven robot with vision feedback, there are two main sources for state estimation errors. First, because of the cable tension and nonlinearities in the transmission kinematics which are difficult to model, the estimation of robot joint angles is inaccurate which leads to an incorrect $\hat{\mathcal{J}}$. Second, in the control feedback, the estimation of ${}^{\mathrm{B}}\mathbf{T}_{\mathrm{C}}$ may not be accurate or become corrupted during the operation. These error sources might provide inaccurate information for motion control (see experiment II result). We demonstrate that the ORL algorithm can successfully address these issues and provides a superior control performance and robustness to these errors.

2.2.3 Online Residual Learning Algorithm

1) Algorithm: To improve the model-based controller (i.e., CVS) discussed in Sec. 2.2.2, an online residual learning approach (the ORL algorithm) is proposed to be incorporated with controller (2.7). Fig. 2.3 shows the comprehensive framework for autonomous control of the Raven IV robot using the ORL algorithm. The proposed algorithm takes into account the temporal changes of the robot end-effector pose as well as the joints angles to learn and update a linearized mapping. Throughout the chapter, this linear mapping which relates the changes in the end-effector position to the changes in the joints angles is referred to as *Pseudo Jacobian (PJ)* and denoted as $\hat{\mathcal{J}}^p$. Note that RJ and PJ represent the same mapping but with different methodologies; PJ deploys a combination of a model-based and a learning-based approaches (as we shall see later) while RJ uses only a model-based approach. Moreover, it is important to note that PJ does not necessarily calculate the (unknown) true robot Jacobian. Rather, it estimates arbitrary parameters that ensure

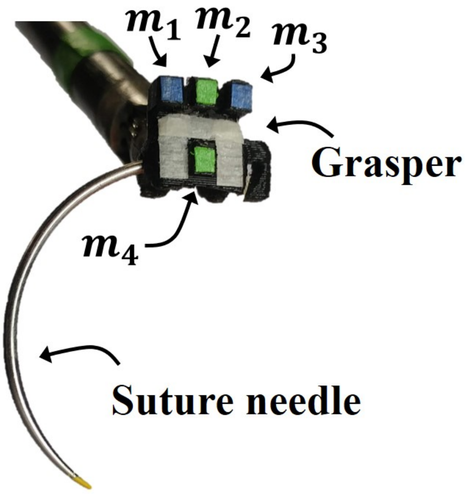


Figure 2.4: Needle grasper with four colored markers.

the asymptotic convergence to the desired values.

In our robotic setup, a 3D-printed grasper with four color markers is installed on the end-effector to provide vision feedback (see Fig. 2.4). The relationship between the grasper and end-effector which is obtained from a CAD model is denoted as ${}^{\mathrm{Gr}}\mathbf{T}_{\mathrm{E}}$. As a result, the Eq. (2.5) can be expressed as:

$$ {}^{\mathrm{B}}\hat{\mathbf{T}}_{\mathrm{E}} = {}^{\mathrm{B}}\hat{\mathbf{T}}_{\mathrm{C}}{}^{\mathrm{C}}\hat{\mathbf{T}}_{\mathrm{Gr}}{}^{\mathrm{Gr}}\mathbf{T}_{\mathrm{E}} \tag{2.9} $$

Of note, if the grasper is not used, the ${}^{\mathrm{Gr}}\mathbf{T}_{\mathrm{E}}$ would be an Identity matrix. In this case, the ${}^{\mathrm{C}}\hat{\mathbf{T}}_{\mathrm{E}}$ would be directly measured and Eq. (2.5) could be used. In either way, the changes of the end-effector pose between the time t and $t-1$ can be expressed as:

$$ \begin{aligned} \Delta^{\mathrm{B}}\mathbf{x}_{\mathrm{E}} &= \zeta({}^{\mathrm{B}}\hat{\mathbf{T}}_{\mathrm{E},t}) - \zeta({}^{\mathrm{B}}\hat{\mathbf{T}}_{\mathrm{E},t-1}) \\ &= {}^{\mathrm{B}}\hat{\mathbf{x}}_{\mathrm{E},t} - {}^{\mathrm{B}}\hat{\mathbf{x}}_{\mathrm{E},t-1} \end{aligned} \tag{2.10} $$

To relate these changes of the pose to the changes of the joint angles and update PJ, temporal changes of the joints angles is acquired as $\Delta\mathbf{q} = \mathbf{q}_t - \mathbf{q}_{t-1}$. Of note, the changes of the joint angles are in the Joint space and not in the Motor space (see Fig. 2.2). With these two temporal changes of the end-effector pose as well as the joints, an iterative approach to update PJ in real-time is exploited as following. Using the measurements of $\Delta^{\mathrm{B}}\mathbf{x}_{\mathrm{E}}$ and $\Delta\mathbf{q}$ at time t, $\hat{\mathcal{J}}^{p}_{t}$ is calculated from $\Delta^{\mathrm{B}}\mathbf{x}_{\mathrm{E}} = \hat{\mathcal{J}}^{p}_{t}\Delta\mathbf{q}$. Note that this equation has infinite solutions for $\hat{\mathcal{J}}^{p}_{t}$ in general but the ORL algorithm finds an optimal one. By subtracting $\hat{\mathcal{J}}^{p}_{t-1}\Delta\mathbf{q}$ from both sides and defining $\Delta\hat{\mathcal{J}}^{p} = \hat{\mathcal{J}}^{p}_{t} - \hat{\mathcal{J}}^{p}_{t-1}$, we get:

$$\begin{aligned}\Delta^{\mathrm{B}}\mathbf{x}_{\mathrm{E}} - \hat{\mathcal{J}}^{p}_{t-1}\Delta\mathbf{q} &= \hat{\mathcal{J}}^{p}_{t}\Delta\mathbf{q} - \hat{\mathcal{J}}^{p}_{t-1}\Delta\mathbf{q} \\ &= \Delta\hat{\mathcal{J}}^{p}\Delta\mathbf{q}\end{aligned} \tag{2.11}$$

As a result, $\Delta\hat{\mathcal{J}}^{p}$ should satisfy Eq. (2.11). On the other hand, the ORL algorithm requires minimal changes of PJ in order for the robot motion to be smooth. If the changes in PJ are large, the robot commanded pose would differ significantly from the current pose resulting in jerky robot motion or high motor torques which are not desirable. As a result, PJ changes measured by Frobenius norm should be minimized and hence the following constrained optimization problem is formulated:

$$\begin{aligned}&\min_{\Delta\hat{\mathcal{J}}^{p}} \frac{1}{2}\left\|\Delta\hat{\mathcal{J}}^{p}\right\|_{F}^{2} \\ &s.t.\ \Delta\hat{\mathcal{J}}^{p}\Delta\mathbf{q} = \Delta^{\mathrm{B}}\mathbf{x}_{\mathrm{E}} - \hat{\mathcal{J}}^{p}_{t-1}\Delta\mathbf{q}\end{aligned} \tag{2.12}$$

where$\|\cdot\|_{F}$ is the Frobenius norm. An augmented cost function $\boldsymbol{C}$ with Lagrange multipliers ($\boldsymbol{\lambda}$) is constructed to change Eq. (2.12) to an unconstrained optimization problem:

$$\min_{\Delta\hat{\mathcal{J}}^{p}} \boldsymbol{C} = \frac{1}{2}\left\|\Delta\hat{\mathcal{J}}^{p}\right\|_{F}^{2} + \boldsymbol{\lambda}^{\mathsf{T}}\left(\Delta\hat{\mathcal{J}}^{p}\Delta\mathbf{q} - \Delta^{\mathrm{B}}\mathbf{x}_{\mathrm{E}} + \hat{\mathcal{J}}^{p}_{t-1}\Delta\mathbf{q}\right) \tag{2.13}$$

To minimize this augmented cost function, the partial derivative of $\boldsymbol{C}$ with respect to $\Delta\hat{\mathcal{J}}^{p}$ is set to zero:

$$\frac{\partial\boldsymbol{C}}{\partial(\Delta\hat{\mathcal{J}}^{p})} = \Delta\hat{\mathcal{J}}^{p} + \boldsymbol{\lambda}\Delta\mathbf{q}^{\mathsf{T}} = 0 \tag{2.14}$$

With the Eq. (2.11) and Eq. (2.14), $\boldsymbol{\lambda}$ and $\Delta\hat{\mathcal{J}}^{p}$ are solved as:

$$\begin{aligned}\boldsymbol{\lambda} &= -\frac{\Delta^{\mathrm{B}}\mathbf{x}_{\mathrm{E}} - \hat{\mathcal{J}}^{p}_{t-1}\Delta\mathbf{q}}{\Delta\mathbf{q}^{\mathsf{T}}\Delta\mathbf{q}} \\ \Delta\hat{\mathcal{J}}^{p} &= -\boldsymbol{\lambda}\Delta\mathbf{q}^{\mathsf{T}} = \left(\frac{\Delta^{\mathrm{B}}\mathbf{x}_{\mathrm{E}} - \hat{\mathcal{J}}^{p}_{t-1}\Delta\mathbf{q}}{\Delta\mathbf{q}^{\mathsf{T}}\Delta\mathbf{q}}\right)\Delta\mathbf{q}^{\mathsf{T}}\end{aligned} \tag{2.15}$$

As a result, $\hat{\mathcal{J}}^{p}$ can be updated with Eq. (2.15):

$$\hat{\mathcal{J}}^{p}_{t} \leftarrow \hat{\mathcal{J}}^{p}_{t-1} + \left(\frac{\Delta^{\mathrm{B}}\mathbf{x}_{\mathrm{E}} - \hat{\mathcal{J}}^{p}_{t-1}\Delta\mathbf{q}}{\Delta\mathbf{q}^{\mathsf{T}}\Delta\mathbf{q}}\right)\Delta\mathbf{q}^{\mathsf{T}} \tag{2.16}$$

Algorithm 1 Online Residual Learning

```
*register camera frame to robot base frame*
**while** $\mathbf{e} > \mathbf{e}_{threshold}$ **do**
    *measure and estimate robot states,* $\hat{\mathbf{x}}_{E,t}$ *and* $\mathbf{q}_t$
    **if** $\hat{\mathcal{J}}^p$ *is* **not** *initialized* **then**
        *initialize* $\hat{\mathcal{J}}^p$ *with Robot* $\mathcal{J}$
    **else**
        *update* $\hat{\mathcal{J}}^p$ *with update equation* (2.16)
    *replace* $\hat{\mathcal{J}}$ *with* $\hat{\mathcal{J}}^p$ *in Eq.* (2.7) *and calculate* $\dot{\mathbf{q}}$
    *control the robot with* $\dot{\mathbf{q}}$ *in joint space*
    $\hat{\mathbf{x}}_{E,t-1} \leftarrow \hat{\mathbf{x}}_{E,t}$ *and* $\mathbf{q}_{t-1} \leftarrow \mathbf{q}_t$
```

This update equation locally estimates PJ by using the proposed residual which takes the minimal step between the updates. The ORL algorithm is summarized in Algorithm 1. The desired joint velocities are calculated by replacing $\hat{\mathcal{J}}$ with $\hat{\mathcal{J}}_t^p$ and solving the optimization in Eq. (2.7) as:

$$\dot{\mathbf{q}} = \hat{\mathcal{J}}^{p^+}\,{}^{\mathrm{B}}\mathbf{e}_{\mathrm{E}} \tag{2.17}$$

where $\hat{\mathcal{J}}^{p^+}$ is the pseudo-inverse of $\hat{\mathcal{J}}^p$.

2) Convergence: To prove the asymptotic stability of the robot motion error under the proposed controller, we use the Lyapunov stability and definite a positive-definite energy function as:

$$\mathbf{V} = \frac{1}{2}\mathbf{e}^{\intercal}\mathbf{e} \tag{2.18}$$

where **e** is the state error vector defined in Eq. (2.6). Differentiating the Eq. (2.18) with respect to time, we obtain:

$$\begin{aligned}\dot{\mathbf{V}} &= \mathbf{e}^{\intercal}\dot{\mathbf{e}} \\ &= \mathbf{e}^{\intercal}(\dot{\mathbf{x}}_{des} - \dot{\mathbf{x}})\end{aligned} \tag{2.19}$$

Using Eq. (2.17) and the definition of Jacobian in Eq. (2.8), we get

$$\dot{\mathbf{x}} = \hat{\mathcal{J}}^p\hat{\mathcal{J}}^{p^+}\mathbf{e} \tag{2.20}$$

In Eq. (2.19), if the robot is controlled to static points, $\dot{\mathbf{x}}_{des}$ term is zero. With this assumption and substituting Eq. (2.20) into the Eq. (2.19), we get

$$\dot{\mathbf{V}} = -\mathbf{e}^{\intercal}\hat{\mathcal{J}}^{p}\hat{\mathcal{J}}^{p^{+}}\mathbf{e} \leq 0 \tag{2.21}$$

The equality holds when $\mathbf{x}_{des}$ and $\mathbf{x}$ match. From this result, we conclude the asymptotic stability of error, ($\mathbf{e} \rightarrow \mathbf{0}$), under the ORL controller.

2.2.4 Grasper Tracking

To provide real time visual feedback of the robot end-effector, a grasper with four colored markers is designed and 3D-printed. The markers are placed at known locations; two are colored as blue and the other two as green. For image processing and detection of the markers, HSV (Hue, Saturation, and Value) color segmentation of the markers is performed on the rectified images and the center of each markers is selected as their pixel positions. We denote the x, and y center position in the left and right camera images as: $[{}^{I}m^{L}_{x,i}, {}^{I}m^{L}_{y,i}, {}^{I}m^{R}_{x,i}, {}^{I}m^{R}_{y,i}], i \in \{1, 2, 3, 4\}$, where i represents one of the four markers and I denotes image space. The projection matrix $\boldsymbol{Q}$ is constructed with intrinsic parameters of stereo camera as:

$$\boldsymbol{Q} = \begin{bmatrix} 1 & 0 & 0 & -c^{L}_{x} \\ 0 & 1 & 0 & -c^{L}_{y} \\ 0 & 0 & 0 & f \\ 0 & 0 & \frac{-1}{T_x} & \frac{c^{L}_{x}-c^{R}_{x}}{T_x} \end{bmatrix} \tag{2.22}$$

where c^{L}_{x} and c^{L}_{y} are the x and y components of the principal point for the left camera. Similarly, c^{R}_{x} and c^{R}_{y} are the x and y components of the principal point for the right camera. f is the focal length, and T_x is the x component of the translation between stereo images. This projection matrix is used to calculate the 3D position of the i^{th} marker, ${}^{c}\mathbf{m}_i$, with:

$$\mathbf{M}_i = \begin{bmatrix} M_{x,i} \\ M_{y,i} \\ M_{z,i} \\ M_{w,i} \end{bmatrix} = \boldsymbol{Q} \begin{bmatrix} {}^{I}m^{L}_{x,i} \\ {}^{I}m^{L}_{y,i} \\ {}^{I}m^{L}_{x,i} - {}^{I}m^{R}_{x,i} \\ 1 \end{bmatrix} \tag{2.23}$$
$${}^{c}\mathbf{m}_i = [M_{x,i}/M_{w,i}, M_{y,i}/M_{w,i}, M_{z,i}/M_{w,i}]^{\intercal}$$

In order to update PJ matrix continuously, even when some of the markers are not observed by the camera, and to reduce the effects of the noise, a Kalman filter is exploited to estimate the 3D position of the markers. To this end, we define a state vector of the filter composed of the x, y, z positions and velocities of the markers as $\boldsymbol{\eta} = [{}^{c}\mathbf{m}_{x,1}, {}^{c}\dot{\mathbf{m}}_{x,1}, {}^{c}\mathbf{m}_{y,1}, {}^{c}\dot{\mathbf{m}}_{y,1}, \ldots, {}^{c}\mathbf{m}_{z,4}, {}^{c}\dot{\mathbf{m}}_{z,4}]^{\intercal} \in \mathbb{R}^{24}$. The discrete-time dynamics and measurement equations of the filter are:

$$\begin{aligned} \boldsymbol{\eta}_{k+1} &= (\mathcal{H} \otimes \mathbf{I}_3 \otimes \mathbf{I}_4)\boldsymbol{\eta}_k + \mathbf{w}_k \\ \mathbf{y}_k &= (\boldsymbol{\Phi} \otimes \mathbf{I}_4)\boldsymbol{\eta}_k + \mathbf{v}_k \end{aligned} \tag{2.24}$$

In Eq. (2.24), $\otimes$ is the Kronecker matrix product operator, k denotes the time instance, and $\mathbf{y}_k \in \mathbb{R}^{12}$ is the measurement vector of 3D positions of the markers. Moreover, the $\mathbf{w}_k \in \mathbb{R}^{24}$ and $\mathbf{v}_k \in \mathbb{R}^{12}$ denote the system noise and measurement noise vectors assumed to be additive white noise with zero mean and covariance $\mathbf{W}_k$ and $\mathbf{V}_k$ respectively (i.e., $\mathbf{w}_k \sim N(0, \mathbf{W}_k)$, $\mathbf{v}_k \sim N(0, \mathbf{V}_k)$). The $\mathcal{H}$ and $\boldsymbol{\Phi}$ matrices are defined as:

$$\mathcal{H} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \boldsymbol{\Phi} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix} \tag{2.25}$$

where Δt is time period of filter, which is 33 ms and identical to the update rate of the stereo camera. Using these equations and the formulated Kalman filter algorithm [49], the best estimate of the state vector of the marker positions is obtained at each time step.

With the estimated positions of markers $[{}^{c}\hat{\mathbf{m}}_1, {}^{c}\hat{\mathbf{m}}_2, {}^{c}\hat{\mathbf{m}}_3, {}^{c}\hat{\mathbf{m}}_4]$, we can formulate a constrained optimization problem to find an estimated pose of the grasper in the camera frame (${}^{c}\hat{\mathbf{R}}_{Gr}$) as:

$$\begin{aligned} {}^{c}\hat{\mathbf{R}}_{Gr} &= \arg\min_{\mathbf{R}} \left\|\mathbf{R} - {}^{c}\hat{\mathbf{R}}\right\|_{F} \\ & s.t.\ \mathbf{R}\mathbf{R}^{\intercal} = \mathbf{I}_3 \end{aligned} \tag{2.26}$$

where ${}^{c}\hat{\mathbf{R}}$ is the measured rotation matrix of the grasper in the camera frame as:

$${}^{c}\hat{\mathbf{R}} = \begin{bmatrix} \frac{{}^{c}\hat{\mathbf{m}}_4 - {}^{c}\hat{\mathbf{m}}_2}{\|{}^{c}\hat{\mathbf{m}}_4 - {}^{c}\hat{\mathbf{m}}_2\|} & \frac{{}^{c}\hat{\mathbf{m}}_3 - {}^{c}\hat{\mathbf{m}}_2}{\|{}^{c}\hat{\mathbf{m}}_3 - {}^{c}\hat{\mathbf{m}}_2\|} & {}^{c}\hat{\mathbf{z}}_{Gr} \end{bmatrix} \tag{2.27}$$

Note that Eq. (2.26) is formulated because the ${}^{c}\hat{\mathbf{R}}$ might not be orthogonal due to the measurement noise. In Eq. (2.27), ${}^{c}\hat{\mathbf{z}}_{Gr}$ is the normal vector of the grasper plane which is obtained with the following optimization:

$$\begin{aligned} {}^{c}\hat{\mathbf{z}}_{Gr} &= \arg\min_{\mathbf{z}} \left\| \begin{bmatrix} {}^{c}\hat{\mathbf{m}}_1^{\intercal} & 1 \\ {}^{c}\hat{\mathbf{m}}_2^{\intercal} & 1 \\ {}^{c}\hat{\mathbf{m}}_3^{\intercal} & 1 \\ {}^{c}\hat{\mathbf{m}}_4^{\intercal} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ -1 \end{bmatrix} \right\|_2 \\ s.t.\ & \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \mathbf{z} < 0 \\ & \|\mathbf{z}\|_2 = 1 \end{aligned} \tag{2.28}$$

The first constraint is to ensure that the normal vector points towards the camera while the second constraint assures that it is normalized. With the results of the optimizations in Eq. (2.26) and Eq. (2.28), and by assigning ${}^{c}\hat{\mathbf{p}}_{Gr} = {}^{c}\hat{\mathbf{m}}_4$, the pose vector of the grasper in the camera frame is calculated as:

$${}^{c}\hat{\mathbf{x}}_{Gr} = \zeta({}^{c}\hat{\mathbf{T}}_{Gr}) = \zeta\left(\begin{bmatrix} {}^{c}\hat{\mathbf{R}}_{Gr} & {}^{c}\hat{\mathbf{p}}_{Gr} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}\right) \tag{2.29}$$

2.2.5 Hand-Eye Calibration

The constant rigid transformation between the stereo camera frame and the robot base frame, denoted by ${}^{C}\mathbf{T}_{B}$, is estimated as follows. First, the estimated grasper pose in the camera frame

${}^{C}\hat{\mathbf{T}}_{Gr}$ (calculated in Sec. 2.2.4) and the known ${}^{Gr}\mathbf{T}_{E}$ (from the CAD model) are used to calculate ${}^{C}\hat{\mathbf{T}}_{E}$. Second, a constrained optimization is formulated with the estimated ${}^{C}\hat{\mathbf{T}}_{E}$ and the forward kinematics ${}^{B}\mathbf{T}_{E}$. To reduce the noise effects, m number of measurements are exploited in the optimization as:

$$\begin{gathered}{}^{C}\hat{\mathbf{T}}_{B} = \begin{bmatrix} \mathbf{R}^{*} & \mathbf{p}^{*} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} = \underset{\{\mathbf{R},\mathbf{p}\}}{\arg\min}\ \frac{1}{m}\sum_{i=1}^{m} \left\| \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} {}^{B}\mathbf{T}_{E,j} - {}^{C}\hat{\mathbf{T}}_{E,j} \right\|_{F} \\ s.t.\ \mathbf{R}\mathbf{R}^{\intercal} = \mathbf{I}_{3} \end{gathered} \tag{2.30}$$

where j denotes the j^{th} measurement sample. This optimization is solved using the method described in [50].

2.3 Experiment

To verify the efficacy of the proposed online residual learning controller, we designed and implemented three sets of experiments. In *Experiment I*, we examined the convergence of the algorithm for different sets of initial and desired configurations. In *Experiment II*, the robustness of the algorithm to hand-eye calibration errors was assessed. In *Experiment III*, the performance of the ORL controller for autonomous operation of two exemplary surgical subtasks, Needle Transfer (NT) and Tissue Debridement (TD), was tested. These tasks were chosen because they are frequently performed in surgery and require high control precision for a successful completion. All experiments were performed using the cable-driven Raven IV surgical robotic system along with a stereo vision system. We evaluated the performance of the system based on the repeatability and convergence of the algorithm (Experiment I). Furthermore, we assessed the robustness of the algorithm from the number of successful convergence under various levels of hand-eye calibration errors (Experiment II). Finally, we demonstrated successful completion of automated surgical subtasks using the proposed controller across a wide range of robot workspace (Experiment III).

2.3.1 Experiment Setup

2.3.1.1 Robotic system

We deployed a Raven IV open platform surgical robotic system [3] in our experiments as shown in Fig. 2.1. Raven IV has two pairs of robotic arms to allow collaborative operations in surgery. In our experiments, however, only one pair of arms each equipped with a da Vinci needle driver are exploited. Each arm has 7 DoF consisting of a 3 DoF spherical positioning mechanism and a 4 DoF instrument (3 DoF articulated wrist + 1 DoF grip). To enable the vision feedback, a stereo camera (Blackfly-BFLY-U3-13S2C, Point Grey Research) with 30 Hz frame rate and 644×482 pixels image resolution is deployed.

2.3.1.2 Grasper

To enable a 6 DoF vision feedback of the robot end-effector, we designed a grasper which can be installed on the instrument jaws (see Fig. 2.4). The grasper has four colored markers to provide a robust pose estimation of the end-effector based on the camera measurements and the grasper's CAD model. Furthermore, the structure of the grasper was designed so that the suture needle can be securely grasped by the robot end-effector. This secure grasp is essential to obtain an accurate needle pose estimation during NT experiments which will be further explained in Experiment III.

2.3.2 Experiment Design and Procedure

2.3.2.1 Experiment I (Convergence)

We mathematically proved the asymptotic stability of the end-effector pose error under the proposed controller (see section 2.2.3). To confirm these results experimentally, we tested the step response of algorithm on the Raven IV system. In addition, the experiments were performed for three different pairs of initial and desired robot configurations. These configurations were randomly sampled from the robot's workspace. This demonstrates that the convergence is agnostic to a particular robot configuration or certain motion trajectories. Moreover, the repeatability of the algorithm was verified by performing three trials for each initial and desired configuration pair.

2.3.2.2 Experiment II (Robustness to Hand-Eye Calibration)

One major caveat of the model-based visual servo controllers (i.e., CVS) is high sensitivity to calibration parameters [46]. Of note, the hand-eye calibration may become corrupted during surgery because of frequent endoscope motion or system fault. As a result, in the context of automation in surgical robotics, the model-based visual servoing can lead to undesirable robot motion under inaccurate or corrupted calibration which puts the patient at high risk. As mentioned previously, the ORL algorithm has higher robustness to hand-eye calibration errors compared to the model-based controllers. To confirm this experimentally, four sets of step response experiments with various levels of incorrect hand-eye calibration information were designed. In the experiment set k ($k = 1, 2, ..., 4$), the hand-eye calibration (i.e., robot-camera registration matrix) was corrupted so that all axes are rotated $\alpha_{rot} = 10 \times k$ degrees with respect to the fixed true calibration. We refer to α_{rot} as *calibration error*. For example, in the experiment set 3, all axes are rotated $\alpha_{rot} = 30$ degrees. For each level of the calibration error, the algorithm was tested for three randomly selected initial and desired robot configuration pairs. For each pair, three trials were performed as well. As a result, a total of 36 (4 calibration errors $\times$ 3 configurations $\times$ 3 trials) trials were conducted in the Experiment II.

2.3.2.3 Experiment III (Clinical Application)

To demonstrate the applications of the proposed controller in real surgical scenarios, automation of two surgical subtasks (NT and TD) is considered. NT serves various purposes in the suturing task such as reorienting the needle or placing the needle in an appropriate arm. TD is a tedious surgical subtask in which dead tissue is removed to help the recovery of the remaining healthy part. Both subtasks are good candidates for automation since they are performed frequently during soft tissue surgeries. The main challenge, however, is that automation of these subtasks requires high accuracy control strategies. For example, autonomous grasping of the suture needle with sub-millimeter thickness at appropriate position and angle may not be feasible using the current cable-driven surgical robots with inaccurate kinematic controller.

To assess the performance of our algorithm in this context and across the entire robot workspace, 5 sets of experiments for each NT and TD were performed. For NT experiments, the ORL controller was deployed for autonomous grasp and transfer of the needle at five different configurations. Grasping the needle was performed in two steps by first controlling the robot to an *intermediate point* which has the same orientation as the *grasping point* but is located at a different position (see Fig. 2.10 for illustration of the points). In the second step, the robot was translated to the grasping point to complete the task. Of note, this two-stage needle grasping was crucial to avoid undesirable collisions between the robot arm and the needle. The orientation of the grasping point is selected in such a way that the end-effector is aligned with the tangent line of the needle at the grasping point (see Fig. 2.10). The position and orientation of the grasping point on the needle is inferred from (i) the grasper information from the vision and (ii) the kinematic relation between the needle and the grasper.

For TD experiments, the tissue phantom was placed at five random locations and the robot was guided to grasp, move, and drop the tissue pieces. The location of the tissue phantom and the desired drop point were obtained with a computer vision algorithm as shown in Fig. 2.12a.

2.4 Results

2.4.0.1 Experiment I (Convergence)

The robot was commanded to be positioned at three desired configurations using the proposed algorithm. Fig. 2.5 shows a robot trajectory for one of the configurations. For each desired configuration, the experiment was repeated three times to assess the repeatability of the performance and

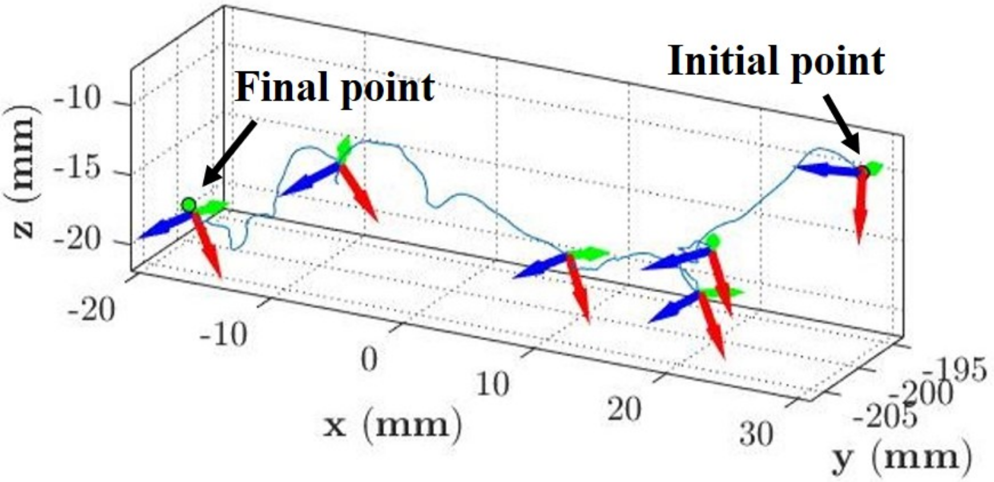


Figure 2.5: A representative robot trajectory in Experiment I. The light blue line depicts the robot position and the coordinates show the robot orientation along the trajectory.

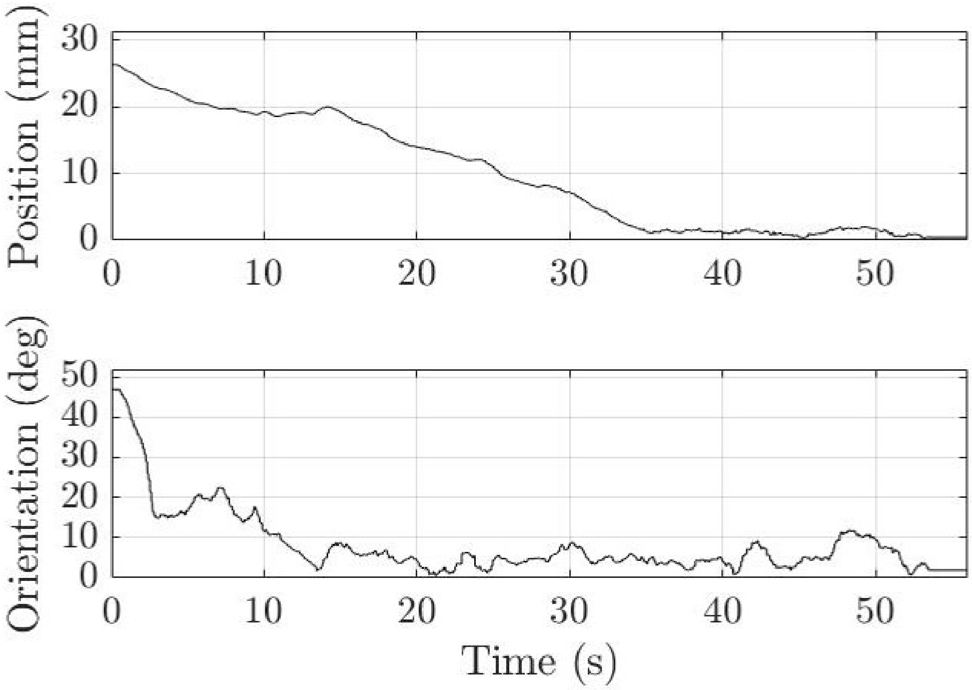


Figure 2.6: The position and orientation errors of an exemplary trial in Experiment I.

all converged successfully. The root mean square errors (RMSEs) of the position and orientation is shown in Fig. 2.6. As it can be seen, both position and orientation errors asymptotically converged. To compare PJ from the proposed ORL algorithm with RJ from the robot, the Frobenius norm of changes of the two along a given trajectory are plotted in Fig. 2.7. Of note, the ORL algorithm updates PJ matrix with minimal changes in Frobenius sense. As it can be seen, the changes in PJ is one order of magnitude smaller than the actual changes in RJ.

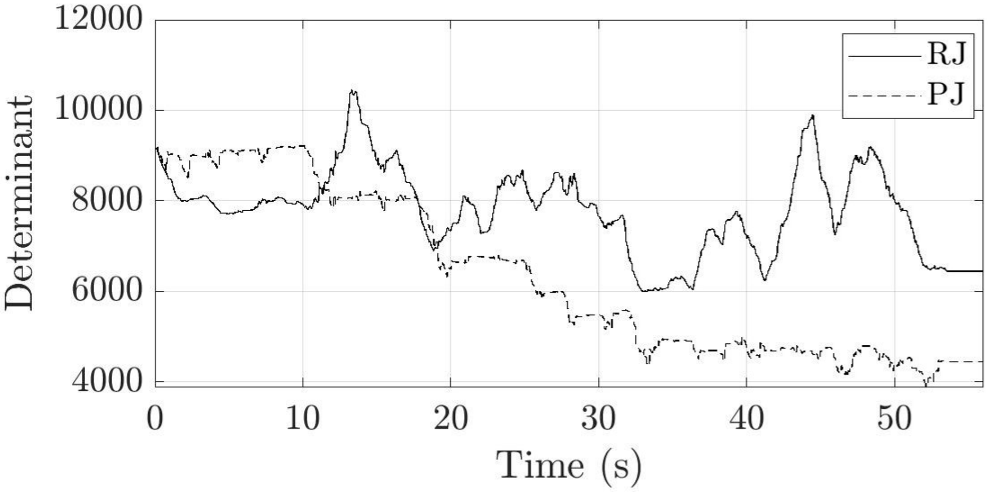
2.4.0.2 Experiment II (Robustness to Hand-Eye Calibration)

The robot was controlled to three different configurations under two control strategies: (i) the ORL algorithm and (ii) a model-based CVS controller [51]. Of note, the model-based visual servo schemes suffer from high sensitivity to calibration parameters and hence are used here as a benchmark for robustness measure. Four levels of hand-eye calibration errors $\in \{10^\circ, 20^\circ, 30^\circ, 40^\circ\}$ were used to systematically corrupt the true hand-eye calibration. To confirm the repeatability of the results, each experiment was repeated three times for each configuration and each controller. Table 2.1 summarizes the success rate of the convergence ($\frac{\text{\# of successful convergence}}{\text{\# of trials}}$) for each controller.

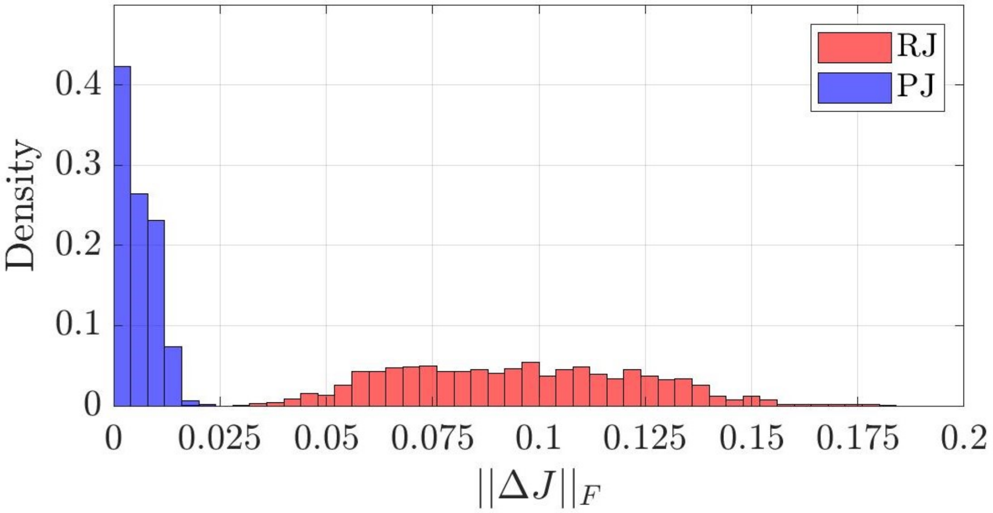
Furthermore, Fig. 2.8 depicts the determinant of PJ along all trajectories for different configurations and various calibration errors. Fig. 2.9 shows the RMSEs of the position and orientation for the model-based and the ORL controllers in presence of 20 degrees calibration error.

2.4.0.3 Experiment III (Clinical Application)

The ORL controller was deployed for both NT and TD experiments. The NT experiments were performed for 5 different configurations of the needle. Fig. 2.10 and Fig. 2.11 demonstrate snapshots of the experiments for all five configurations. Note that for brevity purposes, only 3 snapshots are shown for each configuration in Fig. 2.11. The TD experiments were performed for 5 different initial locations of the tissue phantom. Fig. 2.12 and Fig. 2.13 demonstrate snapshots of the experiments for all five locations. Three snapshots for each configuration are shown in Fig. 2.13.

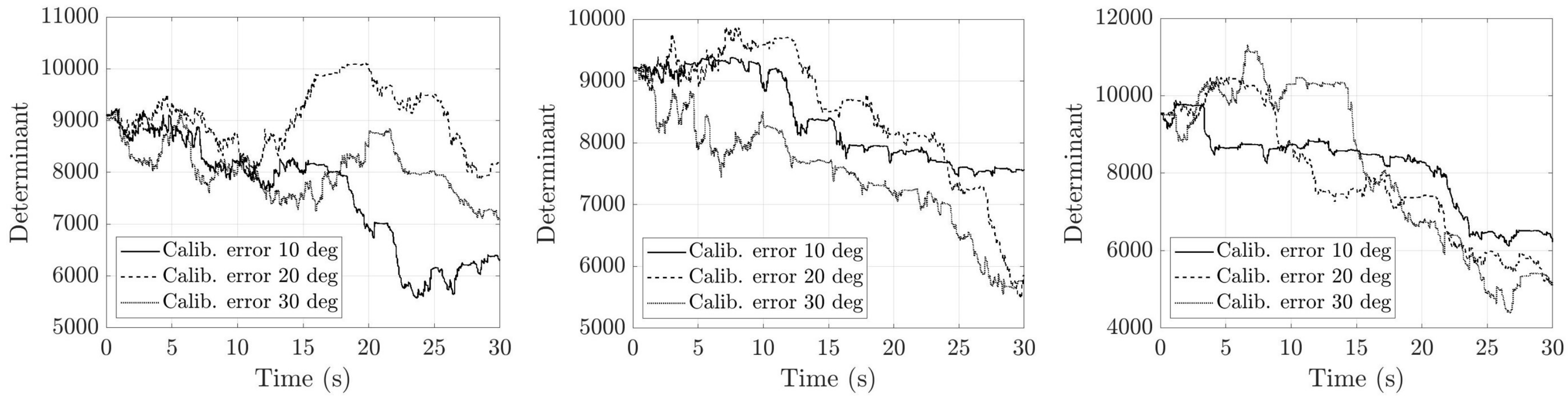


(a)



(b)

Figure 2.7: (a) RJ and PJ determinants (b) histogram of temporal variations in RJ and PJ. Both figures are for a trajectory in Experiment I.



(a)

(b)

(c)

Figure 2.8: Determinant of PJ with various hand-eye calibration errors for (a) Config. 1 (b) Config. 2 (c) Config. 3, in Exp. II.

Table 2.1: The success rate of the convergence for the model-based and the ORL controllers under various calibration errors.

Calib. error	Controller	Config. 1	Config. 2	Config. 3
10 deg	Model-based	0/3	3/3	0/3
	ORL	3/3	3/3	3/3
20 deg	Model-based	0/3	0/3	0/3
	ORL	3/3	3/3	3/3
30 deg	Model-based	0/3	0/3	0/3
	ORL	3/3	3/3	3/3
40 deg	Model-based	0/3	0/3	0/3
	ORL	0/3	0/3	3/3

2.5 Discussion

The performance of the proposed ORL algorithm is evaluated based on the error convergence, robustness to calibration errors, and successful completion of autonomous surgical subtasks. A Lyapunov function was used to show the convergence of the algorithm theoretically. For experimental evaluation, the tracking results of a step reference signal for three different configurations each with three trials illustrated the convergence of the algorithm. Fig. 2.5 and Fig. 2.6 show a robot trajectory and RMSE signals (position and orientation) of the robot under the ORL controller. As it can be seen, the error signals asymptotically converge to zero. It should be re-emphasized that this 6 DoF positioning task is very challenging for cable-driven robots due to unknown cable coupling, friction, and hard-to-model transmission kinematics which results in inaccurate joints angle estimates. Nevertheless, the proposed ORL controller is able to converge successfully. Of note, we tested the algorithm for different configurations across the robot's workspace to show that the convergence is configuration agnostic. Moreover, the fact that the error converged to zero for all three trials of a given robot initial and desired configuration confirms that these results are

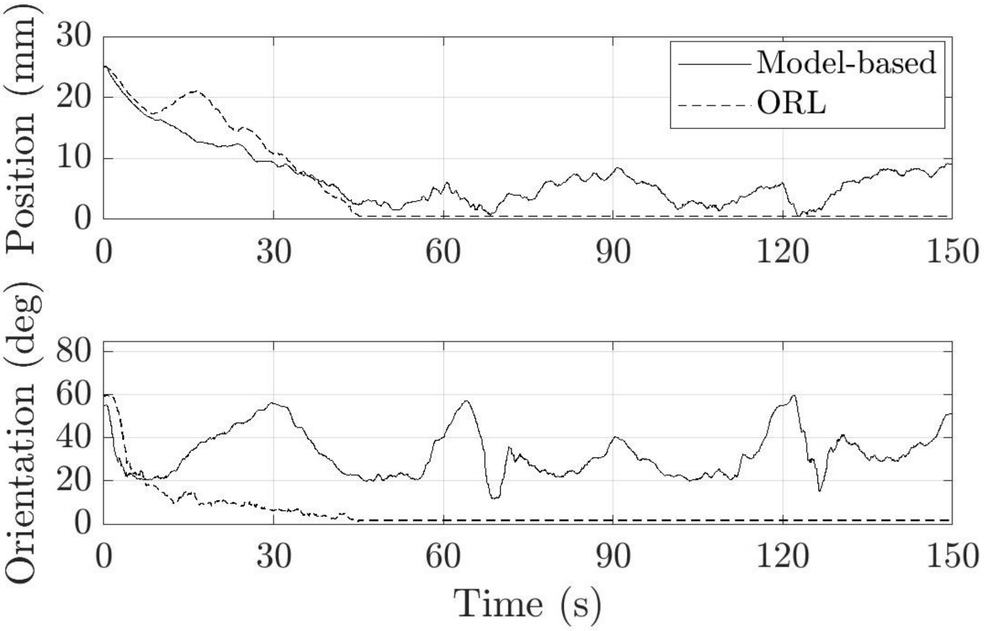
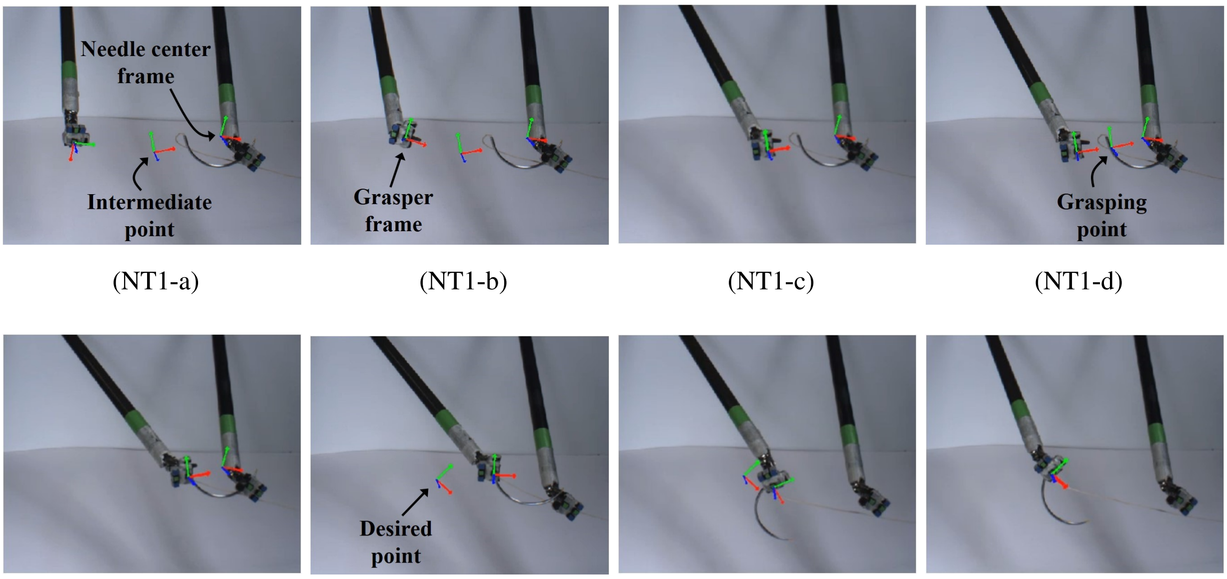


Figure 2.9: RMSEs of position and orientation for the model-based and the ORL controllers for 20 degrees calibration error.

repeatable. Additionally, the calculated RMSEs of position and orientation are 0.38 mm and 1.64 degree across all trials. This result, in fact, is a drastic improvement for the Raven IV internal kinematic controller with position and orientation errors of 26 mm and 20.6 degrees respectively [28]. Based on these results, it is concluded that the proposed ORL controller achieved excellent accuracy performance and improved the control accuracy by factors of 50 for position and 10 for orientation.

Fig. 2.7a compares the determinants of PJ against RJ along a sample robot trajectory. Of note, it should be re-emphasized that the Jacobian from the proposed controller (i.e., PJ) is not necessarily the true robot Jacobian. Rather, it is a locally linearized mapping (joint angles to robot end-effector) which ensures asymptotic convergence of the robot motion. For example, as shown in Fig. 2.7, the determinant of PJ and RJ are different along the trajectory. Therefore, while PJ might not have physical interpretation nor estimate the true Jacobian, it successfully drives the system to desired configurations. Moreover, Fig. 2.7b depicts the normalized histogram of the Frobenius



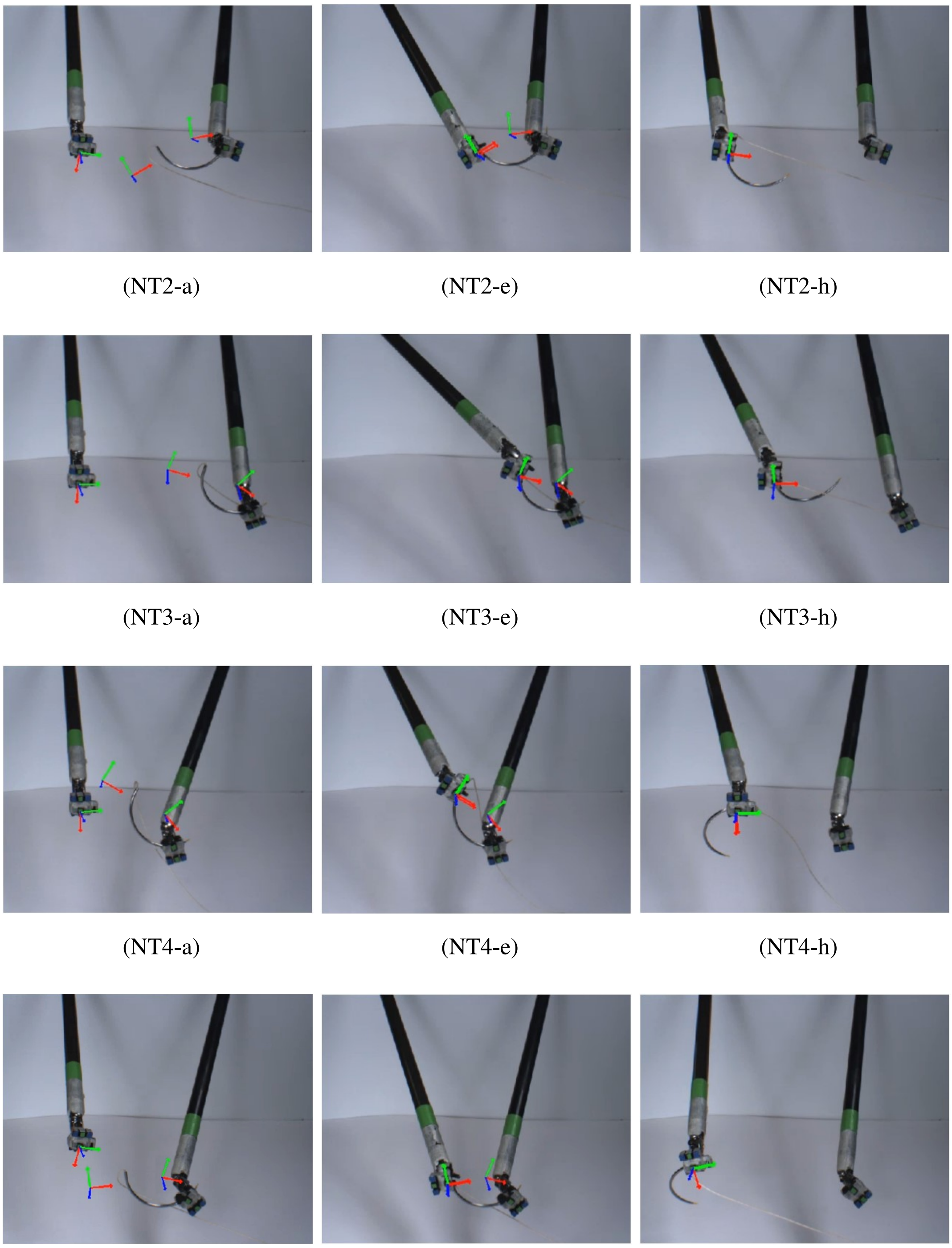
(NT1-a) (NT1-b) (NT1-c) (NT1-d)

(NT1-e) (NT1-f) (NT1-g) (NT1-h)

Figure 2.10: Representative snapshots of the needle transfer experiment. (NT1-a) initial configuration, (NT1-b) moving towards the intermediate point, (NT1-c) reaching the intermediate point, (NT1-d) setting the grasping point, (NT1-e) reaching the grasping point and closing the gripper of the left arm, (NT1-f) releasing the needle by the right arm and setting the left arm's desired point, (NT1-g) moving towards the desired point, (NT1-h) reaching the desired point with the grasped needle.

norm of ΔJ for both PJ and RJ. As it can be seen, the $||\Delta J||_F$ of PJ is much smaller than the $||\Delta J||_F$ of RJ (average ratio $\frac{\left\|\Delta\hat{\mathcal{J}}^p\right\|_F}{\left\|\Delta\hat{\mathcal{J}}\right\|_F} = 0.06$). Hence, the experiment results are inline with the theoretical derivation of the ORL update function, Eq. (2.15), where the $||\Delta J||_F$ is minimized at each iteration.

The ORL algorithm initializes PJ with RJ. This can be seen in Fig. 2.7a, as the determinants of both PJ and RJ start with the same value. This initialization provides an advantage over other learning-based controllers such as UVS whose performance is highly dependent on a particular choice of initialization. For example, initial coarse estimation of the Jacobian in UVS may result in instability, particularly at the beginning of the servoing [52]. Such undesirable motion behavior was not observed for the ORL algorithm as there was no need to find a good initial estimate



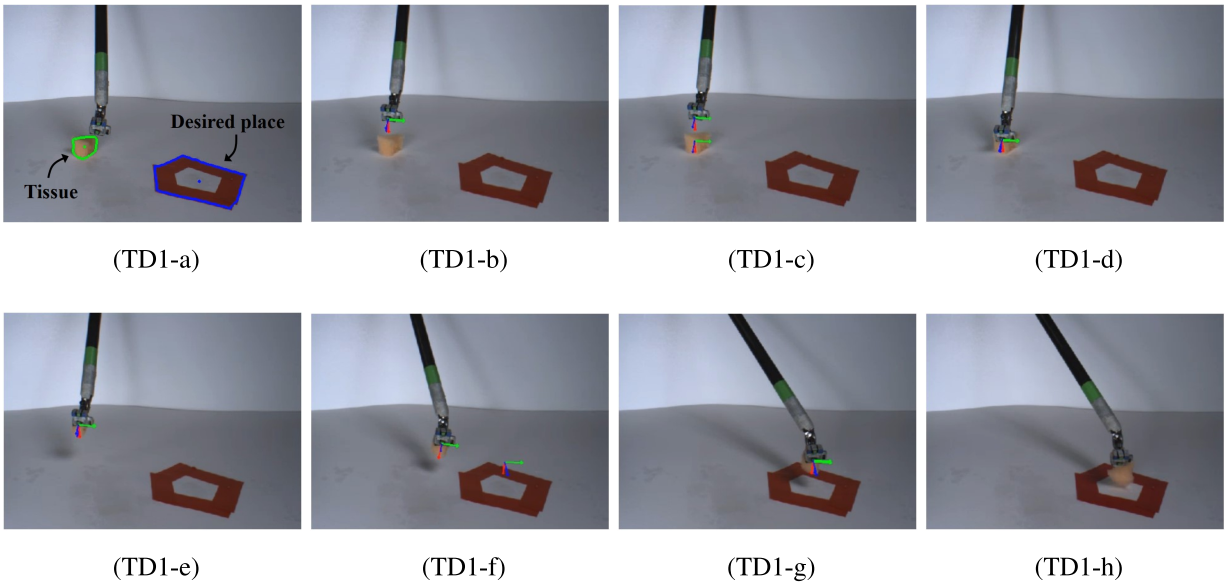
(NT2-a) (NT2-e) (NT2-h)

(NT3-a) (NT3-e) (NT3-h)

(NT4-a) (NT4-e) (NT4-h)

(NT5-a) (NT5-e) (NT5-h)

Figure 2.11: Illustration of needle transfer experiments with four different needle configurations.



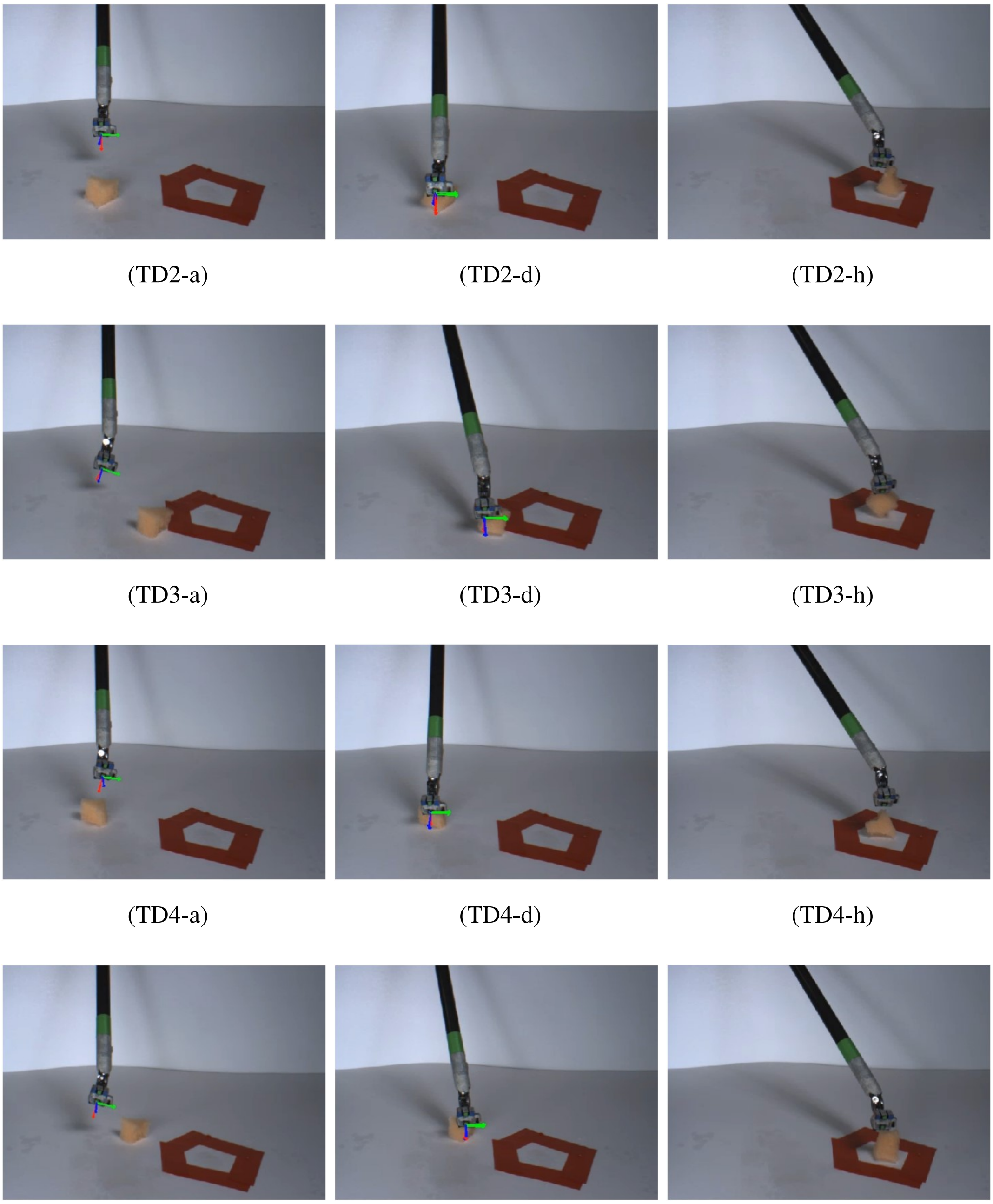
(TD1-a) (TD1-b) (TD1-c) (TD1-d)

(TD1-e) (TD1-f) (TD1-g) (TD1-h)

Figure 2.12: Representative snapshots of the tissue debridement experiment. (TD1-a) initial configuration, (TD1-b) reaching the intermediate point, (TD1-c) setting the desired point on the tissue, (TD1-d) reaching the tissue and closing the gripper, (TD1-e) lifting the tissue, (TD1-f) moving towards the desired point, (TD1-g) reaching the desired point, (TD1-h) dropping the tissue.

of PJ. As a result, a notable power of the ORL controller is its ability to combine the model-based approach (i.e., initialization to RJ) with the learning-based approach (i.e., real-time residual learning).

One measure to evaluate the robustness of the controllers to hand-eye calibration errors is obtaining the number of trials in which the model-based and ORL algorithms converged successfully. This data was obtained in Experiment II and is reported in Table 2.1. As can be seen, the model-based controller converged for only one configuration (out of three) when the calibration error was 10 degrees and did not converge for larger calibration errors. On the other hand, the ORL algorithm converged for all configurations up to calibration errors of 30 degrees and for one configuration at 40 degrees. An exemplary convergence results (i.e., 20 degrees calibration error) of the model-based and ORL controller are shown in Fig. 2.9. As it can be seen, the ORL controller asymptotically converged while the model-based oscillated around the desired configu-



(TD2-a) (TD2-d) (TD2-h)

(TD3-a) (TD3-d) (TD3-h)

(TD4-a) (TD4-d) (TD4-h)

(TD5-a) (TD5-d) (TD5-h)

Figure 2.13: Illustration of tissue debridement experiments with four different initial configurations.

ration. Moreover, Fig. 2.8 demonstrates variations of PJ for different levels of error and shows that the algorithm adaptively changes PJ to compensate for calibration errors. These results confirm the superior performance of the proposed controller and validate its robustness to calibration errors up to 30 degrees. Of note, accurate hand-eye calibration information is cumbersome to obtain and is susceptible to corruption during surgery due to environment disturbances. The fact that the CVS controller is not robust to the smallest calibration error (10 degrees) may suggest that accurate hand-eye calibration is even more crucial for cable-driven systems as the kinematics models are already inaccurate. This further highlights the potential of the ORL controller as a safer and more reliable solution for the vision-based autonomous control of surgical manipulators.

The repeatability of the ORL convergence and robustness is shown by performing three trials for each configuration in experiments I and II. Moreover, it is very important to test autonomous controllers across different configurations for the cable-driven surgical manipulators. This is because the nonlinearities such as cable tensioning and coupling as well as friction are configuration dependent [26]. In other words, some trajectories or desired points might be more difficult to achieve compared to other ones. This was the main motivation for evaluating the ORL controller for three configurations selected from different parts of the state space.

The clinical applications of the ORL algorithm were shown in the experiment III. NT and TD subtasks were chosen as they occur frequently during soft tissue surgeries and their autonomous implementation requires an accurate robot controller. Our results confirmed that the ORL algorithm enabled the Raven IV cable-driven robot to successfully perform both tasks across a wide range of robot workspace.

2.6 Conclusion

In this chapter, we propose a novel control algorithm based on online residual learning scheme to improve the control accuracy and robustness of cable-driven surgical robotic systems. The ORL algorithm is a data-driven approach in which a linearized mapping between the robot end-effector and joint angles is updated on-the-fly based on the difference between the predicted and actual motion of the end-effector. This real-time adaptation results in an improved accuracy and robustness of the controller. The proposed algorithm combines the model-based approach with the learning-based approach constituting a hybrid method which is sample efficient and robust to modeling errors. In fact, such framework allows the ORL controller to compensate for various errors which can run the CVS controller imprecise or diverging. One example of such error is kinematic nonlinearities such as cable tensioning which are difficult to model and might change during an operation [26]. Another example includes hand-eye calibration which might be inaccurate or become corrupted during surgery. We assessed error convergence, position accuracy, and robustness to calibration errors of the algorithm using the Raven IV surgical system. The results indicate that the ORL controller improves the accuracy of the Raven IV kinematic controller by factors of 50 and 10 for position and orientation respectively. Furthermore, the results demonstrate that the ORL has robustness of up to 30 degrees to calibration errors. Lastly, we show successful implementation of the ORL algorithm for autonomous operation of two surgical subtasks including needle transfer and tissue debridement.

CHAPTER 3

Autonomous Tissue Manipulation Using Learning-Based Model Predictive Control

A supplementary video can be found at: *http://bionics.seas.ucla.edu/research/surgeryproject17.html*

3.1 Introduction

Automation in surgical robotics is part of a vision that will redefine the role of the surgeon in the operating room. It will shift the surgeons toward the decision making role while the vast majority of the manipulations will be conducted via a surgical robot. As part of this vision, research is directed at automating subtasks that serve as building blocks of many of the surgical procedures such as suturing [9, 53, 54], tumor resection [55], bone cutting [56], and drilling [57]. Among the many surgical subtasks, tissue manipulation is one of the tasks that is most frequently performed. More specifically, when a surgeon wants to connect two different tissues or close an incision, both sides of the tissue should be placed with respect to each other in a way that enables homogeneous suture distance for improved healing [58]. However, tissue manipulation presents a complex dynamics and hence is particularly challenging to automate given the lack of a model which predicts its behavior [59]. Furthermore, indirect manipulation of interest points on the tissue makes it more difficult. Tissue manipulation falls under the broader research problem of deformable object manipulation.

There are in general two methods to approach the problem of tissue manipulation, namely model-based and model-free control. For model-based control method, a control law was suggested that could position a deformable object based on a spring-mass model and uncertainty [60]. In another study, a nonlinear finite element model was used to estimate the motion of soft tissue and parameters are updated using the difference between estimation and actual data [61]. In [62], a PID controller was used with a model of a deformable object. The model-based manipulation of deformable objects is well summarized in [63]. For the model-free method, real time optimization framework utilizing rank-one Jacobian update with vision feedback has been used for manipulating a kidney [64], a deformable phantom tissue [65], and soft objects [66]. Furthermore, this model-free method has been expanded to manipulate a compliant object under unknown internal and external disturbances [21]. A learned variable impedance control that trades off between force and position trajectories extracted from demonstrations is proposed for deformable objects manipulation [67]. In another study, linear actuators were controlled to apply external force to soft tissue to position a target feature while a needle is injected [68]. Lastly, robotic manipulation and grasping of deformable objects are comprehensively covered in [69].

The reported research focuses on the task of manipulating tissue to place specified points on the tissue (tissue points) at desired positions in the image frame as described in Fig. 3.1. In operating rooms, tissue points can be selected as tissue features identifiable via recognition of common patterns on the tissue. However, as part of this reported research, colored markers are used for robust tracking of the points with a vision algorithm. In the task, the tissue points are simultaneously indirectly manipulated by the robot arms grasping the tissue at manipulation points. This task is complicated by the complex dynamics between the motions of the robot arms and tissue points. This research effort proposes a learning-based model predictive control (MPC) framework to solve the dynamics and manipulate the soft tissue. Two learning approaches are compared. One is a reinforcement learning (RL) method where the robot learns the dynamics of the tissue after exploring by itself. The other is a learning from demonstration framework (LfD) that initializes the dynamics of the tissue by studying human expert demonstrations.

Compared to previous model-based approaches that require a model for each manipulated object, the algorithms proposed in this study use a simple neural network to learn the dynamics in

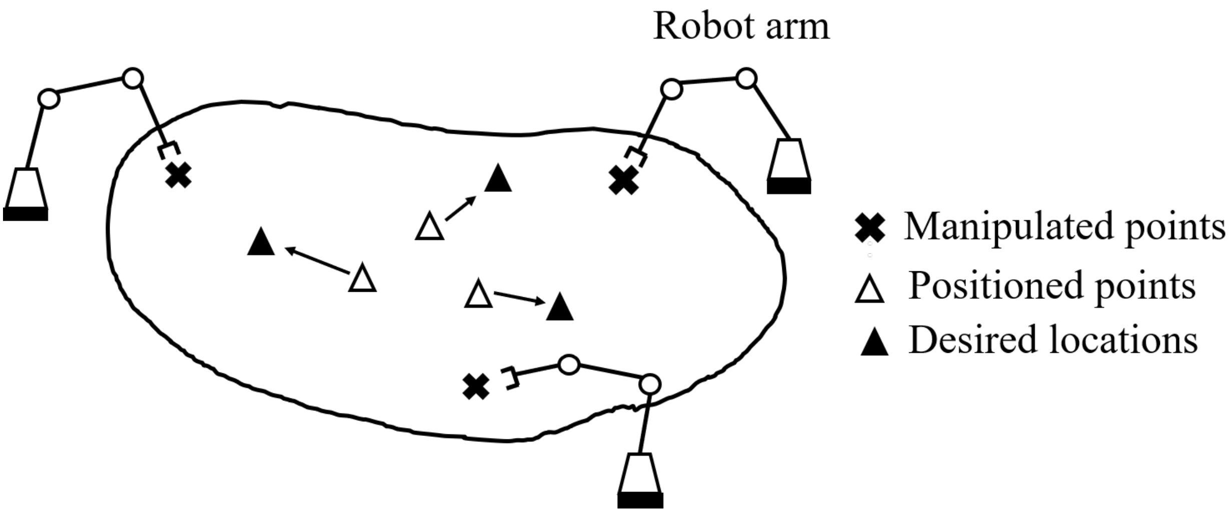


Figure 3.1: Task description of indirect soft tissue manipulation

image space by exploiting the universality of neural network with hidden layers to describe any function. This provides flexibility of the algorithm to be applicable to any object, even those with different physical properties. As opposed to the reviewed model-free approaches using linearization, the proposed algorithms directly learn nonlinear dynamics of tissue in image space and control the robot with nonlinear optimization. This allows avoidance of local optima that may occur due to the physical constraints of the environment, by predicting future steps from learned dynamics. Moreover, the proposed LfD algorithm provides a framework to incorporate human demonstrations which leads to initialization of tissue dynamics and great controller performance even in the initial learning phase. The demonstrations can be easily acquired by recording the scene of robotic tissue manipulation in teleoperation mode.

3.2 Methods

3.2.1 Algorithms

RL has shown great success in many applications where learning missing pieces, e.g. dynamics, of a task is necessary to find an optimal policy [70, 71, 72]. RL is a technique used by artificial agents or robots to learn strategies to optimize expected cumulative reward by collecting data through trial-and-error. As opposed to model-free RL, model-based RL is used in this work because of its high sample efficiency [73] which is desirable for robotic applications where collecting data with physical systems is expensive. Model-based RL updates a dynamics function with data samples collected by trial and error and has an internal controller to calculate control inputs. It optimizes a reward or cost function by applying the learned dynamics to the internal controller. In this work, model-based RL with internal MPC is used because it has been shown to successfully control robotic systems for a variety of tasks. An under-actuated legged robot was controlled in image space [74]. An inverted pendulum was controlled in image space using an MPC paired with a learned deep dynamical model that predicts future images of the system [75].

3.2.1.1 Assumptions

We developed all algorithms based on the following assumptions:

- Vision feedback of robot and tissue features is always available. The robot and tissue features are never occluded.
- The task begins after the robot grasps the manipulation points, and there is no slip between the grippers and the tissue.

3.2.1.2 Model Predictive Control

MPC is a control scheme that predicts future states by forward propagation using the current states, inputs, and dynamics equation in order to output the set of future inputs that result in optimal costs [76]. The MPC has proven its ability to control complex mechanical systems [77]. The MPC for tissue manipulation is formulated in this work with the following equations:

$$\begin{aligned}
&\underset{\{u_t,\cdots,u_{t+h}\}}{\arg\min}\left\|\vec{p}^{\,T,des}_{t+h+1} - \vec{p}^{\,T,curr}_{t+h+1}\right\|_2^2 \\
&s.t. \quad \vec{v}^{\,T}_{t+h} = f(\vec{p}^{\,T}_{t+h}, \vec{p}^{\,R}_{t+h}, \vec{u}^{\,R}_{t+h}) \\
&\qquad\quad \vec{p}^{\,T}_{t+h+1} = \vec{p}^{\,T}_{t+h} + \int_0^{\Delta t} \vec{v}^{\,T}_{t+h} dt \\
&\qquad\quad \vec{p}^{\,R}_{t+h+1} = \vec{p}^{\,R}_{t+h} + \int_0^{\Delta t} \vec{u}^{\,R}_{t+h} dt \\
&\qquad\quad h = 0, \cdots, H-1
\end{aligned} \tag{3.1}$$

where superscript T and R are used to designate the tissue points and robot wrists. f is a learned dynamics (Fig. 3.2), u is an input specifying movement of the robot in image space, Δt is control period, and H is the maximum number of steps in the time horizon. $\vec{p}^T$ and $\vec{v}^T$ are position and velocity vectors defined in image space, of all tissue points, and are each $\in R^{2*\{\#\ of\ tissue\ points\}}$. In the same manner, $\vec{p}^R \in R^{2*\{\#\ of\ robots\}}$ and $\vec{v}^R \in R^{2*\{\#\ of\ robots\}}$. The cost function is formulated to reduce the Euclidean distance between the tissue points and desired points at a time instance. If the tissue points are close to their desired positions, it is not necessary to use all inputs in the input horizon. Thus, the optimal number of inputs in the input horizon is also found in equations (3.1). As a result, the output from the MPC formulated in equations (3.1) is a set $\{u^*_t, \cdots, u^*_{t+h^*}\}$.

3.2.1.3 Adaptive MPC

Accurate modeling of dynamics is crucial for successful application of MPC. However, defining the dynamics of a complex system is challenging. In order to address this challenge, adaptive MPC that updates the dynamics using learning algorithms was suggested [78, 79]. In this work, a neural network is used to find the dynamics for the MPC. The input vector for the dynamics neural network is a vector that is composed of positions of the robot wrists, positions of the tissue points, and the control inputs for the robots. The output is the velocities of the tissue points. The structure of the neural network structure is shown in Fig. 3.2.

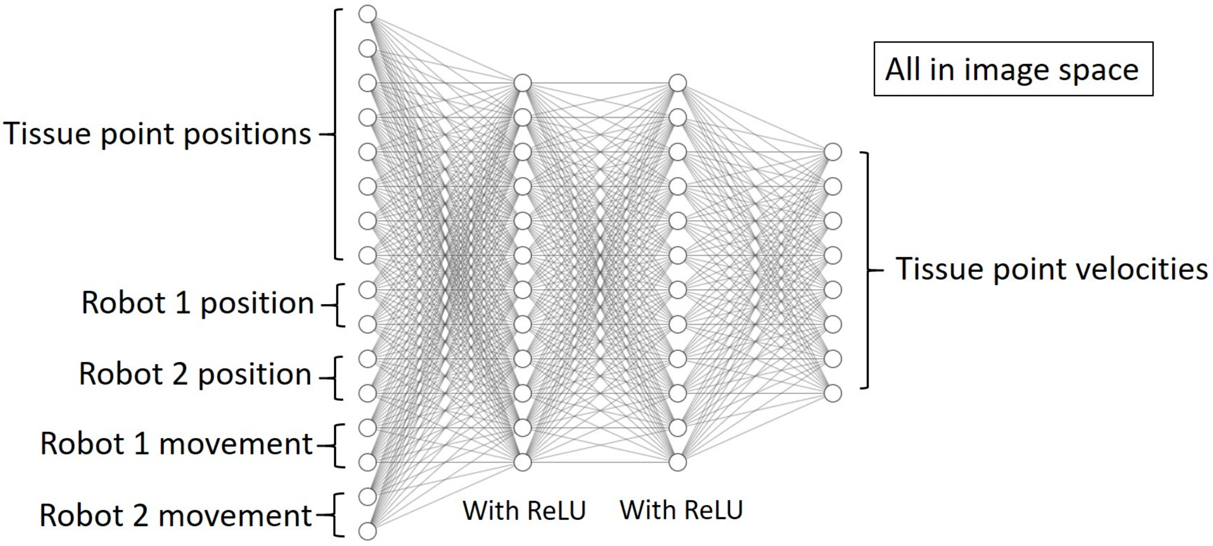


Figure 3.2: The structure of dynamics neural network

An optimal control sequence, which is an output from the equation (3.1), can be obtained in two ways: optimization by error back propagation [75] or generation of random input candidates [74]. The research presented in this study uses the latter approach for calculating the optimal number of steps in the time horizon h^* and the corresponding control sequence. After that, for each control period, the desired robot wrist positions in image space are updated with the first input in the optimal control sequence. The robot position is controlled to match the desired positions via visual servoing.

3.2.1.4 Model-based Reinforcement Learning

The reinforcement learning algorithm using model predictive control is shown in Fig. 3.3 and summarized in Algorithm 2. After initialization of the neural network variables, a computer vision algorithm extracts the positions of the robot wrists and the tissue points from an image. ϵ-greedy approach is used to force the robot to explore randomly at the beginning, but gradually optimize policy as dynamics are learned. For ϵ-greedy behavior in RL, ϵ is decreased from 1 to 0.1 linearly as a function of the number of robot actions taken. If ϵ is greater than a random number generated

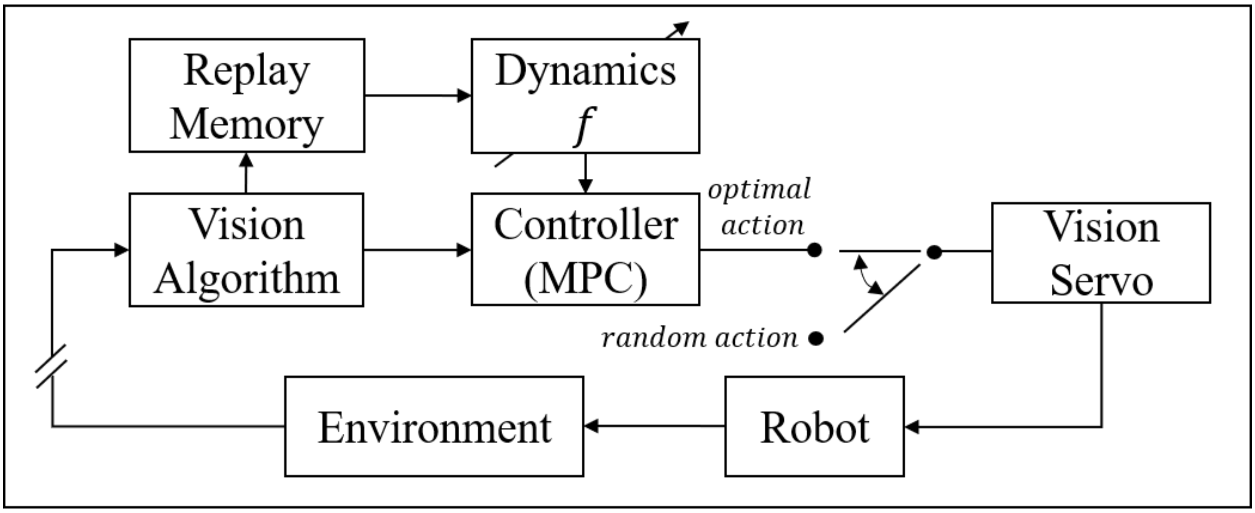


Figure 3.3: Block diagram of Reinforcement Learning.

Algorithm 2 Model-based Reinforcement Learning

```
initialize neural network variables
while action number < exploration number do
    extract $p_t^T$ and $p_t^R$ in image
    if $frame_{curr} - frame_{prev} > \Delta t$ then
        with $\epsilon$, choose optimal or random action
        update $p_{t+1}^{R,des}$ in image
        save experience to replay memory
        choose mini batch and train dynamics model
    visual servoing of robots
```

between 0 and 1, a random action is taken. Otherwise, the optimal action based on the MPC is taken. Each action set, $u \in \{[0,0],[1,0],[-1,0],[0,1],[0,-1]\}$ is multiplied by a scale factor, $step$, in image space that determines the step size of the robots in pixels. The actions in u, in order, correspond to the robot stopping, moving left, moving right, moving upward, and moving downward in image space. We have found that step size of the robots' movement and control period should be properly selected to learn meaningful information. After the action is determined, desired robot positions are updated and visual servo is performed to control the robots. In the next frame, the algorithm again obtains positions, and calculates velocities of the robot wrists and tissue points by taking the difference between previous and current positions. An experience set that contains the previous positions and velocities is saved to the replay memory. Training of the dynamics model starts after collecting more than a predefined number of experience sets. Random but fixed size sets of experience sets are selected from the replay memory and used to train the network. This process is repeated until the number of total actions reaches a predefined exploration number. After this learning period, the robot always chooses the optimal action based on the learned dynamics.

3.2.1.5 Learning from Demonstrations

Self exploration can be time-consuming and dangerous when the robot does not have any prior knowledge of the environment. As such, learning the dynamics from scratch using RL is inadvisable for tissue manipulation in clinical environments. However, if human experts such as surgeons can demonstrate the task to the robots, and if the robots can learn from these demonstrations, it would be safer. LfD is actively studied in the field of robotics because of its many strengths [80]. Furthermore, it has been shown that the learning process in model-based reinforcement learning can be accelerated from demonstrations [81]. These demonstrations could be obtained, for tissue manipulation during surgery, by surgeons recording videos that capture the screen of the teleoperation console which is actively used for controlling surgical robots [82]. Thus, an LfD algorithm that initializes the dynamics using experts' demonstrations is proposed in Fig. 3.4. We assume that demonstrations are in video formats that consist of a sequence of images and that the video captures teleoperation of surgical robots by human experts.

The LfD algorithm is described in Algorithm 3. In the first phase of the LfD algorithm, images from demonstrations are fed to the vision algorithm, and the positions and velocities of the robot wrists and tissue points are extracted. Experience sets are saved to the demonstration replay memory and the dynamics neural network is trained with this memory. After this training phase with

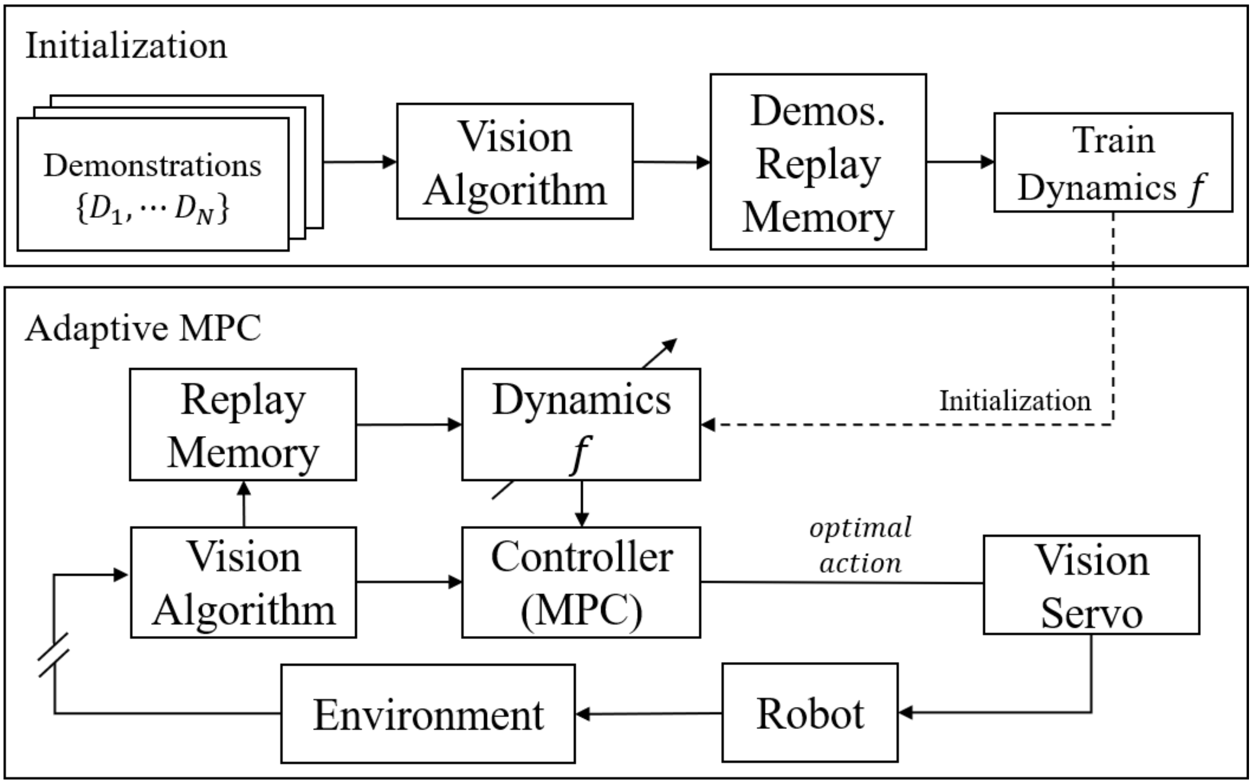


Figure 3.4: Block diagram of Learning from Demonstration.

demonstrations, the dynamics neural network in MPC is initialized with the trained network. This research found that if demonstrations for only one specific set of desired tissue point positions are given, robots can only properly locate the tissue points to desired positions slightly different than in the demonstrations. In order to reach significantly different sets of desired tissue point positions, exploration is necessary. However, if demonstrations encompass a wide range of desired positions and workspaces of the robots, the robots can finish tasks even without exploration.

3.2.1.6 Computer Vision Algorithm

Positions of the robots and tissue features in image space are extracted through a computer vision algorithm. Robot wrist and tissue point positions are recognized by colored features installed at appropriate locations. The position of each component is calculated as the average of the con-

Algorithm 3 Learning from Demonstrations

```
initialize neural network variables
for Number of frames in demonstrations do
    extract $p^T$ and $p^R$ in image
    save experience to replay memory
train dynamics network with replay memory
initialize dynamics of MPC with the trained network
while $error > threshold$ do
    extract $p^T_t$ and $p^R_t$ in image
    if $frame_{curr} - frame_{prev} > \Delta t$ then
        choose optimal action
        update $p^{R,des}_{t+1}$ in image
        save experience to replay memory
        choose mini batch and train dynamics model
    visual servoing of robots
```

tour point positions that can be obtained after color segmentation and morphological operations. This research used the OpenCV library for processing these operations [83]. The tissue points are labeled based on the prior information of the configuration. The motions of the robot wrists are restricted to a two-dimensional square workspace to prevent occlusion of the tissue points. However, this workspace restriction limits the dexterity of the robots.

3.2.1.7 Learning Algorithm Hyperparameters

The dynamics neural network was chosen to have two hidden layers with 12 elements each. Rectified Linear Unit (ReLU) is used for the activation function for both hidden layers. Four tissue points are used in this research effort, resulting in input and output vector sizes of 16 and 8 respectively. The method presented in this research effort can be easily scalable to different number of tissue points and robot arms. Four tissue points and two grasp points are chosen to demonstrate the viability of the proposed algorithm even when an exact solution does not exist because there is insufficient controllability. Weights of the networks are initialized with random numbers from a normal distribution. Learning rate was set to 0.01 and batch size to 200 for RL. Adam optimizer was used to train the neural networks [13]. To stay within computational and physical limitations of the computer and robot, the control period, Δt, was set to 0.5 sec for both RL and LfD algorithms. $Step$ was set to 5 during the learning process in RL. For RL in simulation, the episode was reset every 1,000 actions and the robot was allowed to explore the state space until it reached 5,000 actions.

3.2.2 Simulation

To verify the performance of the controller, a tissue manipulation simulation was designed. Fig. 3.5 demonstrates the environment of the simulation. We used CHAI3D open-platform simulation [84]. The GEL module is used to describe the motion of the soft tissue. The simulated tissue consists of a predefined number of spheres as illustrated in Fig. 3.5. The physical properties of the soft tissue can be set by mass, spring, and damper coefficients for the nodes that form the skeleton structure of the soft tissue, and these physical properties determine the tissue dynamics. The dynamics of the simulation update at a frequency of 1kHz. The movements of the tissue elements are determined by the library of GEL based on the given external forces. External attraction forces between two manipulation points and two robot grippers are generated proportional to the distance between them. The positions of elements on the boundary of the tissue were fixed for internal stability of the tissue. The two manipulation points, four tissue points, and the desired positions of the tissue points are predetermined at the beginning of the simulation. Green markers are attached to the tissue points and wrists of the robots are colored as blue. Human operators can demonstrate bimanual manipulation of the simulated tissue with two phantom omni [85].

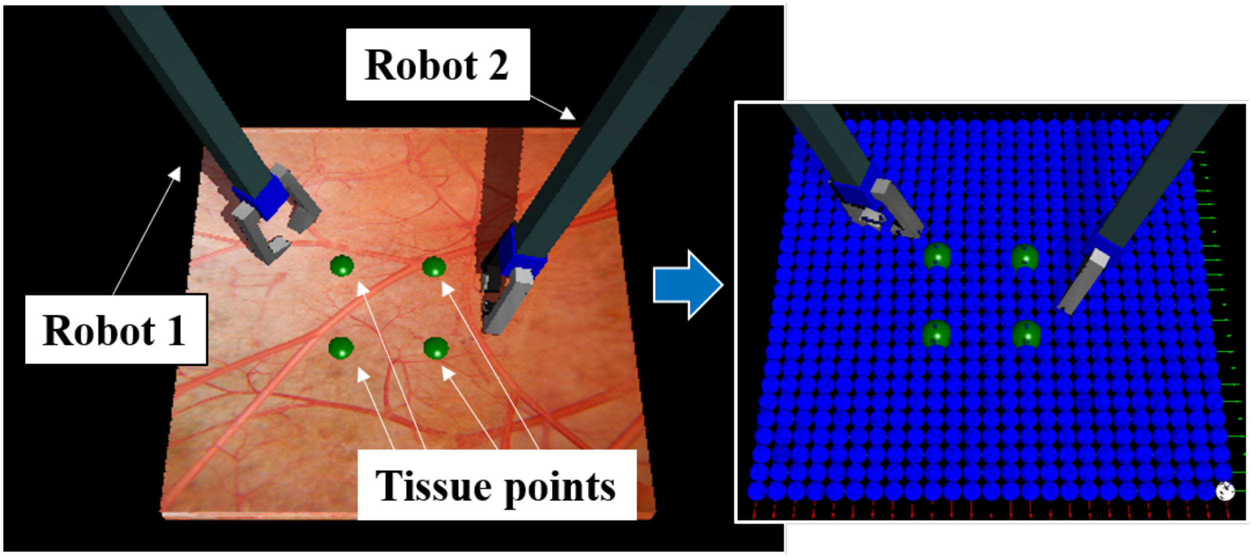


Figure 3.5: Simulation environment and skeleton structure of soft tissue.

3.2.3 Surgical Robot Experiment

3.2.3.1 Raven IV

Experiments were performed with the Raven IV, an open platform for surgical robot research [82]. The Raven IV possesses two pairs of cable-driven surgical robotic arms, each with 7-degrees-of-freedom (DoF) including the grippers. In the experiments conducted, only one pair of robot arms was used. A surgeon or separate surgical automation algorithm could independently perform other tasks, such as suturing, with the remaining two robot arms.

3.2.3.2 Experiment Environment

The environment of the experiment is shown in Fig. 3.6. The manipulation object is made of highly elastic colored latex, and is used to emulate tissue. Four clips are used to fix the object while the robot performs the task. Blue tape is attached to the manipulation object to represent the tissue points and facilitate tracking using the computer vision algorithm. Although a stereo camera (Blackfly-BFLY-U3-13S2C, Point Grey Research) is shown in Fig. 3.6, it is used in single camera

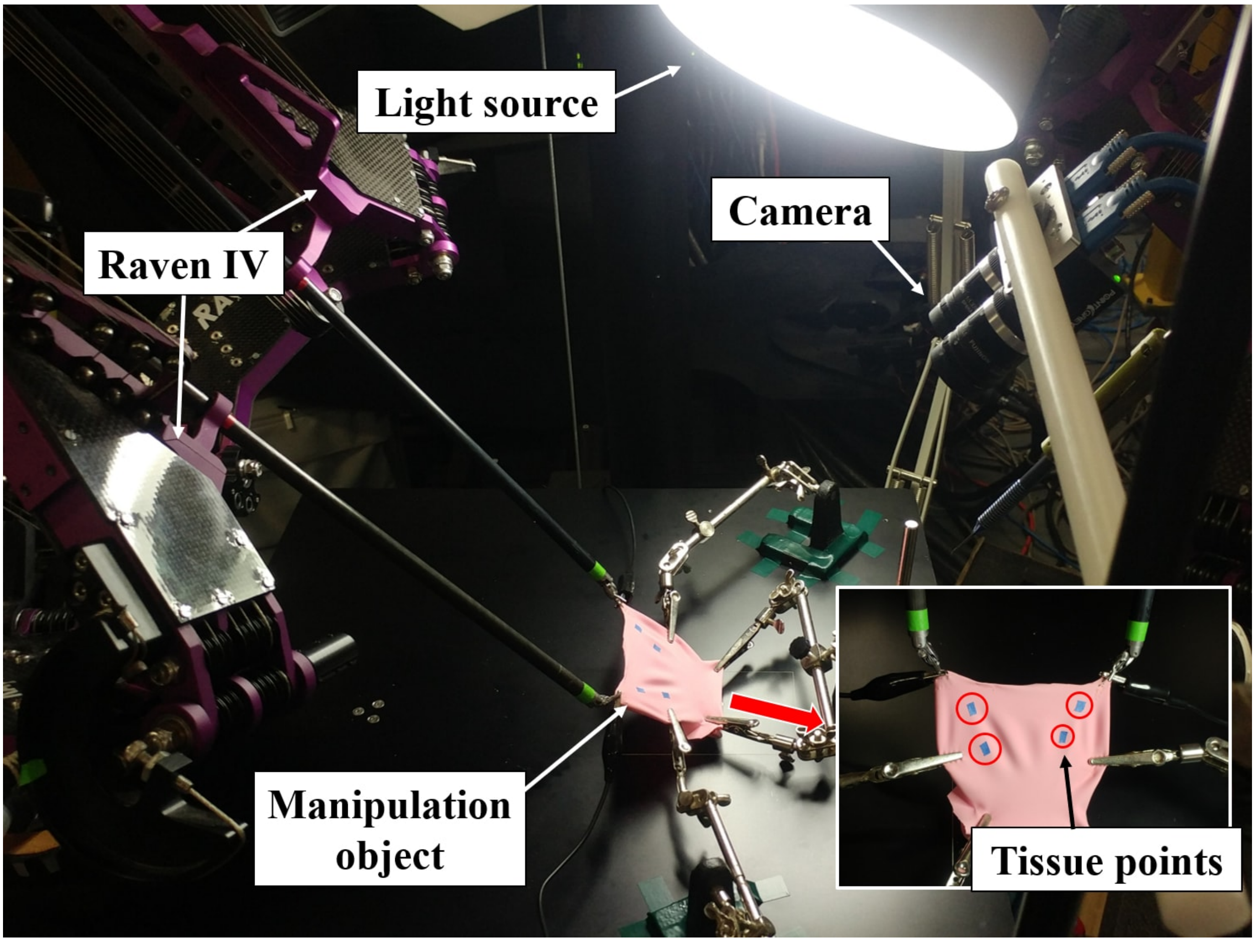


Figure 3.6: Tissue manipulation experiment environment with the Raven IV surgical robotic system.

mode. The original resolution of the camera is 1288x964 but it was reduced to 644x482 before processing the computer vision algorithm. The frame rate of the camera is 30 Hz. For robustness of the computer vision algorithm, a light source is installed.

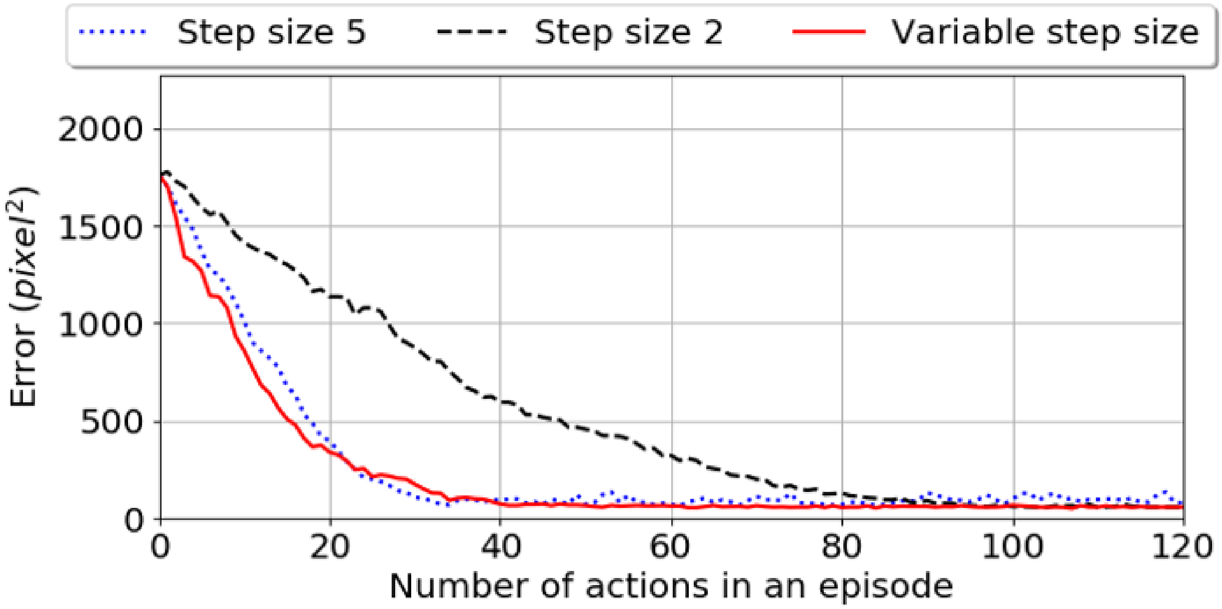


Figure 3.7: Positioning error versus action number in simulation experiments with varied *step* used in the fully trained controller from RL.

3.3 Results and Discussion

3.3.1 Simulation

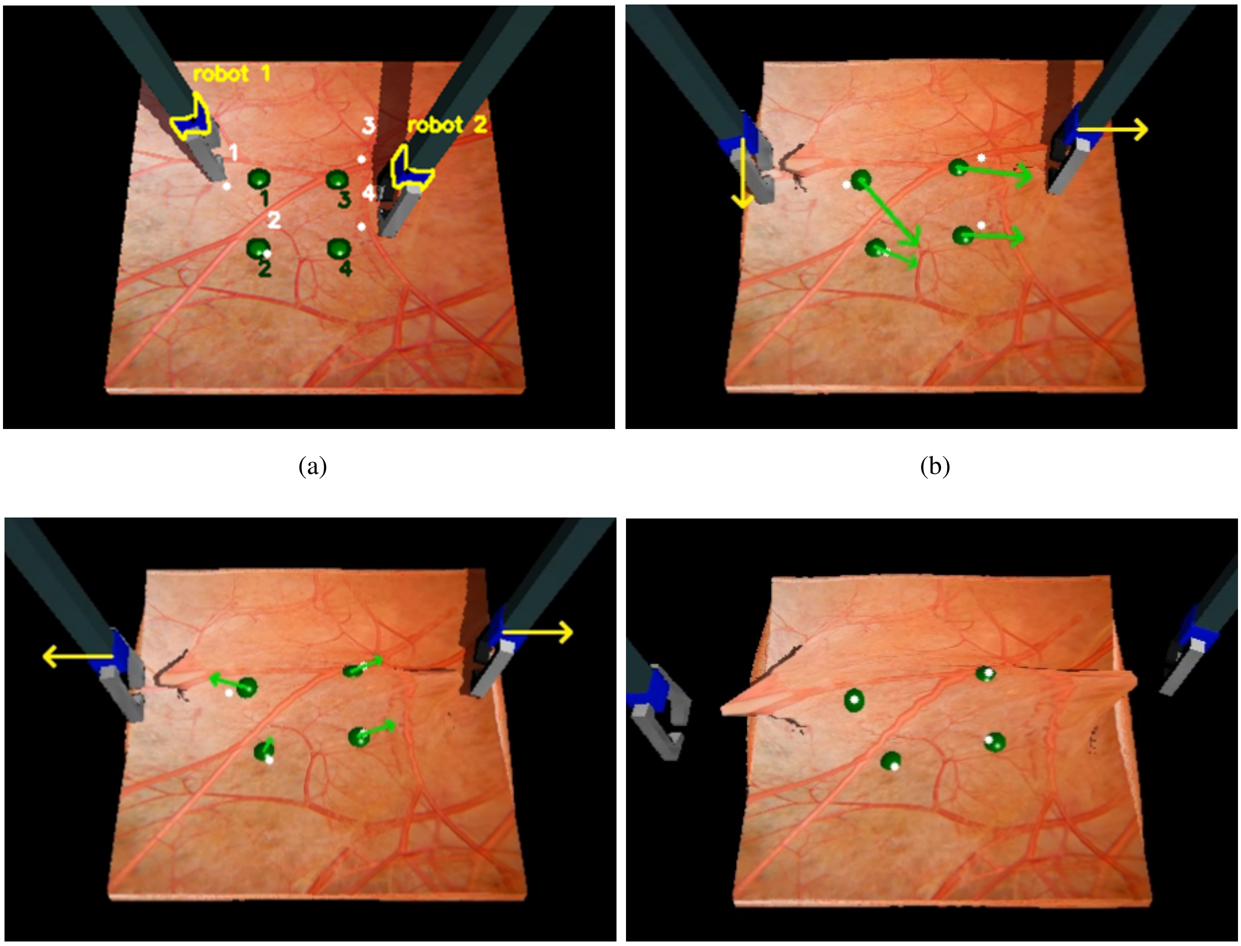
Both RL and LfD algorithms were implemented in the simulation and were evaluated. For the MPC, the maximum number of steps in the time horizon was set to 5, and 5,000 sets of robot action candidates were generated and evaluated during each control period.

3.3.1.1 Effects of Step Size

Fig. 3.7 illustrates the positioning error of the tissue points in an episode with different step sizes. As is shown, the neural network functions and the fully trained controller from the RL algorithm successfully minimizes error regardless of step size. Steady state error is not zero because it is not always feasible to place the four tissue points at exactly their desired positions using only two robot arms. As can be seen, depending on step size, the decay rate and steady state of the error vary. When the step size is large (*step*=5), error decreases quickly but steady state error oscillates because the appropriate step size near the destination is less than the predetermined step size. Alternatively, small step size (*step*=2) results in slow error decay rate but stable and smaller steady state error. Therefore, a variable step size is used. When error is greater than 150, a large step size is used to quickly decrease error. When error is between 150 and 70, a small step size is used to stabilize and reduce the steady state error. Below 70 error, *step* is further reduced to unity (actions move the robot arms single pixels). Both RL and LfD algorithms use the variable step size. The sequence of the simulation is illustrated in Fig. 3.8. The initial configuration is shown in Fig. 3.8a, with the desired tissue point positions labeled 1 through 4. As the simulation progresses in Figs. 3.8b-c, the robot actions (yellow arrows) are input to the learned dynamics to predict the motion of the tissue points (green arrows). Eventually, the error between the actual and desired tissue point positions is minimized as shown in Fig. 3.8d, and the simulation ends.

3.3.1.2 Simulation RL vs. LfD

To compare the RL and LfD algorithms in simulation, three demonstrations were collected from one expert and used to train the dynamics network. We compare the performance of the controller with LfD and at different stages of RL in Fig. 3.9. For this comparison, RL fully exploited optimal actions based on the current dynamics it has at each learning stage. As expected, RL does not perform well until it has been thoroughly trained. We also observe that controller performance does not necessarily improve during the process of learning until dynamics are well understood. This is evident in a comparison of RL 40% and RL 20% in Fig. 3.9. However, LfD is able to perform the task immediately after initialization when desired tissue point positions are not far from the ones in the demonstrations. It was found that if a single demonstration covers a wide range of the workspace, the single demonstration is sufficient for moving tissue points to a variety of desired positions.



(a)

(b)

(c)

(d)

Figure 3.8: Illustration of sequence of simulation experiment. (a) Initial configuration with computer vision results marked as yellow (robot wrists), green (tissue points), and white (desired tissue point positions). (b) Learned dynamics when left robot moves downward and right robot moves right, yellow arrows scaled 10 times show movement of robot and green arrows scaled 30 times visualize predicted motion of tissue points by the robots' motions. (c) Learned dynamics when left robot moves left and right robot moves right. (d) Final configuration showing tissue points located at the desired positions.

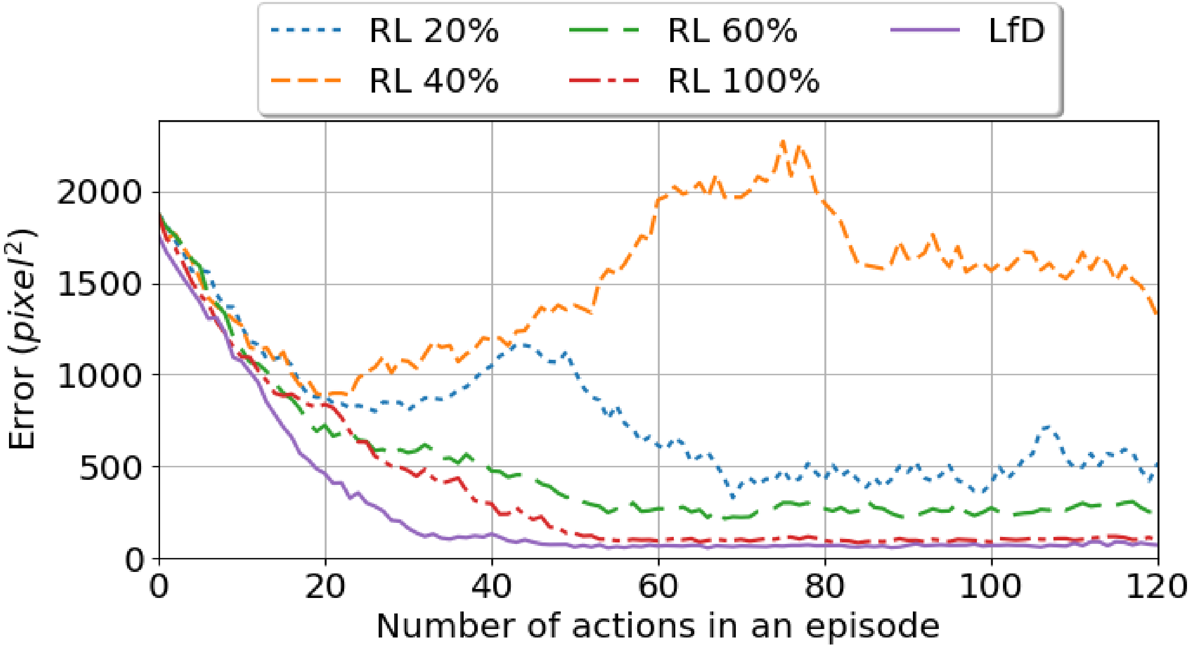
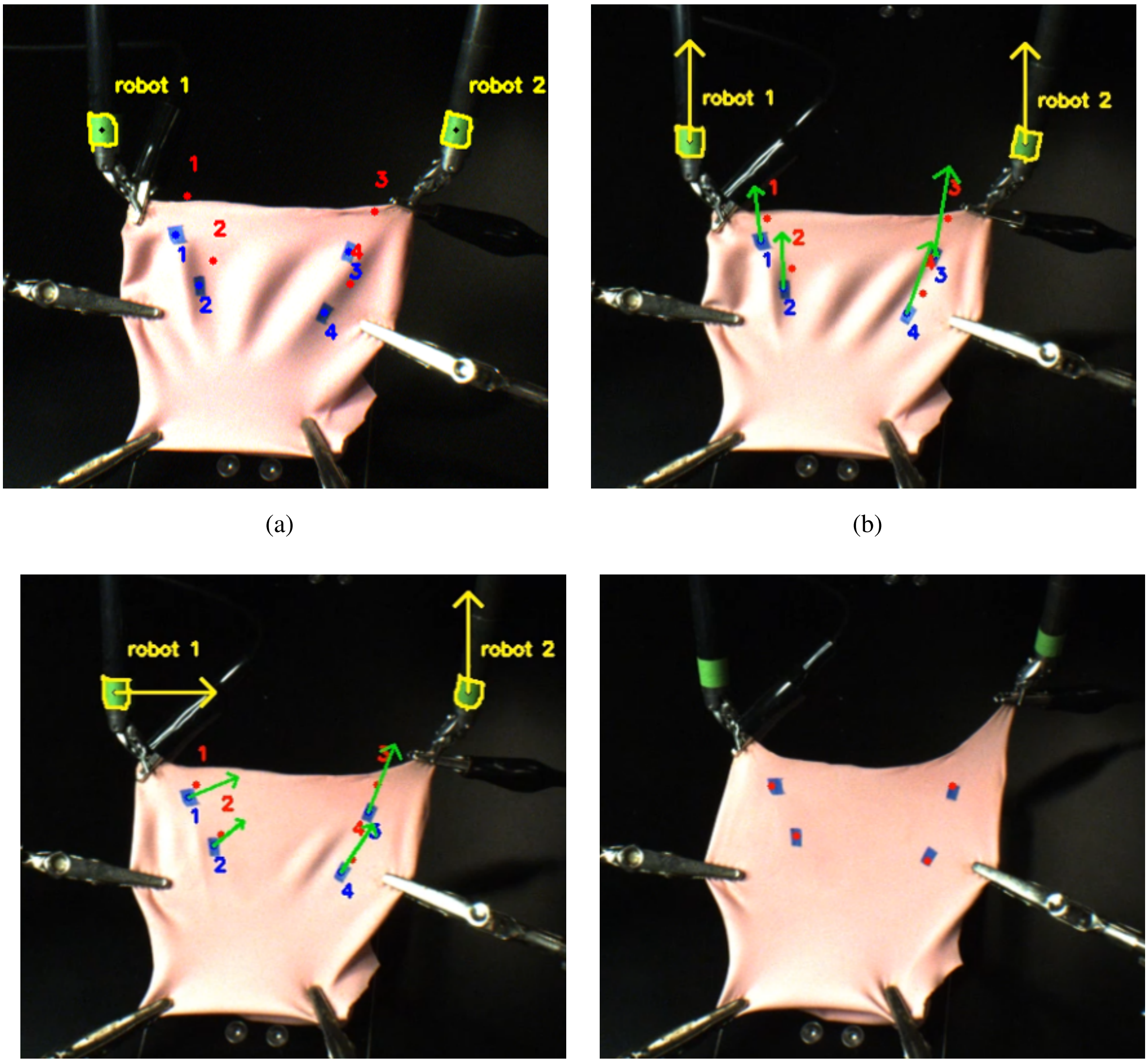


Figure 3.9: Comparison of LfD and RL at multiple learning stages. RL applied its learned dynamics at each stage and fully exploited optimal actions.

3.3.2 Surgical Robot Experiment

From the simulation results in the initial states of training the RL without LfD, it was judged that RL is potentially hazardous and too time consuming to apply to physical systems. However, based on the results of the simulation, it was observed that the initial policy from LfD is meaningful enough to perform the task on the Raven IV. For the MPC, the maximum number of steps in the time horizon was set to 12, and 10,000 sets of robot action candidates were generated and evaluated during each control period.

Two demonstrations were collected by an operator controlling the Raven IV in teleoperation mode with camera feedback, and used to initialize the neural network. We repeated the robot experiment three times with similar initial configurations of manipulation points, initial tissue point positions, and desired tissue point positions. A sequence of captured images representative of these experiments is demonstrated in the Fig. 3.10. The operation of the robot was stopped when it



(a)

(b)

(c)

(d)

Figure 3.10: Illustration of the robot experiment sequentially from (a) to (d). (a) Shows the initial configuration and computer vision algorithm results, red dots are desired tissue points positions. (b) Illustrates learned dynamics when both robots move upward, yellow arrows scaled 10 times represent the motion of robots and green arrows scaled 10 times visualize predicted motions of the tissue points. (c) Shows learned dynamics when left robot moves right and right robot moves upward. (d) Final configuration of experiment when the task is finished.

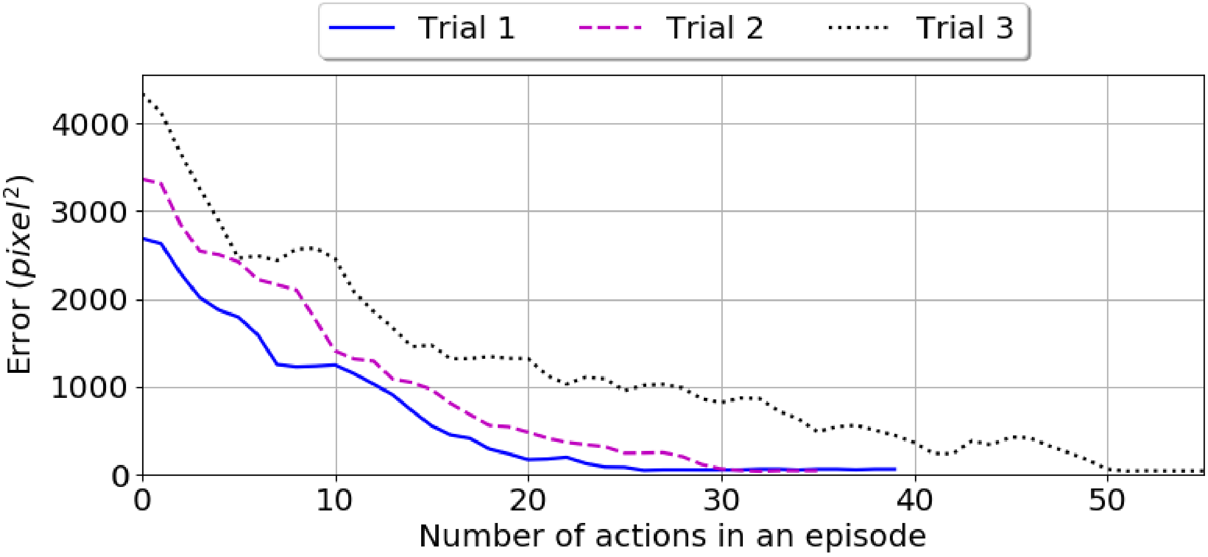


Figure 3.11: Experiment results of LfD with Raven IV. The experiments were repeated three times on similar initial configurations.

placed the four tissue points to the desired tissue point positions with a total error of less than 50 pixels squared as shown in Fig. 3.11. Note that there are slight differences in initial errors among the experiments because initial tissue point positions cannot be replicated exactly.

3.4 Conclusion

In this research effort, RL and LfD algorithms have been presented for automating the soft tissue manipulation task. Experiments on the simulation showed that both algorithms could accomplish the task. The results demonstrate that LfD boosts the learning process by dramatically reducing the amount of exploration required for the neural network to correctly learn the dynamics. The LfD algorithm was implemented on the Raven IV surgical robot, and the robot successfully placed the tissue points to their desired positions. This shows that the LfD algorithm can result in a good initial policy for accomplishing the task in real environments when relevant workspaces are covered in demonstrations. The authors believe that the capability of the LfD algorithm could be expanded significantly if given additional demonstration data that more fully captures the robot's workspace.

There are limitations to the proposed approach that will be addressed by future studies. In the simulation and experiments, the workspace of the robots was constrained to prevent visual occlusion. However, in order to utilize the full capabilities of robots, an algorithm that can avoid the occlusion may be developed. In addition, the algorithm should be expanded to three-dimensional tissue manipulation. Furthermore, the LfD framework presented in this work provides good initial dynamics based on the assumption that the relationship between the camera frame and environment frame is fixed. There should be further research to efficiently use the demonstration data provided in different camera frames than the task environment.

In conclusion, it is anticipated the future research along similar lines will eventually lead to the introduction of automation into surgery clinically in which subtasks of the surgical procedure will be fully automated. This approach is likely to unify across the field that will eventually lead to improved patents' outcome.

CHAPTER 4

Semi-Automated Lens Extraction with Semantic Segmentation of Optical Coherence Tomography Images

4.1 Introduction

Cataracts are the progressive clouding of the natural lens of the eye and represent the leading cause of blindness and visual impairment in the world [86]. Cataracts can be treated by removal of the opaque lens through cataract surgery, which is the most frequently performed surgical procedure in the United States, totaling approximately three million operations per year [87]. In cataract surgery, the opaque lens is extracted and replaced with an intraocular lens implant through several surgical steps including corneal incision, capsulorhexis, nucleus removal, cortical material removal, capsular bag polishing, and implant injection.

While technologies such as femtosecond laser systems can improve the eye-preparation steps, lens extraction—the most delicate and dangerous step—continues to be manually performed. Safe, effective lens removal is challenged by the physiological limitations of a human surgeon including hand tremor [88] and limited resolution of depth sensing [89]. In particular, the posterior capsule (PC) is a delicate and thin (approximately 4–9 μm) membrane which is optically translucent and difficult to visualize [90]. A surgeon is liable to misinterpret shadows and other indirect visual indications of the surgical instrument position, thereby increasing risk of PC rupture, one of the most common complications of cataract surgery [91].

However, the motion and stability requirements of cataract surgery are not prohibitive to the application of robotic surgical systems. The incorporation of robotic systems has recently found widespread use throughout many fields such as urology, gynecology, and general surgery with the development of systems such as the da Vinci Surgical System and the Raven-II open surgical platform [3]. In the field of ophthalmology, several teleoperated robotic systems have been developed and tested on *in vivo* models including human patients. Examples include the Preceyes Surgical System [92, 93] from Preceyes BV as well as the Mynutia intraocular surgical system [94, 95]. Both systems have demonstrated the capability of performing a range of teleoperated vitreoretinal surgical procedures including membrane peeling, subretinal injection, and retinal vein cannulation. However, intraocular robotic systems which have focused on performing procedures specific to cataract surgery are rare, and none have been demonstrated in an automated fashion. In contrast to vitreoretinal procedures, cataract extraction presents a less structured and more dynamic workspace, where the highly mobile lens material is subjected to unpredictable fluid forces, presenting unique challenges for an automated robotic system. The Intraocular Robotic Interventional Surgical System (IRISS) developed at UCLA is one system which has been used to demonstrate semi-automated lens extraction on *ex vivo* pig eyes using optical coherence tomography as visual feedback to guide the robotic system [96, 97].

To overcome the limited sensing capability of a surgeon during cataract surgery, OCT has been incorporated into the IRISS to localize the tool and the surrounding anatomy [96]. Through OCT-based visualization (Fig. 4.1), the tool position relative to intraocular anatomical structures can be understood during surgical procedures and maneuvers can be more safely executed. In [96], an automatic tool insertion method and a trajectory generation algorithm were developed using the parametric model of the eye and fitting the model to OCT B-scan and volume scan data. Although this study introduced automation into important steps of the lens extraction, the remaining lens pieces at the end of the procedure were manually localized in the camera and OCT frames in order to remove them. The data acquired by OCT suffers from a low signal-to-noise ratio and is corrupted by granular interference inherent to the acquisition process (commonly referred to as speckle noise). Conventional image-processing techniques suffer from the presence of speckle noise, and while methods to reduce the speckle noise have been investigated (e.g., [98]),

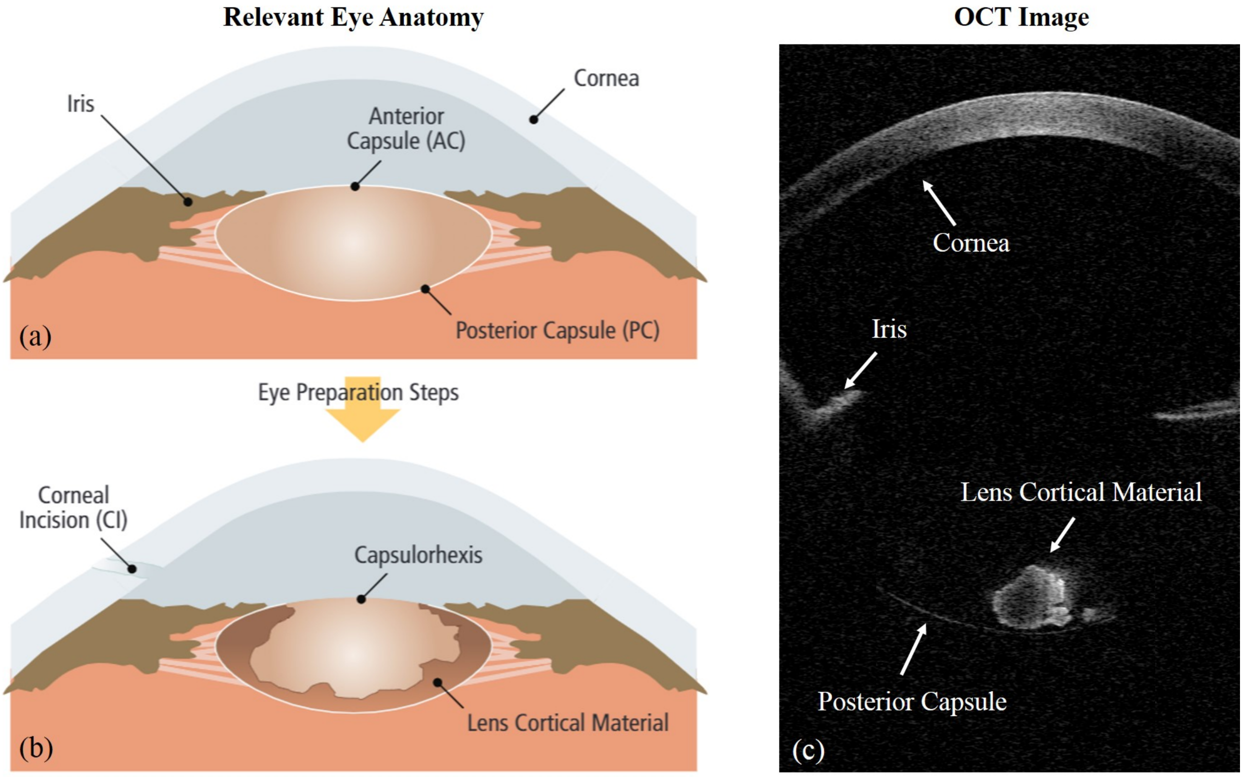


Figure 4.1: Relevant eye anatomy and the corresponding OCT B-scan. (a) the normal eye before any procedure (b) the eye status after the preparation (c) OCT B-scan corresponding to (b). It was generated by merging two B-scans in different depth of scan for the tutorial purpose.

the challenge to extract useful information from OCT data remains.

Recent advancements in deep learning have demonstrated success in computer-vision problems. Among deep-learning architectures, convolutional neural networks (CNNs), motivated by how the brain processes visual information, have shown promising performance in many image classification problems [10]. CNNs extract visual features from a given dataset using multiple channels and layers of convolution layers. As the structure of a CNN gets deeper, the higher level visual features are learned. These visual features representing the dataset are used to classify an image into an object or to generate a pixel-level segmentation map.

Deep learning has been employed in a few aspects of eye surgery for segmentation of OCT images. In [99], a fully convolutional neural network was used to localize the cornea and needle in OCT images. The segmentation algorithm was proposed to be used in deep anterior lamellar keratoplasty surgery and porcine eyes were used to train and validate the performance. Cornet was introduced to segment three corneal interfaces for anterior segment interventions [100]. Other works on segmentation of OCT images mostly focused on retinal layers. ReLayNet was developed to segment retinal layers and fluid for monitoring the degradation of vision quality caused by diabetics [101]. Work on segmenting Bruch's membrane and choroid layer in OCT images to generate the thickness map was presented [102]. Although these works addressed segmentation problems of OCT images for eye, they are not applicable for the lens extraction task because the area of the eye where OCT scanned in those works is not appropriate.

For application to cataract surgery, existing work in image segmentation has been limited to tracking of a surgical instrument in camera images using CNN [103] and an attention-based neural network [104]. However, to achieve safe automated removal of lens material during cataract surgery, the ability to localize relevant intraocular anatomy will be required. For this reason, we present a method to localize intraocular anatomy in OCT images. Furthermore, a framework (Fig. 4.2) to incorporate the developed OCT-segmentation algorithm into the intraocular robotic system is presented. The efficacy of the framework was verified by experiments with *ex vivo* pig eyes. To the best of our knowledge, this is the first demonstration of automated segmentation of OCT images for the removal of lens material and the first to apply this to guidance of a robotic surgical system.

Main Contributions of This Work

- Development of a deep-learning framework to segment intraocular anatomy (lens material, capsule, cornea, and iris) in OCT images.
- Development of a framework for semi-automated robotic extraction of a piece of lens material from a pig eye.

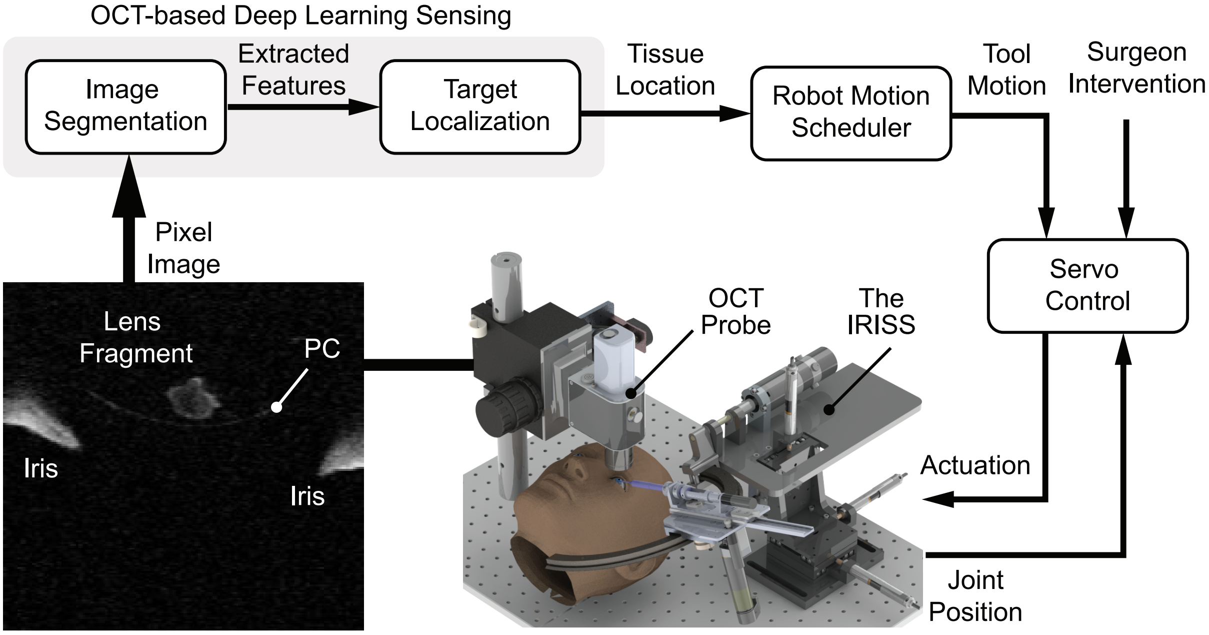


Figure 4.2: The framework for the semi-automated lens extraction using the IRISS and segmentation algorithm.

4.2 Materials and Methods

4.2.1 The IRISS and OCT System

The robotic system used in this work was the IRISS. The IRISS has been used to perform a range of teleoperated intraocular surgical procedures on *ex vivo* pig eyes [105, 106] as well as demonstrate partially automated lens removal on *ex vivo* pig eyes with OCT feedback [96, 97]. Detailed description of the IRISS mechanism and kinematics are provided in previous work [96, 97, 105, 106].

The OCT system used in this work (Telesto II-1060LR, Thorlabs) was capable of acquiring two-dimensional, cross-sectional images (B-scans) and three-dimensional volume scans. The axial resolution of the system was 9.18 µm/px and lateral resolution of 25 µm/px. B-scans were acquired at a width of 10 mm and depth of 9.4 mm while the volume scans were acquired with a volume of 10×10×9.4 mm^3. The automated lens-extraction portion of this work relied on the B-scan data as feedback while the volume scans were used only for evaluation purposes. For this work, the IRISS was mounted with a straight-tip, side port, I/A handpiece (8172 UltraFLOW, Alcon). The I/A handpiece was registered to the robotic workspace according to the calibration process developed in previous work [96].

4.2.2 Pig Eye Preparation

As the eye model, *ex vivo* pig eyes were used (Sioux-Preme Packing, Sioux City, Iowa, USA). The unscalded, enucleated eyes were shipped on ice overnight from pigs butchered the previous day. The eyes were secured by pinning their excess skin into a custom polystyrene holder. Preparation of each eye was performed under a surgical microscope (M840, Leica Microsystems, GmbH). A temporal corneal multiplanar incision was created with a 2.8 mm keratome blade to ensure a watertight wound. Sterile lubricating jelly (MDS032290H, Medline) was injected into the anterior chamber to protect the corneal endothelium. The lubricating jelly has proven itself a good alternative to the more expensive ophthalmic viscoelastic gel and exhibits similar optical and material properties. A cystotome was used to create a central linear cut in the anterior capsule and then pushed to generate a flap, which was manipulated with forceps to create a 6–8 mm diameter continuous curvilinear capsulorhexis. Balanced saline solution was then injected with a cannula attached to a syringe between the outer part of the lens and the capsular bag to achieve their separation. Lens removal was accomplished by slowly aspirating the lens using an I/A handpiece. Lens pieces of various sizes (Section 4.2.3.5 and Table 4.2) were intentionally left in the capsular bag and pushed onto the PC with sterile lubricating jelly, which also helped provide a smoother concave shape to the PC.

4.2.3 Deep Learning-Based Segmentation

4.2.3.1 Data labelling

In this study, we approached the problem of localizing lens material as a semantic segmentation problem. Specifically, given an OCT image $\mathcal{I}$, we wished to find a function $\mathcal{F} : \mathcal{I} \rightarrow \mathcal{L}$ that mapped every pixel in $\mathcal{I}$ to a label $\mathcal{L} \in \{1, \ldots, n_L\}$, where n_L is the number of classes. The segmentation task was a $n_L = 5$ class-classification problem where the classes were the (1) lens, (2) capsule, (3) cornea, (4) iris, and (5) background. The I/A handpiece was included in the lens class because during cortical-material clean-up, the I/A handpiece is commonly occluded by the lens material. In addition, the location of the I/A handpiece can be identified using the forward kinematics of the robot and the known robot-to-OCT registration and therefore can be differentiated from the lens material if necessary.

For the training and validation of the developed algorithms, the OCT-acquired data was manually labeled (Fig. 4.4 and Fig. 4.5). Specifically, the cornea appears as a thick, transparent curve along the anterior segment; the iris appears as hyper-reflective regions on either or both sides of the eye; the lens material appears as amorphous forms within the capsular bag; and the PC is the posterior surface of the capsular bag and appears as a thin, reflective curve. It is important to know the location of the PC due to its fragility and to prevent the I/A handpiece from going near it to avoid breaking the PC—a serious surgical complication. While the cornea and iris are relatively static throughout the lens-extraction procedure, the highly mobile lens material and flexible PC will change location; their localization is essential for performing safe, effective robotic cataract surgery.

4.2.3.2 Deep neural network

The shape and location of intraocular tissue in the acquired data can vary significantly between frames due to fluid turbulence and dynamic deformation of tissue. These challenges decreased our confidence that a model-based approach could produce accurate results. Instead, a deep-learning approach for segmenting the OCT images was applied. Among the various types of deep convolution neural network structures, a fully convolutional neural network was exploited to provide pixel-level segmentation (Fig. 4.3). Its design was inspired by the U-net and FCN-8 structures [107, 108], with an encoder part to extract high-level features and a decoder part to recover the feature channel dimensions to the original input size.

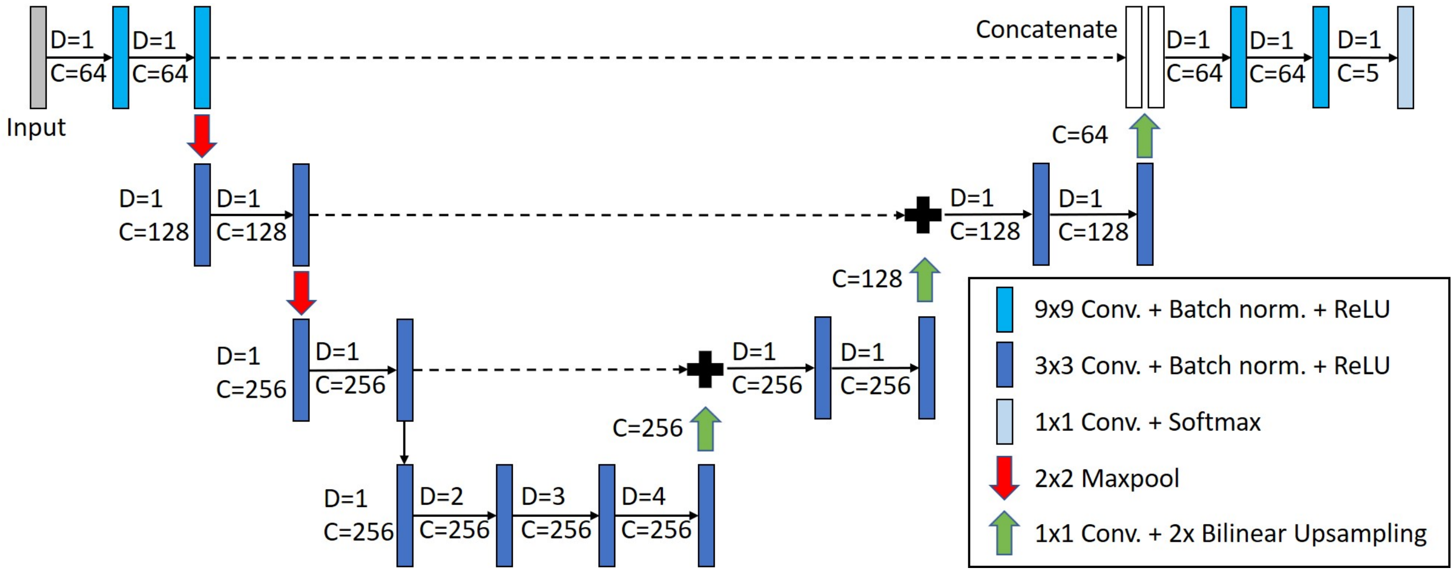


Figure 4.3: Used convolutional neural network. C and D refer the number of channels and dilation rate, respectively.

In each level of abstractions, there is a bridge which links the features map to a decoded channel to incorporate different levels of information into the segmentation. We also used a dilated convolution in the highest level of features to enlarge the receptive field of the filters [109]. To avoid the gridding problem of dilated convolution [110], the dilated rate was increased by one in each feature-extraction level. The filter size of each convolution layer was 3×3 except for the first two convolutions and the batch normalization was followed with a rectified linear unit (ReLU) activation layer. The first two convolutions have 9×9 kernels to extract denser features with a large receptive field. Researchers have suggested to replace kernels with sizes larger than 3×3 with consecutive layers of 3×3 kernels [16]. However, we opted to use the 9×9 convolutions because they provided higher accuracy and a shorter computation time. We compared the performance of the proposed network with the model replacing the 9×9 convolution with four 3×3 convolutions, which had the same size as the receptive field (Section 4.2.3.5). Upsampling in the decoder part consisted of 1×1 convolutions to match the channel numbers in the following layers and the dimension was upsampled with bilinear interpolation.

4.2.3.3 Training

Our dataset is split into training data (809 images) and test data (111 images). The OCT images for training were collected from ten eyes and the images for the test was from a different set of eight eyes.

The developed network was trained using the dice coefficient loss (DCL) and the focal loss (FL). These loss functions were selected to address data imbalance problem as size of each intraocular structures is different (Fig. 4.4 and Fig. 4.5), especially PC takes significantly small portion in the dataset. DCL and FL are defined as:

$$\mathrm{DCL} = 1 - \frac{1}{n_L}\sum_{c=1}^{n_L}\frac{2|P_c * G_c|}{2|P_c * G_c| + |(\mathbf{1} - P_c) * G_c| + |P_c * (\mathbf{1} - G_c)|} \tag{4.1}$$

$$\mathrm{FL} = -\sum_{c=1}^{n_L}\alpha|(\mathbf{1} - P_c)^{\gamma} * G_c * \log(P_c)\,| \tag{4.2}$$

where P is a prediction map from the neural network and G is ground-truth. Both are $\in \mathbb{R}^{H \times W \times n_L}$ when the dimension of the input image is $\in \mathbb{R}^{H \times W}$. c refers to the channel of the prediction and ground-truth. $*$ is an element-wise multiplication and $| \cdot | : \mathbb{R}^{H \times W} \rightarrow \mathbb{R}$ is the summation of all elements. $\mathbf{1} \in \mathbb{R}^{H \times W}$ is a matrix which all elements are 1. The DCL minimizes the ratio between the intersection and union of the prediction and the ground truth. The summation term is 1 when the predictions perfectly match with the ground truth and is subtracted from 1 to make the minimum value of the DCL zero. In Eq. (4.2), the exponential and log operations are element-wise. The FL adds a factor $(\mathbf{1} - P_c)^{\gamma}$ to the cross-entropy loss, which has been suggested to address problems of data imbalance [111]. The added term reduces the loss for well classified classes.

Each image was preprocessed in three steps. First, the first 29 rows in the image is cropped out as artifacts present and it is further half sampled. This step reduces the dimension of the input image from 1024x400 to 498x200. Second, the reduced image is filtered with Gaussian smoothing kernel to reduce the noise and then the mean of pixel values are subtracted. Finally, the pixel values are normalized such that they are in the range between 0 and 1.

An Adam optimizer [13] with the learning rate 1×10^{-4} was used to train the network. The learning rate was reduced by half if the validation accuracy did not increase for three consecutive epochs. The total number of epochs was set to 100 but training was stopped if the performance of the network did not improve for ten epochs. For the focal loss, $\gamma = 2$ as proposed by [111] and $\alpha = 0.2$.

Training data was augmented by horizontally flipping images, vertical translation, and adding Gaussian noise. Vertically flipping images was not used in order to maintain the geometrical meaning of the intraocular structures.

4.2.3.4 Evaluation

The test set has images from different scan depths (anterior and posterior views) and anatomical environments. The performance of the trained neural networks was evaluated based on two metrics:

accuracy and inference time. For the accuracy, two metrics were considered. The first metric was intersection-over-union (IoU) which calculates the ratio between intersection and union area of prediction and ground truth, defined as:

$$\mathrm{IoU} = \frac{1}{L}\sum_{i=1}^{L}\frac{\mathcal{C}_{ii}}{\mathcal{G}_i + \mathcal{P}_i - \mathcal{C}_{ii}} \tag{4.3}$$

where $\mathcal{C}_{ii}$ is the number of pixels whose ground-truth is labeled as class i and inference result is i. $\mathcal{G}_i$ is the number of pixels which is labeled as the class i in ground-truth. $\mathcal{P}_i$ is the number of pixels predicted as class i. The second metric is mean pixel accuracy (MPA). The MPA calculates average percentage of correctly classified pixels for each class. MPA is defined as:

$$\mathrm{MPA} = \frac{1}{L}\sum_{i=1}^{L}\frac{\mathcal{C}_{ii}}{\mathcal{G}_i} \tag{4.4}$$

Inference time was an important factor to consider because its value determines the feasibility of incorporating the framework into a robotic system as real-time feedback. The inference time was the average time to get an inference probability map of 1000 images from the network. The network was implemented in Keras/Tensorflow. All experiments were performed on an Nvidia Geforce GTX 1080 Ti with 11 GB of memory.

4.2.3.5 Segmentation accuracy

The performance of the neural network used in this work is shown in Table 4.1. The model trained with the DCL exhibited higher accuracy than the one trained with the FL. This difference was more pronounced in the posterior capsule detection accuracy. The performance of the developed network was also compared with a model that replaced each of the 9×9 convolutions with four 3×3 convolutions. The models with all 3×3 kernels and trained with the DCL (88.91% in MPA and 77.88% in mean IoU) and FL (82.97% in MPA and 75.34% in mean IoU) had lower accuracy but required longer computation time (36.37 ms).

Figures 4.4 and 4.5 illustrate examples of inference results from the developed network. They show 12 sets of images which consist of input image (OCT B-scan), the inference result, and the

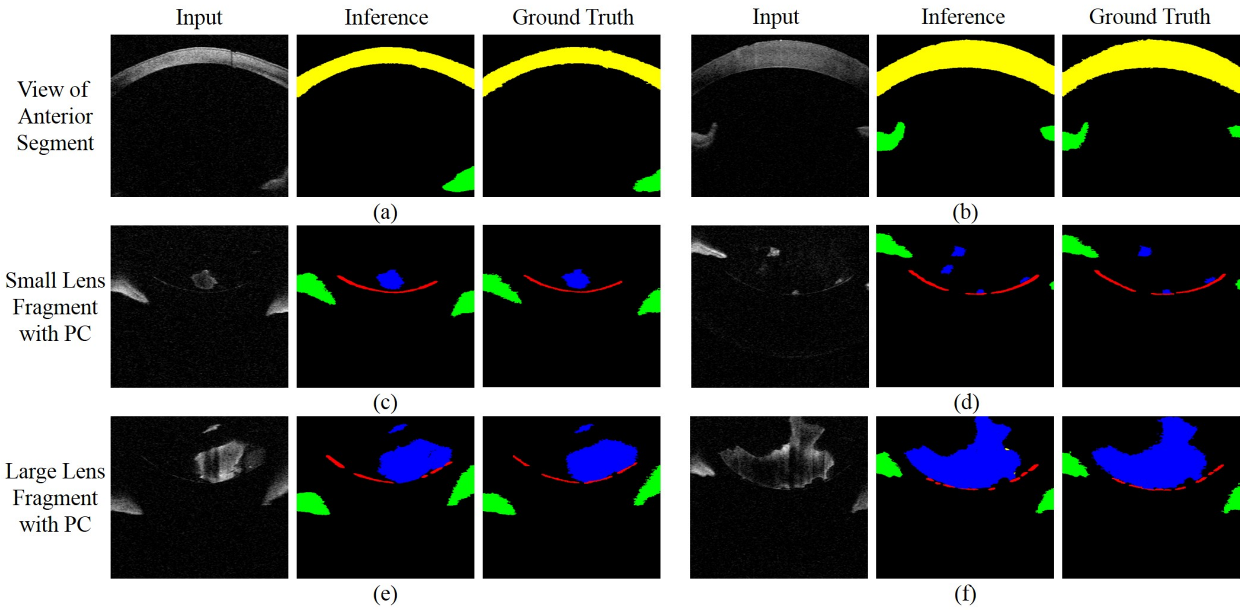


Figure 4.4: Illustration of OCT image segmentation results I. Each set includes input (left), inference result from DNN (middle), and ground-truth (right) images. In the inference and ground-truth images, the segmented intraocular anatomical structures are colored as yellow (cornea), green (iris), blue (lens), and red (posterior capsule). (a) anterior view with cornea and iris, other side of iris is not captured, (b) anterior view, (c,d) small lens pieces with posterior capsule, (e,f) large pieces of lens with posterior capsule.

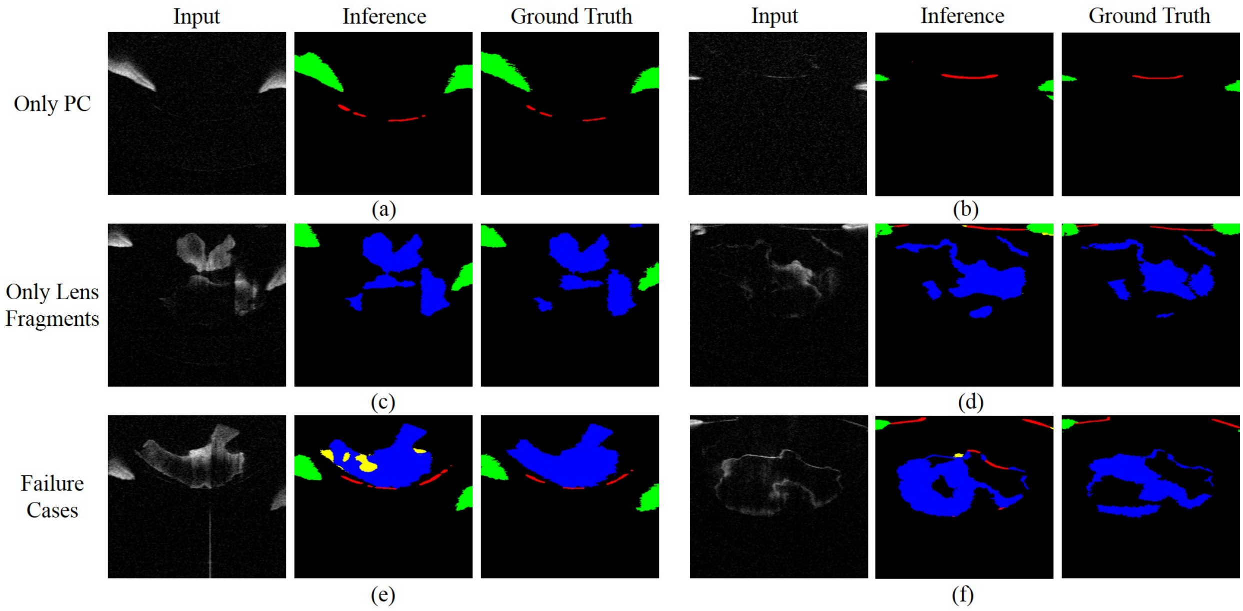


Figure 4.5: Illustration of OCT image segmentation results II. Each set includes input (left), inference result from DNN (middle), and ground-truth (right) images. In the inference and ground-truth images, the segmented intraocular anatomical structures are colored as yellow (cornea), green (iris), blue (lens), and red (posterior capsule). (a,b) posterior capsule without lens, (c) only lens, posterior capsule is not visible, (d) lens and anterior capsule, (e) failure case I, part of big lens piece is classified as cornea, (f) failure case II, inside of the lens piece is not scanned and the part of outline of the lens is classified as the capsule.

Table 4.1: Evaluation metrics of used neural network

Loss functions	DCL	Focal
Mean pixel accuracy (%)	89.83	83.90
Lens pixel accuracy (%)	88.41	89.91
PC pixel acc. (%)	69.91	47.38
Mean IoU (%)	78.20	74.62
Lens IoU (%)	75.03	72.44
PC IoU (%)	46.63	33.68
Inference time (mean ± SD)	31.28±1.25 ms	

SD = standard deviation

ground truth. The shown images cover different cases of the OCT scan such as various size of a lens piece and scan depth. In the test data, the total area of the lens piece varies from 139 pixels (0.13 mm^2) to 20,884 pixels (19.17 mm^2). We defined the lens pieces smaller than 7,000 pixels (6.43 mm^2) as the small lens fragment. The medium lens fragment is the piece with the area between 7,000 pixels (6.43 mm^2) and 14,000 pixels (12.86 mm^2). If it is larger than 14,000 pixels (12.86 mm^2), we refer it the large lens fragment.

4.2.3.6 Phase ambiguity in the OCT system

Like many OCT systems, the data acquired by the device used in this work suffers from phase ambiguity. The phase ambiguity results in image inversion of anatomical structures which are physically closer to the probe than the scanning depth (Fig. 4.6(a)). For the lens-extraction task, where the scanning depth is focused near the PC, the cornea and iris will appear inverted in B-scan images. This inversion introduces difficulty for correct labeling, especially when the inverted structures appear deeper in the scan where the signal-to-noise ratio is lower. To address this problem, we defined an area mask as shown in Fig. 4.6 and this masked area was not considered in

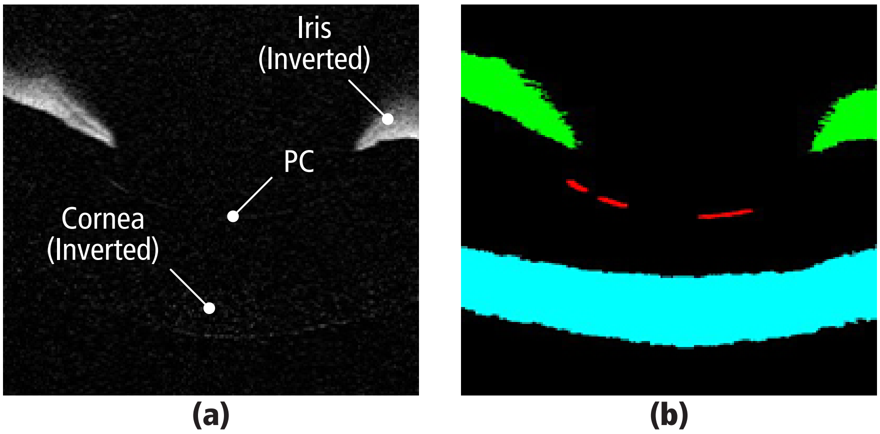


Figure 4.6: (a) A B-scan containing non-inverted PC and inverted iris and cornea. The inverted cornea complicates the learning algorithm and is ignored. (b) Pixel labels for the three anatomical structures in (a). Green: iris, red: PC, and cyan: cornea (ignored area).

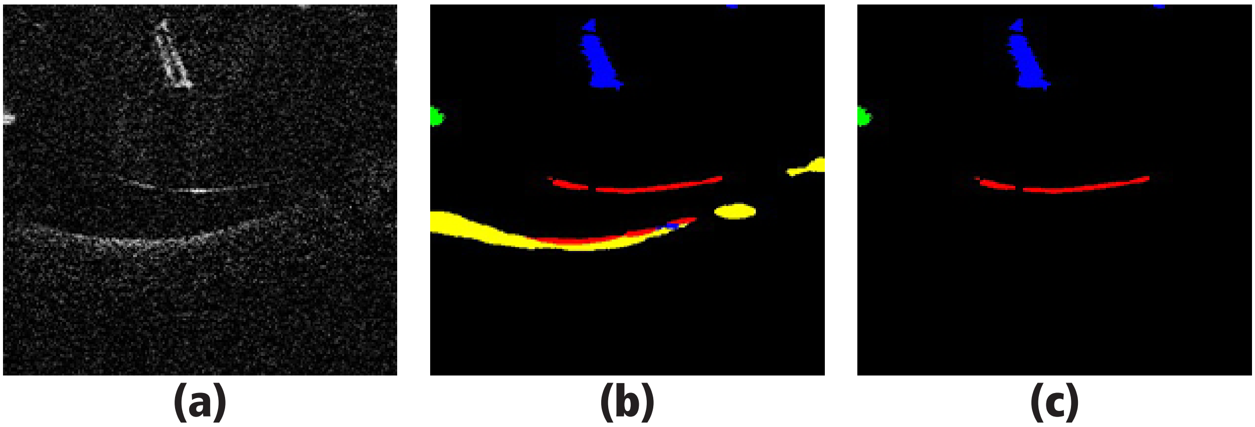


Figure 4.7: (a) B-scan data showing a piece of lens, the PC, and the inverted cornea. Note the inverted cornea appears similar to the PC. (b) Segmentation result without compensation for the inverted cornea. Note the incorrect labeled. (c) Segmentation results after cornea removal. Color labels are blue: lens piece, green: iris, red: PC, and yellow: cornea.

the accuracy tests. We use the knowledge that the cornea is at least several millimeters anterior to the capsular bag and therefore can be safely ignored. During the operation of the supervised lens extraction, the scan depth can be adjusted by physically moving the OCT probe closer or further from the eye. Doing so shifts where the inverted cornea and iris will appear in the B-scan image relative to the PC and lens material. With larger pig eyes, the depth could be adjusted such that these structures are not visible. However, in the case of smaller eyes, this may not be an option. This problem could be solved in two ways. First, the operator can select the ignored area in the image and the segmentation algorithm neglects the chosen area. Second, the location of the inverted cornea is found using the pixel information classified as the cornea. Then, the ignored region could be defined using the predefined two convex shape kernels and user-defined cornea thickness. This is possible because the cornea shape is convex when inverted and does not significantly deform. An example of this case is shown in Fig. 4.7. It is seen that the outline of the cornea is classified as the PC due to the majority of the cornea not being captured because of low signal-to-noise ratio. However, using the cornea information, the inverted cornea can be successfully removed using the post-processing algorithm.

4.3 Experiment and Result

To demonstrate the developed framework, we performed semi-automated detection and extraction of a piece of lens on seven *ex vivo* pig eyes. To begin a trial, a pig eye was manually prepared (Section 4.2.2) as illustrated in Fig. 4.1, fixed to a Styrofoam holder, and placed within the physical workspace of the robotic system. Next, the IRISS was teleoperated to align the tip of the I/A handpiece to the corneal incision, the tool was inserted approximately 1–2 mm into the eye, and the irrigation pressure set to a constant 60 mmHg to maintain intraocular pressure. At this point, the operator acquired a single B-scan image of the lens material by shifting the OCT scanning plane through the eye based on the camera image.

The B-scan data was sent to the image-processing unit and the segmentation performed using the CNN. The largest binary blob of lens material was found and the centroid of the blob was calculated. The pixel coordinates of the centroid were used to represent the location of the detected lens material in the image. The detected location of the lens material was displayed to the operator as an overlay atop the B-scan image. Once the operator confirmed its location, the IRISS was

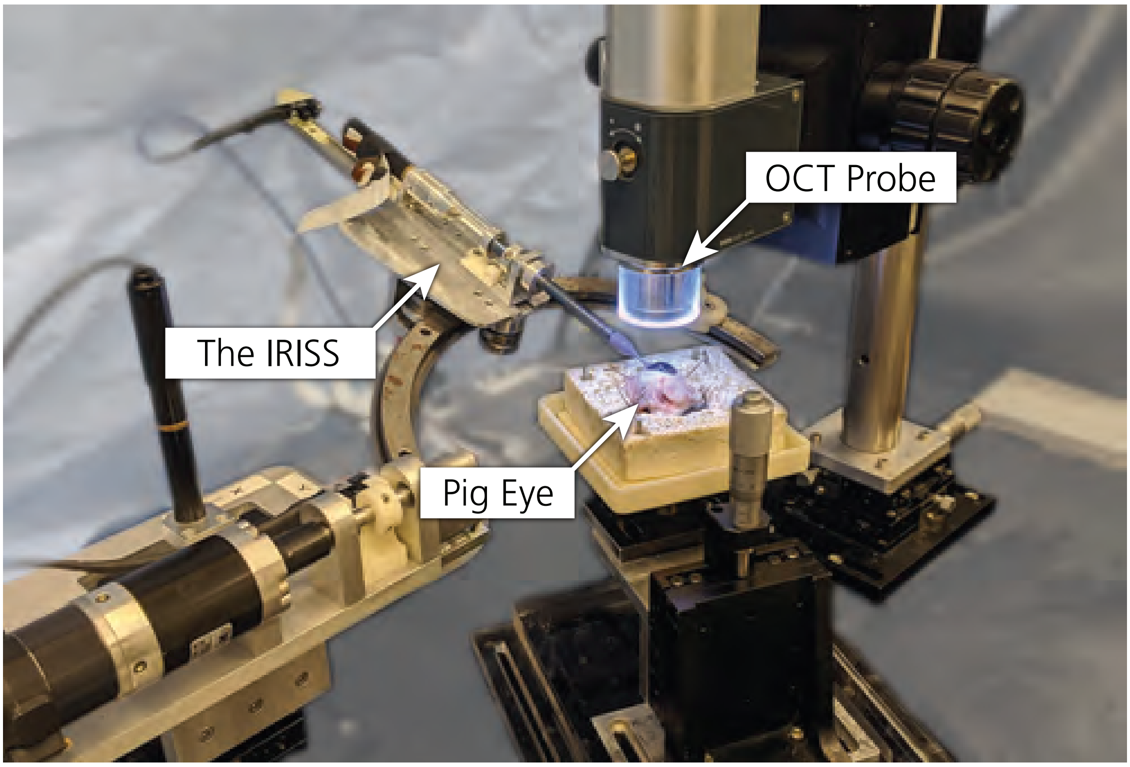


Figure 4.8: Experiment setup with the IRISS and OCT system.

commanded to move towards it with a prescribed $\approx$ 1 mm/s speed. The software architecture allowed for motion abort and rerouting of the path of the I/A handpiece to help account for the dynamic nature of the process.

Once at the lens fragment, the aspiration force was stepped up to 200 mmHg and the piece aspirated (Table 4.2). The aspiration was stopped once the operator deemed the piece had been removed based on feedback from the continuously acquired B-scan imaging (Fig. 4.10) as well as from the camera image. These sequences demonstrate the I/A handpiece was moved to the targeted lens material piece and then successfully aspirated it. After extraction of the lens fragment, complete removal was confirmed by a trained fellow. These steps are summarized in Fig. 4.9.

For post-trial evaluation, OCT volume scans were acquired before and after the lens-extraction operation, outside the scope of the automated procedures (Fig. 4.11). The volume of the piece of lens material (Table 4.2) was calculated by manual segmentation of the OCT volume scan: the

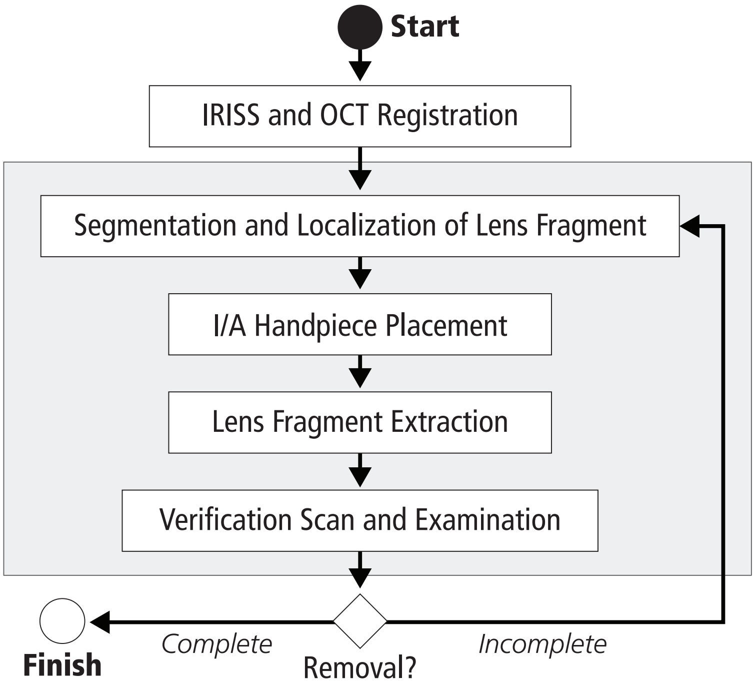

Start
IRISS and OCT Registration
Segmentation and Localization of Lens Fragment
I/A Handpiece Placement
Lens Fragment Extraction
Verification Scan and Examination
Finish
Complete
Removal?
Incomplete

Figure 4.9: Shown is a schematic of the robot-control method with integrated segmentation algorithm.

number of lens-material voxels were summed and then multiplied by the known voxel volume (Section 4.2.1).

Table 4.2: Evaluation metrics of lens extraction

Eye No.	Lens Volume [mm^3]	Time to Aspirate [s]
1	24.18	48.09
2	7.66	2.90
3	8.64	101.42
4	8.78	44.06
5	13.78	290.77
6	41.16	397.18
7	2.90	45.96

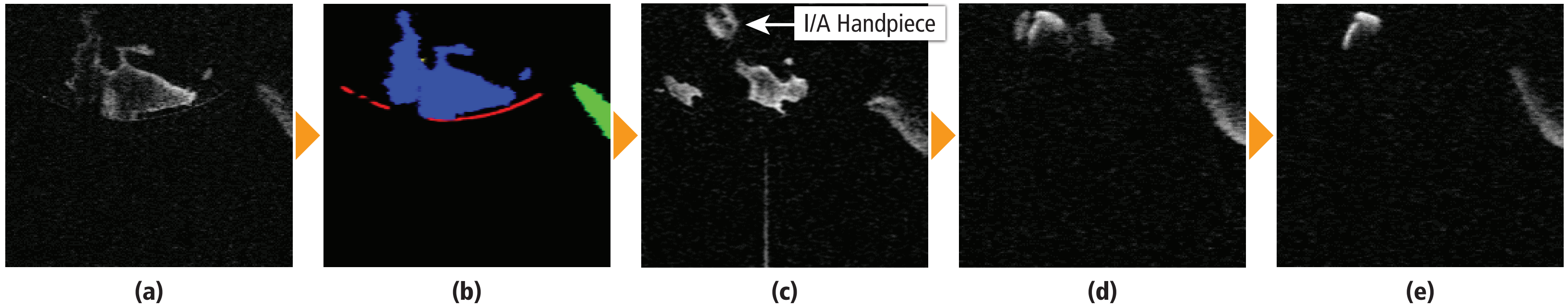


Figure 4.10: OCT B-scans of an exemplary robotic experiment. (a) the selected scan by the operator, (b) segmentation result of (a), (c) the tool is placed on the cortical material, (d) the cortical material is gradually removed by the tool (e) the lens material has been removed.

4.4 Discussion

The proposed model successfully segmented the target anatomical structures across different scenarios, including depth of scan (Fig. 4.4 and Fig. 4.5a–d). Furthermore, the success of the developed method suggests its ability to handle the speckle noise that is common in OCT images.

However, there are at least two failure cases which should be addressed in future work. First, the lens piece can sometimes appear similar (shape and intensity) to the inverted cornea (Fig. 4.5e). Because the OCT data is grayscale and the lens material can take any arbitrary form, the lack of distinctive visual features can result in this misclassification. Second, the internal intensity of the lens piece varies for unknown reasons, sometimes causing the piece to appear solid and at other times hollow (Fig. 4.5f). Because the contour of the lens piece can appear as a thin, hyper-reflective line, the segmentation algorithm may mislabel it as capsule. Three explanations are proposed. (1) The amount of lubricating jelly or water used to prepare the eye may be affecting the intensity of the piece since water is known to attenuate the OCT signal to a greater extent than the jelly. (2) The lens piece may be nucleus material rather than cortical material. (3) The size of the lens piece may affect the overall intensity in the OCT image.

The successful implementation of robotic experiments on 7 pig eyes confirms the efficacy of the proposed framework for the supervised automated lens extraction. In Fig. 4.10, it is shown that the developed neural network detects the iris, cortical material, and posterior capsule successfully in the selected B-scan image. The volume scans shown in Fig. 4.11 confirm that the piece of cortical material on the posterior capsule was removed after the operation. Seven repeated robot experiments on different pig eyes demonstrate the robustness of our method.

It was observed that the time to aspirate the cortical material varied between each operation (Table 4.2), but no clear correlation was found between the volume of the lens material and the time required to aspirate. However, it was observed that the majority of the reported time was spent aspirating the piece as it occluded the vacuum port of the I/A handpiece, and this observation offers some explanation: the tool-tip would oftentimes become clogged and would require a long

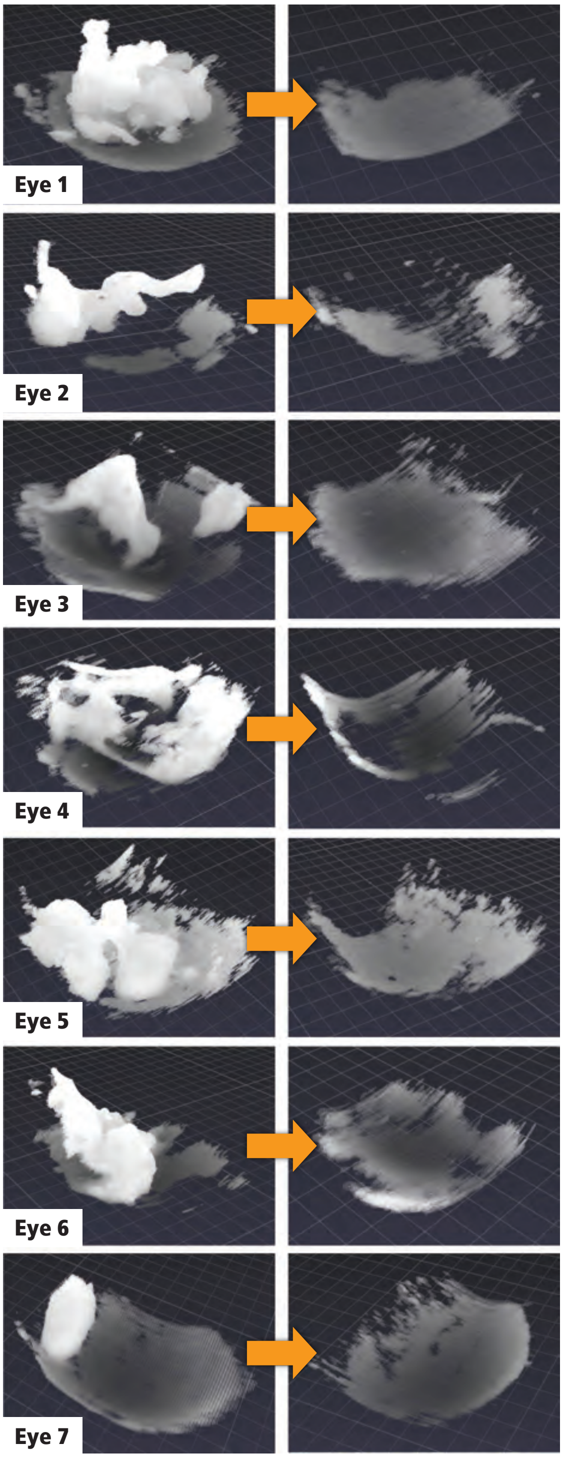


Figure 4.11: Preoperative and postoperative OCT volume scans that illustrate the successful removal of the lens material from each pig eye. The square grid has an edge length of 90.6 μm.

time to clear itself. Therefore, with a tool better suited for removal of soft pig-eye lens material (such as a phacoemulsification probe), it is expected that the time-to-aspirate would correlate with the volume of the lens material.

Future work includes three avenues of study. First, to overcome limitations due to the lack of color information and the large variability in appearance of the lens material, a method which includes adjacent B-scans can be developed. This could allow the segmentation algorithm to incorporate three-dimensional data and could potentially improve the segmentation accuracy. Second, a control method which updates the tool-tip location in response to the internal dynamics of the eye should be investigated as a necessary step towards full automation of the lens-extraction procedure. In reality, the lens material changes shape and location as a function of the aspiration and irrigation forces, and a means to account for these changes will be necessary. Third, the location and shape of the posterior capsule should be incorporated into the path planning of the I/A handpiece. An important safety consideration during cataract removal is to know the location of the posterior capsule and avoid its rupture. By using the segmentation results, a high-level "no fly" zone could be established around the posterior capsule, or the tool-tip position could be adjusted in response to changes in its location.

4.5 Conclusion

In this work, an integrated framework was developed for semi-automated detection and extraction of a piece of lens material in an *ex vivo* pig eye. The developed framework included segmentation of OCT images using a deep-learning approach and guidance of an intraocular robotic surgical system using the segmentation results. A neural network was trained on data from ten pig eyes (809 images) and tested on images from eight eyes (111 images). The framework was experimentally demonstrated on seven pig eyes to verify that the developed method was feasible. This work represents an important step towards fully autonomous, robot-guided cataract extraction.

CHAPTER 5

Conclusion

Introducing the automation into surgery could potentially improve the patient outcome by allowing surgeons to focus on the decision making procedures and providing accurate manipulation of tissue. To deploy the current surgical robots as automation fashion, the robots or surgical systems should have at least three capabilities: 1) high precision 2) path planner 3) scene understanding.

In this dissertation, solutions for the aforementioned problems were developed. First, the vision-based kinematic controller which utilizes both the model of the robot and real-time data-driven approach for cable-driven robots was studied. The experimental results showed the convergence and applicability of the controller to the clinical tasks which require high precision. Second, learning-based path planning algorithms for soft tissue manipulation were presented. The algorithms were tested on the designed simulation and the Raven IV surgical robotic system. The results verified that the algorithms could successfully manipulate the deformable object by learning the dynamics from the experience and demonstrations. Lastly, a semantic segmentation algorithm using deep learning was proposed to localize the intraocular anatomical structures (cornea, iris, lens, and capsule) in optical coherence tomography images. OCT images from 18 pig eyes were collected to train the developed neural network and evaluate the performance of it. Furthermore, this segmentation algorithm was incorporated into the intraocular surgical system to demonstrate the semi-autonomous lens extraction. The experiments on the 7 *ex vivo* pig eyes confirmed the efficacy the developed framework and feasibility of using OCT as guidance for the lens extraction task.

The works presented in this dissertation are on the trajectory towards the profound vision which redefines the role of surgeons as the pure decision maker while the surgical robots perform the manipulation tasks.

REFERENCES

[1] G. S. Guthart and J. K. Salisbury, "The intuitive/sup tm/telesurgery system: overview and application," vol. 1, pp. 618–621, 2000.

[2] I. Intuitive Surgical, "Annual report of intuitive surgical, inc.," 2016.

[3] 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, 2012.

[4] S. Giri and D. K. Sarkar, "Current status of robotic surgery," *Indian Journal of Surgery*, vol. 74, no. 3, pp. 242–247, 2012.

[5] G.-Z. Yang, J. Cambias, K. Cleary, E. Daimler, J. Drake, P. E. Dupont, N. Hata, P. Kazanzides, S. Martel, R. V. Patel, *et al.*, "Medical robotics—regulatory, ethical, and legal considerations for increasing levels of autonomy," *Sci. Robot*, vol. 2, no. 4, p. 8638, 2017.

[6] A. Murali, A. Garg, S. Krishnan, F. T. Pokorny, P. Abbeel, T. Darrell, and K. Goldberg, "Tsc-dl: Unsupervised trajectory segmentation of multi-modal surgical demonstrations with deep learning," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4150–4157, IEEE, 2016.

[7] P. Kazanzides, 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 *2014 IEEE international conference on robotics and automation (ICRA)*, pp. 6434–6439, IEEE, 2014.

[8] 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 *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1202–1209, IEEE, 2015.

[9] S. A. Pedram, P. Ferguson, J. Ma, E. Dutson, and J. Rosen, "Autonomous suturing via surgical robot: An algorithm for optimal selection of needle diameter, shape, and path," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2391–2398, IEEE, 2017.

[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[11] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, pp. 740–755, Springer, 2014.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[13] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[14] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[15] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *International Conference on Learning Representations*, 2015.

[16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[17] A. Shademan, R. S. Decker, J. D. Opfermann, S. Leonard, A. Krieger, and P. C. Kim, “Supervised autonomous robotic soft tissue surgery,” *Science translational medicine*, vol. 8, no. 337, pp. 337ra64–337ra64, 2016.

[18] S. Sen, A. Garg, D. V. Gealy, S. McKinley, Y. Jen, and K. Goldberg, “Automating multi-throw multilateral surgical suturing with a mechanical needle guide and sequential convex optimization,” in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4178–4185, IEEE, 2016.

[19] R. C. Jackson and M. C. Çavuşoğlu, “Needle path planning for autonomous robotic surgical suturing,” in *2013 IEEE International Conference on Robotics and Automation*, pp. 1669–1675, IEEE, 2013.

[20] C. Shin, P. W. Ferguson, S. A. Pedram, J. Ma, E. P. Dutson, and J. Rosen, “Autonomous tissue manipulation via surgical robot using learning based model predictive control,” in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 3875–3881, IEEE, 2019.

[21] F. Alambeigi, Z. Wang, R. Hegeman, Y.-H. Liu, and M. Armand, “A robust data-driven approach for online learning and manipulation of unmodeled 3-d heterogeneous compliant objects,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4140–4147, 2018.

[22] S. A. Pedram, P. W. Ferguson, C. Shin, A. Mehta, E. P. Dutson, F. Alambeigi, and J. Rosen, “Toward synergic learning for autonomous manipulation of deformable tissues via surgical robots: An approximate q-learning approach,” *arXiv preprint arXiv:1910.03398*, 2019.

[23] A. Pandya, L. Reisner, B. King, N. Lucas, A. Composto, M. Klein, and R. Ellis, "A review of camera viewpoint automation in robotic and laparoscopic surgery," *Robotics*, vol. 3, no. 3, pp. 310–329, 2014.

[24] T. S. Ralston and N. J. Sanchez, "Autonomous ultrasound probe and related apparatus and methods," Nov. 17 2016. US Patent App. 14/714,150.

[25] 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.

[26] M. Haghighipanah, Y. Li, M. Miyasaka, and B. Hannaford, "Improving position precision of a servo-controlled elastic cable driven surgical robot using unscented kalman filter," in *2015 IEEE/RSJ Int'l conference on intelligent robots and systems (IROS)*, pp. 2030–2036, 2015.

[27] S. N. Kosari, S. Ramadurai, H. J. Chizeck, and B. Hannaford, "Control and tension estimation of a cable driven mechanism under different tensions," in *ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pp. V06AT07A077–V06AT07A077, American Society of Mechanical Engineers, 2013.

[28] J. Mahler, S. Krishnan, M. Laskey, S. Sen, A. Murali, B. Kehoe, S. Patil, J. Wang, M. Franklin, P. Abbeel, *et al.*, "Learning accurate kinematic control of cable-driven surgical robots using data cleaning and gaussian process regression," in *2014 IEEE International Conference on Automation Science and Engineering (CASE)*, pp. 532–539, 2014.

[29] R. A. Beasley and R. D. Howe, "Increasing accuracy in image-guided robotic surgery through tip tracking and model-based flexion correction," *IEEE Transactions on Robotics*, vol. 25, no. 2, pp. 292–302, 2009.

[30] A. R. Lanfranco, A. E. Castellanos, J. P. Desai, and W. C. Meyers, "Robotic surgery: a current perspective," *Annals of surgery*, vol. 239, no. 1, p. 14, 2004.

[31] R. A. B. R. D. Howe, "Model-based error correction for flexible robotic surgical instruments," in *Robotics: Science and Systems*, pp. 359–364, 2005.

[32] J. Hollerbach, W. Khalil, and M. Gautier, "Model identification," *Springer Handbook of Robotics*, pp. 321–344, 2008.

[33] S. Ramadurai, S. N. Kosari, H. H. King, H. J. Chizeck, and B. Hannaford, "Application of unscented kalman filter to a cable driven surgical robot: A simulation study," in *2012 IEEE International Conference on Robotics and Automation*, pp. 1495–1500, IEEE, 2012.

[34] M. Miyasaka, J. Matheson, A. Lewis, and B. Hannaford, "Measurement of the cable-pulley coulomb and viscous friction for a cable-driven surgical robotic system," in *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 804–810, IEEE, 2015.

[35] M. Miyasaka, M. Haghighipanah, Y. Li, and B. Hannaford, "Hysteresis model of longitudinally loaded cable for cable driven robots and identification of the parameters," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4051–4057, IEEE, 2016.

[36] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 421–436, 2018.

[37] S. Krishnan, R. Fox, I. Stoica, and K. Goldberg, "Ddco: Discovery of deep continuous options for robot learning from demonstrations," *arXiv preprint arXiv:1710.05421*, 2017.

[38] J. Sturm, C. Plagemann, and W. Burgard, "Unsupervised body scheme learning through self-perception," in *2008 IEEE International Conference on Robotics and Automation*, pp. 3328–3333, IEEE, 2008.

[39] P. Pastor, M. Kalakrishnan, J. Binney, J. Kelly, L. Righetti, G. Sukhatme, and S. Schaal, "Learning task error models for manipulation," in *2013 IEEE International Conference on Robotics and Automation*, pp. 2612–2618, IEEE, 2013.

[40] D. Seita, S. Krishnan, R. Fox, S. McKinley, J. Canny, and K. Goldberg, "Fast and reliable autonomous surgical debridement with cable-driven robots using a two-phase calibration procedure," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6651–6658, IEEE, 2018.

[41] S. Aoyagi, A. Kohama, Y. Nakata, Y. Hayano, and M. Suzuki, "Improvement of robot accuracy by calibrating kinematic model using a laser tracking system-compensation of non-geometric errors using neural networks and selection of optimal measuring points using genetic algorithm," in *2010 IEEE/RSJ International conference on intelligent robots and systems*, pp. 5660–5665, IEEE, 2010.

[42] D. Nguyen-Tuong and J. Peters, "Model learning for robot control: a survey," *Cognitive processing*, vol. 12, no. 4, pp. 319–340, 2011.

[43] D. Nguyen-Tuong, M. Seeger, and J. Peters, "Model learning with local gaussian process regression," *Advanced Robotics*, vol. 23, no. 15, pp. 2015–2034, 2009.

[44] G. Fang, X. Wang, K. Wang, K.-H. Lee, J. D. Ho, H.-C. Fu, D. K. C. Fu, and K.-W. Kwok, "Vision-based online learning kinematic control for soft robots using local gaussian process regression," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1194–1201, 2019.

[45] S. Vijayakumar, A. D'souza, and S. Schaal, "Incremental online learning in high dimensions," *Neural computation*, vol. 17, no. 12, pp. 2602–2634, 2005.

[46] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE transactions on robotics and automation*, vol. 12, no. 5, pp. 651–670, 1996.

[47] K. Hashimoto, T. Kimoto, T. Ebine, and H. Kimura, "Manipulator control with image-based visual servo," in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, pp. 2267–2271, IEEE, 1991.

[48] K. Hosoda and M. Asada, "Versatile visual servoing without knowledge of true jacobian," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94)*, vol. 1, pp. 186–193, IEEE, 1994.

[49] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[50] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor Fusion IV: Control Paradigms and Data Structures*, vol. 1611, pp. 586–607, International Society for Optics and Photonics, 1992.

[51] W. J. Wilson, C. W. Hulls, and G. S. Bell, "Relative end-effector control using cartesian position based visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 684–696, 1996.

[52] F. Chaumette, "Potential problems of stability and convergence in image-based and position-based visual servoing," in *The confluence of vision and control*, pp. 66–78, Springer, 1998.

[53] H. Dehghani, S. Farritor, D. Oleynikov, and B. Terry, "Automation of suturing path generation for da vinci-like surgical robotic systems," in *2018 Design of Medical Devices Conference*, pp. V001T07A008–V001T07A008, American Society of Mechanical Engineers, 2018.

[54] S. Leonard, K. L. Wu, Y. Kim, A. Krieger, and P. C. Kim, "Smart tissue anastomosis robot (star): A vision-guided robotics system for laparoscopic suturing," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 4, pp. 1305–1317, 2014.

[55] S. McKinley, A. Garg, S. Sen, D. V. Gealy, J. P. McKinley, Y. Jen, M. Guo, D. Boyd, and K. Goldberg, "An interchangeable surgical instrument system with application to supervised automation of multilateral tumor resection.," in *CASE*, pp. 821–826, 2016.

[56] T. Osa, C. F. Abawi, N. Sugita, H. Chikuda, S. Sugita, H. Ito, T. Moro, Y. Takatori, S. Tanaka, and M. Mitsuishi, "Autonomous penetration detection for bone cutting tool using demonstration-based learning," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pp. 290–296, IEEE, 2014.

[57] 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.

[58] J. Waninger, G. W. Kauffmann, I. A. Shah, and E. H. Farthmann, "Influence of the distance between interrupted sutures and the tension of sutures on the healing of experimental colonic anastomoses," *The American journal of surgery*, vol. 163, no. 3, pp. 319–323, 1992.

[59] N. Famaey and J. V. Sloten, "Soft tissue modelling for applications in virtual surgery and surgical robotics," *Computer methods in biomechanics and biomedical engineering*, vol. 11, no. 4, pp. 351–366, 2008.

[60] S. Hirai and T. Wada, "Indirect simultaneous positioning of deformable objects with multi-pinching fingers based on an uncertain model," *Robotica*, vol. 18, no. 1, pp. 3–11, 2000.

[61] P. Boonvisut and M. C. Çavuşoğlu, "Estimation of soft tissue mechanical parameters from robotic manipulation data," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 5, pp. 1602–1611, 2013.

[62] T. Wada, S. Hirai, S. Kawamura, and N. Kamiji, "Robust manipulation of deformable objects by a simple pid feedback," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 1, pp. 85–90, IEEE, 2001.

[63] P. Jiménez, "Survey on model-based manipulation planning of deformable objects," *Robotics and computer-integrated manufacturing*, vol. 28, no. 2, pp. 154–163, 2012.

[64] F. Alambeigi, Z. Wang, Y.-h. Liu, R. H. Taylor, and M. Armand, "Toward semi-autonomous cryoablation of kidney tumors via model-independent deformable tissue manipulation technique," *Annals of Biomedical Engineering*, pp. 1–13, 2018.

[65] F. Alambeigi, Z. Wang, R. Hegeman, Y.-H. Liu, and M. Armand, "Autonomous data-driven manipulation of unknown anisotropic deformable tissues using unmodelled continuum manipulators," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 254–261, 2019.

[66] D. Navarro-Alarcon, H. M. Yip, Z. Wang, Y.-H. Liu, F. Zhong, T. Zhang, and P. Li, "Automatic 3-d manipulation of soft objects by robotic arms with an adaptive deformation model," *IEEE Transactions on Robotics*, vol. 32, no. 2, pp. 429–441, 2016.

[67] A. X. Lee, H. Lu, A. Gupta, S. Levine, and P. Abbeel, "Learning force-based manipulation of deformable objects from multiple demonstrations," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 177–184, IEEE, 2015.

[68] V. G. Mallapragada, N. Sarkar, and T. K. Podder, "Robot-assisted real-time tumor manipulation for breast biopsy," *IEEE Transactions on Robotics*, vol. 25, no. 2, pp. 316–324, 2009.

[69] F. F. Khalil and P. Payeur, "Dexterous robotic manipulation of deformable objects with multi-sensory feedback-a review," in *Robot Manipulators Trends and Development*, InTech, 2010.

[70] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[71] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, p. 484, 2016.

[72] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[73] Y. Chebotar, K. Hausman, M. Zhang, G. Sukhatme, S. Schaal, and S. Levine, "Combining model-based and model-free updates for trajectory-centric reinforcement learning," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 703–711, JMLR. org, 2017.

[74] A. Nagabandi, G. Yang, T. Asmar, G. Kahn, S. Levine, and R. S. Fearing, "Neural network dynamics models for control of under-actuated legged millirobots," *CoRR*, vol. abs/1711.05253, 2017.

[75] N. Wahlström, T. B. Schön, and M. P. Deisenroth, "From pixels to torques: Policy learning with deep dynamical models," *arXiv preprint arXiv:1502.02251*, 2015.

[76] J. L. Speyer and D. H. Jacobson, *Primer on optimal control theory*, vol. 20. Siam, 2010.

[77] U. Eren, A. Prach, B. B. Koçer, S. V. Raković, E. Kayacan, and B. Açıkmeşe, "Model predictive control in aerospace systems: Current state and opportunities," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 7, pp. 1541–1566, 2017.

[78] R. Hedjar, "Adaptive neural network model predictive control," *International Journal of Innovative Computing, Information and Control*, vol. 9, no. 3, pp. 1245–1257, 2013.

[79] G. Chowdhary, M. Mühlegg, J. P. How, and F. Holzapfel, "Concurrent learning adaptive model predictive control," in *Advances in Aerospace Guidance, Navigation and Control*, pp. 29–47, Springer, 2013.

[80] 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.

[81] S. Schaal, "Learning from demonstration," in *Advances in neural information processing systems*, pp. 1040–1046, 1997.

[82] Z. Li, D. Glozman, D. Milutinovic, and J. Rosen, "Maximizing dexterous workspace and optimal port placement of a multi-arm surgical robot," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 3394–3399, IEEE, 2011.

[83] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[84] F. Conti, F. Barbagli, R. Balaniuk, M. Halg, C. Lu, D. Morris, L. Sentis, J. Warren, O. Khatib, and K. Salisbury, "The chai libraries," in *Proceedings of Eurohaptics 2003*, (Dublin, Ireland), pp. 496–500, 2003.

[85] A. J. Silva, O. A. D. Ramirez, V. P. Vega, and J. P. O. Oliver, "Phantom omni haptic device: Kinematic and manipulability," in *Electronics, Robotics and Automotive Mechanics Conference, 2009. CERMA'09.*, pp. 193–198, IEEE, 2009.

[86] S. Resnikoff, D. Pascolini, D. Etya'Ale, I. Kocur, R. Pararajasegaram, G. P. Pokharel, and S. P. Mariotti, "Global data on visual impairment in the year 2002," *Bulletin of the world health organization*, vol. 82, pp. 844–851, 2004.

[87] N. G. Congdon, D. S. Friedman, and T. Lietman, "Important causes of visual impairment in the world today," *Jama*, vol. 290, no. 15, pp. 2057–2060, 2003.

[88] C. N. Riviere, R. S. Rader, and P. K. Khosla, "Characteristics of hand motion of eye surgeons," in *Proceedings of the 19th Annual International Conference of the IEEE Engineering in Medicine and Biology Society.'Magnificent Milestones and Emerging Opportunities in Medical Engineering'(Cat. No. 97CH36136)*, vol. 4, pp. 1690–1693, IEEE, 1997.

[89] P. B. Hibbard, A. E. Haines, and R. L. Hornsey, "Magnitude, precision, and realism of depth perception in stereoscopic vision," *Cognitive Research: Principles and Implications*, vol. 2, no. 1, p. 25, 2017.

[90] S. Krag and T. T. Andreassen, "Mechanical properties of the human posterior lens capsule," *Investigative ophthalmology & visual science*, vol. 44, no. 2, pp. 691–696, 2003.

[91] M. Zare, M.-A. Javadi, B. Einollahi, A.-R. Baradaran-Rafii, S. Feizi, and V. Kiavash, "Risk factors for posterior capsule rupture and vitreous loss during phacoemulsification," *Journal of ophthalmic & vision research*, vol. 4, no. 4, p. 208, 2009.

[92] M. D. de Smet, J. M. Stassen, T. C. Meenink, T. Janssens, V. Vanheukelom, G. J. Naus, M. J. Beelen, and B. Jonckx, "Release of experimental retinal vein occlusions by direct intraluminal injection of ocriplasmin," *British Journal of Ophthalmology*, vol. 100, no. 12, pp. 1742–1746, 2016.

[93] M. D. de Smet, T. C. Meenink, T. Janssens, V. Vanheukelom, G. J. Naus, M. J. Beelen, C. Meers, B. Jonckx, and J.-M. Stassen, "Robotic assisted cannulation of occluded retinal veins," *PloS one*, vol. 11, no. 9, p. e0162037, 2016.

[94] K. Willekens, A. Gijbels, L. Schoevaerdts, L. Esteveny, T. Janssens, B. Jonckx, J. H. Feyen, C. Meers, D. Reynaerts, E. Vander Poorten, *et al.*, “Robot-assisted retinal vein cannulation in an in vivo porcine retinal vein occlusion model,” *Acta ophthalmologica*, vol. 95, no. 3, pp. 270–275, 2017.

[95] A. Gijbels, J. Smits, L. Schoevaerdts, K. Willekens, E. B. Vander Poorten, P. Stalmans, and D. Reynaerts, “In-human robot-assisted retinal vein cannulation, a world first,” *Annals of biomedical engineering*, vol. 46, no. 10, pp. 1676–1685, 2018.

[96] C.-W. Chen, Y.-H. Lee, M. J. Gerber, H. Cheng, Y.-C. Yang, A. Govetto, A. A. Francone, S. Soatto, W. S. Grundfest, J.-P. Hubschman, *et al.*, “Intraocular robotic interventional surgical system (iriss): Semi-automated oct-guided cataract removal,” *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 14, no. 6, p. e1949, 2018.

[97] C.-W. Chen, M. J. Gerber, A. Francone, M. Lee, A. Govetto, T.-C. Tsao, and J.-P. Hubschman, “Semiautomated optical coherence tomography-guided robotic surgery for porcine lens removal,” *Journal of Cataract and Refractive Surgery*, vol. 45, no. 11, pp. 1665–1669, 2019.

[98] Y. Ma, X. Chen, W. Zhu, X. Cheng, D. Xiang, and F. Shi, “Speckle noise reduction in optical coherence tomography images based on edge-sensitive cgan,” *Biomedical optics express*, vol. 9, no. 11, pp. 5129–5146, 2018.

[99] I. Park, H. K. Kim, W. K. Chung, and K. Kim, “Deep learning based real-time oct image segmentation and correction for robotic needle insertion systems,” *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4517–4524, 2020.

[100] T. S. Mathai, K. L. Lathrop, and J. Galeotti, “Learning to segment corneal tissue interfaces in oct images,” in *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pp. 1432–1436, IEEE, 2019.

[101] A. G. Roy, S. Conjeti, S. P. K. Karri, D. Sheet, A. Katouzian, C. Wachinger, and N. Navab, “Relaynet: retinal layer and fluid segmentation of macular optical coherence tomography using fully convolutional networks,” *Biomedical optics express*, vol. 8, no. 8, pp. 3627–3642, 2017.

[102] S. Masood, R. Fang, P. Li, H. Li, B. Sheng, A. Mathavan, X. Wang, P. Yang, Q. Wu, J. Qin, *et al.*, “Automatic choroid layer segmentation from optical coherence tomography images using deep learning,” *Scientific reports*, vol. 9, no. 1, pp. 1–18, 2019.

[103] D. Zang, G.-B. Bian, Y. Wang, and Z. Li, “An extremely fast and precise convolutional neural network for recognition and localization of cataract surgical tools,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 56–64, Springer, 2019.

[104] Z.-L. Ni, G.-B. Bian, X.-H. Zhou, Z.-G. Hou, X.-L. Xie, C. Wang, Y.-J. Zhou, R.-Q. Li, and Z. Li, "Raunet: Residual attention u-net for semantic segmentation of cataract surgical instruments," in *International Conference on Neural Information Processing*, pp. 139–149, Springer, 2019.

[105] E. Rahimy, J. Wilson, T. Tsao, S. Schwartz, and J. Hubschman, "Robot-assisted intraocular surgery: development of the iriss and feasibility studies in an animal model," *Eye*, vol. 27, no. 8, p. 972, 2013.

[106] J. T. Wilson, M. J. Gerber, S. W. Prince, C.-W. Chen, S. D. Schwartz, J.-P. Hubschman, and T.-C. Tsao, "Intraocular robotic interventional surgical system (iriss): Mechanical design, evaluation, and master–slave manipulation," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 14, no. 1, p. e1842, 2018.

[107] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[108] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

[109] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *International Conference on Learning Representations (ICLR)*, May 2016.

[110] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," in *2018 IEEE winter conference on applications of computer vision (WACV)*, pp. 1451–1460, IEEE, 2018.

[111] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.