Title

Optimization and Design for Automation of Brachytherapy Delivery and Learning Robot-Assisted Surgical Sub-Tasks

Author

Garg, Animesh

Publication Date

2016

Optimization and Design for Automation of Brachytherapy Delivery and Learning Robot-Assisted Surgical Sub-Tasks

by

Animesh Garg

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

in

Engineering – Industrial Engineering and Operations Research

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Ken Goldberg, Co-chair
Professor Alper Atamtúrk, Co-chair
Professor Pieter Abbeel
Professor Laurent El Ghaoui

Summer 2016

Optimization and Design for Automation of Brachytherapy Delivery and Learning Robot-Assisted Surgical Sub-Tasks

Abstract

Optimization and Design for Automation of Brachytherapy Delivery and Learning Robot-Assisted Surgical Sub-Tasks

by

Animesh Garg

Doctor of Philosophy in Engineering – Industrial Engineering and Operations Research

University of California, Berkeley

Professor Ken Goldberg, Co-chair

Professor Alper Atamtúrk, Co-chair

The goal of this dissertation is to enhance automation in healthcare applications, specifically: brachytherapy delivery for cancer treatment and robot-assisted surgery, used for over 500,000 procedures annually in US alone. Brachytherapy uses cutting-edge computer-assisted planning, but assumes a fixed hardware design. On the contrary, Robot-Assisted Surgery uses high-precision state-of-the-art hardware under complete manual control with little automation. This dissertation is a step towards addressing this gap using a combination of optimization and design. Case studies show that performance of autonomous systems can be improved by leveraging the interaction between optimization based algorithms and the design of hardware systems.

For Brachytherapy, this dissertation has developed a new approach for treatment delivery in Intracavitary Brachytherapy using patient specific 3D printed implants and implemented it on a clinical case of oral cancer. We present an algorithm to quantify reachability with straight-line needles for a given anatomy in prostate cancer. This dissertation integrates optimization based needle and dose planning algorithms in interstitial brachytherapy for prostate cancer using two methods of skew-line needle configuration implants: robot-assisted procedures and customized needle guides. The procedures are demonstrated on physical phantoms and performance is compared with an expert physician.

For Robot-Assisted Surgery, this dissertation highlights the interplay between the design of hardware to reduce uncertainty and optimization based motion planning to enable automated multi-throw suturing. We present a sequential convex programming based algorithm for optimizing curved needle trajectories and we also devise a novel mechanical needle guide, SNAP that reduces needle pose uncertainty by 3**x**. This dissertation introduces a novel algorithm, Transition State Clustering (TSC) that extracts the latent task structure from task demonstrations by segmenting robot trajectories using hierarchical clustering and fitting Gaussian mixture model to identify transitions. We extend TSC with deep learning to perform trajectory segmentation for multi-modal data consisting of kinematics and videos.

To my parents – Poonam and Naresh Garg,
for their support and sacrifice.

In a tribute to life and work of,
Dr. Jean Pouliot.

Contents

Acknowledgments

I am deeply grateful and immensely fortunate to have been mentored by and collaborated with many a great people en route to this dissertation.

I am foremost thankful to my advisor Ken Goldberg for his support, motivation and training. Thank you, Ken, for betting on me as a potential advisee and more importantly not giving up on me through the numerous ups and downs. He has left an indelible mark on my professional philosophy: choice of problems, the value of articulate communication, and enthusiasm for out-of-the-box methods, in addition to emphasizing excellence in teaching and leadership.

I am very grateful to Pieter Abbeel for teaching me foundations of robotics and always making time to meet, and offer insightful and genuine advice. He has been a great inspiration and a lesson in commitment to the vision, leading by his own example in his intense work and workout schedules.

I would like to thank Alper Atamtürk for being so kind & open to working with me on optimization problems. I cherish our in-depth discussions, both technical and career-related.

Laurent's convex optimization class has been one of the foundational classes of my graduate training and a focus of my research. I also thank him for being on both the Qualification Exam and Thesis Committee.

This dissertation would not have been complete without the support of exceptional collaborators at UCSF and UC Davis. I deeply appreciate the efforts from Jean Pouliot, Adam Cunha and Dr. Joe Hsu for giving me regular advice, training and education in clinical brachytherapy and access to cutting-edge resources. I want to pay my tribute to Jean Pouliot's life's work on cancer treatment of which this dissertation is only a small part.

I also want to express my gratitude to Dr. Doug Boyd at UC Davis for the many meetings on discussing problem choices for surgical automation, his enthusiastic reviews of our work and his willingness to make time for Berkeley trips to help us despite the hectic schedules.

I would also like to thank Henrik Christensen and Greg Hager. Henrik has been a source of constant academic mentorship and is the reason I ended up at Berkeley. And Greg has shared bits of his immense experience and knowledge in surgical robotics, preempting the problems I might have and guiding me through the academia.

I wish to thank Rhonda Righter and Lee Schruben not only for their great classes that have helped me in graduate studies but also their support and many consultations through the years.

I have benefited immensely from the guidance of postdocs and senior grads I have had the opportunity to work with. Dmitry Berenson was one of the first people at Berkeley who helped calm my anxieties and taught me the ways of research including how to write papers. Sachin Patil was a great mentor and friend – supporting me through the details of my work, critiquing my methods and suggesting existing work from his encyclopedic knowledge of the field.

My graduate experience would not have been half as fun without my close collaboration and even closer friendships with Timmy Siauw and Sanjay Krishnan. I worked with Timmy during the first three years of my Ph.D. on multiple projects relating to Cancer Treatment, spent nights on robot experiments and rode back of trucks for live robot demos. I started working with Sanjay starting my third year and we have since worked on many problems together – often outside both of our expertise. He has been a great source of inspiration to work harder, read more papers and has brought a fresh perspective to my work. My intimate interactions – numerous outings, discussions on research ideas, science fiction and the travails of graduate life – have forged close friendships with Timmy and Sanjay that I will cherish for many years to come.

I have been very fortunate to have had a chance to work with brilliant and interdisciplinary students in Ken's and Pieter's groups, Jeff Mahler, Lauren Miller, Michael Laskey, Stephen McKinley, Karthik Narayan, Teodor Moldovan, Ben Kehoe, John Schulman, David Gealy, Zoe McCarthy. I am very thankful for the intellectual conversations, feedback as well as the many friendships. I would like to mention the travels and discussions with Jeff Mahler, creative collaborations with Stephen McKinley and Lauren Miller. I also want to express my thanks to all the undergrads I worked with over the years. Especially Siddarth Sen, Adithya Murali, Richard Liaw and Yiming Jen have been tremendous in their research efforts and made me proud through their achievements since our collaboration started.

Besides my academic interactions, my friends Aakash, Ashish, Mohit, Vipul, Varun, Patik, Nabankur, Hitesh, Manish, Sahil and Bikash have helped me through life outside the lab. My sister Aditi has been a perennial source of encouragement. And support from Anju, my partner in crime, has been invaluable in keeping me going in times I was low, motivating me to exert and providing impartial criticism.

And finally, my parents, Naresh and Poonam Garg who willingly supported my decision to move to the US, encouraged me through the thick & thin and instilled in me the values which have led to this standing. I will forever be in their debt!

Chapter 1

Introduction

This dissertation applies optimization to two healthcare applications: radiation therapy and surgical robotics. Through case studies in these applications, this dissertation highlights that performance of autonomous systems can be improved by leveraging the interaction between optimization based algorithms and the design of hardware systems. Specifically, this dissertation presents new algorithms and hardware designs for High Dose-Rate Brachytherapy (HDR-BT) treatment for cancer and Subtask automation in Robot-assisted Minimally Invasive Surgery (RMIS).

1.1 High Dose-Rate Brachytherapy

Cancer is a major cause of death across the world with over 14 million cases in 2012 [1]. Among the various treatment options available, radiation therapy has proved to be an effective treatment modality with advancements in targeted delivery. Radiation therapy is used in two-thirds of cancer treatment procedures, either independently or in combination with another treatment modality [2].

Radiation delivery in cancer treatment can be broadly categorized by mode of radiation delivery as *External* and *Internal*. Each year over 500,000 patients are treated with Brachytherapy, a form of internal radiotherapy wherein radioactive sources are located proximal to tumor [3]. Brachytherapy is frequently used for tumors in prostate, cervix, breast, oral, nasopharynx, rectal, and gynecological tumors. Furthermore, brachytherapy is also commonly employed along with other treatment modalities including external radiotherapy and surgical interventions.

Brachytherapy procedures involve the use of radioactive sources temporarily proximal to the tumors to sufficiently irradiate the tumors while limiting radiation damage to healthy organs and tissues. And based on the strength of sources used, the procedures are classified as *Low Dose-Rate* (LDR) and *High Dose-Rate* (HDR). In LDR procedures, sources are placed and often left inside the body for long periods leading to the nomenclature: permanent implants. In contrast, HDR source(s) are threaded into catheters to reach a sequence of pre-planned locations and stay at each location for a short duration, in the order of seconds, and the source is finally retracted from the body. Multiple such dose fractions are delivered for a complete treatment plan over the course of several days. Brachytherapy procedures may be further, classified as *intracavitary*

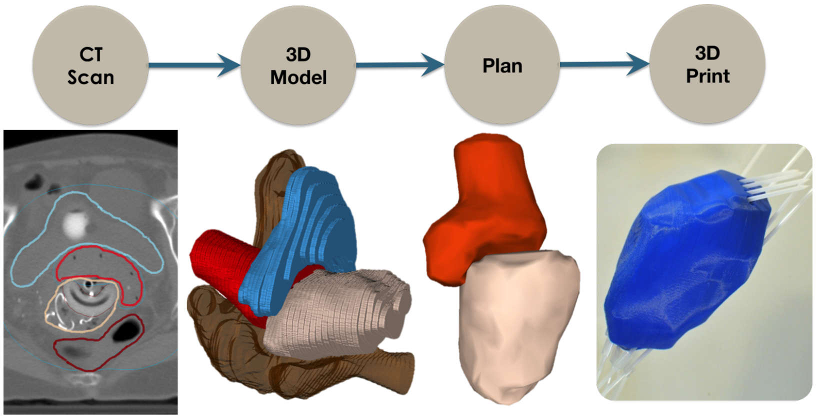


Figure 1.1: Intracavitary Brachytherapy: The sequence the steps in creating a personalized implant. A CT-Scan generates 3D anatomy models which are used for treatment planning and internal channel planning in the cavity. Thereafter, a cavity conformal implant is 3D printed.

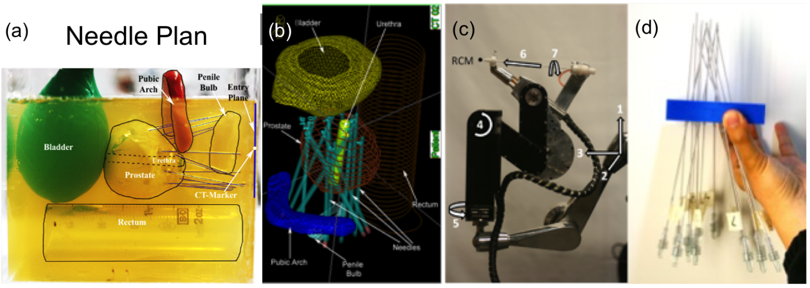


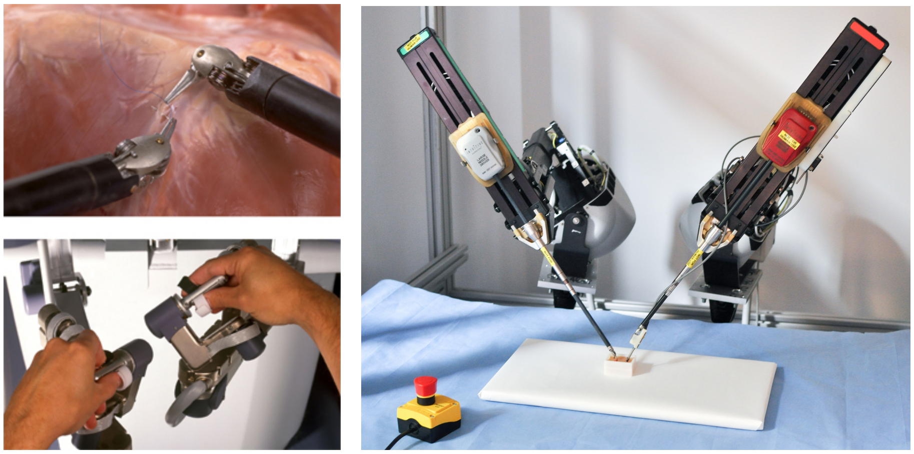
Figure 1.2: Interstitial Brachytherapy: Illustration of a prostate phantom (a) along with a needle plan (b). Two methods of needle configuration implant are shown in: (c) Robot-Assisted, and (d) Customized Needle Guide.

and *interstitial* depending on radioactive source placement inside the tissue or inside naturally occurring body cavities. Based on the tumor location, either form or a combination of both forms of brachytherapy procedures may be used.

This dissertation focuses on automation in HDR brachytherapy, both intracavitary and interstitial forms. The key insight behind this work is that current methods of dose planning implicitly assume that the source placement mechanism is fixed, hence the set of reachable dwell positions is artificially constrained. We present algorithms and systems to circumvent the hardware constraints of current clinical methods with a potential to reduce side effects and improve treatment quality. Specifically, this dissertation has developed a customized implants for treatment delivery in Intracavitary Brachytherapy and implemented it on a clinical case of oral cancer (Figure 1.1) [4]. Moreover, this is also the first to integrate optimization based needle and dose planning algorithms with robot-assisted skew-line needle configuration implants in interstitial brachytherapy for prostate cancer (Figure 1.2) [5].

1.2 Robot-assisted Minimally Invasive Surgery

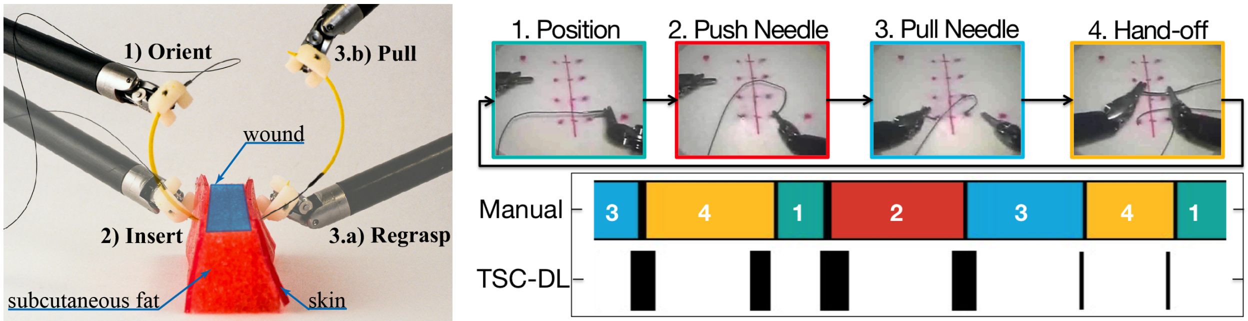
Robot-assisted Minimally Invasive Surgery (RMIS) was used in manual teleoperation mode in over 500,000 procedures worldwide in 2015 with 3600 systems [6]. Robotic surgical assistants (RSAs), such as the da Vinci system from Intuitive Surgical, address the ergonomic constraints of hand-held laparoscopic tools through the use of a master-slave configuration. RSAs provide surgeons with a precision laparoscopic tool that offers higher dexterity, tremor reduction and range of motion with immersive 3D visualization. Clinical RMIS systems have focused on not only general surgery but



(a) Da Vinci Master Console

(b) Da Vinci Research Kit (dVRK)

Figure 1.3: Da Vinci Surgical System: (a) Master console along with the visualization system as seen by the surgeon. (Image: Intuitive Surgical) (b) The image depicts the dVRK set-up with two arms at the UC Berkeley Autolab.



(a) Autonomous Suturing with da Vinci

(b) Unsupervised Trajectory Segmentation: Transition State Clustering

Figure 1.4: Robot-Assisted Suturing: (a) The image show the 4 steps in a single throw in the Suturing subtask on a tissue phantom. (b) Unsupervised Trajectory Segmentation: The first row shows a manual segmentation of the suturing task in 4 steps: (1) Needle Positioning, (2) Needle Pushing, (3) Pulling Needle, (4) Hand-off. *Transition State Clustering* extracts many of the important transitions without labels.

also a number of surgical specialties such as gynecologic surgery, urologic surgery, cardiothoracic surgery, and head and neck surgery.

RMIS has ushered in an era of shorter recuperation time, lesser blood loss, lower patient trauma, and lesser tissue injuryin [7–9]. Surgical outcome studies have shown preference for robot-assisted surgery such as kidneys [10], pancreas [11] hysterectomy [12], and uterine fibroids [13]. Regardless of the benefits, RMIS requires skilled surgeons to perform tediously long procedures with reduced sensory perception during surgical manipulation. Supervised automation in RMIS has the potential to reduce the time required for surgical procedures, reducing the time patients are under anesthesia and associated costs and contention for operating room resources. Partial automation in RMIS has the potential to be the next generation of tools, akin to having an intelligent, personalized, and modular surgical assistant rather than a replacement for the surgeon. The key aim to introduce algorithmic automation, learning and design in RMIS is to enable low-latency telesurgery, reduction in surgeon's training requirement and cognitive load.

This dissertation contributes towards progress in sub-task automation in RMIS. First, we highlight the interplay between the design of hardware solutions, that minimize uncertainty, and optimization based motion planning to enable automated multi-throw suturing on a Da Vinci system (Figure 1.4a) [14]. The system uses a novel mechanical needle guide design to minimize needle pose uncertainty and a sequential convex optimization framework to optimize needle size, needle trajectory and control parameters for two arms.

Furthermore, robot learning from raw trajectory data is challenging due to temporal and spatial inconsistencies. A key problem is extracting conceptual task structure from noisy human demonstrations. This dissertation also presents a novel algorithm, Transition State Clustering (TSC) that extracts the latent task structure(Figure 1.4b). TSC proposes a Switched Linear Dynamical System (SLDS) characterization of the demonstrations [15]; the key insight being that switching events induce a density over the state space. When the model noise and transition noise are Gaussian, this reduces to hierarchical clustering as shown for a Suturing example in Fig 1.4b. These clusters encode two important aspects: (precondition) the state of the robot prior to the transition and (post-condition) the state of the environment when beginning a new segment. This gives a notion of necessary conditions for task success, where a robot has to ensure that it satisfies the post-condition before proceeding.

1.3 Summary of Contributions

An outline of the main contributions of this dissertation is presented briefly here along with the corresponding peer-reviewed publications and co-authors.

1. **Customized Implants for Brachytherapy** This dissertation is the first to present a new approach for Intracavitary Brachytherapy therapy that leverages 3D printing and steerable needle motion planning to create patient-specific bio-compatible implants.

a) **Algorithm**: We present the Channel Layout Algorithm (CLA) computing channels to precisely guide radioactive sources to target dwell positions inside the printed channels. CLA is a probabilistic algorithm based on rapidly-expanding randomized trees (RRT) that outputs a set of curvature-constrained channels inside the implant volume.
b) **Simulated Experiments**: A simulated case of prototypical OB/GYN cervical & vaginal cancer with three treatment options: standardized ring implant (current practice), customized implant with linear channels, and customized implant with curved channels. Results with a two-parameter coverage metric suggest that customized implants with curved channels can offer significant improvement over current practice.
c) **Physical Experiments**: This method has been used on two patients of Oral-cancer in a clinic for post-surgery tumor ablation.

This work was performed in collaboration with UCSF Department of Radiation Oncology and is discussed in Chapter 2. This work was published in:

- Animesh Garg, Sachin Patil, Timmy Siauw, J Adam M Cunha, I Hsu, Pieter Abbeel, Jean Pouliot, and Ken Goldberg. “An Algorithm for Computing Customized 3D Printed Implants with Curvature Constrained Channels for Enhancing Intracavitary Brachytherapy Radiation Delivery”. In: *Proc. IEEE Int. Conf. Au-*

tomation Science and Engg. (CASE). Aug. 2013, pp. 466–473

- Katherine Mellis, Timmy Siauw, Atchar Sudhyadhom, Rajni Sethi, I-Chow Hsu, Jean Pouliot, Animesh Garg, and Ken Goldberg. "Material Evaluation of PC-ISO for Customized, 3D Printed, Gynecologic 192Ir HDR Brachytherapy Applicators". In: *Journal of Applied Clinical Medical Physics (JACMP)* (2014)

2. **Robot-Assisted HDR Brachytherapy** This dissertation presents a first integrated brachytherapy system that leverages optimization based needle and dose planning algorithms with the Acubot-RND needle guiding robot.
   a) **Algorithm** We present a systematic integration of two optimization algorithms: Needle Planning and Inverse Dose Planning with robot-assisted skew-line needle implants to efficiently deliver radiation to the prostate while minimizing trauma to sensitive structures such as the penile bulb.
   b) **Physical Experiments** We evaluate our system on custom designed anatomically-correct phantom models. We perform two robot-assisted implants and compare them with an expert physician. We find that robot-assisted implants achieve an average source placement error of 3.5mm and meet clinical dose requirements without puncturing healthy organs-at-risk (penile bulb).

This work was in collaboration with UCSF Department of Radiation Oncology. It is discussed in Chapter 3 and was published in:

- Animesh Garg, Timmy Siauw, Dmitry Berenson, J Adam M Cunha, I-C Hsu, Jean Pouliot, Dan Stoianovici, and Ken Goldberg. "Robot-Guided Open-Loop Insertion of Skew-Line Needle Arrangements for High Dose Rate Brachytherapy". In: *IEEE Trans. on Automation Science and Engineering (TASE)* 10.4 (Oct. 2013), pp. 948–956
- An initial version of the TASE paper was published in CASE 2012 [17].

3. **Customized Needle Guides for HDR-BT** This dissertation also presents a new a concept of patient-specific 3D printed customized needle guide (CNG) for a skew-line needle configuration output from the needle planning and dose planning optimization algorithms.

   a) **Design** We present an algorithmic method to create patient-specific customized needle guides and also present a workflow to integrate optimization-based planning with CNG.
   b) **Physical Experiments** We performed a similar analysis to evaluate the placement error associated with inserting needles with a custom needle guide and the effect this error has on the number of structures punctured and the ability to meet treatment objectives. We observed that in the 4 physical experiments on phantom models, all four cases met clinical dose requirements with the RMS error for manual insertion without feedback ranging between 2.3 mm to 4.5 mm.

This work was in collaboration with UCSF Department of Radiation Oncology. Chapter 3 describes the details and it was published as an abstract in:

- Timmy Siauw, J. Adam M. Cunha, Animesh Garg, Ken Goldberg, I Hsu, and Jean Pouliot. "Customized Needle Guides for Inserting Non-Parallel Needle Arrangements in Prostate HDR Brachytherapy: A Phantom Study". In: *Brachytherapy* 13 (2014)

4. **Reachability Analysis for Needle Placement in HDR-BT** We quantify the reachability in a given anatomical setup of tumor volume in the presence of a single healthy organ to avoid with all possible skew-line configurations from a pre-specified entry zone.

a) **Algorithm** Assuming linear needles, convex polyhedral representations of entry zone, organs-at-risk and target volume, we present an exact polynomial time algorithm for checking existence and calculation of the non-reachable set in the target volume.

b) **Simlation Experiments** We perform experiments using patient data from 18 brachytherapy cases and found that 11 cases had non-empty occluded volume inside the target ranging from 0.01% to 4.3% of the target volume. We also report a sensitivity analysis showing the change in the occluded volume with dilation of the avoidance volume and entry zone.

This is discussed in Chapter 4 and was published at

- Animesh Garg, Timmy Siauw, Guang Yang, Sachin Patil, J Adam M Cunha, I-Chow Hsu, Jean Pouliot, Alper Atamtürk, and Ken Goldberg. "Exact Reachability Analysis for Planning Skew-Line Needle Arrangements for Automated Brachytherapy". In: *Proc. IEEE Int. Conf. Automation Science and Engg. (CASE).* 2014

5. **Supervised Automation of Muti-Throw Suturing** This dissertation presents a framework for supervised automation of the multi-throw suturing task

a) **Algorithm** We present a sequential convex programming based algorithm for optimizing the choice of needle size and trajectories of these curved needles.

b) **New hardware concept** We present the design for Surgical Angular Needle Positioner (SNAP), a novel mechanical needle guide. We track the needle using vision and achieve real-time pose estimate within 5°error. And the SNAP improves repeatability in needle grasping by 10× and reduced needle pose uncertainty by 3**x**.

c) **Physical Experiments** We evaluate the algorithm and SNAP on a da Vinci Research Kit using tissue phantoms and compare completion time with that of humans from the JIGSAWS dataset [20]. Initial results suggest that the dVRK can perform suturing at 30% of human speed while completing 86% suture throws attempted.

Chapter 5 described the algorithm and design details. The work was published in:

- Siddarth Sen*, Animesh Garg*, David Gealy, Stephen McKinley, Yiming Jen, and Ken Goldberg (*denotes equal contribution). "Autonomous Multiple-Throw Multilateral Surgical Suturing with a Mechanical Needle Guide and Optimization based Needle Planning". In: *Proc. IEEE Int. Conf. Robotics and Automation (ICRA).* 2016

6. **Transition State Clustering for Unsupervised Trajectory Segmentation** This dissertation develops a new unsupervised segmentation algorithm, Transition State Clustering (TSC), which combines results from hybrid dynamical systems and Bayesian non-parametric statistics to segment kinematic recordings of robotic surgical procedures.

a) **Algorithm** We present the TSC algorithm, that models demonstration trajectories as noisy observations of an underlying switched linear dynamical system (SLDS) and clusters them into spatially and temporally similar transition events (i.e., switches in the linear regime). TSC uses a hierarchical Dirichlet Process Gaussian Mixture Model to avoid selecting the number of segments *a priori*.

b) **Evaluation on Real Data** TSC is evaluated against 5 state-of-the-art techniques and we find that TSC recovers the ground truth 49% more accurately these alternatives in the presence of corrupted with process and observation noise. Furthermore, TSC runs 100x times faster than the best performing alternate method. We also evaluated TSC on 67 recordings of the surgical needle passing and suturing. On this dataset with manually annotated visual features, TSC finds 83% of the needle passing transitions and 73% of the suturing transitions annotated by human experts.

TSC models and details are presented in Chapter 6 and these contributions were published at:

- Sanjay Krishnan*, Animesh Garg*, Sachin Patil, Colin Lea, Gregory Hager, Pieter Abbeel, and Ken Goldberg (*denotes equal contribution). "Transition State Clustering: Unsupervised Surgical Trajectory Segmentation For Robot Learning". In: *Int. Symp. on Robotics Research (ISRR)*. Springer STAR. 2015
- An extended version of the ISRR 2015 paper is under review at Int. Journal of Robotics Research 2016.

7. **TSC over visual state space with Deep Learning** We build on the Transition State Clustering algorithm to extend its application to unannotated video and kinematic data to segment trajectories into locally-similar contiguous sections.

a) **Algorithm** We present a novel framework TSC-DL that finds regions of the visual feature space that correlate with transition events using features automatically constructed from layers of pre-trained image classification Deep Convolutional Neural Networks (CNNs).
b) **Evaluation on Real Data** We evaluate TSC-DL on real surgical datasets and observe that it result in up-to a 30.4% improvement in Silhouette score for clustering.

Chapter 7 discusses these extensions and these contributions were published at:

- Animesh Garg*, Sanjay Krishnan*, Adithyavairavan Murali, Trevor Darrell, Pieter Abbeel, and Ken Goldberg (*denotes equal contribution). "On Visual Feature Representations for Transition State Learning in Robotic Task Demonstrations". In: *NIPS 2015 Workshop on Feature Extraction*
- Adithyavairavan Murali*, Animesh Garg*, Sanjay Krishnan*, Florian Pokorny, Pieter Abbeel, Trevor Darrell, and Ken Goldberg (*denotes equal contribution). "TSC-DL: Unsupervised Trajectory Segmentation of Multi-Modal Surgical Demonstrations with Deep Learning". In: *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*. 2016

Part I

Optimization Algorithms for Automation in Brachytherapy

Chapter 2

Customized 3D Printed Implants for Intracavitary Brachytherapy

Overview

Each year, over $500,000$ cancer patients worldwide are treated with brachytherapy [3], a form of internal radiotherapy where small radioactive sources are placed close to tumors (brachys: Greek for proximal). Brachytherapy is widely used to treat cancer in a number of anatomical sites: interstitial locations such as prostate, breast, liver, brain; and intracavitary locations such as nasal and oral cavity, cervix, and the vaginal canal [23]. This chapter presents an algorithm for computing customized 3D-printed implants with curvature constrained channels for enhancing intracavitary brachytherapy radiation delivery.

In current practice for intracavitary brachytherapy, standardized applicators with internal channels are inserted into body cavities to guide the sources. A radioactive source is then guided through the needle or implant channel using an attached wire and precisely controlled by an automated afterloader that causes the source to dwell for specified times at specified points along the needle or channel to deliver the desired radiation dose.

These standardized implants are one-size-fits-all and are prone to shifting inside the body, resulting in suboptimal dosages. As we describe later in this chapter, existing clinical methods employ standardized implants that do not conform to the patient anatomy allowing for relative movement, and only offer a fixed set of possible dwell position options for placing sources (see Figure 2.2a). Furthermore, biological effectiveness requires the prescribed dose be divided into 2-4 iterations and delivered with intervening gaps of 5-6 hours. In existing practice, patients are required to remain immobile over the course of treatment to maintain the geometric positions between anatomy and sources. Another limitation is that treatment quality depends on precisely positioning the sources to sufficiently irradiate the tumors while minimizing radiation delivered to healthy organs and tissues. As noted by Magne et al. [24], “the proper placement of the applicator within vagina is the most important first step to avoid tumor underdosage or excessive dose to critical organs".

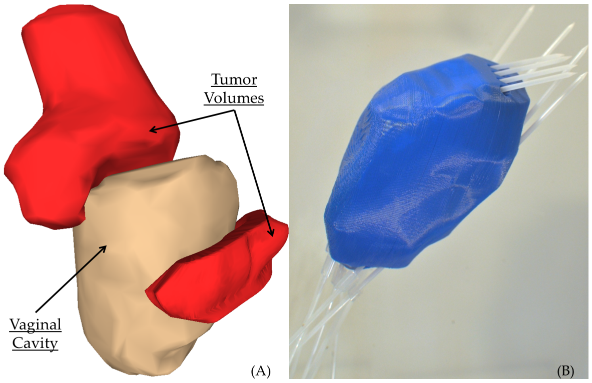


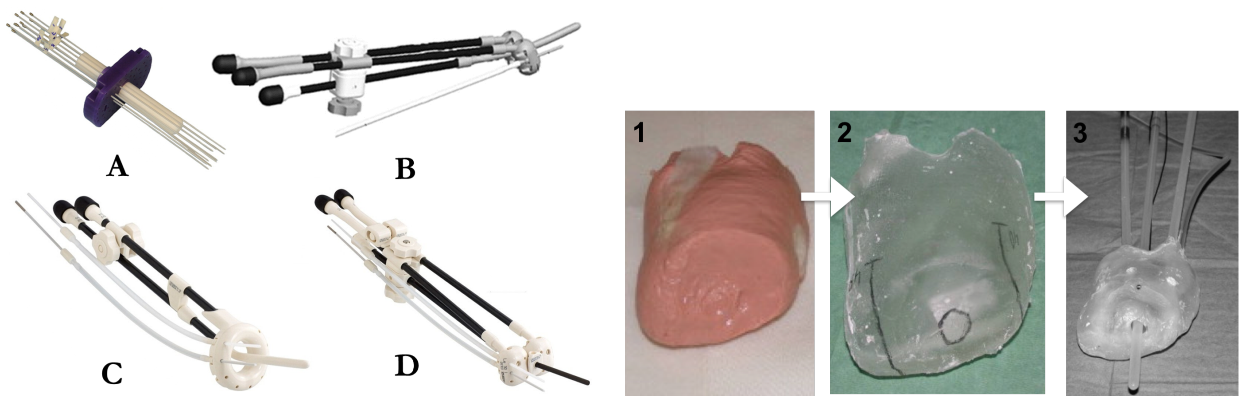
Figure 2.1: Case study for OB/GYN cancer. Left: 3D model of customized implant for treating tumors of the cervix and endometrium of the vaginal cavity. The figure shows an anatomical configuration of the vaginal canal (in buff) with two tumors (in red), one above the cervix (top) and one on the vaginal sidewall. Right: Picture of a customized implant with 11 curvature constrained channels generated by the algorithm (8 catheters are placed for visual aid) for the anatomy on the left. The radioactive source (seed) can be precisely guided through each channel sequentially to precisely deliver treatment to the tumors.

In this chapter, we present a new approach for HDR-BT intracavitary treatment that builds on recent results in 3D printing and steerable needle motion planning to design customized implants with interior curvature-constrained channels that can fit precisely and guide radioactive sources to customized dwell points proximal to cancerous tumors. Such curved channels have potential to reach targets that may not be reachable with existing methods. We present an algorithm for computing curvature constrained channels that fit inside the specified implant geometry and meet dose and delivery requirements. Figure 2.1 illustrates an OB/GYN case study with typical cervical and vaginal tumors (the approach is also relevant to others tumor locations in intracavitary HDR-BT). Comparison with standardized ring implant (current practice) with a two-parameter coverage metric suggest that customized implants with curved channels can offer significant improvement.

2.1 Background and Related Work

There are a number of commercially available implants/applicators for treating cervical and vaginal cancers, for e.g. Fletcher applicators [25], Utrecht applicator [26], Vienna applicator [27] and Mold type applicators [24]. These standardized implants can be combined with linear catheters as illustrated in Figure 2.2a. Used by many radiation oncologists, these intracavitary applicators include an intrauterine tandem and intravaginal ovoids, and produce a pear-shaped dose distribution centered on the cervix, allowing a high dose to be delivered to the cervix while sparing bladder and rectum. Although these systems allow some adaptation to patient anatomy, in correct placement and patient movement (and filling of bladder and bowels) can cause shifts in the applicator position and hence result in undesired doses.

An innovative approach is described by Magne et al. [24], which proposes use of a customized implant created using vaginal impression with plaster that accurately shows the topography and extension of tumors and the specific anatomy of the vagina and cervix as shown in Figure 2.2b. A silicone implant is made using this plaster mold, and two linear catheters and tandem shaft are inserted by the oncologist into the implant. The authors report decreased relative movement of implant while the patient is mobile over three days, thereby improving conformity between planned and delivered dose distributions. Treatment of patients with tumor extensions to the endometrial



(a) Standardized implants.

(b) French Mold Applicator

Figure 2.2: (a) Four standardized templates/applicators/implants for gynecological brachytherapy. (A) Vaginal cylinder applicator with 8 parallel catheters, (B & D) Ovoids applicator, (C) Ring applicator. B,C&D also have a uterine tandem and allow for interstitial catheters. The uterine tandem provides a channel for dwell positions inside the uterine canal. The ring (C) and ovoids (B & D) act as guides for inserting catheters into the tissue surrounding the cervix. (b) The procedure to create a French Mold Applicator [24]. This applicator is prepared manually and unlike our method no computation is involved in channel placement.

tissue of the vaginal wall often requires two separate implants if treated with standard applicators. A custom implant allows the oncologist to account for tumor extensions in a single iteration. The authors report their experience with more than 5000 patients and note that their method has three main advantages: patient tailored treatment, MRI procedure compatibility without image quality disturbance, and increased patient comfort. We note that Magne et al prepare the mold implant manually and correct placement of catheters is highly dependent on oncologist's experience.

Recent advances in 3D printing (also known as additive manufacturing) are poised to have a major impact on many fields as described in introductions by Jacobs [28] and Lipson [29]. Non-toxic, FDA approved materials are allowing 3D printed parts to be used for medical applications [30] such as bone replacement and oral surgery implants.

In contrast to Magne et al., we explore an extension where MRI/CT scans are used to reconstruct precise 3D model of patient anatomy (or the plaster cast may be scanned). This model is provided as input to our algorithm for computing a set of internal curved channels that can be embedded in a biocompatible implant of the same shape as the vaginal volume.

Motion Planning for Needle Steering: A growing body of research has been reported on motion planning for steering needles [31–33]. The objective is to steer a flexible needle with curvature constraints through tissue to internal targets by exploiting asymmetries at the needle tip. Such needles can reach targets that cannot be reached by stiff linear needles. The needle is a nonholonomic system and is related to motion planning for fixed-wing aircraft [34].

Furthermore, the radiation source for HDR brachytherapy for treatment of GYN tumors is typically an ^{192}Ir core embedded in a steel capsule 0.9 mm in diameter and $\sim$5.0 mm in length [35] as shown in Figure 2.3. The cylindrical geometry imposes curvature constraints on the channels;

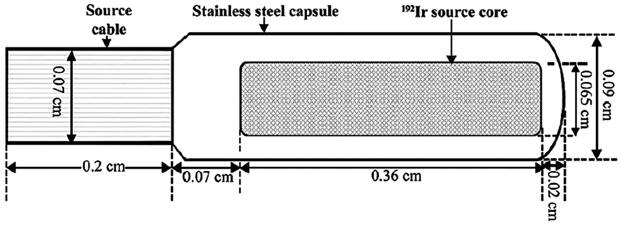


Figure 2.3: Schematic of a typical ^{192}Ir source used in GYN Brachythrapy [35](Permission pending).

given a channel diameter of 2.5 mm, we calculate the minimum local curvature as 10 mm.

Computing a set of internal channels is a similar problem in that curvature is constrained but has the distinct advantage that there is no uncertainty due to tissue properties or needle mechanics: channels can be printed with high accuracy. It is also important that channels do not intersect. We build on prior work by Patil et al. [32] which uses rapidly exploring random trees (RRT) [36] for planning curvature constrained paths for steerable needles [33].

2.2 Problem Statement

The objective is to compute a set of non-intersecting curvature-constrained channels within the implant that reach targets proximal to tumors for delivery of radiation and if needed, a report of which tumor zones cannot be reached.

The input is the registered pre-operative geometry from a combination of 3D scan of the plaster cast and CT (or MRI) scan of the patient. This input includes external geometry of the implant specified as a triangle mesh; the desired entry zone at the base of the implant for all channels; and the locations of tumors and organs-at-risk (OAR) (vaginal wall, cervix, rectum, urethra, bladder, uterus). The channel layout problem can then be stated as below.

Objective: Given a 3D model of the implant volume I, which may include internal voids that will be treated as obstacles for channels, a set of 3D cancerous tumors that require radiation treatment $\mathcal{T}$, a specification of the entry region at the base of the implant E, the maximum allowable entry angle (deviation from normal) α, the minimum radius of curvature of the channel, r_{min}, and the channel diameter, w, corresponding to the width of the catheter carrying the source, the objective is to compute a set of non-intersecting curvature constrained channels $\mathcal{C} = \{C_1, C_2, \ldots, C_N\}$ starting from E that lie within I and are proximal to as much of the set $\mathcal{T}$ as possible.

Coverage Quality Metric The ability to deliver radiation doses depends on the arrangement of potential source dwell points and their proximity to tumors. The radiation dosage is assumed to follow an inverse square law. We measure the quality of an implant by the percentage of tumor volume that is "covered" by the set of dwell points, where coverage is a function of the distance between a dwell point (source) and a tumor point (target). Higher quality reduces the maximum dwell time needed to treat tumors and the potential for hot spots that can harm healthy tissue. The use of alternate dose models such as inverse dose planning [35] is deferred to future work.

To compare implants and channels for a given set of tumors $\mathcal{T}$, we consider the set of reachable dwell positions and how thoroughly they "cover" the set of tumors. Consider a set of reachable dwell positions $\mathcal{S}$ (for instance in the case of 3D printed implants these are evenly spaced inside reachable dwell segments). We discretize the set of tumors into a set of evenly spaced points dT. We quantify the proximity of a dwell position dS from a tumor point dT with the "coverage radius" δ such that: if dS lies within a ball of radius δ centered at dT, then dS is said to cover dT. Hence the cover of dT is the set

$$\texttt{cover}(dT,\delta) = \{dS : \|dS - dT\|_2 \leq \delta, dS \in \mathcal{S}\}. \tag{2.1}$$

It is also helpful to consider cases where tumor points can be covered by multiple dwell points, say n. We define the quality of coverage $\mathcal{Q}(n,\delta)$ as the percentage of tumor volume such that each tumor point dT within that volume $\mathcal{T}'$ ($\mathcal{T}' \subseteq \mathcal{T}$) is covered by at least n dwell positions within a ball of radius δ centered at dT. Hence,

$$\mathcal{Q}(n,\delta) = \frac{1}{|\mathcal{T}|}\int_{\mathcal{T}} I\{|\texttt{cover}(dT,\delta)| \geq n\}\, dT, \tag{2.2}$$

where $I\{\cdot\}$ is the indicator function and $|\cdot|$ is set cardinality. Reaching 100% coverage with smaller radiation radius and more dwell positions can reduce occurrence of hot spots and increase dose conformation to the tumor geometry to spare healthy tissue.

2.3 Channel Layout Algorithm (CLA)

The Channel Layout Algorithm (CLA) is summarized in Algorithm 1. The first step is generating a set of dwell segments proximal to the given set of tumors. Starting from the dwell segment most distal to the entry zone, we use the curvature constraints to construct an RRT backward from the segment toward the entry zone, stopping if/when we find a channel that avoids obstacles. We then treat this channel as an obstacle and consider the next dwell segment until all dwell segments are considered. We describe each step in detail below.

`generate_dwell_segments`$(\cdot)$: We start by computing a candidate set of dwell segments, which are linear segments near tumors that may include multiple potential source dwell positions. We can also consider curved dwell segments and segments in alternate orientations.

Given the set of tumors $\mathcal{T}$ and the implant volume I, we compute the set of dwell segments $\mathcal{D}$ as follows. We discretize the implant volume with a regular voxel grid, where each voxel is a cube of side length equal to the channel width w. The surface of the implant volume is represented as a triangle mesh. We mark all triangles on implant whose surface normals intersect with tumor mesh. All marked triangles are then projected inwards in opposite direction to their surface normals by a distance $w/2$ to account for the channel width. All voxels intersecting with projected triangles are noted.

These voxels represent a discretization of the volume that should ideally be covered with the dwell segments. We then select a set linear segments with a greedy heuristic that covers the marked voxels. In general an optimal solution to this choice is an integer program, also known as the "pencil packing problem," which can in general case be NP-hard [37].

For every dwell segment in $\mathcal{D}$, we compute a channel inside the implant volume that reaches it or a report that no channel can be found. We consider the dwell segments in decreasing order

Algorithm 1: $\mathcal{C} \leftarrow \texttt{channel\_layout}(I, E, \mathcal{T}, r_{\min}, w)$

```
$\mathcal{D} \leftarrow \texttt{generate\_dwell\_segments}(I, \mathcal{T})$
$\mathcal{C} = \emptyset$
forall $d \in \mathcal{D}$ do
    $\mathcal{X} \leftarrow \emptyset$
    $\mathcal{X} \leftarrow \texttt{add\_vertex}(X_d)$
    while $((\mathbf{p}_{new} \in E) \wedge \textit{permissible}(R_{new}))$ do
        $\mathbf{p}_{\text{rand}} \leftarrow \texttt{random\_point\_in\_}\mathbb{R}^3(I, \mathcal{C})$
        $X_{\text{near}} \leftarrow \texttt{nearest\_neighbor}(\mathbf{p}_{\text{rand}}, \mathcal{X}, r_{\min})$
        $X_{\text{new}} \leftarrow \texttt{circular\_arc}(X_{\text{near}}, \mathbf{p}_{\text{rand}})$
        if $\textit{collision\_free}(X_{near}, X_{new}, I, \mathcal{C})$ then
            $\mathcal{X} \leftarrow \texttt{add\_vertex}(X_{\text{new}})$
            $\mathcal{X} \leftarrow \texttt{add\_edge}(X_{\text{near}}, X_{\text{new}})$
    $C_d \leftarrow \texttt{build\_channel}(\mathcal{X}, X_{\text{new}}, w)$
    $\mathcal{C} \leftarrow \mathcal{C} \cup C_d$
return $\mathcal{C}$
```

of distance from the entry region E. The medial axis of each curvature constrained channel can be parameterized as a sequence of circular arcs $\{\Psi_1, \Psi_2, \dots, \Psi_n\}$ in $\mathbb{R}^3$, where each circular arc Ψ_i is parameterized as a tuple $[l_i, \phi_i, r_i]^T$ (Figure 2.4). Here, l_i is the length of the arc, $r_i > r_{\min}$ is the radius of the arc, and ϕ_i is the twist applied to the tangential frame at the end of Ψ_i that rotates the plane containing the arc Ψ_i to the plane that contains the arc Ψ_{i+1}. The channel is constructed by sweeping a circle of diameter w along the medial axis.

Although the channels are constructed in 3D space, the state space of the layout problem comprises of both the position and orientation in $SE(3)$ because of the constraints on the channel curvature. The position and orientation constraint at the end of each dwell segment $d \in \mathcal{D}$ can be described as $X_d = \left[\begin{smallmatrix} R_d & \mathbf{p}_d \\ \mathbf{0} & 1 \end{smallmatrix}\right] \in SE(3)$ comprising of the position $\mathbf{p}_d$ of the end of the segment and rotation matrix R_d encoding the orientation of the dwell segment in 3D. Without loss of generality, we assume that the dwell segment d is oriented along the $\mathbf{z}$-axis of the local coordinate frame attached to the end of dwell segment.

Recent results in motion planning for nonholonomic systems emphasize sampling-based methods such as the Rapidly-exploring Random Tree (RRT) planner [36] where the probability of finding a solution converges to one, if such a solution exists, as the number of samples approaches infinity. We employ this approach building on an algorithm to compute curvature constrained needle paths in $\mathbb{R}^3$ [32]. Given a dwell segment $d \in \mathcal{D}$, we use the planner to compute the medial axis of the channel while staying within the implant volume and avoiding obstacles and the set of existing channels $\mathcal{C}$ in the environment. We plan backward starting from the dwell segment d to the entry region E because the larger entry region is less constrained.

Given initial state X_d and entry region, the algorithm incrementally builds a tree $\mathcal{X}$ over the

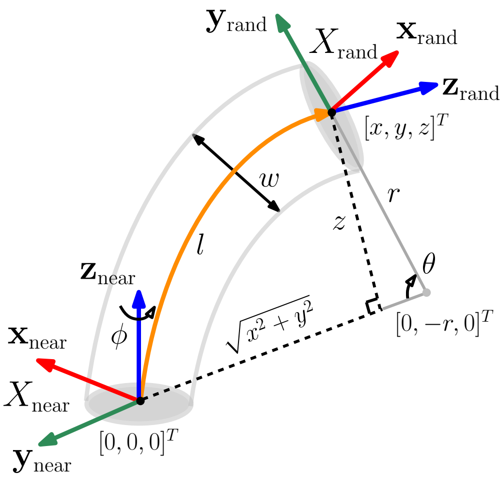


Figure 2.4: The medial axis of each channel is parameterized with a sequence of circular arcs $\{\Psi_1, \Psi_2, \ldots, \Psi_n\}$. We show one such circular arc here (orange) parameterized as a tuple $[l, \phi, r]$. The channel is obtained by sweeping a disk of diameter w along the length of the arc. This arc connects the state $X_{\text{near}} \in SE(3)$ at the nearest tree node to the randomly sampled point $\mathbf{p}_{\text{rand}} \in \mathbb{R}^3$. We assume that the medial axis of the channel is oriented along the local $\mathbf{z}$-axis at each point along the arc. The circular arc is constructed by rotating the local frame X_{near} by an angle θ around a line parallel to the local $\mathbf{x}$-axis and passing through the point $[0, -r, 0]^T, r > r_{\text{min}}$. The rotation ϕ rotates the tangential frame at the end of one circular arc to align it with the plane that contains the subsequent circular arc.

state space, while conforming to nonholonomic motion constraints of the system and avoiding obstacles. As described in Patil et al. [32], building the tree in the $SE(3)$ state space directly is computationally inefficient, so we sample a random point $\mathbf{p}_{\text{rand}} \in \mathbb{R}^3$ rather than a random state $X_{\text{rand}} \in SE(3)$. The planner then identifies a node in the tree X_{near} that is closest to the sample $\mathbf{p}_{\text{rand}}$, as defined by a specified distance metric $\boldsymbol{\rho}[\cdot]$. The sample $\mathbf{p}_{\text{rand}}$ is then connected to X_{near} using a circular arc parameterized by the tuple $[l, \phi, r]^T$. If the circular arc does not collide with the implant volume or existing channels and the minimum clearance from the obstacles is at least the channel width w, we add the arc as an edge in the tree. This process is repeated until either the tree $\mathcal{X}$ connects X_d and E or the available computation time is exceeded, in which case the planner reports that a solution cannot be found. The medial axis of the channel can then be extracted from the tree by traversing backwards from the entry region to the dwell segment that corresponds to the root of the tree.

`random_point_in_`$\mathbb{R}^3(\cdot)$: We sample a random point $\mathbf{p}_{\text{rand}} \in \mathbb{R}^3$ within the implant volume I that is not collision with any of the channels in $\mathcal{C}$. The sampled point can then be connected to a given state $X_{\text{near}} = \left[\begin{smallmatrix} R_{\text{near}} & \mathbf{p}_{\text{near}} \\ \mathbf{0} & 1 \end{smallmatrix}\right]$ directly using a circular arc parameterized by $[l, \phi, r]^T$, where l is the arc length, ϕ is the change in orientation of the node X_{near} around the $\mathbf{z}_{\text{near}}$-axis, and r is the arc radius (Figure 2.4). Let $[x, y, z]^T = R_{\text{near}}^T(\mathbf{p}_{\text{rand}} - \mathbf{p}_{\text{near}})$ be the coordinates of $\mathbf{p}_{\text{rand}}$ in the local coordinate frame of X_{near}. The parameters of the circular arc are then given by:

$$r = \frac{x^2+y^2+z^2}{2\sqrt{x^2+y^2}}, \tag{2.3}$$

$$\phi = \arctan(x, -y), \tag{2.4}$$

$$l = r \arctan(z, r - \sqrt{x^2+y^2}). \tag{2.5}$$

To build toward the entry zone, we incorporate two forms of biasing when constructing the tree. First, a sample from the entry zone with a higher probability than the rest of the implant volume. Second, whenever a new node X_{new} is added to the tree, the planner attempts to connect X_{new} to a randomly sampled point in the entry zone E.

`nearest_neighbor`$(\cdot)$: We use the distance measure proposed by Patil et al. [32] that is customized for nonholonomic systems with curvature constraints to select the tree node that is nearest to the sampled point $\mathbf{p}_{\text{rand}}$. Since the channel has a minimum radius of curvature $r_{\min}$, not all sampled points will be reachable from a given state. The *reachable set* from a state $X_{\text{near}} = \left[\begin{smallmatrix} R_{\text{near}} & \mathbf{p}_{\text{near}} \\ \mathbf{0} & 1 \end{smallmatrix}\right]$ consists of all points that can be connected to $\mathbf{p}_{\text{near}}$ by a circular arc that has a radius $r \geq r_{\min}$ and is tangent to the $\mathbf{z}_{\text{near}}$-axis of the local coordinate frame. We use this definition of the reachable set to define the distance metric $\boldsymbol{\rho}[X_{\text{rand}}, \mathbf{p}_{\text{rand}}]$ as the length of such a circular arc connecting $\mathbf{p}_{\text{rand}}$ and X_{near} if $\mathbf{p}_{\text{rand}}$ is in the reachable set of X_{near}, and infinity otherwise.

$$\boldsymbol{\rho}[X_{\text{rand}}, \mathbf{p}_{\text{rand}}] = \begin{cases} l(\equiv r\theta) & \text{if } r \geq r_{\min} \wedge \theta \geq 0 \\ \infty & \text{otherwise} \end{cases} . \tag{2.6}$$

`circular_arc`$(\cdot)$: Given a circular arc parameterized as $[l, \phi, r]^T$ and a maximum step size Δ to progress at each iteration of the RRT algorithm, we compute the position and orientation of the new node X_{new} by composing a rotation of ϕ around the $\mathbf{z}_{\text{near}}$-axis and then applying a rotation of $\theta = \min\{l, \Delta\}/r$ around a line parallel to the $\mathbf{x}_{\text{near}}$-axis and passing through the point $[0, -r, 0]^T, r > r_{\min}$ in the local coordinate frame of X_{near}.

`collision_free`$(\cdot)$: To enable obstacle avoidance, only collision free arcs are added to the tree. We check if the circular arc connecting X_{near} and X_{new} is collision free by approximating it as a sequence of line segments and checking if all the segments are collision free. Since the obstacle definitions are obtained from segmentation of 3D scans, the obstacle meshes are likely to be non-manifold. We use the SOLID library [38] for detecting collisions with arbitrary, polyhedral obstacles at interactive rates. We also check if the minimum clearance of the circular arc is at least the channel width w from the implant volume and existing channels to ensure that the channel that is constructed around the medial axis of this arc is collision free.

`permissible`$(\cdot)$: Since the catheter carrying the source is inserted through the channels, we want the channel orientation at the entry region E to as close as possible to perpendicular to E. We allow a cone of permissible orientations, i.e., the dot product of the local $\mathbf{z}$-axis at a point on the channel medial axis at the entry region and the normal to the entry region should be less than the maximum allowable entry angle (deviation from normal), α.

`build_channel`$(\cdot)$: A channel is found when the position $\mathbf{p}_{\text{new}}$ of a newly added state X_{new} is found to lie in the entry region E and the orientation R_{new} is permissible. By traversing the tree $\mathcal{X}$ backwards from X_{new} to the root X_d, we obtain a path composed of piecewise circular arcs $\{\Psi_1, \Psi_2, \ldots, \Psi_n\}$ constituting the medial axis of the channel, each with radius $r > r_{\min}$. We build the channel by sweeping a circle of diameter w along the medial axis.

The channel is then added to the list of existing channels $\mathcal{C}$ and the process is repeated for the next most distant dwell segment until all dwell segments $\mathcal{D}$ are considered. As it may not be possible to find solutions for all dwell segments, we report a segment as unreachable if a maximum number of iterations of the RRT algorithm are exceeded and no valid path is found to the entry region E.

2.4 Case Study and Evaluation

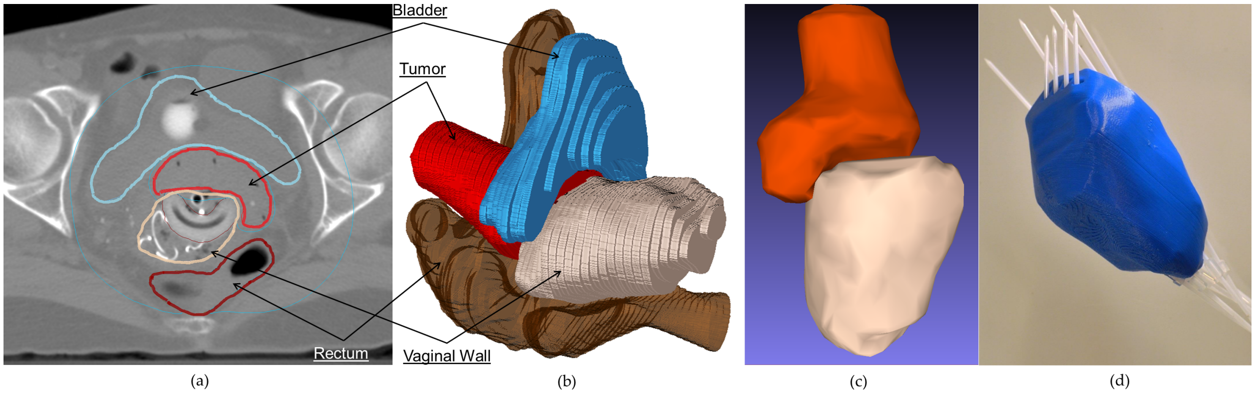
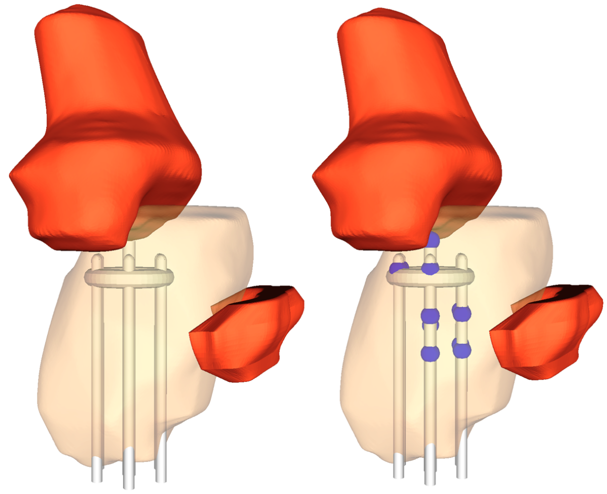


Figure 2.5: This sequence of figures depicts the procedure followed while generating a custom implant. The image in (a) shows a contoured image slice from CT-Scan. The image in (b) shows the 3D model of anatomy reconstructed from the set of contoured CT images. The image in (c) shows smoothed meshes of only the tumor and the vaginal cavity extracted from the 3D model. These meshes are used for generating internal channels. The image in (d) shows a custom 3D printed implant shaped as the corresponding vaginal cavity with catheters inserted in the channels.

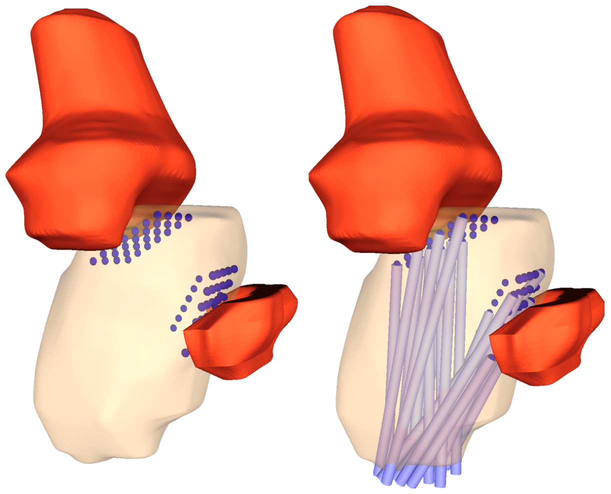
Physical evaluation of the proposed technique requires following a multi-step procedure as illustrated in Figure 2.5. A CT-scan is performed on a patient, and a clinician contours different organs in the CT-scan images. One such CT-image slice with organ contours is shown in Figure 2.5(a). A labeled 3D model of the anatomy is then reconstructed from the CT-scan images as shown in Figure 2.5(b). Thereafter, the tumor and the vaginal cavity are considered separately as shown in Figure 2.5(c) for planning purposes. Using the algorithm described in Section 2.3, internal channels are planned in the vaginal cavity. A modified mesh for vaginal cavity containing internal channels is created and printed using a 3D printer. Figure 2.5(d) shows an image of a custom implant with catheters inserted in the channels. The channels have been extended to be open at distal end of the implant and catheters are shown to exit the implant. In practice, channels will not have an opening at the distal end and catheters will remain in the implant interior.

Case Study Setup: As a case study, we used anonymized data from an actual patient CT-scan from UCSF Mt. Zion Center. A side tumor was also added to anatomy to supplement the complexity of the case. The vaginal cavity was contained in a $70 \times 52 \times 54$mm bounding box with anatomy shown in Figure 2.1. We consider three treatment methods: standardized ring implant (current practice), customized 3D printed implant with linear channels, and customized 3D printed implant with curved channels. We compare them with the coverage quality metric defined in Section 2.2.

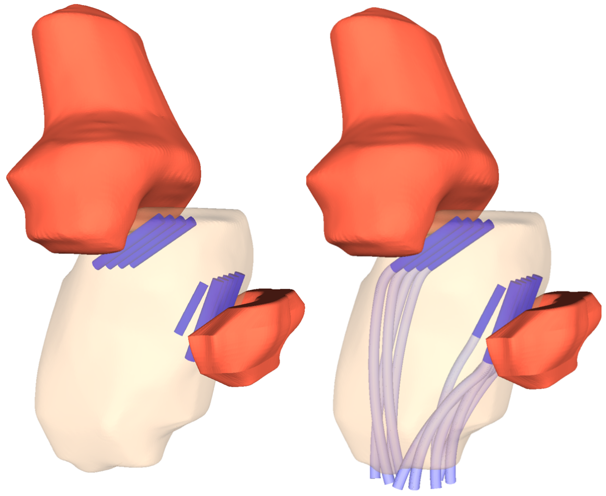
We first consider the standardized ring implant. The left image in Figure 2.6a shows a ring implant placed in the vaginal cavity, containing a toroidal channel running around the interior of the ring and a number (usually 6) of parallel catheters running along the axis of symmetry of the



(a) Standardized ring implant.



(b) 3D Printed implant with only linear channels.



(c) 3D Printed implant with curved channels.

Figure 2.6: (a) Standardized ring implant (white) that cannot conform to patient anatomy. Only 12 dwell positions(in blue) are reachable in this case.
(b) 3D Printed implant with only linear channels: Left: candidate set of 70 dwell positions. Right: only 21 dwell positions can be reached using linear channels.
(c)3D Printed implant with curved channels computed by the CLA algorithm: Left: 11 dwell segments. Right: channels computed by the CLA algorithm.

ring near its outer diameter. A central tube (uterine tandem) passes into the uterine canal via the cervix. In a clinical procedure, the ring implant is placed against the cervix by the physician and then the patient is scanned using either MR or CT imaging. After scanning, a physician segments the anatomical structures and digitizes the positions of the catheters. Using these structures and the set of catheter positions defined by their geometry, dose optimization software determines the best subset of dwell positions and times at each of these positions. The right image in Figure 2.6a shows one such configuration of dwell positions superimposed on the implant.

Next, we consider an alternative technique to the plaster mold proposed by Magne et al. [24], where the channels are manually created by the clinician by pushing linear catheters into the soft material. The right image in the Figure 2.6b shows a set of linear channels (skew lines) that reach as many of the dwell positions as permitted by the size of the entry zone. Finally, we consider the

Table 2.1: The minimum coverage radius (δ in mm) needed to achieve 100% coverage ($\mathcal{Q}$), for 1, 5, 12 and 20 dwell points respectively. The 3D printed implant with curved channels achieves 100% coverage with smaller coverage radius in all cases. The ring implant reaches 12 dwell positions and hence is not applicable for $n > 12$

Tumor Type	n (multiple)	Implant Type		
		Standardized Ring	3D Printed with Linear Channels	3D Printed with Curved Channels
Side Tumor	1	41.35	21.53	21.53
	5	50.61	33.83	22.76
	12	61.10	49.21	26.45
	20	N/A	60.23	33.83
Top Tumor	1	74.32	73.36	73.35
	5	86.00	75.55	73.36
	12	105.12	79.93	74.45
	20	N/A	106.21	76.64

custom implant with curvature-constrained non-linear channels generated by the CLA algorithm as shown in Figure 2.6c.

The standardized ring implant can reach 12 potential radiation source dwell points, the custom implant with linear channels can reach 21 dwell points and the 3D Printed implant with curved channels can reach 70 dwell points (11 dwell segments discretized at 5 mm intervals). We have analyzed the coverage performance for the two tumors (top and side) separately. Table 2.1 lists the values of δ in mm at which coverage quality $\mathcal{Q}$ reaches 100%. Figure 2.7 shows the quality metric as a function of radius for each of three implant types: standardized, custom with linear channels, and custom with curved channels (3D printed). Figure A considers the side tumor with $n = 1$ (number of dwell positions that achieve that indicated quality); Figure B considers the side tumor with $n = 12$. For the side tumor, the 3D printed implant can achieve full tumor coverage ($Q = 100\%$) with lower δ. The effect is less pronounced for the top tumor since a majority of tumor volume is distant from the implant surface.

2.5 Discussion

This chapter presented a new approach to perform intracavitary brachytherapy using 3D printing and present an algorithm for generating curvature-constrained internal non-linear channels. We considered a case-study with an OB/GYN cervical and vaginal cancer to compare three treatment options: standardized implant (current practice), customized implant with linear channels, and customized implant with curved channels. Results with a two-parameter coverage metric, summarized in Section 2.4 and Table 2.1, suggest that customized implants with curved channels can offer significant improvement over current practice, especially for tumor volumes proximal to cavity.

However, we note that none of the intracavitary implants can treat tumor volumes located at a distance of more than 1 cm from the cavity surface. In the case of the Top tumor, use of curved channels provide only a marginal advantage while in the case of proximal Side tumor, curved

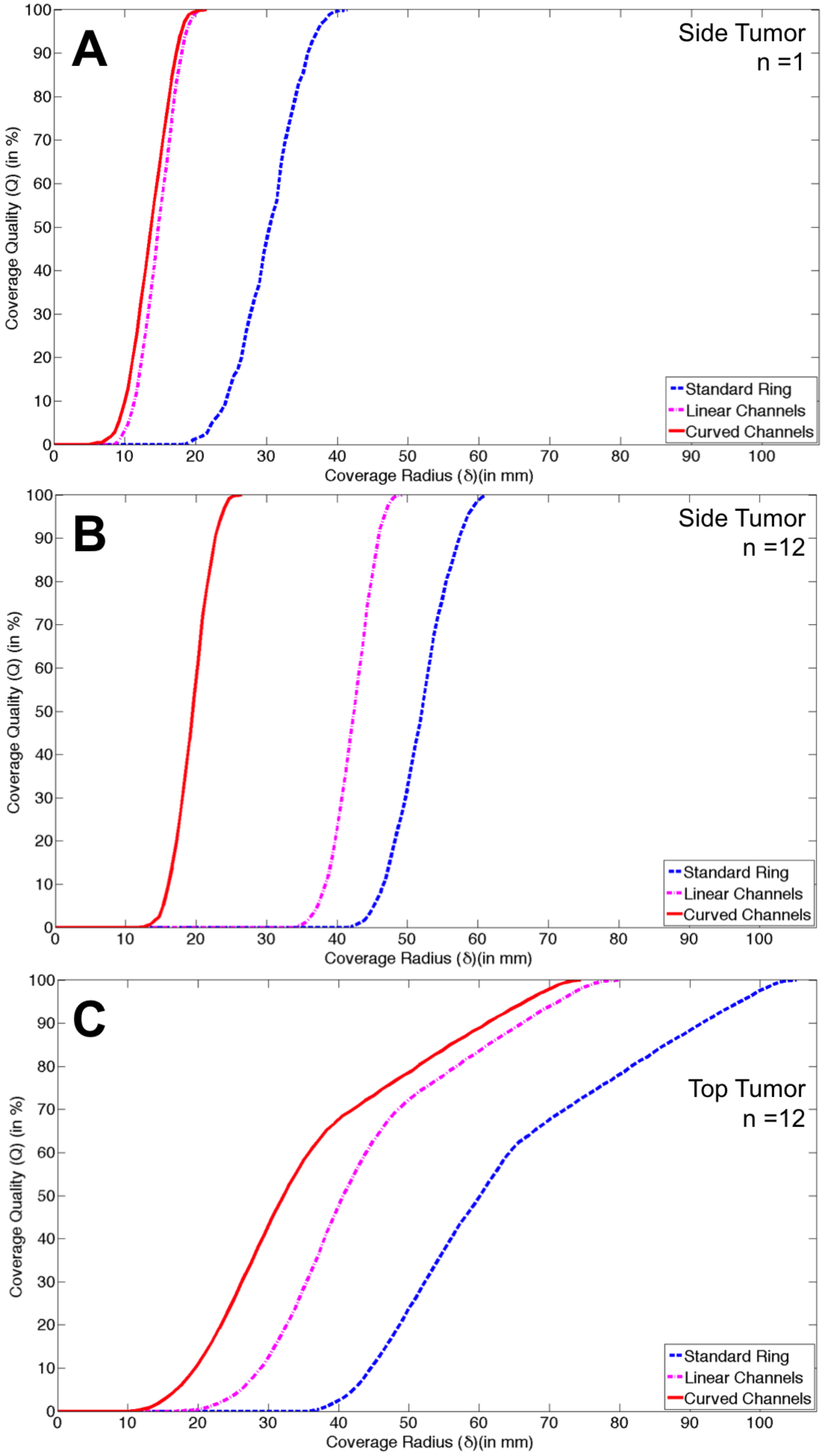


Figure 2.7: Comparison of quality metric $\mathcal{Q}$ (% of tumor volume covered) versus the coverage radius δ for Side Tumor [$n = 1$ (A) , $n = 12$(B)] and Top Tumor [$n = 20$(C)] for the three implants types: standardized ring (current practice), customized implant with linear channels, and customized 3D printed implant with curved channels. As n increases, full tumor coverage ($\mathcal{Q} = 100\%$) is achieved with significantly lower δ in the case of curved channels in comparison to linear channels or standardized implant.

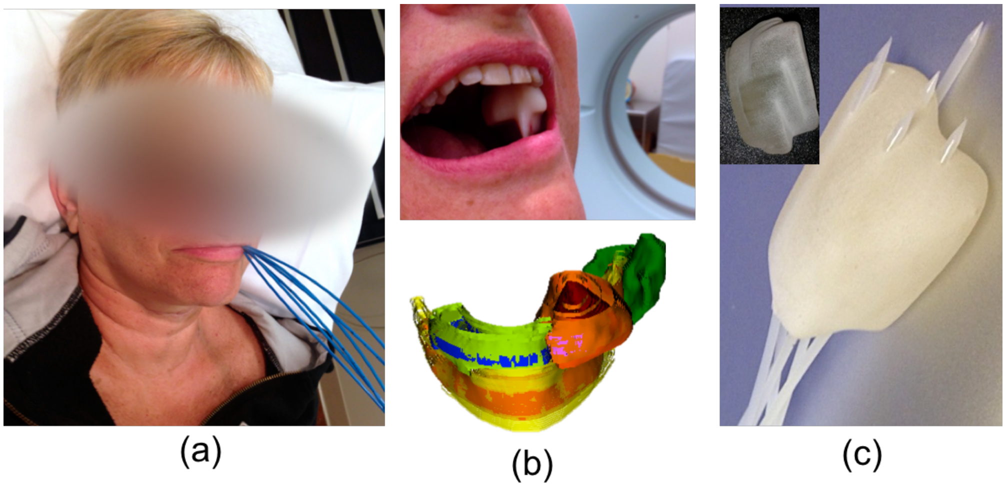


Figure 2.8: This sequence of figures shows the application of customized implants on an oral cancer patient at UCSF clinic in Summer 2014. (a) The image shows the patient with customized implant put in the cavity with catheters connected to the Afterloader device. (b) The image shows the placement and fitting procedure for implant in the left side of the upper jawbone (maxilla). The bottom figure shows the contoured CT-Scan of the patient with the tumor volume displayed in red, (c) The image shows the Class VI biocompatible implant with catheters. Inset image shows a prior iteration of the implant used to fitting procedure.

channels perform significantly better than the other two implant types. Hence a direction of future work is to explore designs of implants that allow placement of interstitial catheters along with intracavitary channels for treatment of tumor volumes farther from the surface.

Such improvements in the coverage metric improve: (a) options for dose planning, which can reduce occurrence of hot spots, and (b) dose conformity with the tumor geometry to spare healthy tissue. We envision that 3D printed implants are clinically viable as outlined below in the potential treatment procedure:

1. Create a cavity model: Use a patient CT/MRI scan along with manual organ contouring. Alternatively, a plaster cast of the cavity can be created.
2. 3D scan the plaster cast noting locations of tumors on surface when possible.
3. Create a 3D printed implant with CT/MRI opaque fiducial markers to improve registration.
4. Scan patient after inserting planning implant: Improve 3D anatomy model with associated fiducial markers embedded to account for post insertion anatomy changes.
5. Compute dose plan and channels in implant using Channel Layout Algorithm (CLA) along with inverse dose planning.
6. 3D print the final custom implant with internal channels.
7. Insert custom implant and deliver treatment using a programmable Afterloader device, which controls the radiation source, over several sessions as needed.

This method has been used for two test patient cases, one of which is depicted in Figure 2.8. The patient in this case was administered with brachytherapy for post-surgery radiation ablation of the tumor bed to reduce possibility of recurrence. The tumor was located in left side of the upper jawbone (maxilla). After the surgical removal of the tumor volume from bone, the resultant cavity was scanned and the aforementioned procedure was followed during treatment delivery.

Chapter 3

Methods for Skew-Line Needle Implants in Interstitial Brachytherapy

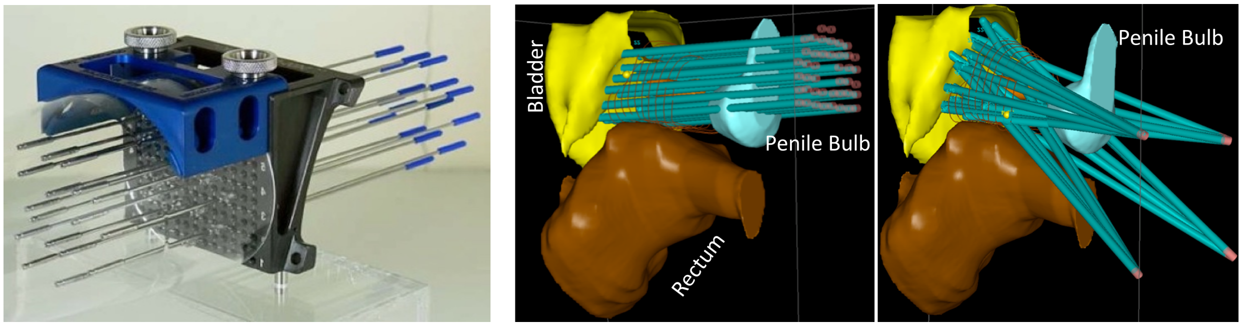
Overview

As introduced in Section 1.1, Brachytherapy is a method of internal radiation therapy where radioactive sources are placed in close proximity to tumor sites for achieving better control over dose distributions to both the tumor and the healthy organs-at-risk. Brachytherapy is an effective treatment for cancers in the prostate, cervix, breast, and other anatomical sites [39]. For prostate cancer, there are two modes of brachytherapy: Prostate Permanent-seed Implant (PPI) and High Dose-Rate (HDR). In PPI-BT, needles implant radioactive seeds with a relatively short half-life (weeks) which are left in the patient after the procedure. In HDR-BT, multiple needles are inserted into the patient. After scanning and planning, a highly radioactive source is automatically moved through each needle using a remote afterloader. The dose distribution is controlled by source dwell times at pre-specified positions along the needles; the source is removed after treatment.

In this chapter, we focus on improving Interstitial HDR-BT treatment delivery for prostate cancer, where current approaches often result in side-effects such as incontinence and impotence [40–42]. Most side-effects result from needle penetration through sensitive structures (urethra, bladder, rectum, penile bulb, cavernous veins, and neuro-vascular bundles) [42–46].

In the previous chapter, we addressed the limitations imposed by standardized implants in intracavitary brachytherapy through the use of patient-specific implants. We demonstrated that radioactive sources can be placed along algorithmically calculated curved channels inside a customized 3D printed implant. In this chapter, we demonstrate similar ideas for addressing the shortcomings of standardized external templates, for guiding linear needles, used in prostate cancer treatment. We demonstrate that a set of linear brachytherapy needles can be accurately placed in a non-parallel (skew-line) pattern to avoid puncturing sensitive organs with the help of a specialized autonomous robot and also with a manual procedure utilizing Customized Needle Guides (CNG).

In the current approach to Interstitial HDR-BT for prostate, hollow needles are inserted into the prostate through the perineum. The insertion is performed manually by the physician using

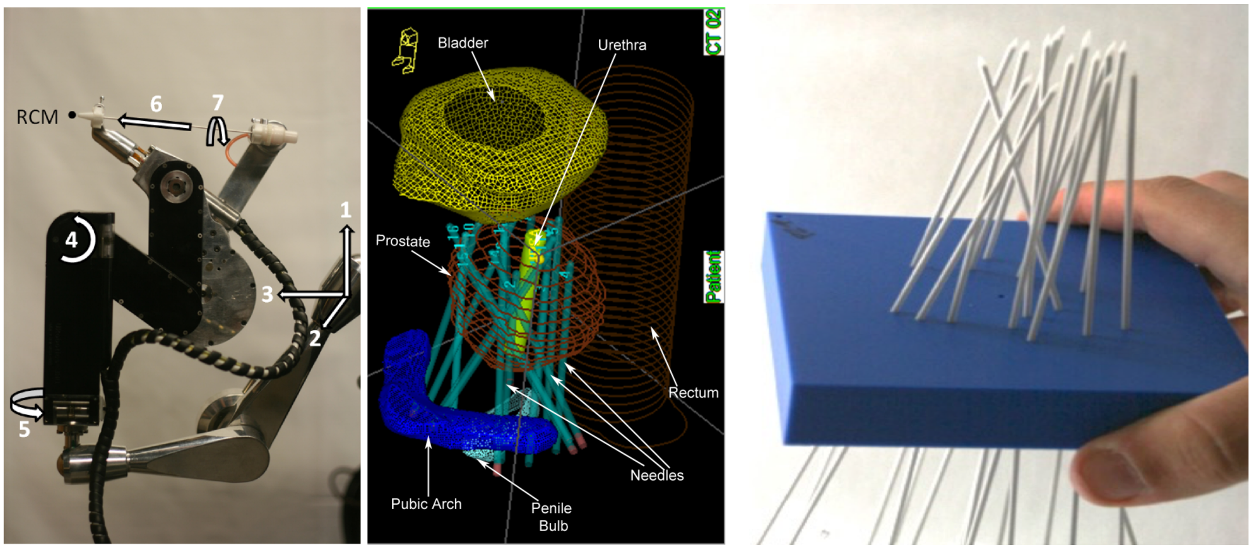


(a) Parallel Needle Template (b) Parallel and Skew-line Needle Configurations

Figure 3.1: (a) The current clinical approach to prostate high dose rate brachytherapy (HDR-BT) uses parallel needles guided by a mechanical template [53]. (a) Left: This approach may prevent needles from reaching prostate volumes blocked by the pubic arch and often require needles to puncture sensitive organs (which can produce long-term side-effects). Right: Skew-line needle arrangements facilitated by robot guidance can avoid puncture by reaching under the pubic arch and can minimize trauma to sensitive organs such as the penile bulb which can produce side effects such as incontinence and impotence.

real-time imaging using a trans-rectal ultrasound probe and a rigid template with parallel holes. As illustrated in Figure 3.1(left), the rigid template requires that all needles be parallel. This restriction often results in puncture of healthy organs such as the penile bulb and related vasculature, and can prevent access to some sections of the prostate due to pubic arch interference. Puncturing healthy tissues also results in trauma related side-effects such as tissue swelling; urinary infections & incontinence; and impotence [47, part 7], [48]. Alternatively, skew-line (non-parallel, non-intersecting) needle arrangements as shown in Figure 3.1(right), can avoid puncturing delicate structures and be angled to reach under the pubic arch. Recently, a “freehand" approach that does not require the template was proposed by physicians to allow skew-line needle arrangements [49]. However, the freehand approach requires a high degree of skill and clinical proficiency. This chapter explores the use of a robot and a customized needle guide to implant skew-line needle arrangements in HDR-BT.

This chapter describes a first integrated brachytherapy system that leverages optimization-based needle and dose planning algorithms with the Acubot-RND needle guiding robot [50] illustrated in Figure 3.2(a) and with customized need guides as shown in Figure 3.2(b). Prior work on the development of IPIP algorithm to compute HDR-BT dose plans [51] and the NPIP algorithm for computing skew-line needle arrangements [52] showed the feasibility of patient-specific skew-line needle arrangements that avoid sensitive organs and meet treatment dose objectives in *simulated* results. We present *physical* experiments with both methods and our results suggest that a human-centered automation system can successfully implant skew-line needle arrangements that avoid puncturing non-prostate structures, meet clinical radiation dose objectives, with mean RMS error between planned and actual dwell points between 2-4 mm.



(a) Acubot-RND and Skew-line Needle Configuration (b) Customized Needle Guide (CNG)

Figure 3.2: Two methods of human-centered automation of treatment delivery in HDR-BT for prostate cancer. (a) Robot-Assisted: The left figure shows the 7-DoF Acubot-RND robot used for this study. It has a 3-DoF Cartesian stage (1,2 and 3), a 2 DoF rotating center of motion (4 and 5), needle insertion (6) and needle rotation (7). The right figure shows a skew-line needle arrangement implanted by the robot system into a phantom as viewed after CT-Scan. (b) Customized Needle Guide (CNG): These patient specific needle-guides are 3D-printed after Needle configuration planning for that anatomy instance. The figure shows an example of one such needle-guide with needles in place for illustration.

3.1 Background and Related Work

The clinical HDR-BT workflow has six main steps: pre-implant patient scanning, needle planning, needle insertion, post-implant patient scanning, dose planning, and dose delivery. An excellent overview of prostate cancer treatment is provided by Salembier and Hoskin [53]. Existing research has explored planning systems for computing optimal dose distributions for both PPI and HDR-BT [51, 54–58]. Since the set of possible dose distributions depends on the implanted needle arrangement, planning systems like Prostate Implant Planning Engine for Radiotherapy (PIPER) [54] and Hybrid Inverse Planning and Optimization (HIPO) [58] incorporate the positioning of needles into their dose planning model.

However, these approaches were developed for the standard parallel needle template, which has a smaller search space: fewer than 100 candidate parallel needles in contrast to 200-300 candidates for skew-line needles. In contrast to active needle steering using bevel-tips or cannuli [59–65], this chapter explores how a symmetric (diamond-tip) needle can be steered to the desired configuration within the tissue by precisely positioning and orienting its primary axis outside the body.

Automation in Needle Implants Prior research in automated needle insertion has explored devices that address the clinical challenges of space constraints and safety requirements for needle insertion robots specially designed for prostate brachytherapy with trans-rectal ultrasound guidance [66–69]. Several of these devices can potentially insert skew-line needles, but they focus on PPI-BT and are not fully integrated with needle planners[70–74].

Similarly, Roy et al. [75] explored the use of precision machining for making external templates as guides for linear needle configurations. But unlike our method, these templates and paths were not generated algorithmically. The Acubot-RND was designed for PPI-BT and is operated by a manual joystick [50, 76, 77]. In this chapter, we describe a modified version of the Acubot-RND with an interface to our needle planning software.

A recent study by Long et al. [78] used the PROSPER image-guided robotic brachytherapy system [69] to perform multiple needle insertions into a gelatin phantom using intra-operative feedback from a 3-D ultrasound system. As noted in the Discussion section, we obtain similar error values without using ultrasound feedback.

This chapter focuses on HDR-BT and integrates automated needle planning system with open-loop robot guided insertion using the Acubot-RND. The needle and dose planning systems are discussed in Section 3.2 and the modifications to the Acubot-RND are discussed in Section 3.3. This is a revised and expanded version of a paper presented at the IEEE International Conference on Automation Science and Engineering (CASE) [17]. This paper is rewritten throughout, with an expanded related work section and detailed analysis of random vs. systematic error.

3.2 Problem Statement

The objective is to compute a set of non-intersecting skew-line needle configuration for a given prostate anatomy that meets the clinical dose requirements and then implant the configuration with a novice assisted by needle insertion robot.

The RTOG-0321 clinical protocol [79] established recommendations for a set of dosimetric indices that are correlated with positive patient outcomes. In these indices, $\mathrm{V}_d^{\mathrm{s}}$, is the volume of structure *s* that receives at least *d* percent (eg., 75%, 100%, 150%) of a specified reference radiation dose (typically 950 cGy).

For the prostate, the value of $\mathrm{V}_d^{\mathrm{Prostate}}$ is specified as a percentage of the total prostate volume, thus $\mathrm{V}_{100}^{\mathrm{Prostate}} >= 90\%$ specifies that at least 90% of the prostate volume should receive at least 100% of the specified reference radiation dose. For other structures such as the bladder, penile bulb, rectum, and urethra, $\mathrm{V}_d^{\mathrm{s}}$ is specified in cubic centimeters, thus $\mathrm{V}_{125}^{\mathrm{Urethra}} <= 0.1$ cc specifies that no more than 1 cc of the urethra should receive more than 125% of the reference dose. The RTOG-0321 recommendations are summarized in the second column of Tables 3.1 and 3.3. Note that $\mathrm{V}_{200}^{\mathrm{Body}} = 0$ cc specifies that no non-organ volume of the body should receive 200% of reference radiation dose.

The treatment requires a sequence of steps: A 3D model of patient anatomy is obtained from a CT scan and manually segmented into organs. We then (1) plan a needle arrangement, if such exists, that lies within the workspace of the robot, avoids non-prostate organs/structures, and meets RTOG-0321 dose requirements, (2) transform this plan into a set of corresponding robot set-points so that each needle starting position and orientation guides a human novice who inserts needles to the indicated depth. (3) perform a second CT scan, compute a dose plan for the actual needle arrangement and report RTOG-0321 dose indices.

In addition to meeting the clinical dose requirements, the performance of both proposed solutions, i.e. implants performed with the robot and customized needle guides, are measured on the following two criteria:

Trauma Metric To quantify the damage to sensitive organs and structures, we propose a trauma metric equal to the total intersection volume: The trauma metric for structure s is:

$$T^s = \sum_k A_k L_k^s,$$

where A_k is the cross sectional area of needle k and L_k^s is the length of needle k puncturing structure s. The needles have a circular cross sections, hence, $A_k = \pi d^2/4$ in mm^2, where d is needle diameter.

Measurement of Needle Placement Error We also measure the total, systematic, and random errors between planned and actual needle arrangements in physical experiments. We sample needle position at 1 mm intervals producing 60 sample points per needle. Using the same sampling procedure for planned and actual needle configurations, we generate two sets of corresponding points: a set of planned points P and set of actual points A.

The total placement error was computed as the root mean squared error (RMSE) between the corresponding points in the planned and actual arrangements. We decompose total error into systematic and random components by computing the least-squares rigid transformation between the pairs of point sets [80]. Specifically, we compute the rotation matrix, R, and the translation vector, T, which minimizes the least-squares error over the whole point set,

$$\sum \|P - (RA + T)\|_2^2,$$

, where P is the vector of planned points and A is the vector of actual points. The associated translations and rotation values define the systematic error. The α, β, and γ values are the rotations in the Euler angles reported in degrees. The Euler angles are computed as:

$$\alpha = \sin^{-1}(r_{1,3}), \beta = \cos^{-1}\left(\frac{r_{1,1}}{\cos(\alpha)}\right), \gamma = \cos^{-1}\left(\frac{r_{3,3}}{\cos(\alpha)}\right),$$

where $r_{i,j}$ is the element of R in the i$^{\text{th}}$ row and the j$^{\text{th}}$ column.

Planning Skew-Line Needle Arrangements and Dose Distributions

Skew-Line Needle Planning To plan skew-line needle arrangements and dose plans, we modified the NPIP needle planning algorithm [52] to use a more comprehensive sample set of candidate needles and we incorporated it with the IPIP dose planning algorithm [51]. These references include details on these planners with experiments and sensitivity analysis.

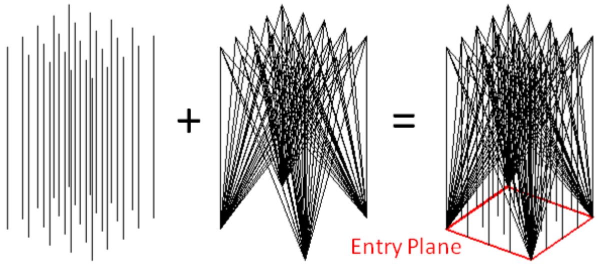


Figure 3.3: The candidate needle set is the set of needles that are available during needle planning. As shown in the figure, the candidate needle set for this study consisted of: parallel lines, and skew-lines. The entry plane, which represents the bounded region on the perineum within which needles can enter the phantom is also depicted.

NPIP accepts as input patient anatomy, the prostate target, obstacles such as the pubic arch and penile bulb, and the defined needle entry zone to search for an arrangement of skew-line needles that: (1) includes approximately 16 needles (the standard at the UCSF clinic), (2) avoids the pubic arch bone and other sensitive organs, (3) offers dwell points that can deliver a dose plan that meets RTOG-0321 dose objectives, (4) minimizes for the trauma metric.

The planner uses integer programming and it is not complete (guaranteed to find such an arrangement if one exists) nor does it always produce an optimal solution. NPIP was modified to use non-uniform sampling to generate the candidate needle set and an additional constraint: all needles in the solution must have a mutual clearance of γ. The parameter γ specifies the distance between the medial axes of a pair of needles. For a nonintersecting needle pair, $\gamma \geq d$, where d is the needle diameter. We chose a conservative value of $\gamma = 2d$ to allow for deviations during insertion.

The prostate volume is discretized into a rectangular grid of sample points, with a spacing of 4 mm in the x- and y-directions and 3 mm in the z-direction (the inter-plane CT sample distance). This produced approximately 1000 points for each case. NPIP takes as input this set of sample points and a user-specified parameter, δ. NPIP generates a candidate needle set (line segments) and searches for a subset of these candidate needles where every point within the prostate is within δ of at least one needle. A high value of δ allows needles to cover more volume, producing needle arrangements with fewer needles. To normalize across the prostate volume, we set $\delta = 33\%$ of the radius of a sphere with equivalent volume to the prostate and iteratively increased or decreased it to obtain a solution with 16 needles. NPIP uses heuristics to solve an integer program so there are no time or performance guarantees, but for the cases we considered, NPIP computed solutions within 120 seconds.

Dose Planning The needle arrangements computed by NPIP are given as input to the Inverse Planning by Integer Program (IPIP) dose planning algorithm [51]. Given the set of needles, IPIP computes a set of dwell times (spaced 5 mm apart within each needle) for the radioactive source that maximizes $\mathrm{V}_{100}^{\mathrm{Prostate}}$ subject to the RTOG-0321 dose requirements. For the three phantom cases, we studied, IPIP found solutions within 10 seconds with values as reported in Table 3.1.

3.3 Robot-Assisted Needle Insertion with the Acubot-RND

The Acubot-RND robot system was designed and constructed at Johns Hopkins University to guide needle insertion for permanent-seed (PPI-BT) treatment [50]. Hardware specifications for the

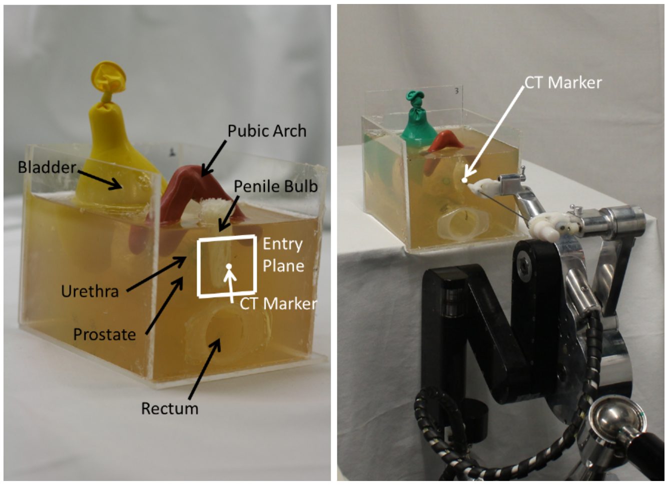


Figure 3.4: Prostate phantom (5in x 5in x 8in) (left) and insertion setup (right). The anatomy modeled in the phantom includes: prostate; urethra; bladder; penile bulb; pubic arch; and rectum. A CT marker is centered on the square entry zone for calibration. As shown in the right image, the Acubot-RND is registered to the CT-marker.

Acubot-RND, including spatial resolutions and maximum ranges for each degree of freedom are reported in Fichtinger et al [68].

As shown in Figure 3.2 (a), the Acubot-RND is a 7-DoF robot with three stages: The first is the 3-DoF Cartesian Positioning Stage (CPS), The second is the 2-DoF Rotating Center of Motion (RCM) that sets needle angle keeping the needle tip position fixed, and the third is the 2-DoF Rotating Needle Driver Module (RND) that can rotate and insert needles automatically.

The phantom is draped during the experiments. For this study, we position the first stage manually during calibration and we send computed commands to the second stage to orient the needle prior to insertion. We then send a command to the third stage to insert the needle to a pre-specified end point without feedback. At this point a human novice with no clinical experience manually retracts each needle leaving behind a stylet in tissue.

Digital Interface

The needle entry plane with CT marker defines the coordinate frame. We modified the Acubot-RND, augmenting the manual joystick operation with a digital interface that allows commanding specific offsets in tip position from the center of the entry zone, and specific pairs of angular offsets from the normal to the plane of needle entry zone.

A needle plan defines a set of i needles, each specified with two points: $\underline{p}_0^i$ in the entry plane, and $\underline{p}_1^i$ at the desired distal tip of the inserted needle, where x and y components of $\underline{p}^i$ span the entry plane in horizontal and vertical directions; and the z component points into the phantom volume. The insertion depth for needle i is d_i, the Euclidean distance between the points. The angles for angle needle i, defined as rotations in the associated planes, are:

$$\theta_{xz} = atan2\left(x_1 - x_0, z_1 - z_0\right)$$

$$\theta_{yz} = atan2\left(y_1 - y_0, z_1 - z_0\right)$$

These angles are specified as joint angles for the RCM.

3.3.1 Setup of Physical Experiments

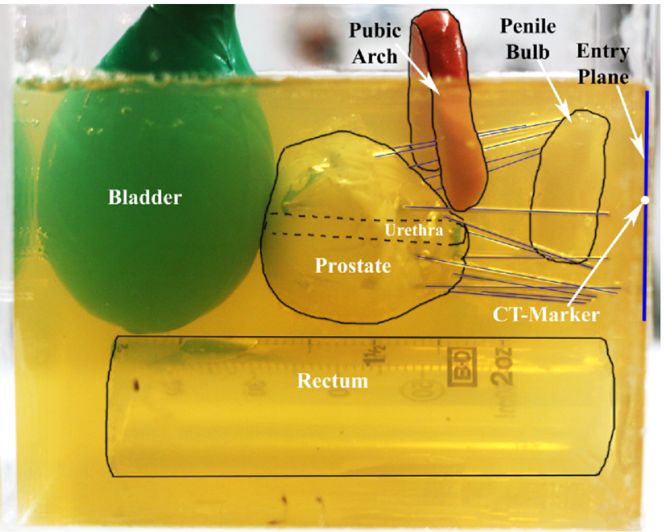


Figure 3.5: Anatomically-correct phantom Ph1 with robot-implanted needle configuration A1. The organ boundaries and actual needles positions are highlighted. No sensitive structures were punctured.

Prostate Phantoms To evaluate the performance of the NPIP and IPIP algorithms and robot hardware, we constructed three nearly identical physical phantoms in the clinic at UCSF: Ph1, Ph2, and Ph3. Each phantom includes an anatomically-correct layout of organ structures of similar density as human tissue and suspended in a translucent gelatin medium that had a strong CT contrast to the mixture used for the prostate, bladder, and penile bulb. Harder bone structures like that of the pubic arch are constructed from modeling clay to prevent puncturing. The prostate, bladder, and penile bulb were made from small, latex water balloons filled with a water-milk-gelatin mixture. A plastic straw segment was inserted into the prostate balloon to simulate the urethra length within the prostate. The rectum was simulated by a 1 inch (2.5 cm) diameter plastic tube. The organ structures include urethra, prostate, bladder, penile bulb, pubic arch and rectum as shown in Figures 3.4 and 3.5.

Each phantom also had a square entry zone of dimension 45 mm, consistent with clinical practice as shown in Figure 3.4, relative to an example candidate needle set in Figure 3.3. We performed end-to-end needle insertion procedures with 16 needles on each phantom using the robot for the first two (Ph1 and Ph2) and an expert human physician for the third phantom (Ph3).

Procedure Workflow Each experiment includes these steps (with step 2 omitted for the expert human physician who used his clinical intuition to determine a needle plan):

1. Perform first CT-Scan and 3D segmentation of organs.
2. Plan desired Needle configuration using NPIP and calculate dose plan IPIP.
3. Implant Needles with robot or with expert human.
4. Perform second CT-scan of phantom with needles.
5. Perform dose planning using IPIP.

3.3.2 Needle Implant Experiments

A side view of an implanted phantom Ph1 is shown with needle configuration A1 in Figure 3.5. A robot-assisted implant of needles was performed on two phantoms, Ph1 and Ph2. The needle entry zone is a square on the surface of the phantom centered on the CT marker. As in typical clinical cases, the entry zone is 45 mm $\times$ 45 mm as shown in Figures 3.4 and 3.5. We place a radio-opaque CT-marker at the center of each entry zone to register the coordinate system of the planning algorithm with the robot.

Pre-Implant Scanning

CT scans of tissue phantoms, before and after all 16 needles are inserted, were taken in 3 mm thick slices. The contoured prostate volumes for the three phantoms were 39 cc, 32 cc, and 37 cc. The total phantom volume was 750 cc. The organs in the phantom and the CT marker were contoured in 3D using the Nucletron Oncentra® Dynamic Planning Environment. Using Oncentra, we added a 2 mm margin to the outer contour of the penile bulb. These 3D organ models were exported to NPIP and IPIP. A reference dose of 950 cGy is commonly prescribed for prostate HDR-BT; we used this level as the reference in all cases.

Needle and Dose Planning

For Ph1 and Ph2, there were 287 and 229 candidate needles respectively. NPIP used a δ value of 6.5 mm for Ph1 and 6.0 mm for Ph2 to produce solutions with 16 needles. γ value was chosen to be twice the needle diameter, 4 mm. For Ph1 and Ph2, we define two needle arrangements the planned needle arrangements, P1 and P2, and the actual needle arrangements, A1 and A2.

All computation was performed using Matlab R2011a on a Lenovo ThinkPad with an Intel i5-2410M processor and 4GB of RAM. The integer program optimization was done using the Matlab interface for the Mosek Optimization Toolbox v.6. The complete run for planning using NPIP less than 70 seconds for both Ph1 and Ph1; and IPIP runs took 10 seconds for both Ph1 and Ph2.

Robot Experiments on Prostate Phantoms Ph1 and Ph2

After the initial CT scan, the robot and phantom are clamped to a worktable, leveled, and manually calibrated as follows (1) the robot is manually moved to an initial state with first needle tip at the registration mark and aligned normal to the entry plane by moving to specified x and y offsets and confirming that it just touches the surface at each point. Figure 3.4 shows the Acubot-RND and phantom in such an initial state. We used a standard 18-gauge diamond-tip brachytherapy needle (COOK Biotech) of length 15 cm and 2 mm diameter hollow sheath that housed a rigid stylet. To implant needle arrangements in Ph1 and Ph2, the Acubot-RND was brought into each specified position and orientation where a needle was inserted by the robot until the pre-specified depth in phantom tissue. The insertion depth was marked on a stylet and it was manually pushed through the hollow needle in the phantom by the novice operator, and the needle is retracted to leave the stylet in the phantom. The stylets were used as a proxy for needles in phantom to minimize interference to the robot during subsequent needle insertions.

Expert Human Physician Experiment on Prostate Phantom Ph3

Collaborator Dr. I-Chow Hsu is a certified radiation oncologist at UCSF with a specialization in brachytherapy and over 18 years of clinical experience. He performed insertion on Ph3 for comparison. We performed a CT scan of Ph3 as above. Dr. Hsu used his expert intuition to determine a needle plan. He inserted 16 standard HDR-BT needles into phantom Ph3 under trans-rectal ultrasound (TRUS) guidance using the UCSF-developed "freehand” technique [49]. A HAWK 2102 EXL TRUS system from B-K Medical was used for ultrasound imaging.

Post-Implant CT Scan

After executing all implants, another CT scan is performed on the phantom. The needles are segmented and organs are contoured to determine the needle configuration actually implanted, A_i.

3.3.3 Results

The expert human physician experiment was completed in under 15 minutes. Each robot experiment required approximately 45 minutes due to calibration and slow needle insertion speeds by the novice. We also note that the expert human physician had the benefit of ultrasound feedback while the needle insertions for the robot experiments were performed without ultrasound or visual feedback. Figure 3.6 shows the cross-section of the needle arrangements implanted by the expert (left) and by a novice with the robot guide (right).

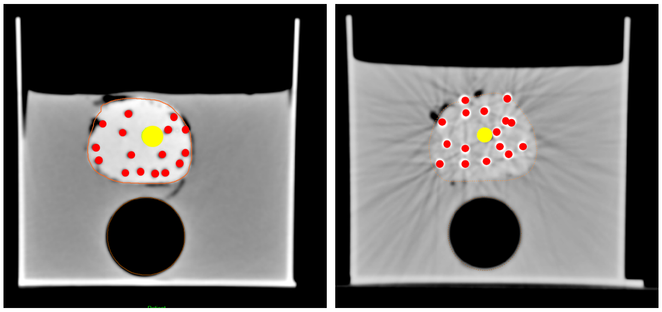


Figure 3.6: Cross-sectional view of an actual needle arrangement inserted by an expert human physician without the robot (left) and one inserted by a novice human guided by the robot (right). Both are considered successful as they meet the RTOG dose objectives without penetrating the penile bulb.

Dose Indices and Trauma Metric

The RTOG-0321 clinical requirements and results from all three experiments, planned and actual for the robot, and actual for the human, are summarized in Table 3.1. For all 3 cases, clinical requirements were met and performance with the robot was comparable to that of an expert human physician.

It is worth noting that the difference between the values obtained from planned vs actual needle arrangements is relatively minor. Although actual needles could puncture the penile bulb due to

Table 3.1: This Table lists the clinical dose index and trauma metrics, the RTOG-0321 requirements, and the values from each experiment, Ph1 and Ph2 using the robot, and Ph3 by an expert human physician. Columns P1 and A1 are the dose values achieved by IPIP for the planned and actual needle arrangements respectively for Ph1. The same for P2, A2, and Ph2. A3 for the third phantom Ph3 is based on the needles as actually implanted by the expert human physician (who did not plan a needle arrangement).

		Phantom 1		Phantom 2		Phantom 3
Metric	RTOG Req.	P1	A1	P2	A2	A3
$V_{100}^{\text{Prostate}}$	$\geq 90\%$	99.0	97.0	96.0	96.0	98.0
$V_{150}^{\text{Prostate}}$	$\leq 45\%$	39.0	40.0	40.0	37.0	37.0
V_{75}^{Bladder}	≤ 1 cc	0.00	0.00	0.30	0.80	0.30
V_{100}^{Bladder}	$= 0$ cc	0.00	0.00	0.00	0.00	0.00
V_{75}^{Bulb}	≤ 1 cc	0.00	0.00	0.00	0.00	0.00
V_{100}^{Bulb}	$= 0$ cc	0.00	0.00	0.00	0.00	0.00
V_{75}^{Rectum}	≤ 1 cc	0.06	0.00	0.00	0.00	0.00
V_{100}^{Rectum}	$= 0$ cc	0.00	0.00	0.00	0.00	0.00
V_{125}^{Urethra}	≤ 0.1 cc	0.06	0.05	0.04	0.06	0.07
V_{150}^{Urethra}	$= 0$ cc	0.00	0.00	0.00	0.00	0.00
V_{200}^{Body}	$= 0$ cc	0.00	0.00	0.00	0.00	0.00
T^{Bulb}	min	0.00	0.00	0.00	0.00	0.00

placement error, the puncture volume in all planned and actual cases, for the robot and the human, was zero (0 cc). Also, no needles intersected the pubic arch.

A notable exception in dose is the difference in V_{75}^{Bladder} values for P2 and A2 which were 0.3 cc and 0.8 cc, respectively. They are both below the clinically acceptable limit for this criterion: 1 cc. This discrepancy is due to some needles not being inserted completely to the intended end-points into the prostate. This is mainly due to placement error in the manual step of the needle insertion. Since no dwell positions are available at the apex of the prostate, IPIP increases the dwell times at the distal ends of the needles to achieve coverage, but this produces a higher-than-desired dose to the bladder.

Needle Placement Error

We next consider the total, systematic, and random errors between planned and actual needle arrangements in the two robot experiments (there is no planned needle arrangement for the third experiment). Table 3.2 summarizes mean, min, and max RMS error (RMSE) along each dimension and d, the Euclidean distance. For Ph1 and Ph2, the total RMS errors were 2.6 mm and 4.3 mm respectively.

The errors for Phantom 1 and Phantom 2 are shown in Table 3.2. The random error is the residual error after the actual points are compensated by the least square transformation. Note that

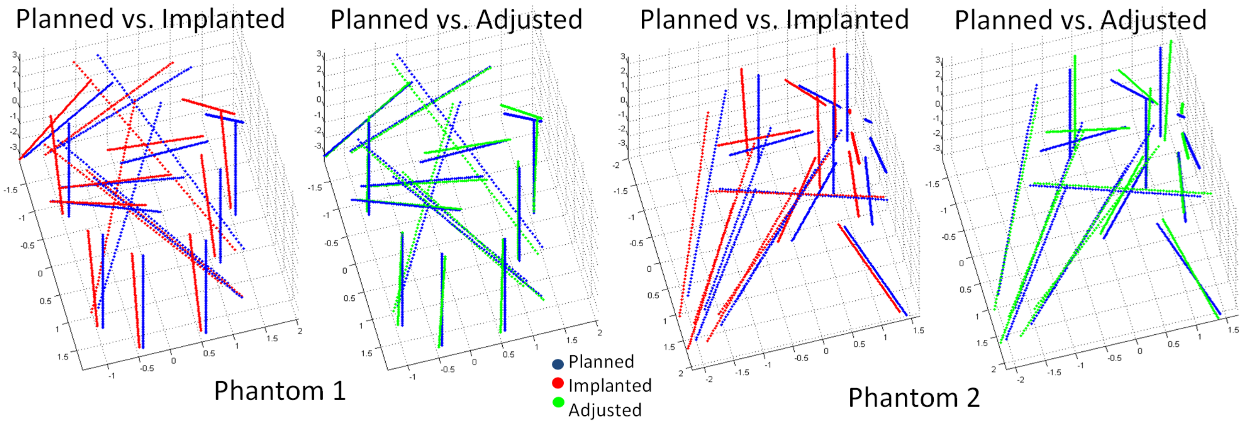


Figure 3.7: Robot-Assisted Implants: Superposition of planned (blue) and implanted (red) needle arrangement for Phantom 1 and Phantom 2. Although no sensitive structure was punctured in the implanted needle arrangement and all dose objectives were met, there was non-zero placement error. The placement error was separated into systematic and random error. Upon compensation for the systematic error, the adjusted needle arrangement (green) fits better to the planned configuration.

	Phantom 1			Phantom 2		
Total RMS	Min	Max	Mean	Min	Max	Mean
x	0.5	2.5	1.4	0.8	2.7	1.9
y	0.2	2.5	1.6	0.7	3.2	2.3
z	0.1	3.0	1.5	0.9	5.3	3.1
d	1.3	4.1	2.6	2.0	6.3	4.3
Random Error						
x	0.0	0.8	0.5	0.1	2.3	1.2
y	0.1	1.1	0.5	0.1	1.8	1.1
z	0.1	2.3	1.2	0.0	5.1	1.8
d	0.2	2.5	1.4	0.8	5.2	2.4
Systematic Error						
ε_x	1.2			0.9		
ε_y	1.4			2.2		
ε_z	0.8			2.4		
α	1.8			1.3		
β	-0.9			0.8		
γ	1.6			3.5		

Table 3.2: Error Analysis: Total errors are RMS errors (in mm) measured in phantoms post-implant. Random errors are RMS errors (in mm) after compensation for systematic error. The x-,y- and z- rows list RMS errors in each direction. d is the overall RMS error. Systematic errors are obtained by least square point set matching. (ε_i in *mm* and angles in *degrees*)

systematic and random components do not sum to the total error due to rotations. Total random error for Ph1 and Ph2 are 1.4 mm and 2.4 mm respectively. Table 3.2 summarizes the results.

For qualitative comparison, the superposition of the planned (blue) and implanted (red) needles is shown in Figure 3.7, as well as the planned and adjusted needle arrangements (green).

3.4 Computationally Designed Custom Needle Guides

Recent advances in 3D ultrasound technology and rapid prototyping allow a low-cost alternative framework to robot-assisted implants for performing skew line needle insertion in brachytherapy. The main insight in this work is skew line needle implant based on a pre-computed plan by a human is not feasible due to intricate angles involved. However, a Customized Needle Guide (CNG) can be created for each patient that matches the needle plan, and this needle guide can then be used in a manner similar to standard templates (Figure 3.1) currently used in the clinic.

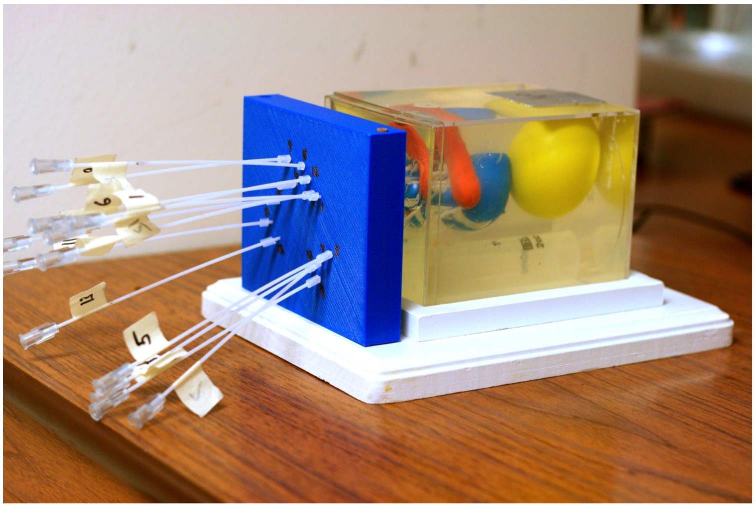
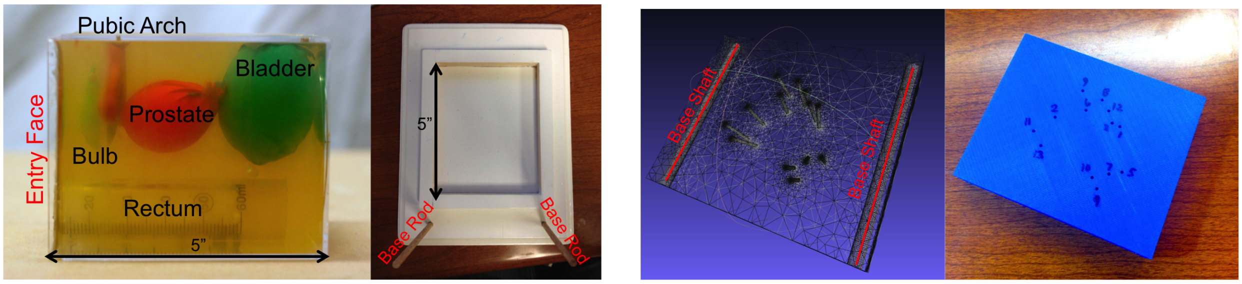


Figure 3.8: Recent advances in real time 3D ultrasound, EM tracking, needle position optimization, and rapid prototyping provide the means to radically change the prostate high dose rate (HDR) brachytherapy workflow. In this study, we evaluate the placement errors associated with inserting needles using a custom needle guide for optimized skew-line needle arrangements, which is one part of this workflow. The figure shows a prostate phantom with skew-line needle configuration implanted with a computationally designed needle guide.

The use of 3D printed needle-guides provides a scalable patient-specific method to perform the HDR-BT without the requirement of high skill in needle placement, as required by the “freehand" method [49], or high capital cost of a robotic system. This method enables needle implants by a novice with the placement errors comparable to the robotic system as described later in this section.

This section will describe a specific part of the 5-step procedure of the HDR-BT procedure as described in Section 3.3.1. Specifically, we evaluate the placement error associated with inserting needles according to a custom needle guide and the effect this error has on the number of structures punctured and the ability to meet treatment objectives.

Figure 3.8 shows a custom needle guide that was used to implant one of the phantoms. The experiments in this section do not use any real-time feedback to alter the needle arrangement or avoid puncturing non-target structures. And the evaluation procedure is similar to the robot-assisted implants in the previous section, through comparisons based on placement error, critical structure puncturing, and the final dose distribution.



(a) Setup of Physical Experiments (b) Two stages in computation of Needle Guide

Figure 3.9: (a) We performed our needle insertion study on prostate gelatin phantoms (left). The regions of interest included the prostate, urethra, bladder, rectum, penile bulb, and pubic arch. The phantom was registered to the needle guide using a base built specifically for this study (right). The base contained two alignment dowel pins, which mated with the custom needle guide (CNG). This figure shows the base from top view. (b) The template mesh contained cylindrical channels for each of the needles and two alignment holes base to mate with the alignment dowel pins in the base (right). An STL mesh file is then created and used for 3D-printing the CNG.

3.4.1 Setup of Physical Experiments

The physical experimental setup and the procedure for phantom preparation are similar to the experiments with Robot-Assisted Needle Implants as described in Section 3.3.1. Step 3 in the aforementioned procedure is now replaced with a novice using a customized needle guide (CNG). We created four gelatin phantoms to simulate prostate cancer cases. These phantoms cases were labeled C1, C2, C3, and C4, respectively. Figure 3.9a shows one of the phantoms used in this study with labels for the relevant anatomical structures. The contoured prostate volumes for each phantom were 27, 26, 31, and 32 cm^3, respectively. These measurements are obtained from pre-implant CT Scan and digitization of the anatomical structures. Similar to robot-assisted implants (Figure 3.4), a global coordinate frame is also defined such that entry face (Figure 3.9a) forms the x-y plane and z-axis is defined along the depth in the phantom measured from the entry face.

In addition to the use of phantoms, in this case, a base was constructed to register the phantom to the planning system and the customized needle guide. The base was constructed from wood to avoid CT artifacts that could interfere with structure identification and segmentation. The base was designed such that the insertion face of the phantom and the CNG were parallel with a 2 cm offset, and the bottom of the phantom and the CNG were at the same height. The phantom was immobilized by side supports that were tight to the phantom's outer housing. The base was designed to hold an extruded, rectangular needle-guide measuring 12 cm wide × 10 cm tall × 2 cm thick. The CNG was registered in position by two alignment dowels pins that were perpendicular to the base and 11 cm apart. These dowel pins were constructed from 3/8 in. (4.8 mm) wooden shafts embedded into the base. Each dowel pin had an associated mating channel in the CNG.

3.4.2 Needle Implants with Customized Needle Guide

Needle Planning

After following a similar pre-implant CT-Scan of each prostate phantom, a 3D model of the anatomical structures is created by manually contouring of the relevant tumors volumes and organs-at-risk. Thereafter, a needle arrangement was computed for each anatomical structure set using Needle Planning by Integer Program (NPIP)[52]. NPIP requires an entry zone definition, i.e. planar segment, through which needles can be inserted into the body. In practice, the entry zone for needles is on the surface of the perineum. Since the perineum was not modeled in our phantoms, the entry zone was taken to be a 6 × 6 cm square, perpendicular to the z-axis, centered at the center of mass of the contoured housing of the phantom.

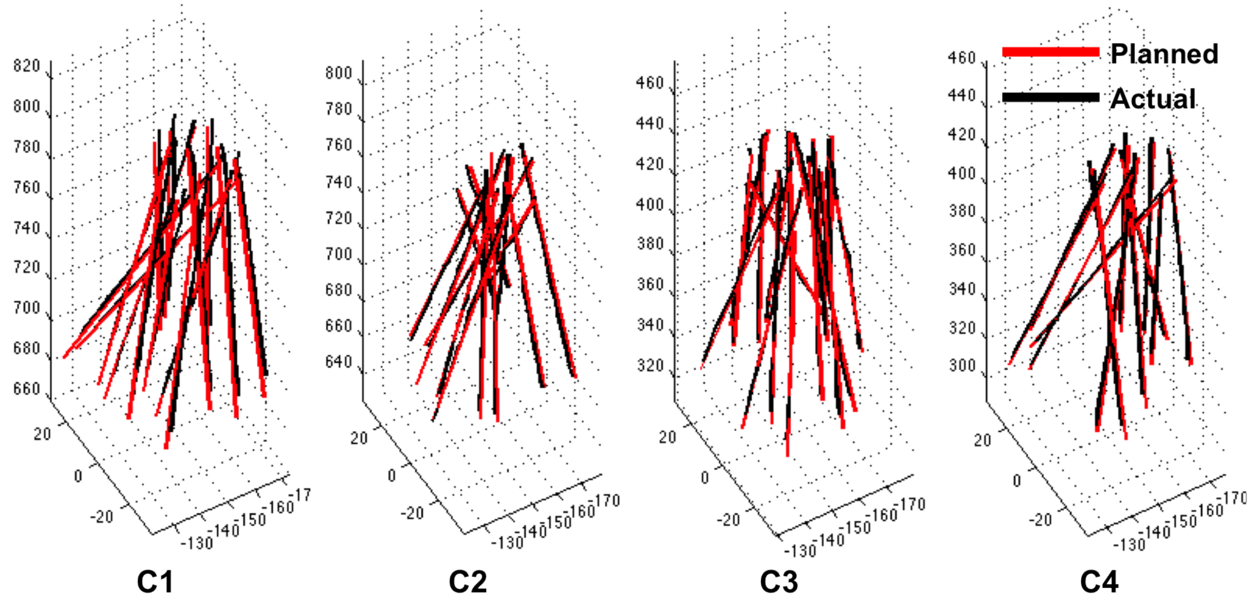
For this study, the δ parameter was initially set to 32.5% of the prostate radius, which was defined to be the radius of a sphere with equivalent volume to the prostate. This selection of δ was known to produce needle arrangements with approximately 14-16 needles [52], where 16 needles are the standard number of needles used at the UCSF clinic for HDR-BT. The final needle arrangement is evaluated by computing a dose plan for it using Inverse Planning by Integer Program (IPIP), which is a dose planning algorithm presented in Siauw et al. [51]. If the dose plan met the target coverage requirement and constraints on the dose to organs-at-risk, then the needle arrangement was finalized. Otherwise, the δ parameter was tuned during the planning phase to achieve a dose plan that met clinical dose objectives and constraints. The planned needle arrangements were labeled P1, P2, P3, and P4, for C1, C2, C3, and C4, respectively.

Computational Design of Customized Needle Guide

Once a needle arrangement was computed, a triangle mesh was generated of a needle-guide that would guide needles to that arrangement and interface with the base. The surface mesh was generated using in-house software specifically designed for this use-case. The CNG was designed to be 12 ×10 × 2 cm with a working area (i.e. space available for needles) of 10 × 10 cm in the x-y plane, centered on the template, and the front face of the CNG was expected to be 2 cm from the entry face of the phantom. For each needle in the skew line needle configuration output from the NPIP, a cylindrical mesh was intersected with the mesh of the CNG. A cylindrical hole, along the direction of the needle, was added to the CNG mesh as a result of the mesh difference operation. The holes were created 2 mm in diameter for 1.8 mm diameter needles.

The CNG was designed to mate perfectly with to the alignment dowel pins in the base, which prevented movement between the base and the CNG. The CNG had two circular channels to interface with the pins in the base. The channels were parallel to the y-axis with centers 1 cm from either side on the x-axis, and centered on the template along the z-axis. These channels were designed to be 6 mm in diameter with a small clearance to fit on 4.8 mm alignment pins.

The triangle mesh was written to a Surface Tesselation Language (STL) file, which is a standard file format for input to 3D printers. Each CNG was printed from the STL files using a uPrint SE Plus (uPrint). The uPrint has a minimum slice resolution of 0.254 mm, which was used for this study, and it has a building volume size of 8 × 8 × 6 in. (204 × 204 × 152 mm). The needle



C1 **C2** **C3** **C4**

Figure 3.10: Planned needle arrangements (red) and actual needle arrangements (black) for each case. The units of the axis are in millimeters. Although the RMS errors are large, the needles are quite close, even difficult to tell apart, and most of the error is in the z-direction, which is along the needle insertion direction. We expect these errors to be reduced when needles are inserted in the context of a brachytherapy workflow with real-time ultrasound feedback, electromagnetic tracking, and software integration.

guides were printed in ABS plastic, and the support material used was an SR-30 soluble material. The process of printing with a low-density setting took approximately 4 hours and needed another 2 hours in a lye bath to dissolve the support material.

Needle Implants on Prostate Phantoms C1, C2, C3, C4

To perform the needle insertion, the phantom and the CNG were registered using the base. A visual interface was designed to assist with the needle insertion step. This interface plotted a diagram of the back side of the template, with needle numbers and insertion depths. Each hole on the CNG was now assigned an identifying number and marked. Standard HDR-BT needles (catheters with steel stylets) were now numbered correspondingly in preparation. Furthermore, rubber stoppers cut from 1/16 in. diameter (1.6 mm) rubber tubes were placed on each needle at the insertion depth specified by the needle insertion interface. The insertion depth was measured from the back of the needle tip.

The needles were inserted one at a time up to the stoppers. The needles were twisted back and forth as they were inserted to facilitate the puncturing of the prostate membrane. There was no ultrasound imaging during needle insertion, and no attempt was made to deviate from the needle arrangement set by the CNG and NPIP plan. The needle implant process took approximately 5 minutes in each case.

Table 3.3: Dosimetric data, trauma metrics, and needle placement errors for Needle Configuration Implants with Customized Needle Guides. C1-C4 are the four prostate phantoms. P# & A# represent planned and actual needle arrangements

			C1		C2		C3		C4	
		unit	P1	A1	P2	A2	P3	A3	P4	A4
V^{Prostate}		cc	27	27	26	28	31	33	32	33
Needles		#	14		14	(13)	13		15	
$V^{\text{Prostate}}_{100}$	≥ 90	%	95	94	96	97	95	95	96	94
$V^{\text{Prostate}}_{150}$	≤ 45	%	39	38	35	26	39	42	35	36
V^{Urethra}_{125}	≤ 0.1	cc	0.07	0.07	0.06	0.04	0.08	0.07	0.06	0.08
V^{Urethra}_{150}	$= 0$	cc	0	0	0	0	0	0	0	0
V^{Bladder}_{75}	≤ 1	cc	0.16	0	0.48	0.85	0.30	0.52	0.90	0.62
V^{Bladder}_{100}	$= 0$	cc	0	0	0	0	0	0	0	0
V^{Rectum}_{75}	≤ 1	cc	0	0	0	0	0	0	0	0
V^{Rectum}_{100}	$= 0$	cc	0	0	0	0	0	0	0	0
V^{Bulb}_{75}	≤ 1	cc	0	0	0	0.05	0	0	0	0
V^{Bulb}_{100}	$= 0$	cc	0	0	0	0	0	0	0	0
V^{Body}_{100}	$= 0$	cc	0	0	0	0	0	0	0	0
T^{Bulb}		mm^3	0	0	0	28	0	0	0	0
RMSE_x		mm	0.7	(0.7)	1.4	(1.2)	1.2	(1.1)	0.6	(0.6)
RMSE_y		mm	1.2	(0.5)	0.9	(0.9)	0.9	(0.6)	0.7	(0.6)
RMSE_z		mm	3.4	(1.8)	2.5	(2.4)	4.2	(2.4)	2.1	(1.7)
RMSE		mm	3.6	(1.9)	3.0	(2.8)	4.5	(2.8)	2.3	(1.9)

Once the needles were inserted, the inner metal stylets of the needles were removed. The phantom, base, template, and needles were CT scanned. The anatomical structures and needles were segmented in Oncentra. The coordinate systems between the pre- and post-implant CT-scans were registered together using the 4 CT-markers on the of the base. Finally, the registration transformation between these two point sets was computed using the Coherent Point Drift Algorithm [81]. The actual implanted needle arrangements were labeled A1, A2, A3, and A4, for C1, C2, C3, and C4, respectively.

3.4.3 Results

We evaluated the volumetric dose distribution and the placement error of the needles for each planned and actual needle arrangement. We also computed a trauma metric for each structure, similar to the robot-assisted implants (Section 3.3), which is defined as the structure volume displaced by a puncturing needle.

Dose Indices and Trauma Metric

The final planned needle arrangements, labeled P1, P2, P3, and P4, for C1, C2, C3, and C4 respectively. The final number of needles in the arrangements computed for each phantom was 14, 14, 13, and 15, and the target coverage was 95%, 96%, 95%, and 96%, respectively. All dose constraints were met for the planned arrangements. The relevant numerical results for this study are shown in Table 3.3.

In our experiments, three phantoms were successfully implanted without puncturing any non-target structure. However, in one of the cases, the penile bulb and urethra were punctured and we noted this case as a failure (C2). However, for completeness, we compute a dose plan for C2 wherein all the successful needles are available for source placement leaving out the failed needle.

A satisfactory dose distribution was achieved for every case after dose planning. The maximum difference in target coverage between the planned and implanted needle arrangement was 2%. The planned and actual needle arrangements for each case are shown in Figure 3.10.

Needle Placement Error

The total placement errors were 3.6 mm, 3.0 mm, 4.5 mm, and 2.3 mm, for C1 through C4 respectively. We also computed the rigid transformation between the planned and actual needle arrangement and the associated errors, which are shown in parenthesis in Table 3.3. The RMSE denoting the random error component after the planned and actual arrangements were registered together was 1.9 mm, 2.8 mm, 2.8 mm, and 1.9 mm for C1 through C4, respectively.

We note that the majority of the difference between the non-registered and registered RMSE came from a systematic error in the z-direction. Furthermore, it is also worth noting that on average this error is larger than those achieved with robot-assisted implants in the previous section 3.3. The majority of this error is in the z-direction, which is primarily along the needle insertion direction and least restricted by the CNG.

3.5 Discussion

Chapter Summary This chapter describes the system architecture, algorithms, hardware, and experiments with a human-centered automation system for inserting skew line needle arrangements for HDR-BT. We report results with an open-loop robot guide system that uses CT scans before insertion and does not use sensor feedback during insertion. We also report a similar experiment with needle implants by a novice using Customized 3D printed Needle Guides. For comparison, we also present results from an experiment performed by an expert human physician using ultrasound guidance. These results, in a controlled experimental setup with phantom tissues, suggest that skew line needle arrangements can be planned and executed with a robot guide and patient specific needle guide to achieve the RTOG-0321 clinical treatment objectives while avoiding puncture of sensitive structures such as the penile bulb.

Our results suggest that skew line needle guides can achieve high needle insertion accuracy in the x- and y-direction but have large errors in the z-direction. There are two likely causes for high error in the z-direction. Since the needle insertion direction is primarily in the z-direction, the errors in the z-direction could be caused by uncertainty in needle insertion depth due to reinserting the needles, or the error could be caused by uncertainty in finding the needle tip during segmentation since the scans were taken in 3 mm slices in the z-direction. The needle placement errors are not likely caused by errors in registration since performing a least-squares transformation between the planned and actual needle arrangements did not eliminate the majority of the error.

Limitations We note that Long et al. [78] used the PROSPER robot system (developed for PPI-BT), to insert glass bead markers into a gelatin prostate phantom. After an initial insertion, the needle tip and target bead were measured using 3D ultrasound and needle tip was adjusted along the insertion axis until the error was minimized. Using such intra-operative feedback, the PROSPER system achieved position errors of 2.7 mm. This error, between needle tips and target points, is relevant for PPI-BT. For HDR-BT, we report RMS error along the entire needle which contains dwell positions. We were able to achieve RMS errors of 2.6 – 4.3 mm for the robot and 2.3 – 4.5 mm for the CNG, which is comparable to the error achieved in the closed-loop PROSPER system.

We will explore how calibration can be enhanced with additional CT markers to reduce systematic error and perform experiments to explore how needle insertion order and needle rotation (rifling) may affect needle insertion accuracy. We will also explore how feedback control can be used during insertion.

Some studies like [82, 83] have explored the use of MRI for real-time scanning. Tovar-Arriaga et al. [84] and Ji [85] proposed workflows for needle insertion using CT and MRI feedback respectively. [86, 87] have studied the accuracy of needle placements in real-time MRI tracking. Real-time feedback from either CT or MRI has to deal with trade-off between spatial resolution and temporal resolution. CT can be used for feedback, but it results in radiation exposure to the patient. MRI (magnetic resonance imaging) is relatively slow, requires that all needles and guiding equipment be non-ferrous, and has issues with image warping in larger imaging volumes. As Ultrasound is safe and provides real-time imaging, we will explore how it can be incorporated into active needle guidance.

Chapter 4

Reachability Analysis for Needle Placement in Interstitial Brachytherapy

Overview

Prostate cancer has a high incidence among men, accounting for 14.4% of all cancers diagnosed and 5.1% of all cancer deaths (8.2 million) [88], and brachytherapy is a common and successful treatment modality for prostate cancer. In HDR-brachytherapy for prostate, the radioactive source is delivered to the tumor site using an arrangement of temporarily inserted needles. Currently, this needle arrangement is restricted to only parallel needles by a needle guide; often entailing punctures in healthy organs to reach the target, which in turn may lead to trauma and side effects.

In the previous chapter, we discussed a systematic integration of optimization based needle planning [52] and dose planning [51] algorithms with two methods of skew line needle configuration implants [5, 18]. We operated under the hypothesis that the use of optimized skew line needle arrangements can achieve above par radiation dose distribution without puncturing healthy organs.

In this chapter, we quantify the reachability in a given anatomical setup of tumor volume in the presence of a single healthy organ to avoid with all possible skew-line configurations from a pre-specified entry zone. The goal of this dissertation is to introduce automation into the brachytherapy treatment process and this analysis advances the understanding of treatment planning optimization. This analysis can be used to guide the selection of candidate needles and to identify which subset of the target volume may not be reachable.

In overview, the needle planning problem is to determine a set of non-intersecting line segments that: (a) originate in a pre-specified entry zone, (b) do not puncture any avoidance volume, and (c) provide spatial coverage of target volume. “Spatial coverage” implies that every point in target volume is within a prespecified distance (δ) of at least one needle. NPIP poses this as a set cover problem and optimizes over a large set of randomly generated candidate needles to select a minimal subset providing spatial coverage. Each candidate needle originates in Entry Zone, passes through Target Volume without intersecting Avoidance Volume.

However, a random generation may result in “voids" in the target volume that are not spatially

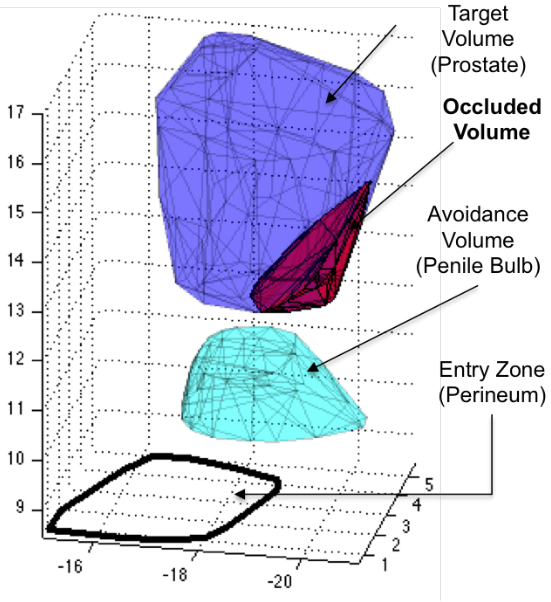


Figure 4.1: This figure shows a target volume (blue), an avoidance volume (cyan), and an entry zone (black), which represent the prostate, penile bulb, and perineum, respectively, in case of prostate brachytherapy. This chapter develops an exact algorithm for finding an occluded volume (red), which is the region inside the target volume that cannot be reached by any straight line originating in the entry zone without intersecting the avoidance volume, or showing that no such volume exists. This algorithm could be an important component of needle planning algorithms for computing brachytherapy needle arrangements.

covered by any candidate needle. This can lead to an inadequate dose to the target and diminish the efficacy of treatment. These voids can be a result of inadequate sampling or because the volume is “occluded", i.e. it is unreachable from the entry zone without intersecting avoidance volume. Figure 4.1 shows a target volume, avoidance volume, and entry zone, which represent the prostate, penile bulb, and perineum, respectively, in the case of prostate brachytherapy along with a resulting occluded volume in this geometry.

Contributions Occluded volume calculation is a step towards systematic needle selection to guarantee the absence of voids in the reachable region and providing complete coverage in the unreachable region. This chapter develops an algorithm for exact characterization of an occluded volume. We model this problem as a linear program which encodes the geometric properties of the prostate. Specifically, contributions in this chapter are:

1. An exact algorithm for finding an occluded volume that exploits the geometric structure and constraints of prostate brachytherapy,
2. A check for the existence of an occluded volume that is a polynomial time in the number of vertices describing the relevant volumes.

We also perform a sensitivity analysis with the motivation that exact contouring of avoidance volume/entry zone is often not possible due to a combination of limitation of imaging and conservative clinical estimates of anatomy contours. And a small uncertainty can result in a significant occluded volume. We use our algorithm to find the occluded volume or show that none exists, for data sets from prostate brachytherapy patients treated at the UCSF Mt. Zion clinic. Given the uncertainty in defining anatomical structures, the effect of systematic perturbations of the avoidance volume and entry zone on occluded volume are analyzed.

4.1 Background and Related Work

This chapter examines the problem of checking the existence of occluded volume in a polyhedral complex. The presented algorithm exploits assumptions about the geometry of prostate brachytherapy, which we use to simplify the problem statement compared to a general occlusion calculation problem. We develop a linear program that is feasible if and only if an occluded volume exists. Thus the existence of the occluded volume can be checked in polynomial time in the total number of vertices (i.e., of the target volume, avoidance volume, and entry zone) without having to perform expensive polyhedron intersection calculations.

Reachability Calculation Many problems in computer graphics, computational geometry, robotics, and automation require an exact reachability (or visibility) computation in the presence of obstacles in 3D. Scherzer et al. [89] provide a survey in shadow computations. These are often performed at a pixel level exploiting parallel computing for handling real-time rendering of large discrete models. CGAL [90] provides efficient algorithms for such calculations. Recent work has also focused on sub-pixel shadow mapping for handling issues like jagged shadows and aliasing has been discussed in [91, 92]. Research in automation has looked at similar problems regarding polyhedral assembly [93] and viewpoint selection [94].

Automated Needle Planning As discussed in the previous chapter, in HDR-brachytherapy for prostate, an arrangement of hollow needles is inserted into the prostate through the perineum, the patch of skin between the testicles and anus. Radiation dose is delivered to the prostate by sequentially threading a radioactive source through each needle. The dose distribution is controlled by halting the source at pre-specified locations along each needle for some dwell time.

Most clinicians insert needles using a rigid template with parallel holes as a needle guide [53]. However, this template restricts the possible insertion locations and directions, which can make puncturing obstructions such as penile bulb unavoidable. Puncturing healthy tissues results in trauma related side-effects such as tissue swelling; urinary infections & incontinence; and impotence [47, part 7], [48]. It has been shown that manually created skew-line (i.e. non-parallel, non-intersecting) needle arrangements [46] can produce dose plans on or above par with standard clinical patterns. The development of needle planning configurations is shown in Figure 4.2.

Siauw et al. [52] recently proposed a novel algorithm, NPIP, which computes patient-specific needle arrangements based on skew line segments. Since these needles can be inserted at a continuum of angles, it is non-intuitive for a human operator to implant the configuration. However, such a skew line needle arrangement can be implanted using an automated needle insertion as described in Chapter 3.

Limitations of Needle Planning Dose planning has three main steps – candidate needle generation, needle selection, and dose planning. The candidate needle set is generated by randomly sampling line segments that originate from the entry zone are inside a projection of the prostate. The needle selection step finds a subset of candidate needle, such that every point in the target is within a distance parameter (δ) from at least one needle in the subset. The heuristic helps ensure

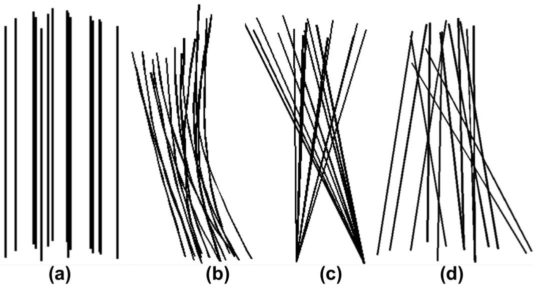


Figure 4.2: The figure shows the progression in body of work in needle configurations in prostate Brachytherapy. Clinical practitioners regularly use needle templates with parallel needles as in (a) [53], which were improved upon by the freehand technique in (b) [49]. Fireworks needle configurations as in (c) were proposed with skew needles to avoid puncture in healthy organs [63]. And latest needle planning system coupled with automated needle insertion allows use of any point in the entry zone as in (d) [52].

that the target can be covered with adequate dose during dose planning. Dose planning is done using Inverse Planning by Integer Program (IPIP) [51], which maximizes dose coverage of the target using only the needles in the subset within dose limits to healthy tissue.

Often there are points in the target volume that cannot be covered by any needle in the candidate needle set (i.e. they are not within the user-specified distance from any candidate needle). Currently, these points are ignored in the needle selection step. This simplification does not significantly impact the final solution if the number of ignored points is small. However, for a given geometric configuration and a parameter δ, it is desirable to provide a guarantee of complete coverage of the target or a proof that no needle arrangement can achieve complete coverage.

The algorithm presented in this chapter for calculation of an exact occluded volume is a step towards this coverage guarantee. Such a guarantee could provide bounds on treatment quality. Our approach would result in a complete algorithm as opposed to the heuristic approach from NPIP. Using the occluded volume, an informed selection of needles can be made for subsequent dose planning. For instance, systematic placement of needles around the occluded volume can achieve desired dose coverage despite incomplete spatial coverage and unreachability of the occluded volume.

4.2 Problem Statement

Given a target volume $\mathcal{T}$, an avoidance volume $\mathcal{A}$, and an entry zone $\mathcal{E}$, we are interested in finding an analytic description of an “occluded volume” $\mathcal{O}$, contained within the target volume $\mathcal{T}$ or show that no such “occluded volume" $\mathcal{O}$ exists. The latter case implies that every point in the target volume $\mathcal{T}$ can be reached by a line segment from *some* point in the entry zone $\mathcal{E}$ without intersecting the avoidance volume $\mathcal{A}$.

In the context of brachytherapy, absence of an occluded volume implies that at least one needle can reach every point inside the target (prostate) from the pre-specified entry zone.

Assumptions: We have assumed that the objects in the environment can be represented as finite polyhedra, and hence are convex. In the case of brachytherapy, the organs are contoured by a physician and polyhedral representation results in the convex hull of the contour. We work with a planar region $\mathcal{E}$, represented by its extreme points. Furthermore, there is no intersection between either of the objects: $\mathcal{T}$, $\mathcal{A}$ and $\mathcal{E}$.

Stated formally, we are interested in finding $\mathcal{O} \subseteq \mathcal{T}$, such that for each $p \in \mathcal{O}$, there exists a $\lambda \in [0,1]$ such that: $[\lambda e + (1-\lambda)p] \in \mathcal{A}$, for all $e \in \mathcal{E}$, or show that no such region $\mathcal{O}$ exists for the given configuration of $\mathcal{T}$, $\mathcal{A}$ and $\mathcal{E}$.

Every polyhedral object in the environment can be specified completely by the vertex set of its convex hull. Let a convex polyhedral avoidance volume $\mathcal{A}$ be represented by the vertex set A of its convex hull. The sets T and E are defined similarly using $\mathcal{T}$ and $\mathcal{E}$. Further, defining $J := \{j : A_j \in A\}$ as the index set of points in A. The index sets $K := \{k : T_k \in T\}$ and $I := \{i : E_i \in E\}$ are defined similarly for T and E, respectively.

Definition: A truncated occlusion cone C_e is the polyhedron which characterizes the occluded volume generated by a convex set of vertices A as viewed from a point $e \notin Conv(\mathcal{A})$. As shown in Figure 4.3, C_e consists of the faces of $\mathcal{A}$ visible from e and all points behind them generated as a conic hull of rays corresponding to visible extreme points. Formally,

$$C_e = \left\{ x \in \mathbb{R}^n : x = e + \sum_{j \in J} \lambda_j (A_j - e), \quad \sum_{j \in J} \lambda_j \geq 1, \quad \lambda_j \geq 0, \forall j \in J \right\} \tag{4.1}$$

In the case of a non-planar specification of $\mathcal{E}$ we can find a suitable projection on a plane. In the case of non-convex objects, we can represent them as a disjoint union of convex sets. The algorithm outputs an occluded volume for each convex subset and the union of these results in final occluded volume. We note that in this case, the final occlusion region may not be necessarily convex.

4.3 Occlusion Volume Calculation Algorithm

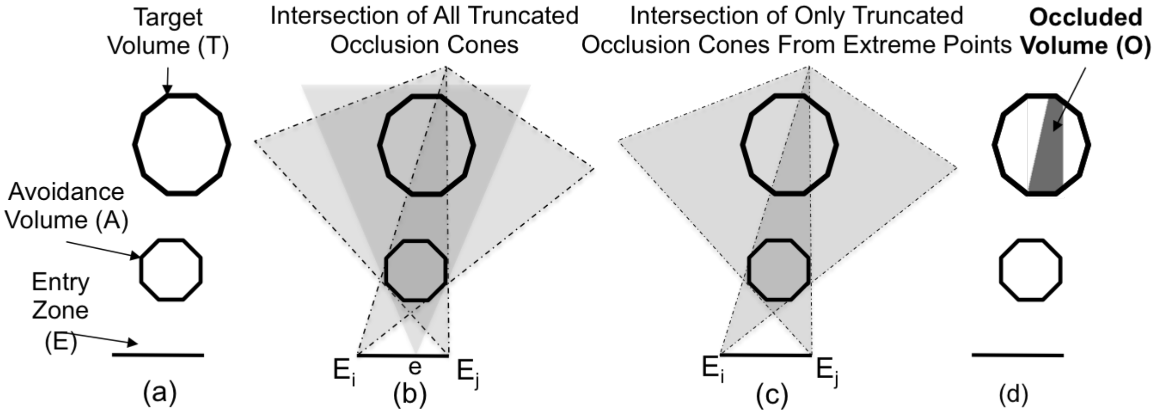


Figure 4.3: The figure illustrates the proposed algorithm for calculation of Occluded volume $\mathcal{O}$. (a) We begin with input geometry with three volumes of interest: Target ($\mathcal{T}$), Avoidance ($\mathcal{A}$) and Entry Zone ($\mathcal{E}$). Fig (b) shows possible occlusion cones $C_e, \forall e \in E$ from different locations at entry zone. (c) However only cones generated from extreme points (C_{E_i}) of the entry region need to considered to calculate occluded volume. Lastly as in (d) The polyhedral representation of $\mathcal{O}$ is output as the intersection of $\mathcal{T} \bigcap \left(\bigcap C_{E_i} \right)$

The proposed algorithm exploits the observation that every point p in the intersection of truncated occlusion cones C_i generated from vertices E_i of a convex closed region $\mathcal{E}$ also lies in the truncated occlusion cone C_e generated from any other point e in region $\mathcal{E}$. Stating the above formally:

$$\begin{aligned}
&\forall p \in \bigcap_{i\in I} C_{E_i},\ \exists\bar{\lambda} \in \mathbb{R}^{A}:\ \sum_{j\in J}\bar{\lambda}_j \geq 1;\ \bar{\lambda} \geq 0\\
&\text{such that: } p = e + \sum_{j\in J}\bar{\lambda}_j(A_j - e) \quad \text{i.e.} \quad p \in C_e, \\
&\text{where } e = \sum_{i\in I}\mu_i E_i,\ \sum_{i\in I}\mu_i = 1,\ \mu_i \geq 0, \forall i \in I
\end{aligned} \tag{4.2}$$

This entails that any point in $\mathcal{T}$ occluded from E_i, $\forall i \in I$ is also occluded from any point in $\mathcal{E}$. Every point $p \notin \bigcap C_i$ is classified as visible, since $p \in C_i$ for at least one $i \in I$. Also, every point $p \in C_e$ is either visible from at least one E_i, i.e. $p \notin C_i$, or lies in the occluded volume $\mathcal{O}$, if it exists, i.e. $p \in \bigcap C_i$.

An intuitive explanation for this observation is if a point $p \in \bigcap C_i$, then none of the E_i's can see p. Moreover, by the reversibility of visibility, p can't see any of the E_i's either. Since $\mathcal{E}$ is convex, hence by extension p should not be able to see any other point in $\mathcal{E}$. Moving the viewing perspective to any point $e \in \mathcal{E}$, and by a similar argument as above, any $e \in \mathcal{E}$ can't see p either. This observation significantly improves the computational effort in the calculation of the occluded volume, since we only need to generate truncated occlusion cones on the finitely many vertices of the convex region E, instead of every point in E, or a discretization thereof.

Hereafter, we build a linear system of equations to check for the existence of an occluded region $\mathcal{O}$. The system 4.3 is feasible if $\mathcal{O}$ is non-empty and infeasible otherwise. If the system of equations returns feasibility, we calculate the polyhedral representation of $\mathcal{O}$ using an iterative method for intersecting polyhedrons.

4.3.1 Check for Existence of Occlusion Volume $\mathcal{O}$

Assume $\mathcal{O}$ is non-empty, and let a point $x \in \mathcal{O}$. Then by definition, x must have the following two properties:
(1) It must be a convex combination of the vertices of $\mathcal{T}$ (since $\mathcal{O} \subseteq \mathcal{T}$).
(2) The line segment between x and some point $e \in \mathcal{E}$ must have at least one point that is a linear combination of the vertices of $\mathcal{A}$.

$$\begin{aligned}
x &= \sum_{k\in K}\mu_k T_k,\\
&\sum_{k\in K}\mu_k = 1, \quad \mu_i \geq 0, \forall i \in I,
\end{aligned}$$

which is simply the definition of convex combination representing the first condition.

While the second condition is represented by $x \in C_e$. In other words, the conic hull of A from e is the set of rays originating at e that have at least one point in A, i.e., at least one point that is a linear combination of the vertices of $\mathcal{A}$. So by definition, x must be in the intersection of $C_e, \forall\, e \in \mathcal{E}$.

$$x \in \bigcap_{e \in \mathcal{E}} C_e.$$

To simplify our representation of this property of x, which requires an intersection of an infinite number of conic hulls, it is sufficient to find the intersection of the conic hulls of A around E_i, $i \in |E|$, the vertices of $\mathcal{E}$. That is,

$$\bigcap_{e \in \mathcal{E}} C_e = \bigcap_{i \in I} C_{E_i}$$

We can then represent the second property of x using

$$x = E_i + \sum_{j \in J} \lambda_{ij}(A_j - E_i), \quad \forall i \in I$$

$$\sum_{j \in J} \lambda_{ij} \geq 1,\ \forall i \in I, \lambda \geq 0,$$

which states that x must be in the conic hull of A originating at each E_i, $i \in I$. We note that the condition: $\sum_j \lambda_{ij} \geq 1, \forall i \in I$, enforces that only the points situated beyond the visible boundary of the avoidance region $\mathcal{A}$ are included.

Put together, we get our linear system, (LS).

$$\begin{aligned}
\text{(LS)}\ x = {} & E_i + \sum_{j \in J} \lambda_{ij}(A_j - E_i), \forall i \in I \\
x = {} & \sum_{k \in K} \mu_k T_k, \\
& \sum_{k \in K} \mu_k = 1, \mu \geq 0 \\
& \sum_{j \in J} \lambda_{ij} \geq 1,\ \forall i \in I,\ \lambda_{ij} \geq 0, \forall i \in I, \forall j \in J.
\end{aligned} \tag{4.3}$$

By construction, any x value that satisfies (LS) cannot be connected to a point in $\mathcal{E}$ by a line segment that does not intersect $\mathcal{A}$. Therefore, the feasible region of x in (LS) is exactly $\mathcal{O}$, the occluded volume. Consequently, if (LS) is infeasible, then $\mathcal{O}$ is empty and the entire target volume can be reached by at least one line segment originating in the entry zone that does not intersect the avoidance volume. If (LS) is feasible, then $\mathcal{O}$ is not empty, and it is worth noting that since the feasible region is made up from a set of linear constraints and is bounded by $\mathcal{T}$, that $\mathcal{O}$ is a polytope when it is not empty.

4.3.2 Calculation of Occlusion Volume $\mathcal{O}$

We use Algorithm 2 to obtain the polytope $\mathcal{O}$. Polytopes can be described using inequalities (H-polytopes) or vertices (V-polytopes). H-polytopes can be converted to V-polytopes using vertex

Algorithm 2: `Get Occluded Volume` $\mathcal{O}$

```
Solve the system 4.3 as LP with zero cost
if LP is infeasible then
    no occluded volume exists
else if LP is feasible then                    // exists occluded volume
    foreach $i \in I$ do
        compute the smallest cone $C_{E_i}$ pointed at $E_i$ covering $A_j, \forall j \in J$
    compute the intersection $C$ of cones $C_{E_i}, \forall i \in I$
    compute the Occluded Volume $O$ as the intersection between $C$ and target $T$.
```

enumeration and vice-versa by facet enumeration. In this case, the input is in the form of V-polytopes while output as H-polytope for a complete representation of $\mathcal{O}$. A preliminary approach for calculation of intersection of convex polyhedra is to start by converting the input polyhedra to H-representation. Then, the intersection is a redundancy removal problem in the union of inequality systems. The minimal H-representation of the union results in H-polytope $\mathcal{O}$. Furthermore, to get vertices of $\mathcal{O}$, we can solve a vertex enumeration problem.

4.3.3 Complexity Analysis

The linear system of equations for checking existence of a occluded volume can be solved as an LP. Assuming A, T and E have m_1, m_2 and m_3 points respectively and dimension of space being n. Furthermore, also assuming non-degeneracy we have: $n \leq m_1, m_2, m_3$. Then, the system 4.3 has $(n + m_1 + m_2 m_3)$ variables and $(n(m_2 + 1) + m_3 + 1)$ constraints apart from the non-negativity constraints. LP has known polynomial time complexity [95].

Furthermore, in case there exists an occluded volume or the linear system 4.3 returns feasible solution, we calculate truncated occlusion cones $C_{E_i}, \forall i \in I$. Thereafter, a sequence of convex polyhedron intersections are performed to obtain $C := \bigcap C_{E_i}$. Every occlusion cone C_e has $O(m_2)$ extreme rays. We perform m_3 such intersection operations for calculating $C := \bigcap C_{E_i}$, and thereafter one more intersection with target volume T to get obtain O. Hence the complexity for the intersections is in $O(m_2 m_3)$. Polyhedral intersection, in general, is shown to be NP-Hard [96]. However, using the above approach, solving for redundancy in the union of k H-polytopes is an LP for convex polyhedra. An intersection of two such occlusion cones can be performed in polynomial time for low dimensions [97].

4.4 Experiments with UCSF Prostate Cancer Data

4.4.1 Preprocessing Clinical Dataset

We tested our algorithm on 18 anonymized anatomy cases taken from patients previously treated at UCSF [51]. This data set contained surface points on the prostate and the penile bulb, which were termed as the target volume $\mathcal{T}$)and avoidance volume $\mathcal{A}$, respectively. The prostate volumes ranged from 27 to 97 cm^3 and the penile bulb volumes ranged from 1 to 11 cm^3.

A specification of the entry zone did not exist in the data set because segmenting an entry zone is not a standard practice in HDR-brachytherapy for the prostate. In clinical procedures, the available entry zone near the perineum is visually assessed by the physician during needle insertion. We used the needles entry points in the actual cases as the available region. For this chapter, the entry zone was defined as the convex hull of these needle entry locations projected on a plane parallel to the ground (x-y plane) and located 2 cm below the penile bulb. This is a reasonable anatomical proxy for the perineum since the distance between the penile bulb and the perineum in human anatomy is also ~ 2 cm. This is a conservative estimate of the possible entry zone; however, it ensures that we restrict the entry zone to an area used by the physician. We note that the entry zone to allow for skew-line needles could, in fact, be larger than this, which would reduce occlusion. The entry zone area ranged from 7 to 13 cm^2.

4.4.2 Occluded Volume Analysis

Table 4.1 lists the size of occluded volumes for the different cases in the dataset. For all cases, we checked for the existence of an occluded volume in the prostate where needles cannot reach without puncturing the penile bulb. We found that 11 out of 18 patients had a non-empty occluded volume. Among the patients with occlusion, the size of the occluded volume ranged from 0.06 cm^3 to 2.4 cm^3 or 0.01% to 4.3% of the respective target volume. The running time ranged from 4 s to 9 s. It is worth noting that time on the order of seconds is inconsequential to the overall brachytherapy workflow.

We implemented our algorithm in Matlab. We used the Matlab interface for Multi-Parametric Toolbox 3 (MPT) [98] to calculate extreme points of a convex bounded polyhedra and to perform polyhedral intersection operation. MPT toolbox implements polyhedral intersection queries in low dimension using the approach described in section 4.3.2. The algorithm was run on a computer with OS/X 10.9.2m 2.7 GHz Intel core i7, and 16 GB memory.

4.4.3 Sensitivity Analysis

The size of the penile bulb has substantial uncertainty in its definition owing to difficulty in its identification on a CT scan. We used our algorithm for computing the occluded volume in each patient's target volume for various dilations of the nominal penile bulb. The dilations were computed by expanding the vertices of the penile bulb about its centroid by a dilation factor (α) that ranged from 0.75 to 1.25 in increments of 0.05, where $\alpha = 1$ results in no change (i.e. its original

Table 4.1: Results from 18 Patient Cases listing Target volume T, Avoidance Region Volume A, Occluded Volume O in absolute and as percentage of Target volume. The last column lists time required for computation in seconds.

Px	T (cm^3)	A [cm^3 (% T)]	O [cm^3 (% T)]	Time (s)
1	33	8 (23)	1.00 (3.10)	8
2	27	5 (19)	0.00 (0.00)	7
3	44	10 (23)	1.80 (4.10)	7
4	31	4 (14)	0.26 (0.86)	5
5	39	3 (8)	1.00 (2.60)	4
6	55	7 (14)	2.40 (4.30)	9
7	58	3 (5)	0.00 (0.00)	4
8	38	4 (11)	0.00 (0.00)	6
9	31	10 (34)	0.00 (0.00)	8
10	46	7 (15)	0.06 (0.12)	6
11	74	4 (6)	0.00 (0.00)	5
12	32	4 (11)	0.15 (0.48)	5
13	97	1 (1)	0.00 (0.00)	3
14	66	10 (15)	2.30 (3.50)	7
15	40	4 (10)	0.00 (0.00)	4
16	50	6 (12)	0.09 (0.18)	5
17	45	8 (17)	0.85 (1.90)	9
18	27	9 (34)	0.003 (0.01)	9

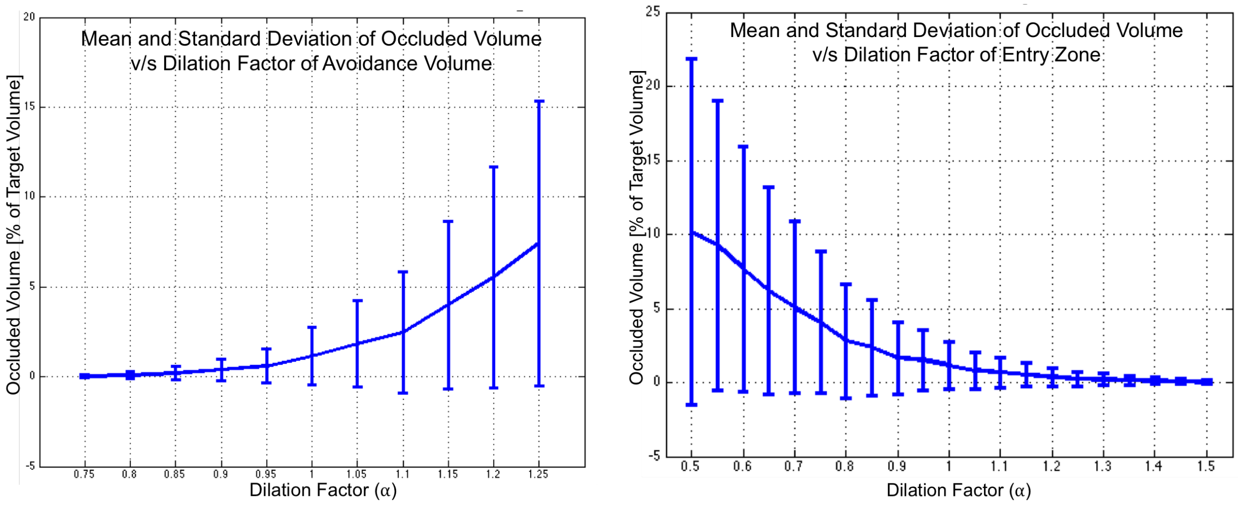
size). For each iteration, we recorded the size of the occluded volume, the volume of the dilated bulb, and the running time of the algorithm.

Likewise, entry zone specification used in the dataset is a conservative estimate. Dilation of the entry zone is prone to the introduction (or removal) of occlusion. We have performed calculations of occluded volumes for various dilations of the entry zone about its centroid by a dilation factor (α) in $[0.5, 1.5]$ in increments of 0.05.

Figures 4.4a and 4.4b show variation in existence and size of occluded volume as a % of target volume for various values of dilation factor α in case of penile bulb and entry zone, respectively. The error bars for each α value represent the measure of the standard deviation for the whole data set. An error bar corresponding to an occluded volume $< 0\%$ implies no occlusion exists. Figure 4.5a shows the change in occluded volume with avoidance volume (penile bulb) dilation in one particular anatomical case. Similarly, Figure 4.5b shows the change in occluded volume with entry zone dilation for another case in the data set.

4.5 Discussion

The occluded volume in the target, where needles starting in the entry region cannot reach without puncturing the avoidance volume (penile bulb), if it exists, is found for all 18 cases in the data set. Computational results listed in Table 4.1 reveal that $\sim 60\%$ (11 out of 18) of cases in the



(a) (b)

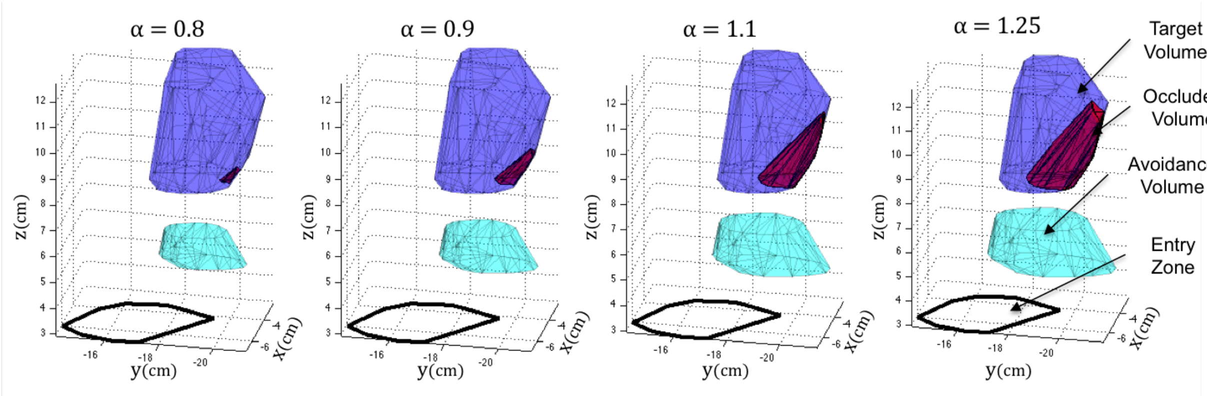
Figure 4.4: The figure illustrates the change in normalized occluded volume v/s change in dilation factor (α) for (a) avoidance volume and (b) Entry Zone. The error bars show the standard deviation across the patients. An occluded volume value $< 0\%$ implies no occlusion exists. The graphs illustrate a possibility of large changes in occluded volume with small uncertainty in specification of either avoidance volume or entry zone.

data set have non-zero occluded volume. Owing to the variance in location and size of the organs and the entry zone, the results ranged from no occluded volume to (4.3%) of the prostate being occluded. In practice, an occluded volume inside the target volume could lead to voids in needle coverage, which results in areas that are difficult to cover with adequate dose and thus make the overall treatment of the patient less effective.

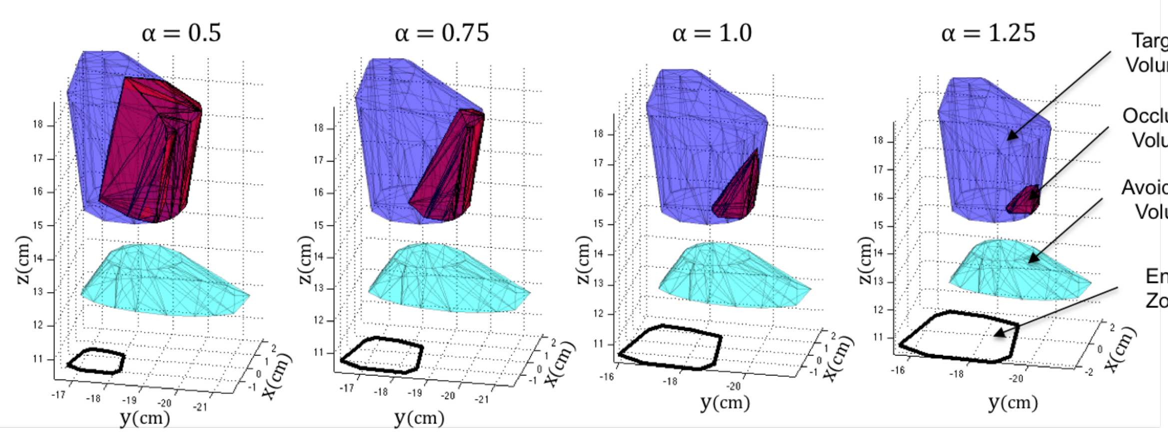
Furthermore, the effect of dilation on the avoidance volume and entry zone to occluded volume emphasizes the importance the proposed approach. As noted earlier, identification of true boundaries of the penile bulb may be difficult, and hence a conservative estimate may be made. We note from Figure 4.4a that the occluded volume is fairly sensitive to dilation of avoidance volume. Conservative errors of 5% in the specification of margins of avoidance volume may result in $> 5\%$ of the target volume being occluded in some cases.

Moreover, the entry zone is not always precisely defined. A similar analysis shows that the occluded volume is relatively less sensitive to entry zone perturbation than to avoidance volume perturbation. As noted in Figure 4.4b, a 10% contraction in entry zone results in ∼5% occlusion in target volume. While on the other hand, at 40% expansion of entry zone, all cases result in 0 occluded volume.

Figures 4.5a and 4.5b provide a qualitative review of occlusion events and resulting occluded volumes for a particular case highlighting the rapid growth of occluded volume with dilation of avoidance volume and entry zone respectively. The errors in estimating avoidance volume and entry zone may not always be isotropic as the centroidal dilation in the sensitivity analysis. We observe that occlusion is often resultant of obstruction from a small subset of neighboring faces.



(a) Occluded volume grows with the avoidance volume.



(b) Occluded volume grows as the entry zone shrinks and a non-zero occlusion exists at even at $\alpha = 1.25$.

Figure 4.5: Sensitivity analysis on the occluded volume calculation yields insight on the occluded volume. Uncertainty in specification of either entry zone or avoidance volume can result in large changes in occluded volumes. The figure shows variation in occluded volume with dilation (by factor α) in the avoidance volume and entry zone in two anatomy instances.

However, prior knowledge of this obstructing neighborhood is not available, necessitating a check in all directions.

Moreover, in the case of non-zero occluded volume, we could perform a similar perturbation in the entry region vertices cyclically instead of isotropic dilation. This would result in the identification of directions, expansion in which would result in an elimination of occlusion.

Limitations: The complexity of the algorithm is proportional to a number of vertex points in the convex hull of the polyhedra. A high fidelity representation would thus result in slow computations. Furthermore, for time critical applications, faster implementation of polyhedral intersection operation with use of parallel computing may be explored.

Currently, the algorithm assumes convex anatomical structures. While in the case of a non-convex avoidance volume, a pre-processing step would section it into disjoint convex subsets. Then our procedure can then be used in parallel for each subset, and performing a union operation in the end. The framework can be generalized to have more than one avoidance volumes, for instance, the pubic arch in the case of prostate brachytherapy. But in both cases, the union operation results in a heuristic without a guarantee on the absence of occlusion.

Future Work: In addition to checking the feasibility of a brachytherapy procedure without puncturing healthy organs, the output of the algorithm can be used to improve treatment planning. In case the target is not completely visible, we can calculate the extreme points of the entry zone which can reach the boundary of the occluded volume in the target. Needles starting from these points can be adaptively re-weighted for higher “importance" in the needle subset selection optimization of the NPIP algorithm [52].

Moreover, the algorithm is also applicable to other applications with reachability calculations: such as visual inspection, robotic spray painting, and other clinical procedures such as biopsy. A kidney biopsy is one such problem where the physician is interested in reaching a particular region inside kidney without puncturing healthy organs.

Chapter Summary

This chapter considers the problem of exact reachability analysis for checking the existence of occluded volume in a target in the presence of an avoidance volume not reachable from an entry zone, where all three regions of interest are modeled as convex polyhedra. The proposed algorithm performs the check in polynomial time and if it exists returns the polyhedral representation of the occluded volume. This analysis forms a basis for identification of candidate needle set generation and needle subset selection for treatment radiotherapy treatment planning in brachytherapy. We have shown computational results on actual anatomical cases of prostate cancer and quantified the occluded volume in each case. We have found that 11 out of 18 cases have non-empty occluded volume ranging from 0.01% to 4.3% of the target volume. Such occlusions can lead voids in radiation coverage, thus reducing the effectiveness of overall treatment. Furthermore, we have conducted a sensitivity analysis to study the change in occluded volume with dilation of the avoidance volume and the entry zone to emphasize that small uncertainty in the specification of regions of interest may result in large changes in occluded volumes in the target.

Part II

Learning Sub-tasks in Robot-Assisted Minimally Invasive Surgery

Chapter 5

Multi-Throw Suturing: A case study in Sub-Task Automation

Overview

Robot-Assisted Minimally Invasive Surgery (RMIS) was used in manual teleoperation mode in over 500,000 procedures worldwide in 2015 with 3600 systems [6]. Robotic surgical assistants (RSAs), such as the da Vinci system from Intuitive Surgical, address the ergonomic constraints of hand-held laparoscopic tools by providing a master-slave interface with 3D immersive visualization. RSAs provide surgeons with a precision laparoscopic tool that offers higher dexterity and range of motion. As noted in [6], – “an RSA utilizes computational, robotic and imaging technologies to enable improved patient outcomes compared to other surgical and non-surgical therapies. da Vinci Surgery is aimed towards advancing the critical surgical ideals of entering the body less invasively, seeing anatomy more clearly, interacting with the tissue more precisely and building surgical skills." Clinical RMIS systems have focused on not only general surgery but also on other surgical specialties such as gynecologic, urologic, cardiothoracic, and head & neck.

RMIS has ushered in an era of shorter recuperation time, lower patient trauma, and lesser tissue injury [7–9, 12]. Regardless of the benefits, RMIS requires skilled surgeons to perform tediously long procedures with reduced sensory perception during surgical manipulation. Moreover, RSAs are currently operated by surgeons using pure teleoperation with little to no intelligent behavior by the system. However, automation of manipulation tasks in surgery such as suturing has the potential to assist surgeons, reduce tedium and fatigue, and facilitate supervised autonomy for remote telesurgery.

Automation of multilateral manipulation has the potential to reduce the time required for surgical procedures, reducing the time patients are under anesthesia and associated costs and contention for O.R. resources. Autonomous manipulation of deformable materials with two or more arms is of particular interest with a potential for surgical robot systems to be configured with more than two arms to perform tasks that may be difficult via dual-arm teleoperation. Multilateral manipulation is also necessary for common surgical tasks such as cutting and suturing; hand-off of tissue

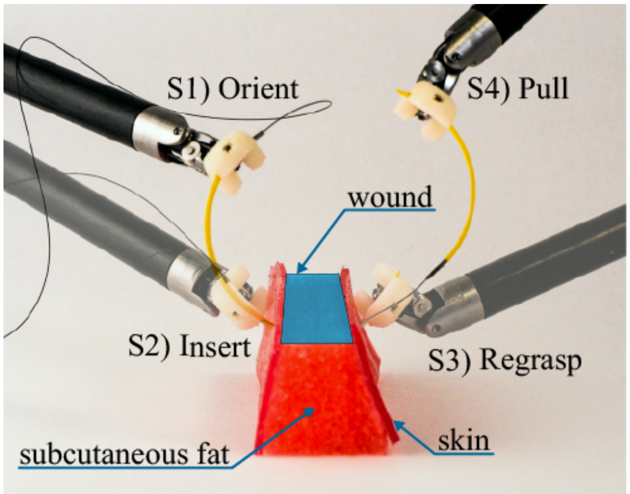


Figure 5.1: *Suturing Sub-Task Automation in RMIS*: Each throw in Multi-throw Suturing (MTS) includes five steps: (S1) Needle placement in desired position and orientation by first actuator, (S2) Needle insertion through tissue by first actuator, (S3) Needle grasp by second actuator, (S4) Needle and thread pull until thread is taut, and (S5) Needle transfer back to first actuator. (note: S5 is not illustrated in this time-lapse image).

or tools between arms is common as each arm has limited dexterity and a workspace that may not cover the entire body cavity.

Sub-task Automation One approach towards automation in RMIS is to study sub-task automation similar to the training modules created for surgical training. The Fundamental Skills of Robotic Surgery (FSRS) defines a representative set of procedures for surgical training and evaluation [99].

This chapter focuses on one of the tasks from the FSRS curriculum: Multi-Throw Suturing (MTS). Each MTS throw includes five steps as illustrated in Figure 5.1. A curved needle with suture thread is repeatedly pushed through a pair of tissue boundaries with one actuator, then pulled through with a second actuator until the thread is taut, then is transferred back to the first actuator to begin the next throw / suture [100, 101]. In Robot-Assisted Minimally Invasive Surgery (RMIS), MTS is a tedious subtask and it can be difficult for a non-expert surgeon to maintain proper needle pose during insertion and transfer as haptic feedback is not available.

In this chapter, we present a systematic approach towards automating MTS with new hardware and a novel optimization algorithm. Our approach includes (1) a mechanical device, the Suture Needle Angular Positioner (SNAP), designed to align and hold the needle in a known orientation, (2) computer vision system to track needle pose, and (3) a sequential convex optimization formulation of needle motion planning. Initial results suggest that SNAP can reduce error in needle orientation by 3x and that the combined system can successfully complete 86% of attempted throws at 30% the speed of human operators [20].

5.1 Background and Related Work

RSAs are being used for many tumorectomy interventions within the abdominal and thoracic cavities such as prostectomy and hysterectomy [9, 102] as described in reviews of recent developments in semi-autonomous and autonomous execution of surgical procedures by Moustris et al. [103] and Kranzfelder et al. [104].

Automated Suturing: Automation of suturing has been studied in the context of hierarchical models for multi-step task planning [105], multilateral manipulation of needle and suture [106], and interaction with deformable tissue [107, 108].

While each of these studies made significant contributions as outlined below, challenges in combining the steps to achieve autonomy in longer tasks has not been sufficiently addressed. Kang et al. devised a specialized stitching device for RMIS which is capable of tying a knot [105]. Mayer et al. used a recurrent neural net as part of a controller to learn knot tying with three industrial arms using motion primitives from human demonstrations [109]. Van den Berg et al. used iterative learning for performing knot tying at super-human speeds [110]. More recently, Schulman et al. used a learning by demonstration approach to warp recorded expert demonstrations and perform suturing in simulation and on a scaled-up robotic setup [111]. Padoy et al. showed the execution of collaborative human-robot suturing, but the key sections requiring interactions such as needle insertion and hand-off were performed manually [112]. Similarly, Staub et al. automated needle insertion into tissue for single-throw suturing [106].

Prior work in surgical automation has modeled the basis set of surgical motions as the “Language of Surgery" composed of surgemes (Hager et al.) [113]. Recent works have also explored the use of learning techniques to infer surgeme transitions from demonstration data [114, 115]. Many of the FSRS procedures, including MTS, are decomposable into long sequences of simpler sub-tasks. This decomposition allows the parametrization and building of Finite State Machines (FSM) for complex procedures using a learning by observation approach, for tasks such as tissue debridement [116], pattern cutting [117], and tumor localization & resection [118].

Suture Needle Path Planning: Some preceding studies use a needle path of fixed curvature. Jackson et al. used a reference trajectory to create an analytical solution allowing for needle insertion without considering uncertainty or robot pose constraints [108]. However, needles do not always follow their natural curvature. Interaction with tissue may deflect the needle, and end point pose constraints necessitate non-orthogonal exit. The use of optimization-based planning has the potential to address these limitations. Recent results in motion planning have shown that Sequential Convex Programming (SCP) based planning, such as [119] can be both faster and more successful in finding solutions than sampling-based planners. This chapter formulates suture needle path planning as a curvature constrained SCP based optimization problem.

This chapter builds on prior work in optimization-based planning [120, 121], sub-task level segmentation of demonstrations [15, 115], gripper mounted interchangeable tools [118], and building robust finite state machines [117]. This is one of the first studies which automate multi-throw suturing along with a concurrent and independent study by Shademan et al. [122].

5.2 Problem Statement

The objective is to compute a suturing plan for a given wound geometry, i.e., choice of needle curvature, number of sutures needed, and trajectory plan for each suture. After the planning the system has to perform the sutures under closed loop-visual feedback.

The success of suturing is highly sensitive to needle pose uncertainty at entry point. Uncertainty in needle pose during insertion can result in tissue injury due to skin penetration at undesirable angles or the lack of sufficiently deep needle insertion to hold the suture securely. As illustrated by the several error cases in Figure 5.2, it is essential to maintain proper needle pose during insertion and handover to avoid dropping the needle or damaging tissue. Since the needle is thin and highly reflective, it is difficult to accurately detect its position and orientation with computer vision as noted in [108, 123–125]. Several medical device manufacturers offer needle-alignment devices for manual laparoscopic applications [126, 127] but, to the best of our knowledge, these are not available for RSAs.

Surgeons follow suturing task guidelines such as entering the tissue orthogonally, minimizing tissue-needle wrench, choosing the correct needle size for adequate suture depth and inserting the needle to a sufficient depth to ensure needle protrusion for needle re-grasp. While a needle would follow a constant curvature path through rigid objects, tissue is deformable. Thus we model the needle path to allow bounded rotations about the needle tip while the needle is inserted. However, needle paths that do not follow the natural curvature of the needle can result in tissue damage, hence we define a bounded deviation (γ) from needle curvature (κ) that can be visualized as a cone at each point as illustrated in Figure 5.3. We monotonically reduce γ as the needle progresses to minimize tissue damage.

Assumptions: We assume that tissue is homogeneous and deformable. Real-time tracking and planning are used to account for departures from needle pose estimates during needle insertion. We assume that the needle is rigidly held in the gripper and can only move forward in the tangential direction of the tip. However, bounded reorientation of the needle tip is permitted as it is inserted through tissue. We assume that our system has access to a continuous range of needle sizes. In practice, needles vary in length in increments of 1 mm and vary in three different fractions of a circle.

Input: The wound shape is provided as input, with the points $\mathcal{M} = [M_1, M_2, \ldots, M_D] \in \mathbb{R}^3$ representing the wound surface as a spline. The system is also provided with suture depth d, suture width l, and a pair of entry/exit poses ($P_i, P_f \in SE(3)$) for the first throw as illustrated in Figure 5.2. Further, we are also given suture pitch w – distance between consecutive suture throws.

Output: The system needs to find a set of suture throws $\mathcal{S}$, where $\forall\ S_j \in \mathcal{S}$, we need to calculate an optimized sequence of needle tip poses $\mathcal{X}_j \in SE(3)$ satisfying the suture depth and suture width constraints or report that no such path plan exists. The system also needs to choose a needle curvature and length. The entry and exit positions at each suture throw S_j are obtained by linearly interpolating P_i, P_f along the spline while keeping the orientation constant.

Curvature Constrained Kinematic Model

The needle trajectory is discretized into time intervals $\mathcal{T} = \{0, 1, \ldots, T\}$, where the needle moves a fixed length (Δ) at each time step. At each time step the needle's pose is parametrized as $X_t \in SE(3)$.

We model the needle trajectory as a sequence of $T-1$ circular arcs with curvature κ_t between every consecutive pair of needle poses ($X_t,\ X_{t+1}$). We model our control of the needle at each time step as a rotation and insertion where at each time step the pose X_t is propagated a distance

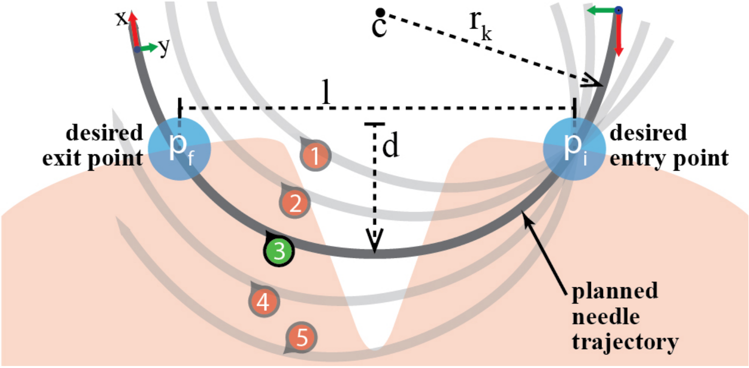


Figure 5.2: The needle trajectory labeled (3) shows the desired trajectory along with poses at entry and exit points from the tissue. The success of suturing depends on the correct orientation of needle with respect to the tissue. For example, uncertainty in needle pose at entry point may result in the needle not connecting opposite tissue sides (1), not making sufficiently deep insertion to hold the suture securely (2), not having enough length of needle at the other end to enable re-grasping (4), or passing completely under the wound and not exiting the tissue at all (5).

Δ to X_{t+1}. Although a needle naturally follows a path of constant curvature, the needle tip can be reoriented at each time step to change the local curvature by $\bar{\gamma}_t$. Thus at each time step the path curvature κ_t can be expressed as $\kappa_t = \kappa + \bar{\gamma}_t$ where κ is the curvature of the needle and $\bar{\gamma}_t$ is the change in curvature applied at each time step. The transformation between consecutive needle poses can be represented a twist in $\mathfrak{se}(3)$ $\quad u_t = [\,\Delta\; 0\; 0\; 0\; \Delta\; \kappa_t\; 0\,]$.

The Lie group $SE(3)$ and the corresponding algebra $\mathfrak{se}(3)$ are related by the exponential and log maps $\exp : \mathfrak{se}(3) \rightarrow SE(3)$ and $\log : SE(3) \rightarrow \mathfrak{se}(3)$. Closed form expressions exist to compute these maps efficiently. Given an incremental twist $x = [p_x\; p_y\; p_z\; r_x\; r_y\; r_z]^T \in \mathbb{R}^6$, the corresponding Lie algebra element is given by the mapping $^\wedge : \mathbb{R}^6 \rightarrow \mathfrak{se}(3)$ as

$$x^\wedge = \begin{bmatrix} 0 & -r_x & r_y & p_x \\ r_z & 0 & -r_x & p_y \\ -r_y & r_x & 0 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The reverse mapping $^\vee : \mathfrak{se}(3) \rightarrow \mathbb{R}^6$ can be used to recover the twist, x from an element of $\mathfrak{se}(3)$. Poses between consecutive time steps can then be related as: $X_{t+1} = \exp(u_t^\wedge) \cdot X_t$.

5.3 Our Approach for Needle Planning and Manipulation

5.3.1 Suture Needle Path Planning

The Suture needle Path Planning (SPP) problem can be formulated as a non-convex, curvature constrained motion planning problem solved with a series of locally convex approximations using sequential convex programming (SCP). We begin by presenting the problem formulation.

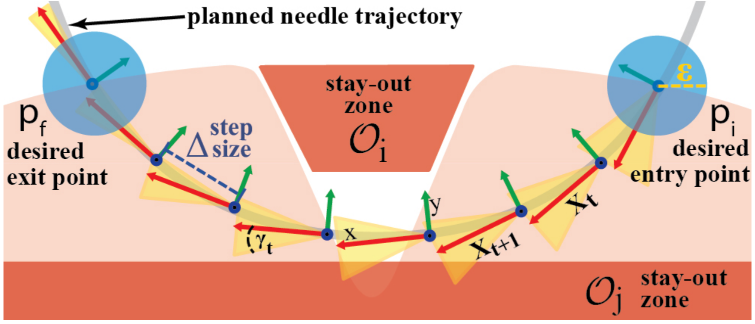


Figure 5.3: The optimization steps and non-holonomic motion at each time-step. The figure shows stay-out zones $\mathcal{O}_i$, trajectory poses X_t, step-size Δ, needle radius r, and γ-cone of allowed rotation at each X_t.

Optimization Model:

For notational convenience we concatenate the states from all time steps as $\mathcal{X} = \{X_t : t \in \mathcal{T}\}$ and control variables as $\mathcal{U} = \{\kappa, \Delta, \gamma_t : t \in \mathcal{T}\}$

$$SPP : \underset{\mathcal{X},\mathcal{U}}{\text{minimize}} \quad \alpha_\Delta \mathcal{C}_\Delta + \alpha_I C_I \tag{5.1}$$

$$\text{s.t.} \quad \log(X_{t+1} \cdot (\exp(u_t) \cdot X_t)^{-1})^\vee = 0_6 \tag{5.2}$$

$$|\bar{\gamma}_t| \le \gamma_t \qquad \forall t \tag{5.3}$$

$$T\Delta + 2l_g - \frac{2\pi l_n}{\kappa} \le 0 \tag{5.4}$$

$$\text{sd}(X_t, \mathcal{O}_i) \ge d_s, \qquad \forall i \tag{5.5}$$

$$X_0 \in \mathcal{B}(p_i, \varepsilon),\ X_T \in \mathcal{B}(p_f, \varepsilon) \tag{5.6}$$

Each term in the above formulation is described below:

Costs (Eqn. 2): We assume the volume of the needle in tissue is proportional to tissue trauma and hence we penalize longer trajectories such that $C_\Delta = T\Delta$, the length of the trajectory. Furthermore, surgical guidelines suggest that the needle entry pose should be orthogonal to the tissue surface. C_I penalizes deviations from an orthogonal start pose. The weights α_Δ and α_I are parameters that are tuned in the optimization.

Kinematic Constraints (Eqns. 3, 4): The kinematic constraint in Eqn. 1 can be transformed using the exponential log map into the standard equality constraint in Eqn. 3. Eqn. 4 bounds the magnitude of $\bar{\gamma}_t$ to minimize tissue damage. We select γ_t to be monotonically decreasing with t because needle rotations away from its natural curvature cause greater damage the further the needle is inserted into the tissue.

Needle Length Constraints (Eqn. 5): The length of the insertion trajectory ($T\Delta$) is constrained to be less than the length of the needle ($2\pi l_n/\kappa$) and should allow for grippers to hold the needle on both ends ($2l_g$).

Collision Constraints (Eqn. 6): We impose constraints to ensure that our trajectory avoids collisions with pre-defined stay out zones. We ensure that the signed distance between each X_{t+1} and each convex mesh in $\mathcal{O}$ is greater than a safety margin parameter d_s. The stay out zones can be non-convex meshes that can be decomposed into convex sub meshes [128], $\mathcal{O} = \{\mathcal{O}_1, \ldots, \mathcal{O}_i\}$.

Entry and Exit Point Constraints (Eqn. (7)): We constrain the start and end poses of the trajectory to be within an ε-Ball of the calculated entry (p_i) and exit (p_f) poses. This can be

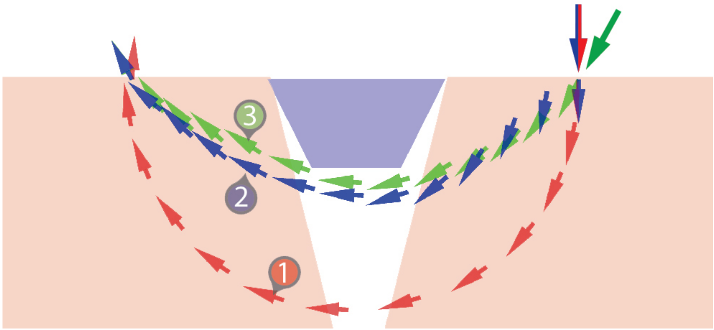


Figure 5.4: *Constraints of Trajectory Optimization*: The side view of three needle trajectories generated by SPP. Trajectory 1 and 3 are constant curvature trajectories whereas trajectory 2 is a variable curvature trajectory.

expressed as $\log(p_i \cdot X_0^{-1})^\vee \leq \varepsilon \cdot \mathbf{1}_6$ for the start pose of the trajectory. The end pose constraint follows a symmetric formulation.

We note that a constant of Δ is chosen for all time steps instead of having a different Δ_t for each time. as the latter is experimentally found to disagree numerically with the findings of Duan et al. [120].

Trajectory Optimization

Sequential Convex Programming (SCP) is a general approach for solving constrained, non-convex optimization problems. We refer the reader to [129] for the details of SCP-based motion planning are described.

Figure 5.4 shows the SPP output for three different sets of pose constraints. For #1, we restrict rotation about needle tip ($\gamma_t = 0, \forall t$). Coupled with the orthogonality constraint at entry/exit, this results in a constant curvature path along the needle radius. For #2, orthogonality is enforced only at entry pose, and γ_t is set to a monotonically decreasing sequence in t. This results in rotations about the needle tip that achieve an asymmetric trajectory satisfying pose constraints at entry. We also demonstrate a case with no pose constraints in #3, resulting in the shortest path trajectory, but with oblique entry angles.

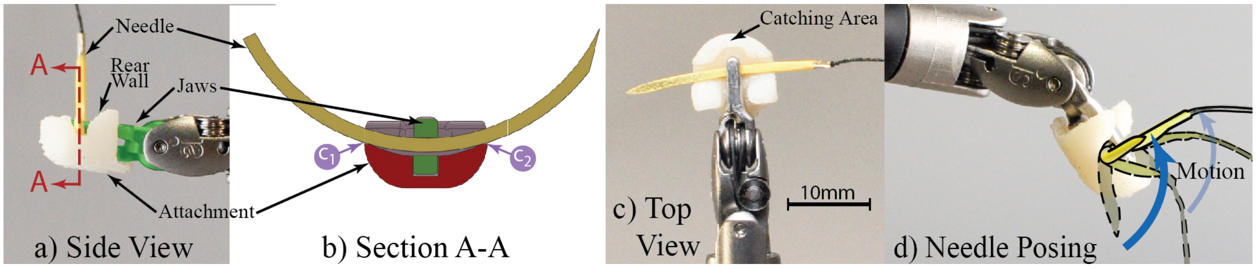
5.3.2 Reducing Needle Pose Uncertainty

As stated in Section 5.2 and Figure 5.2, tissue damage is minimized with orthogonal needle entry and motions that are tangential to the needle tip. These guidelines require accurate needle pose estimates at the needle entry point and robust needle grasps.

Suture Needle Angular Positioner (SNAP)

Commercially available RMIS needle drivers allow handling of a variety of needle sizes, however, an analysis of suturing trials in JIGSAWS dataset [20] reveals that multiple pairs of hand-offs are required for correct needle orientation. This is because the motion of a needle held within the needle driver jaws is not fully constrained. The flat gripper surface allows rotation and translation along the length of the needle, which can be hard to control without haptic or visual feedback.

There have been some commercial efforts to mitigate back-and-forth hand-offs and uncertainty in laparoscopic surgery through passively orienting the needle on gripper closure using a "self-



a) Side View b) Section A-A c) Top View d) Needle Posing

Figure 5.5: This figure illustrates the design and function of the 3D-printed Suture Needle Angular Positioner (SNAP). Figures (a) and (b) show a convex depression in which needle rests upon gripper closure. Figure (d) shows a time-lapse figure of the gripper closing action on needle orientation.

righting” gripper jaw design [126, 127]. However, these are not designed for automation and require a complete tool redesign.

We develop a design for a low-cost Suture Needle Angular Positioner (SNAP) for dVRK Classic 8 mm Needle Driver with 6 mm jaws, which works to guide and passively orient a curved needle into a stable pose upon closure of gripper jaws as illustrated in Figure 5.5(d). SNAP reduces needle pose uncertainty along two rotational axes as shown in Section 5.5. This allows for higher tolerance in relative positioning during needle hand-off, and which relaxes the accuracy requirements of needle tracking.

Mode of Operation: SNAP is mounted axially on one of the needle driver jaws. It is designed to guide the needle towards a groove running perpendicular to the length of the gripper jaws Figure 5.5 (b), (c). Upon closing the jaws, the needle rolls to a stable pose, passing through contact points C_1 and C_2 as shown in the section view in Figure 5.5(b).

The size of the needle gripper is parametrized by the distance between contact points C_1 and C_2 which is dependent on the curvature of the needle, that is a needle with a larger radius needs a wider contact grasp to enable the needle rolling upon jaw closure. As illustrated in Figure 5.5 (a), SNAP has a rear-wall that allows the gripper to overshoot during the pre-grasp approach. It also has a needle *catching area* in the front (Figure 5.5 (c)) that guides the needle into the groove, compensating for undershoot during pre-grasp. Both of the above features increase the robustness of needle manipulation.

The SNAP is fabricated from ABS plastic using a *Stratasys uPrint* 3D printer. For an 8 mm classic needle driver, using a $\frac{3}{8}$ circumference, 39 mm length needle, we designed the SNAP with $C_1 - C_2$ span of 10 mm. Through experimental evaluation, we improved upon the SNAP design to include a larger rear wall. This enabled a wider jaw opening during approach allowing for larger tolerance in needle pose uncertainty.

Real Time Needle Tracking

We have developed a real-time needle tracking system to provide closed loop feedback during the suturing process as summarized in Figure 5.6. Due to tissue and tool specularity, perception using RGB-D sensing is not feasible. Our system provides 3D needle pose estimates using a custom built stereo camera pair, composed of two Prosilica GigE GC1290C cameras with 6 mm focal

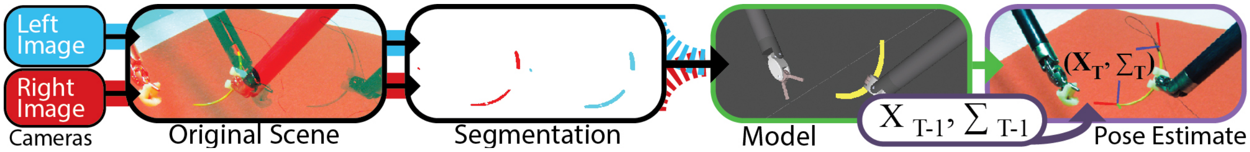


Figure 5.6: This figure shows an overview of the needle tracking pipeline, from stereo images to the final needle pose estimate overlaid onto the original scene. We fuse a Kalman Filter estimate with current camera estimate to compute the final estimate. The tracking system is robust to outliers and missing data in the segmentation masks.

length lenses. The needle tracking algorithm is implemented as a ROS node that publishes real-time estimates of the needle's pose. The tracking system works with partial occlusion for instance when the needle is inside the tissue or behind the robot arms.

We use a model-based tracking system leveraging the needle shape and color. The first step in the process is *Needle Segmentation*. We paint the needle in yellow color to assist in foreground and background separation. And finally, an HSV (Hue, Saturation, Value) separation is used to identify the needle in a cluttered environment with the open-source OpenCV library and create a set of image plane points $\mathcal{P}_I$.

We leverage the circular shape of the surgical needles and their elliptical projection. We create a small set of parametrically sampled points along the length of needle model $\mathcal{P}_M$, $|\mathcal{P}_M| = 12$, and then use affine point set registration to fit the $\mathcal{P}_I$ to $\mathcal{P}_M$. We model the non-linear registration problem as point set matching. This creates robustness to outliers, missing data due to occlusions, and noisy data from incorrect segmentation masks. We use the Matlab library CPD2 for solving the registration problem [81].

Using the ellipse fits on the image pair, we generate a dense set of corresponding points along the needle. This creates a robust disparity map of 3D points on the needle. A plane is then fit to the 3D points, providing a normal vector, while an average tangential direction is calculated using the three points on the end of the needle. Using the end point of the needle and these two vectors, a pose $p_n \in \mathbb{R}^6$ is generated. We use a Kalman filter to smooth needle tip pose estimates.

The use of industrial Prosilica cameras with a wide baseline necessitated the use of a large workspace and consequently larger than average needles in order to enable robust needle tracking. Laparoscopic cameras have a smaller baseline and smaller field of view compared to our setup. The proposed tracking system should be transferable to a laparoscopic setup allowing the use of much smaller needles.

5.4 Multi-Throw Suturing: System Integration

We present a closed loop Finite State Machine (FSM) for multi-throw suturing with needle orientation tracking and multilateral needle hand-off as illustrated in Figure 5.7. Given the registration of the tissue phantom in the camera frame, a multi-throw suture plan is generated. The SPP algorithm is used to generate needle trajectories and a suggested needle curvature. Each throw in the task consists of the following sequence of sub-tasks which were segmented on the basis of manual

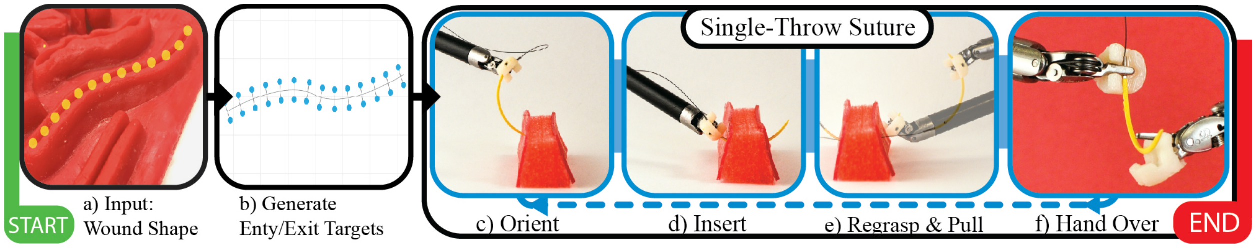


Figure 5.7: The figure outlines the Multi-Throw Suturing Finite State Machine. First, the surgeon specifies a suture path with wound width & depth and suture pitch. The system then computes the number of suture throws required, and generates entry & exit points, and optimized trajectories along with required needle size for each throw of the MTS. Each of the steps S1-S5 (see Figure 5.1) are repeated with visual feedback for each suture throw until all suture throws are completed.

surgeme labels for suturing in the JIGSAWS dataset:

S1. Needle Orientation: The system generates pose estimates for both the front tip of the needle, N_T, and the tail connected to the suture thread, N_S. Starting with the needle held in the right gripper at N_S, the system creates an initial pose estimate. Using this estimate, the robot aligns the needle with the camera's image plane, allowing for an occlusion-free view of the needle and an improved pose estimate.

S2. Needle Insertion: The system executes a trajectory for N_T using the planner described in the previous section. We note that at this point, suture path can be re-planned after every user-specified rolling time horizon.

S3. Needle Grasp: After the right arm guides the needle through tissue, the left arm grasps the needle at N_T and pulls the needle tangentially to the needle tip, rotating around the center of curvature of the needle in order to minimize tissue trauma.

S4. Needle Pull: Once the needle is completely outside the tissue, it is pulled away sufficiently to tighten the suture. The system estimates how much slack is available in the suture thread by modeling the length of thread between consecutive entry points as a helical loop with a radius equal to the radius of the needle and pitch equal to the suture pitch.This provides a conservative estimate of how much slack is lost in each throw and the system uses it to decrease the distance the needle is pulled away after each throw.

S5. Needle Hand-Off: Our needle tracking algorithm estimates the pose of the needle end N_S while it is grasped at N_T. Similar to step (S1), the left arm aligns the needle with the image plane to improve the needle pose estimate. This estimate is used to align the needle with the right arm in order to grasp the needle at N_S and perform the next suture throw.

Due to inherent pose errors in camera-robot registration and robot kinematics, the hand-off process is performed by simultaneously engaging the right arm at N_S while disengaging the left arm at N_T. A slight error in coordination will result in failed transfer due to stresses generated on the needle. The use of SNAP on both gripper ends facilitates this process because the grooves provide a space resulting in a partial cage instead of force closure during the hand-off as described in Section 5.3.2.

5.5 Physical Experiments with DVRK

dVRK: Hardware and Software

We use the Intuitive Surgical da Vinci Research Kit (dVRK) surgical robot assistant as in [117], along with open-source electronics and software developed by WPI and Johns Hopkins University [130]. We use a pair of 8mm Needle Drivers with each gripper having one Suture Needle Angular Positioner (SNAP). The software system is integrated with ROS and allows direct robot pose space control, working in Cartesian space instead of commanding motor torques.

Experimental Evaluation of Needle Tracking

The size and shape of needles make it difficult to obtain ground truth pose estimates using techniques like fiducial-based motion capture. Instead, we designed an experiment to indirectly verify the efficacy of our needle tracking system. The robot holds the needle rigidly in its gripper and moves the needle to random positions in the workspace. Note that the relative pose of the needle with respect to the gripper position never changes. At each random position, the robot pauses and uses the needle tracking system to compute the needle's relative pose with respect to the gripper pose (estimated from kinematics). Poses at 20 different random locations were recorded. Table I shows the standard deviation in the x,y,z (in mm) and in the roll, pitch, and yaw (in degrees) respectively in the needle's relative pose. The low error in every dimension suggests that our estimates of the needle's relative pose are nearly identical at each random location. This matches with the ground truth that the needle's relative pose never changes. The errors reported are not due to the needle tracker alone, but the composite error produced from needle tracking, camera-robot registration, and robot kinematics. However, the errors provide an upper bound on the needle tracking error and are representative of error that our system tolerates.

Table 5.1: Error in Relative Needle Pose (Over 20 Trials)

	Position (mm)			Orientation (degrees)		
	x	y	z	Yaw	Pitch	Roll
Std. Dev	2.182	1.23	1.54	2.495	4.699	4.329

Evaluation of Suture Needle Angular Positioner (SNAP)

1. Stationary Needle Pick up: In this experiment, we evaluate the SNAP's ability to reduce variation in needle grasp pose. This variation is the result of small natural perturbations in the needle starting pose and noise in the robot's kinematic chain. In each trial, a needle is placed in the same location and the robot is provided a constant known grasp pose to initiate pick up. Once the needle is grasped, the robot brings the needle to a known location and the needle's pose is recorded using our needle tracker. We repeat this process over ten trials both with and without SNAP. The standard deviations in each degree of freedom of the needle's pose are presented in Table 5.2. The SNAP reduced needle pose variation in both position and orientation, in some cases by over one order of magnitude.

Table 5.2: SNAP Evaluation: The following table lists the performance improvement in needle pose estimate through the use of Suture Needle Angular Positioner. We present numerical results for both Stationary and Perturbed grasp evaluations, and observe at least a 3x improvement in Needle pose estimate

Stationary Grasp Orientation	**Error (Standard Deviation)**						
	Successful Grasps	x (mm)	y (mm)	z (mm)	yaw (deg)	pitch (deg)	roll (deg)
Without SNAP	100%	2.511	1.434	4.838	20.547	7.584	6.472
With SNAP	100%	0.199	0.158	0.177	0.926	1.094	0.664
Perturbed Grasp Orientation	**Error (Standard Deviation)**						
	Successful Grasps	x (mm)	y (mm)	z (mm)	yaw (deg)	pitch (deg)	roll (deg)
Without SNAP	100%	2.01	2.59	5.95	15.54	12.74	7.62
With SNAP	91.66%	1.58	1.15	1.19	5.55	3.97	6.34

2. *Perturbed Needle Pick up:* In the second experiment, we intentionally perturb the orientation of the robot's grasp pose to evaluate robustness to uncertainty and variation in grasp orientation. Experiment 2 is a variation of experiment 1 where the commanded grasp pose is perturbed from -30 degrees to 30 degrees in yaw, pitch, and roll. The perturbations are applied in increments of 10 degrees independently in each axis resulting in 19 trials total. Our results show that the use of SNAP results in a 3x reduction in needle pose uncertainty over the standard Needle Driver.

Robot Experiments: Four-Throw Suturing Task

We used a suturing phantom made with foam to mimic subcutaneous fat tissue with a layer of 1mm thick skin using (shore hardness 2A) *DragonSkin 10 Medium* Silicone Rubber (*Smooth-On*). The soft tissue phantom deforms during needle insertion to introduce uncertainty. The mechanical design of the dVRK robotic arms ensures that the arms do not move at the point where they would enter a human body ensuring that the kinematic motions of our system remain feasible in-vivo in a minimally invasive surgical (MIS) setting. Due to the wide baseline of our stereo cameras, the size of our phantom, needles, and workspace were constrained to be larger than those found in a nominal MIS setting.

In this experiment, the system tries to complete a closed loop four throw suturing task similar to the suturing task found in the JIGSAWS data-set [20]. We initialized the system with entry and exit poses on opposite surfaces of the tissue phantom and with a desired suture depth. Our system generates insertion trajectories and based on the output optimal needle curvature we selected a 39 mm long, 3/8 reverse cutting needle to perform the suturing throws. For each trial we record time to completion as well as the failure mode if necessary. The robot moves at a top speed of 3cm/s. The results of each trial are found in Tables 5.3 and 5.4.

A video of the procedure is available at: `https://youtu.be/z1ehShXFToc`

Table 5.3: Results for Four-Throw Suturing. 14 trials were performed, with a 50% success rate. For failed states, “N.I" represents incorrect needle orientation or insertion, “G.P." represents incorrect needle re-grasp and pull after insertion, and “H.O" represents a failure in needle hand-off respectively. The test setup was varied with translation of simulated wound along the wound axis.

Trial	4-Throw Success	Number of Throws Completed (Attempted)	Failure Mode	Translation in X	Suture Pitch	Total Time(s)
1	Failure	1 (2)	G.P.	-3mm	3mm	-
2	Failure	2 (3)	G.P.	-3mm	3mm	-
3	Failure	3 (4)	G.P.	-2mm	3mm	-
4	Success	4 (4)		-1mm	3mm	387
5	Success	4 (4)		0mm	3mm	380
6	Success	4 (4)		0mm	3mm	380
7	Success	4 (4)		0mm	3mm	383
8	Failure	2 (3)	H.O.	1mm	3mm	-
9	Failure	2 (3)	N.I.	1mm	3mm	-
10	Failure	3 (4)	G.P.	2mm	3mm	-
11	Success	4 (4)		3mm	3mm	393
12	Success	4 (4)		4mm	3mm	383
13	Success	4 (4)		5mm	3mm	382
14	Failure	3 (4)	G.P.	6mm	3mm	-
Mean	50%	3.14				384
Std Dev		1.027		Single Throw Success Rate: 86.3%		

Table 5.4: This table compares the performance of our autonomous suturing system with different skill levels of surgeons in the JIGSAWS dataset[20]

Operator Mode	Average Time for 1-Throw (s)	Average time for 4-throw Task (s)
Expert	19.03	87.02
Intermediate	18.57	87.89
Novice	32.14	136.85
Autonomous (Our Approach)	112.33	383.00

5.6 Discussion

The methods and experiments in this chapter affirm that the system presented can computationally plan and execute the multi-throw suturing task with four throws in closed-loop operation. The combination of this needle tracking system and the SNAP enables our system to minimize and be robust to needle pose uncertainty. This allows our system to perform multilateral needle hand-off, enabling the execution of multi-throw suturing.

Limitations However, we note that the system completes on average 3.14 of the intended 4 throws, with a 50% completion rate for the four-throw task. It is worth noting that 5 out of the 7 failures were due to incorrect needle re-grasp and pulling after the insertion step. Some of these failures were due to incorrect needle estimate after the needle exits the tissue in unexpected locations. The visual needle tracker could not recognize the needle due to large occlusions. Additional failures were due to the entanglement of the suture thread during the needle pulling.

The slow speed of the task execution is partly because of the larger workspace as compared to the setup in JIGSAWS data [20]. Furthermore, moving to align the needle in camera for improving needle pose also contributes to the delay. We will work to improve the real-time visual estimate of the needle pose without the need for explicit alignment in front of the camera. Future work will focus on improving needle pose estimation with significant occlusions along with receding horizon re-planning during the needle insertion to reduce error in needle re-grasp. We will also evaluate the use of swept needle volume as objective cost and explore augmenting the needle state with needle pose belief for uncertainty compensation through optimization re-planning.

Chapter Summary This chapter presents initial results toward automating MTS with a combination of new hardware and a novel optimization algorithm. The chapter describes the mechanical device, the Suture Needle Angular Positioner (SNAP), designed to align and hold the needle in a known orientation, and an SCP formulation of needle motion planning. The system completes 86.3% of individual suture throws attempted at approx. 30% of the average speed of manually teleoperated demonstrations as listed in Tables 5.3 and 5.4. Our results also show that the proposed needle tracking system can provide robust estimates of needle pose in near real-time with an empirical error of up to 5 degrees. Furthermore, the use of SNAP improves repeatability in needle grasping by 10× and grasping is robust to up to 30 degrees error in needle estimate.

Chapter 6

Transition State Clustering: Unsupervised Surgical Trajectory Segmentation For Robot Learning

Overview

The adoption of robot-assisted minimally invasive surgery (RMIS) is generating datasets of kinematic and video recordings of surgical procedures. This data can facilitate robot learning from demonstrations [117], surgical training and assessment [20, 131], and automation [116, 132]. Segmenting this demonstration data into meaningful sub-trajectories can benefit learning since individual segments are often less complex, have lower variance, and it is easier to remove outliers.

In the previous chapter on automation of multi-throw suturing task, we presented a method for constructing a finite state machine with explicit error handling designed in the state machine. This approach is similar to the approach adopted our attempts for automation in pattern cutting [117] and tumor resection [133]. While an FSM is a reliable method to characterize behaviors that are active in particular regimes, scaling this approach to more complex tasks with unmodeled dynamics is a challenge. In this chapter, we address this limitation through performing unsupervised trajectory segmentation from trajectory demonstrations to recover task-structure which can then be used to construct more robust FSMs.

However, even in a consistent data collection environment, such as teleoperation on identical tissue phantoms, surgical trajectories can vary significantly both spatially and temporally. These trajectories are further corrupted by random noise, spurious motions, and looping actions where a surgeon repeatedly retries a motion until success. The primary challenge in surgical trajectory segmentation is to identify consistent segments across a dataset of demonstrations of the same procedure in the presence of such disturbances.

Trajectory Segmentation algorithms fall into two broad categories: (1) supervised approaches that learn from manual annotations or match sub-sequences to pre-defined dictionaries of primitives [115, 134–136], and (2) unsupervised approaches that infer the latent parameters of some

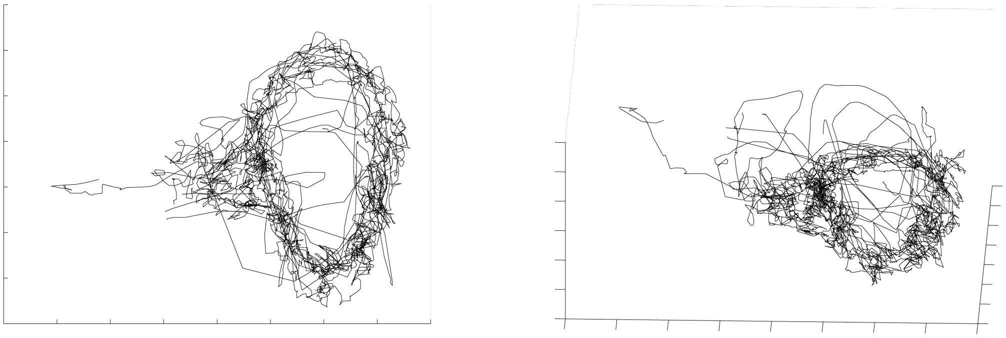


Figure 6.1: We plot 10 trajectories of the end-effector (x,y,z) positions on an identical circle cutting task on the dVRK. This plot illustrates the variability of demonstrations even when the task is identical. The goal of TSC is to identify consistent transition structure in the demonstrations, and our experiments in Section 6.5.5 illustrate the results.

underlying generative process [137–140]. *Consistency* and *supervisory burden* are a key concern in supervised segmentation as it is often difficult to precisely characterize what defines a segment and labeling can be time-consuming. Similarly, it may be unclear how to specify a dictionary of primitives at the correct level of abstraction. To the best of our knowledge, prior work in the surgical setting has been supervised, and we draw from several studies of unsupervised segmentation in non-surgical settings to develop a new unsupervised approach [137–140]. Unsupervised approaches largely apply clustering or local regression models to identify locally similar states.

Transition State Clustering (TSC) is a novel algorithm that builds on these ideas with a two-phase algorithm that first identifies *transitions states*, defined as consecutive time-steps assigned to different segments, and then, *clusters* spatially and temporally similar transition states across demonstrations with a non-parametric mixture model. This chapter focuses on segmenting trajectories derived from kinematic and video recordings of surgical robots in teleoperation. While the TSC algorithm can in principle be relevant to problems in other domains, we restrict the scope of this chapter to robotic tasks.

Consider two trajectories $\mathbf{x}_1$ and $\mathbf{x}_2$. $\mathbf{x}_1$ and $\mathbf{x}_2$ may represent two different linear paths to the same needle insertion transition. That needle insertion is a invariant across $\mathbf{x}_1$ and $\mathbf{x}_2$. TSC attempts to infer such invariants using spatial and temporal clusters of transitions detected by changes in motion. *The crucial insight is that the sequence of transition events often has a consistent partial order across demonstrations–even if the motion between those transition events is very different.* This model can also be coupled with a series of merging and pruning steps to ensure that only the most important transition state clusters are retained.

Contributions We present a new unsupervised segmentation algorithm called Transition State Clustering (TSC), which given a set of discrete-time trajectories identifies a common transition structure to segment the trajectories. We treat each demonstration trajectory as generated from a locally linear dynamical system with i.i.d process noise $\mathbf{w}_t$:

$$\mathbf{x}_{t+1} = A_t\mathbf{x}_t + \mathbf{w}_t : A_t \in \{A_1, ..., A_m\} \tag{6.1}$$

A transition states is a state at which the dynamics matrix changes at the next time-step $A_t \neq A_{t+1}$. We model the set of all transition states as sampled from a probability density generated from a hierarchical nonparametric Bayesian model, where the number of regions is determined by a Dirichlet Process. A series of merging and pruning steps remove outliers.

We evaluate TSC on real surgical data from the JIGSAWS surgical training dataset consisting of joint-space trajectories and video from a fixed camera [20], and a synthetic example consisting of randomly generated motions of a point robot with variable levels of noise. On the synthetic examples, we evaluate the ability of five alternative algorithms, Gaussian Mixture Model, Gaussian Hidden Markov Model, Coresets, Gaussian Hidden Semi-Markov Model, and an Autoregressive Hidden Markov Model, to return a segmentation that has a strong one-to-one correspondence with a known ground truth. Our experimental results suggest that TSC recovers this ground truth with greater accuracy than the alternatives, especially under low-frequency process noise. For the surgical experiments, TSC extends to state-spaces the include derived features from computer vision which are manually derived in this work. We present the TSC model and presents experiments in which we manually label the video stream with two features: a binary variable identifying object grasp events and a scalar variable indicating surface penetration depth. On 67 kinematic and video of the surgical needle passing and suturing tasks from the JIGSAWS surgical training dataset [20] TSC finds 83% of the needle passing transitions and 73% of the suturing transitions found by human experts.

6.1 Background and Related Work

TSC exploits the structure of repeated demonstrations by first identifying transitions and then correlating them spatially and temporally across different demonstrations.

6.1.1 Segmentation in Surgical Robotics

To the best of our knowledge, prior work in surgical robotics has only considered supervised segmentation using either segmented examples or a pre-defined dictionary called surgemes. For example, given manually segmented videos, [141] use features from both the videos and kinematic data to classify surgical motions. Similarly, [142] use manually segmented examples as training for segmentation and recognition of surgical tasks based on archived cataract surgery videos. Several studies use the surgemes to bootstrap learning segmentation, however, this involved time-consuming the process of identifying surgemes in existing data sources for use as training and testing data [115, 134–136]).

6.1.2 Unsupervised Segmentation Models

Unlike supervised segmentation techniques, unsupervised techniques do not use labels or dictionaries. As such, a key differentiating factor is the underlying probabilistic model for segmentation.

Consider a continuous time vector-valued trajectory, which is a sequence of T vectors $\mathbf{x}_t$ in some vector-space $\mathbb{R}^p$.

Gaussian Mixture Models: Many unsupervised segmentation techniques are based on Gaussian Mixture models (GMM). GMMs are particularly intuitive, as shown by Ghahramani and Jordan [143] that GMMs are a form of local linear regression; linearizing around the mixture means.

Lee et al. [138] identify segmentation points in a trajectory by fitting a GMM directly to the data. That is, the sequence $[\mathbf{x}_1, ..., \mathbf{x}_T]$ is modeled as a sample from a GMM, and they assign each $\mathbf{x}_t$ to the most likely mixture component. They tune the number of mixture components using the Bayesian Information Criterion and apply PCA dimensionality reduction before applying the GMM for tractability. By treating the sequence as a sample from a GMM, this approach does not consider dynamics and correlation between demonstrations.

One solution is to draw from the dynamical systems literature [144, 145] and model the trajectory $\mathbf{x}_t$ as a noisy autonomous dynamical system:

$$\dot{\mathbf{x}} = \xi(\mathbf{x}) + \mathbf{w}$$

It can be shown to linearize this system in the same way as in Lee et al. [138], but instead of applying the GMM to the set of states $\mathbf{x}_t$, we have to apply a GMM to samples of $\binom{x}{\dot{x}}$ or $\binom{\mathbf{x}_t}{\mathbf{x}_{t+1}}$ in discrete-time. This approach is used by Krüger et al. [139], and like TSC they also use a Dirichlet process prior to learning locally linear dynamics. Transition State Clustering builds on these GMM approaches with a two-phase algorithm that first identifies transitions states in a way similar to [139] and then, clusters spatially and temporally similar transition states across demonstrations with a non-parametric mixture model under the assumption that each demonstration follows the same partial order of transitions up to noise and loops.

Hidden Markov Models: One approach to deal with spatial and temporal variation is to model the high-level progression of a task as a finite-state Markov Chain [146–149], for example, primitive A progresses to B with probability 0.75 and to C with probability 0.25. Such a model is a class of Hidden Markov Models since this Markov chain is not directly observed. Given the current state of this Markov chain, the system will either ”emit” different states (e.g., Gaussian HMM) or have different dynamics (e.g., Autoregressive HMM). This logic has been extended to more complex transition dynamics such as the Hidden Semi-Markov Model, which additionally models the amount of time spent in a given state [150]. There is also the related ”workflow” HMM proposed by [151]. The HMM, HSMM, and their variants impose a probabilistic grammar on transitions, and the inference algorithm estimates the transition probabilities from data. Accordingly, they can be sensitive to hyperparameters such as the number of segments, the amount of data, and noise [152]. The problem of robustness in GMM+HMM (or closely related variants) has been addressed using down-weighting transient states [153] and sparsification [154, 155].

There have been several Bayesian extensions to these models, which model the time-series as a stochastic process and learn the parameters with MCMC or Stochastic Variational Inference. [156] proposed Beta Process-Autoregressive-Hidden Markov Model, which was applied by [140] in robotics. This model is fits an autoregressive model to time-series, where $\mathbf{x}_{t+1}$ is a linear function of states $\mathbf{x}_{t-k}, \ldots, \mathbf{x}_t$. The linear function switches according to an HMM with states parametrized by a Beta-Bernoulli model (i.e., Beta Process). While HMMs and GMMs also draw from the Bayesian literature, they differ from recently proposed Bayesian segmentation models as they are typically solved more efficiently with analytic expectation maximization algorithms. Similarly, TSC is motivated by Bayesian non-parametrics, there are several features of the algorithm that are motivated by frequentist statistics (e.g., outlier rejection using merging and pruning).

We evaluate against these approaches (GMM+HMM, ARHMM, and HSMM) in our experiments and find that TSC is more robust to un-modeled noise. The intuition is that Hidden Markov methods make the implicit assumption that ”the low-level dynamics within a segment are more structured and predictable than the higher-level dynamics that govern transitions between segments” [157]. In contrast, TSC explores the converse, suppose the low-level dynamics are uncertain and noisy, but the high-level dynamics follows a consistent partial order of events across demonstrations and these events are spatially and temporally correlated.

Coresets: A coreset is defined as a query-dependent compression of a dataset D, such that running the query q results in a provably approximate result. This idea can be used to devise a locally linear segmentation technique [158–160]. The query q is the solution to the k-line segment problem, which fits a k-line segment to a trajectory. The idea is to find a compressed dataset such that the lines can be accurately reconstructed. The main benefit is that this results in segmentation with provable properties, such as sample complexity and convergence. However, the models used in prior work do not consider switched linear dynamics, non-parametrics when choosing k, or robustness to loops.

6.2 Problem Statement

This section describes the problem setting, assumptions, and notation. The objective is to segment each of the given demonstration trajectories based on transition states recovered from the switched-linear dynamics model.

6.2.1 Demonstrations

Let $D = \{d_i\}$ be a set of demonstrations. Each demonstration d_i is a discrete-time sequence of T_i state vectors in a state-space $\mathbb{R}^p$. Associated with D is a set of k transitions which are informally defined as state-space and temporal conditions that trigger a change in motion, in the following section, we will precisely characterize this definition. Thus, each demonstration d_i can be represented as a sequence of labeled transitions $\{1,...,k\}$, e.g., $d_1 = [S_1, S_2, S_4]$, $d_2 = [S_3, S_1, S_4]$. The goal of Transition State Clustering is to learn the sequence of transitions that consistently occur across all demonstrations e.g., $[S_1, S_4]$, and associate them with states in a trajectory.

6.2.2 Regularity

Without regularity assumptions on the demonstrations, there may not be a meaningful common structure. For example, we could observe:

$$d_1 = [S_1, S_3, S_5],\ d_2 = [S_2, S_4, S_6],$$

where there does not exist any overlap between d_1 and d_2. Therefore, we assume, the set of demonstrations is *regular*, meaning there exists a non-empty sequence of transition U^* such that the partial order defined by the elements in the sequence (i.e., S_1 happens before S_2 and S_3) is satisfied by every U_i. For example,

$$U_1 = [S_1, S_3, S_4],\ U_2 = [S_1, S_1, S_2, S_4],\ U^* = [S_1, S_4]$$

An example of an irregular demonstration set is

$$U_1 = [S_1, S_3, S_4],\ U_2 = [S_2, S_5],\ U^* \text{ no solution}$$

Intuitively, this condition states that there have to be a consistent ordering of transitions over all demonstrations up to some additional transitions (e.g., spurious actions). We will show that we can extend this condition such that only a fraction ρ of the demonstrations need to be regular; thereby *pruning* the inconsistent transitions.

6.2.3 Looping

Loops, or repetitions of an action until the desired outcome, are common in surgical demonstrations. For example, a surgeon may attempt to insert a needle 2-3 times. For example, let us assume that a surgeon is attempting to insert a needle and fails to do so 2 times. If in all other demonstrations, there is only a single transition representing needle insertion (i.e., transition ”1”), we might detect multiple transitions S_1' and $S_1^\dagger$ demonstration with loops. Ideally, we would like to compact these repeated motions into a single transition:

$$U_1 = [S_1, S_3, S_4],\ U_2 = [S_1, S_1', S_1^\dagger, S_3, S_4],\ U^* = [S_1, S_3, S_4]$$

We assume that these loops are modeled as repeated transitions, which is justified in our experimental datasets. This assumption may seem at odds with our argument that surgical demonstrations are highly variable. We find that while the motions between transitions are variable and noisy, up-to loops and extra transitions, the high-level sequence of transitions is relatively consistent. In the future, we hope to explore more complex models for failure and retrial, and we believe that variants of our approach can be applied in conjunction with Hidden Markov Models.

6.2.4 Problem Formalization

Other than the regularity assumptions above and the implicit assumptions about local linearity discussed in the next section, we make no assumptions about the nature of the trajectories given to

TSC. In this chapter, we focus on segmentation for surgical robotics applications such as subtask automation on the da Vinci surgical robot. However, TSC is broadly applicable to other robotics domains and data collected from other dynamical systems.

In our experiments, we define the state-space to be the 6-DOF orientation of the robot end-effector. However, it is worth noting the kinematic state of the robot may not be sufficient to describe the system or its interaction with the environment. We additionally provide TSC with features constructed from the video. Suppose at every time t, there is a feature vector v_t composed of discrete and continuous features. Then, we define an augmented state of both the robot kinematic state and the features denoted is:

$$\mathbf{x}_t = \begin{pmatrix} c_t \\ v_t \end{pmatrix}$$

Consider the following features:

1. *Grasp*. 0 if empty, 1 otherwise.
2. *Needle Penetration*. We use an estimate of the penetration depth based on the robot kinematics to encode this feature. If there is no penetration (as detected by video), the value is 0, otherwise, the value of penetration is the robot's z position.

Our goal with these features was to illustrate that TSC applies to general state-spaces as well as spatial ones, and not to address the perception problem. These features were constructed via manual annotation, where the Grasp and Needle Penetration were identified by reviewing the videos and marking the frames at which they occurred. We completely characterize TSC without such features on a synthetic dataset against alternatives, and for the surgical data present results with and without these features.

Problem 1 (Transition State Clustering) *Given a set of regular demonstrations D, partition each* $d_i \in D$ *into a sequence of sub-trajectories defined by transitions* $[S_1^{(i)},\ldots S_k^{(i)}]$. *Each transition should correspond to exactly one other transition in at least a fraction of* ρ *demonstrations.*

6.3 A Probabilistic Model For Transitions

In this section, we formalize the definition of transitions and transition states.

6.3.1 Demonstrations as Dynamical Systems

Each d_i is a trajectory $[\mathbf{x}_1,\ldots,\mathbf{x}_T]$. We model each demonstration as a realization of a noisy dynamical system governed by the dynamics ξ and i.i.d Gaussian white noise:

$$\mathbf{x}_{t+1} = \xi(\mathbf{x}_t) + w_t \tag{6.2}$$

We assume that ξ is locally-linear and can be modeled as switched linear dynamical system. That is, there exists m $d \times d$ matrices $\{A^{(1)},\ldots,A^{(m)}\}$:

$$\mathbf{x}_{t+1} = A_t\mathbf{x}_t + \mathbf{w}_t : A_t \in \{A^{(1)},\ldots,A^{(m)}\} \tag{6.3}$$

A *transition states* is defined as a state at which the dynamics matrix changes at the next time-step $A_t \neq A_{t+1}$.

6.3.2 Transition State Distributions

Over all of the demonstrations D, there is a corresponding set Γ of all transition states. We model the set Γ as samples from an underlying parameterized distribution over the state space $x \in \mathbf{R}^p$ and time $t \in \mathbb{R}_+$.

$$\Gamma \sim f_\theta(x,t)$$

As the name suggests, Transition State Clustering fits mixture models to f_θ, and this has the interpretation of correlating transition events spatially and temporally. Depending on how we choose to define this joint distribution, we can model different phenomena. We use different hierarchies of Gaussian Mixture Models.

Time-Invariant Transition State Model: The most straight-forward approach is to consider an f that is independent of time. This means that:

$$\forall t,t' \in [0,T] : f_\theta(x,t) = f_\theta(x,t')$$

Then, we can model the distribution as a GMM over just the state-space:

$$f_\theta(x) = GMM(\pi, \{\mu_1,...,\mu_k\}, \{\Sigma_1,...,\Sigma_k\})$$

However, this model cannot handle trajectories that cross over the same state multiple times, e.g., a figure 8 trajectory.

Time-Varying Transition State Model: We can extend the above model to consider time-varying distributions. We do this by splitting the distribution into a product of two components, one that is time-invariant and one that depends on time conditioned on the current state. This is a natural consequence of the chain rule where we can decompose $f_\theta(x,t)$ into two independently parametrized densities p,q:

$$f_\theta(x,t) = p_{\theta_p}(x) \cdot q_{\theta_q}(t \mid x)$$

If we model p as a GMM, then for every x will be drawn from one of the $\{1,...,k\}$ mixture components. We then make a simplifying assumption that this mixture component is a sufficient statistic for q_{θ_q}. Let $z \in \{1,...,k\}$ be this mixture component, then, we can apply the time-invariant model from above for p, and we can apply a separate GMM for q conditioned on each possible z:

$$q_{\theta_q}(t \mid z) = GMM(\lambda, \{\mu_1,...,\mu_{l_z}\}, \{\sigma_1,...,\sigma_{l_z}\})$$

In other words, a GMM models the spatial transition states distribution, and within each Gaussian, the states are further drawn from a GMM over time. The resulting mixture model for f has $\sum_{i=1}^{k} l_i$ components.

Multi-modal Transition State Model: The same logic can be used to model multiple sensing modalities (e.g., kinematics, vision). Let $\binom{x}{v}$ be a state-space constructed of kinematics and visual features $\mathbf{x}$ and $\mathbf{v}$ respectively. Consider the following decomposition p, q, r:

$$f_\theta(\mathbf{x}, t) = p_{\theta_p}(x) \cdot q_{\theta_q}(v \mid x) \cdot r_{\theta_r}(t \mid x, v)$$

As in the time-varying case p, q, r are each modeled as GMMs conditioned on the the mixture component of $\mathbf{x}$ and $\mathbf{x}, \mathbf{v}$ respectively.

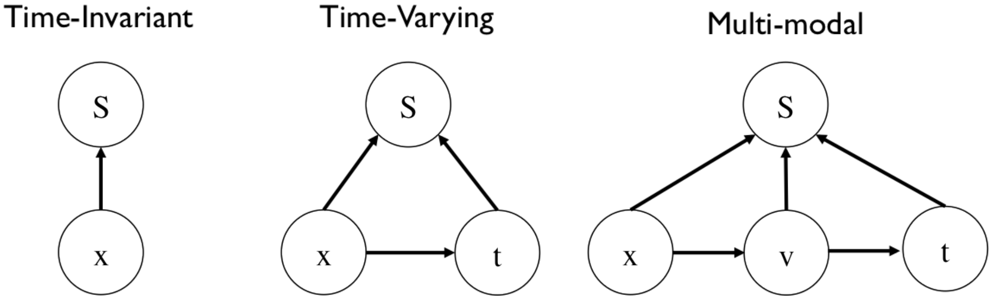


Figure 6.2: Transition State Clustering models the set of transition states S as a sample from a mixture model that depends on the state x, the time t, and other features v. Different probabilistic models can capture different phenomena

6.3.3 Feedback Model

The proposed model describes systems controlled with linear state feedback controllers to the centroids of the k targets $[\mu_1, ..., \mu_k]$. We can show that the Transition State Clustering model naturally follows from a sequence of stable linear full-state feedback controllers sequentially controlling the system to each μ_i (up to some tolerance defined by α).

Consider a single target μ_i. Suppose, we model the robot's trajectory in feature space as a linear dynamical system with a fixed dynamics. Let A_r model the robot's linear dynamics and B_r model the robot's control matrix:

$$\mathbf{x}_{t+1} = A_r \mathbf{x}_t + B_r \mathbf{u}_t + \mathbf{w}_t.$$

The robot applies a linear feedback controller with gain G_i, regulating around the target state μ_i. This can be represented as the following system (by setting $u(t) = -G_i \hat{\mathbf{x}}$):

$$\hat{\mathbf{x}}_t = \mathbf{x}_t - \mu_i.$$

$$\hat{\mathbf{x}}_{t+1} = (A_r - B_r C_i)\hat{\mathbf{x}}_t + \mathbf{w}_t.$$

If this system is stable, it will converge to the state $\hat{\mathbf{x}}_t = \mathbf{0}$ which is $\mathbf{x}_t = \mu_i$ as $t \to \infty$. However, since this is a finite time problem, we model a stopping condition, namely, the system is close enough to $\mathbf{0}$. For some z_α (e.g., in 1 dimension 95% quantiles are $Z_{5\%} = 1.96$):

$$\hat{\mathbf{x}}_t^T \Sigma_i^{-1} \hat{\mathbf{x}}_t \leq z_\alpha.$$

Algorithm 3: The Transition State Clustering Algorithm

```
1 Input: $D$ demonstrations, $\ell$ a window size, $\rho$ pruning parameter, $\delta$ compaction parameter,
    and $\alpha$ a Dirichlet Process concentration prior.
2 Define: $\mathbf{n}_t^{(\ell)} = [\mathbf{x}_{t-\ell}, ..., \mathbf{x}_t]^\intercal$
3 Fit a mixture model to $\mathbf{n}_t$ using DP-GMM assigning each state to its most likely component.
4 Transition states are when $\mathbf{n}_t$ has a different most likely mixture component than $\mathbf{n}_{t+1}$.
5 Fit a mixture model to the set of transition states in the state-space using DP-GMM.
6 Conditioned on each possible state-space mixture component, apply DP-GMM to the set of
    times.
7 Assign every time step to its most likely mixture component, prune mixture components that
    do not have at least 1 observation from a fraction $\rho$ demonstrations.
8 Merge together transitions that are within $\delta$ $L_2$-distance after Dynamic Time Warping.
9 Return: A set of transitions, which are regions of the state-space and temporal intervals
    defined by Gaussian sub-level sets.
```

If the robot's trajectory was modeled as a sequence $1...k$ of such controllers, we would observe the that the set of transition states Γ would be described as a mixture model around each of the targets $[\mu_1, ..., \mu_k]$. The GMM is a tractable mixture model that approximates this distribution.

6.4 Transition State Clustering Algorithm

In this section, we describe the hierarchical clustering process of TSC, which is a two-phase algorithm that first identifies *transition states*, defined as consecutive time-steps assigned to different segments, and then, *clusters* spatially and temporally similar transition states across demonstrations with a non-parametric mixture model under the assumption that each demonstration follows the same partial order of transitions up-to noise and loops. The algorithm is summarized in Algorithm 3.

6.4.1 Non-Parametric Mixture Models

Hyper-parameter selection is a known problem in mixture models. Recent results in Bayesian statistics can mitigate some of these problems by defining a soft prior of the number of mixtures. Consider the process of drawing samples from a Gaussian Mixture Models (GMM). We first sample some c from a categorical distribution, one that takes on values from (1...k), with probabilities ϕ, where ϕ is a K dimensional simplex:

$$c \sim cat(k, \phi)$$

Then, conditioned on the event $\{c = i\}$, we sample from a multivariate Gaussian distribution:

$$\mathbf{x}_i \sim N(\mu_i, \Sigma_i)$$

We can see that sampling a GMM is a two-stage process of first sampling from the categorical distribution and then conditioning on that sample.

The key insight of Bayesian non-parametrics is to add another level (or multiple levels) to this model. The Dirichlet Process (DP) defines a distribution over discrete distributions; in other words, a categorical distribution with certain probabilities and setting of k itself is a sample from a DP [161]. To sample from the Dirichlet Process-GMM model, one must first sample from the DP, then sample from the categorical distribution, and finally sample from the Gaussian:

$$(K,\phi) \sim DP(H,\alpha) \qquad c \sim cat(K,\phi) \qquad \mathbf{x} \sim N(\mu_i,\Sigma_i)$$

The parameters of this model can be solved with variational Expectation Maximization. We denote this entire clustering method in the remainder of this work as DP-GMM. DP-GMM is applied in multiple steps of the TSC algorithm including both transition identification and state clustering.

6.4.2 Transition States Identification

The first step is to identify a set of transition states for each demonstration in D. Suppose there was only one regime, then this would be a linear regression problem:

$$\arg\min_{A} \|AX_t - X_{t+1}\|$$

where $X_t = [\mathbf{x}_1,\ldots,\mathbf{x}_T] \in \mathbb{R}^{p\times T}$ with each column as the state at time t: $\mathbf{x}_t \in \mathbb{R}^p$. Generalizing to multiple regimes, [145] showed that fitting a jointly Gaussian model to $\mathbf{n}_t = \binom{\mathbf{x}_{t+1}}{\mathbf{x}_t}$ is equivalent to Bayesian Linear Regression, and a number of others have applied similar techniques [144, 148]. This general logic defines a family of estimators, where we can define $\mathbf{n}_t^{(\ell)}$ as:

$$\mathbf{n}_t^{(\ell)} = [\mathbf{x}_{t-\ell},\ldots,\mathbf{x}_t]^{\intercal}$$

In our experiments, unless otherwise noted, we use $\ell = 1$.

Therefore, to fit a switched linear dynamical system model, we can fit a DP-GMM model to $\mathbf{n}_t$. Each $\mathbf{n}_t$ is assigned to a most likely mixture component (i.e., cluster). To find transition states, we move along a trajectory from $t = 1,\ldots,t_f$, and find states at which $\mathbf{n}_t$ has a different most likely mixture component than n_{t+1}. These points mark transitions. The result is a set Γ of transition states across all demonstrations.

Linearization with GMMs Using a GMM (and by extension a DP-GMM) to detect switches in local linearity is an approximate algorithm that has been applied in several prior works [144, 145, 162].

Consider the following dynamical system:

$$\mathbf{x}_{t+1} = \xi(\mathbf{x}_t) + \mathbf{w}_t$$

where $\mathbf{w}_t$ is unit-variance i.i.d Gaussian noise $N(0,I)$. Let us first focus on linear systems. If ξ is linear, then the problem of learning ξ reduces to linear regression:

$$\arg\min_{A} \sum_{t=1}^{T-1} \|A\mathbf{x}_t - \mathbf{x}_{t+1}\|.$$

Alternatively, we can think about this linear regression probabilistically. Let us first consider the following proposition:

Proposition 1 *Consider the one-step dynamics of a linear system. If we let* $\mathbf{x}_t \sim N(\mu,\Sigma)$*, then* $\binom{\mathbf{x}_t}{\mathbf{x}_{t+1}}$ *is a multivariate Gaussian.*

Following from this idea, if we let p define a distribution over $\mathbf{x}_{t+1}$ and $\mathbf{x}_t$:

$$p(\mathbf{x}_{t+1},\mathbf{x}_t) \sim N(\mu,\Sigma)$$

For multivariate Gaussians the conditional expectation is a linear estimate, and we can see that it is equivalent to the regression above:

$$\arg\min_{A} \sum_{t=1}^{T-1} \|A\mathbf{x}_t - \mathbf{x}_{t+1}\| = \mathbf{E}[\mathbf{x}_{t+1} \mid \mathbf{x}_t].$$

The GMM model allows us to extend this line of reasoning to consider more complicated ξ. If ξ is non-linear p will almost certainly not be Gaussian. However, GMM models can model complex distributions in terms of Gaussian Mixture Components:

$$p(\mathbf{x}_{t+1},\mathbf{x}_t) \sim GMM(k)$$

where k denotes the number of mixture components. The interesting part about this mixture distribution is that locally, it models the dynamics as before. Conditioned on particular Gaussian component i the conditional expectation is:

$$\mathbf{E}[\mathbf{x}_{t+1} \mid \mathbf{x}_t, i \in 1...k].$$

As before, conditional expectations of Gaussian random variables are linear, with some additional weighting $\phi(i \mid x_t, x_{t+1})$:

$$\arg\min_{A_i} \sum_{t=1}^{T-1} \phi(i \mid \mathbf{x}_t,\mathbf{x}_{t+1}) \cdot \|A_i\mathbf{x}_t - \mathbf{x}_{t+1}\|.$$

where $\phi(i \mid \mathbf{x}_t,\mathbf{x}_{t+1})$ is the probability of a tuple $(\mathbf{x}_{t+1},\mathbf{x}_t)$ of belonging to ith component, and this can be thought of as a likelihood of belonging to a given locally linear model.

6.4.3 Learning The Transition State Distribution

Now, we learn the parameters for the time-varying transition state distribution.

DP-GMM in State-Space: First, we fit a DP-GMM to the spatial distribution of transition states. There are numerous transition states at different locations in the state-space. If we model the states at transition states as drawn from a DP-GMM model:

$$x_t \sim GMM(\pi, \{\mu_1, ..., \mu_k\}, \{\Sigma_1, ..., \Sigma_k\})$$

Then, we can apply the DP-GMM again to group the state vectors at the transition states. After fitting the GMM, each $x \in \mathcal{X}$ will have a $\hat{z} \in \{0, 1, ..., k\}$ associated with it, which is the most likely mixture component from which it is generated.

DP-GMM in Time: Using this $\hat{z}$, we apply the second level of DP-GMM fitting over the time axis. Without temporal localization, the transitions may be ambiguous. For example, in circle cutting, the robot may pass over a point twice in the same task. Conditioned on $\hat{z} = i$, we model the times which change points occur as drawn from a GMM $t \sim GMM(\pi, \{\mu_1, ..., \mu_{l_i}\}, \{\Sigma_1, ..., \Sigma_{l_i}\})$, and then we can apply DP-GMM to the set of times. Intuitively, this can be viewed as a hierarchical clustering process that groups together events that happen at similar times during the demonstrations. The result is a distribution that models spatially and temporally similar transitions.

Interpreting the Distribution: The above model defines a mixture distribution with $m = \sum_{i=0}^{k} l_i$ components, where k is the number of state-space components, and conditioned on each state-space component i there are l_i time-axis components. If there are m total mixture components for the distribution $\{C_1, ..., C_m\}$. Each mixture component defines a Gaussian over the state-space and a distribution that is conditionally Gaussian over time. The quantiles of each component distribution will define an ordered sequence of regions $[\rho_1, ..., \rho_k]$ over the state-space (i.e., its sublevel set of the state-space Gaussian bounded by z_α and ordered by mean the time Gaussian).

6.4.4 Outlier Rejection and Loop Compaction

Next, we describe our approach to improving resilience to spurious actions and loops.

Transition State Pruning: We consider the problem of outlier transitions, ones that appear only in a few demonstrations. Each transition state will have a most likely mixture component $\hat{z} \in \{1, ..., m\}$. Mixture components whose constituent transition states come from fewer than a fraction ρ demonstrations are *pruned*. ρ should be set based on the expected rarity of outliers. For example, if ρ is 100% then the only mixture components that are found are those with at least one transition state from every demonstration (i.e., the regularity assumption). If ρ is less than 100%, then it means that every mixture component must cover some subset of the demonstrations. In our experiments, we set the parameter ρ to 80% and show the results with and without this step.

Transition State Compaction: Once we have applied pruning, the next step is to remove transition states that correspond to looping actions, which are prevalent in surgical demonstrations. We model this behavior as consecutive transition states that have the same state-space GMM mixture component. We apply this step after pruning to take advantage of the removal of outlier mixtures during the looping process.

The key question is how to differentiate between repetitions that are part of the demonstration and ones that correspond to looping actions–the sequence might contain repetitions not due to looping. As a heuristic, we threshold the L_2 distance between consecutive segments with repeated transitions. If the L_2 distance is low, we know that the consecutive segments are happening in a similar location as well. In our datasets, this was empirically found to be a good indication of looping behavior.

For each demonstration, we define a segment $\mathbf{s}^{(j)}[t]$ of states between each transition states. The challenge is that $\mathbf{s}^{(j)}[t]$ and $\mathbf{s}^{(j+1)}[t]$ may have a different number of observations and may be at different time scales. To address this challenge, we apply Dynamic Time Warping (DTW). Since segments are locally similar up to small time variations, DTW can find a most-likely time alignment of the two segments.

Let $\mathbf{s}^{(j+1)}[t^*]$ be a time aligned (w.r.t to $\mathbf{s}^{(j)}$) version of $\mathbf{s}^{(j+1)}$. Then, after alignment, we define the L_2 metric between the two segments:

$$d(j, j+1) = \frac{1}{T}\sum_{t=0}^{T}(\mathbf{s}^{(j)}[i] - \mathbf{s}^{(j+1)}[i^*])^2$$

When $d \leq \delta$, we compact two consecutive segments. δ is chosen empirically and a larger δ leads to a sparser distribution of transition states, and smaller δ leads to more transition states. For our needle passing and suturing experiments, we set δ to correspond to the distance between two suture/needle insertion points–thus, differentiating between repetitions at the same point vs. at others. However, since we are removing points from a time-series, this requires us to adjust the time scale. Thus, from every following observation, we shift the time stamp back by the length of the compacted segments.

6.5 Evaluation: Synthetic and Real Data

We present the results evaluating TSC on a synthetic dataset and three real data sets of kinematic and visual recordings of surgical training tasks on the dVRK.

6.5.1 Synthetic Example

One of the challenges in evaluating segmentation techniques on real datasets is that the ground truth is often not known. Comparing different segmentation models can be challenging due to differing segmentation criteria. We developed a synthetic dataset generator for segmentation and compared several algorithms on the generated dataset. Note, we do not intend this to be a comprehensive evaluation of the accuracy of the different techniques, but more a characterization of the approaches on a locally linear example to study the key tradeoffs. The primary purpose of our experiment is to evaluate the following hypothesis: TSC more accurately recovers the ground truth when the data is corrupted with observation noise and model noise.

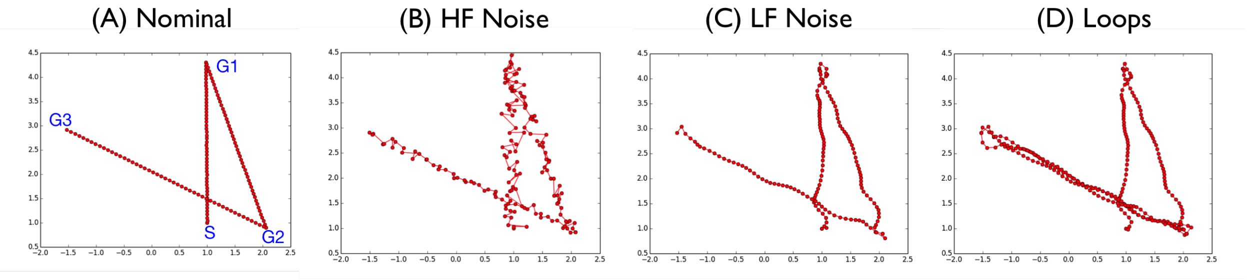

(A) Nominal (B) HF Noise (C) LF Noise (D) Loops

Figure 6.3: One of 20 instances with random goal points $G1$, $G2$, $G3$. (a) Observations from a simulated demonstration with three regimes, (b) Observations corrupted with Gaussian white sensor noise, (c) Observations corrupted with low frequency process noise, and (d) Observations corrupted with an inserted loop. See Figure 6.8 for evaluation on loops.

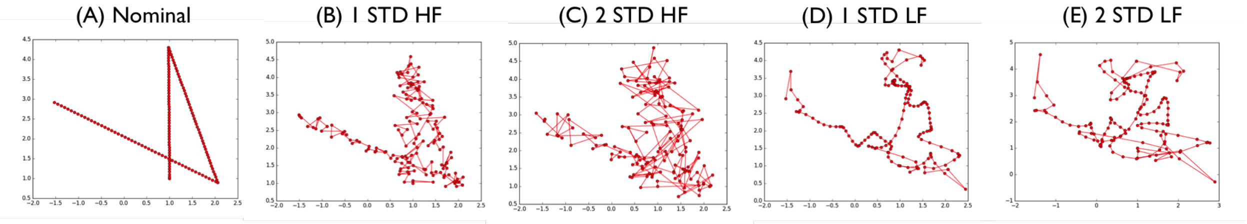

(A) Nominal (B) 1 STD HF (C) 2 STD HF (D) 1 STD LF (E) 2 STD LF

Figure 6.4: (a) Nominal trajectory, (b) 1 std. of high frequency observation noise, (c) 2 std. of high frequency observation noise, (d) 1 std. of low frequency process noise, and (e) 2 std. of low frequency process noise.

Setup We model the motion of a holonomic point robot with two-dimensional position state (x,y) between k goal points $\{g_1, ..., g_k\}$. We apply position control to guide the robot to the targets and without disturbance this motion is linear (Figure 6.3a). We add various types of disturbances (and in varying amounts) including Gaussian observation noise, low-frequency process noise, and repetitive loops (Figure 6.3b-d). We report noise values in terms of standard deviations. Figure 6.4 illustrates the relative magnitudes. A demonstration d_i is a sample from the following system.

Task: Every segmentation algorithm will be evaluated in its ability to identify the $k-1$ segments (i.e., the paths between the goal points). Furthermore, we evaluate algorithms on random instances of this task. In the beginning, we select 3 random goal points. From a fixed initial position, we control the point robot to the points with position control. Without any disturbance, this follows a linear motion. For a given noise setting, we sample demonstrations from this system and apply/evaluate each algorithm. We present results aggregated over 20 such random instances. This is important since many of the segmentation algorithms proposed in literature have some crucial hyper-parameters, and we present results with a *single* choice of parameters averaged over multiple tasks. This way, the hyper-parameter tuning cannot overfit to any given instance of the problem and has to be valid for the entire class of tasks. We believe that this is important since tuning these hyper-parameters in practice (i.e., not in simulation) is challenging since there is no ground truth. The experimental code is available at `http://berkeleyautomation.github.io/tsc/`.

Alternate Algorithms for comparison

We compare TSC against alternative algorithms which explicitly find (or approximately find) locally linear segments. It is important to reiterate that different segmentation techniques optimize different objectives, and this benchmark aims to characterize the performance on a common task.

1. *(GMM)* We compare to a version of the approach proposed by [138]. In this technique, we apply a GMM to a vector of states augmented with the current time. The authors cite [143] to argue that this is a form of local linear regression. In [138], the authors use Bayesian Information Criterion (BIC) to optimize the hyper-parameter of the number of mixture components. In our experiments, we set the parameter to the optimal choice of 3 without automatic tuning.
2. *(GMM+HMM)* A natural extension to this model is to enforce a transition structure on the regimes with a latent Markov Chain [146–149]. We use the same state vector as above, without time augmentation as this is handled by the HMM. We fit the model using the forward-backward (or Baum-Welsch) algorithm.
3. *Coresets* We evaluate against a standard coreset model [159, 160], and the particular variant is implemented with weighted k-means. We applied this to the same augmented state vector as in the previously mentioned GMM.
4. *HSMM* We evaluated a Gaussian Hidden Semi-Markov Model as used in [140]. We directly applied this model to the demonstrations with no augmentation or normalization of features. This was implemented with the package `pyhsmm`. We directly applied this model to the demonstrations with no augmentation as in the GMM approaches.
5. *AR-HMM* We evaluated a Bayesian Autoregressive HMM model as used in [140]. This was implemented with the packages *pybasicbayes*[1] and *pyhsmm-autoregressive*[2].

Evaluation Metric

There is considerable debate on metrics to evaluate the accuracy of unsupervised segmentation and activity recognition techniques, e.g. frame accuracy [163], hamming distance [156]. Typically, these metrics have two steps: (1) segments to ground truth correspondence, and (2) then measuring the similarity between corresponded segments. We have made this feature extensible and evaluated some different accuracy metrics (Jaccard Similarity, Frame Accuracy, Segment Accuracy, Intersection over Union). We found that the following procedure led to the most insightful results–differentiating the different techniques.

In the first phase, we match segments in our predicted sequence to those in the ground truth. We do this with a procedure identical to the one proposed in [163]. We define a bipartite graph of predicted segments to ground truth segments, and add weighted edges where weights represent the overlap between a predicted segment and a ground truth segment (i.e, the recall over time-steps). Each predicted segment is matched to its highest weighted ground truth segment. Each predicted segment is assigned to exactly one ground-truth segment, while ground-truth segments may have none, one, or more corresponding predictions.

---

[1] pybasicbayes: `https://github.com/mattjj/pybasicbayes`

[2] pyhsmm-autoregressive: `https://github.com/mattjj/pyhsmm-autoregressive`

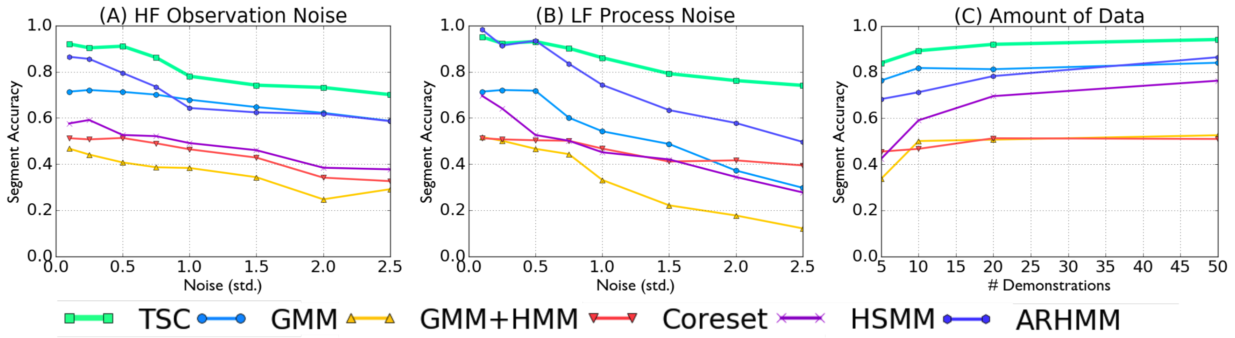


Figure 6.5: [Experiment 1] Linear Trajectories: Each data point represents 20 random instances of a 3-segment problem with varying levels of high-frequency noise, low-frequency noise, and demonstrations. We measure the segmentation accuracy for the compared approaches. (A) TSC finds more a accurate segmentation than all of the alternatives even under significant high-frequency observation noise, (B) TSC is more robust low-frequency process noise than the alternatives, (C) the Bayesian techniques solved with MCMC (ARHMM, HSMM) are more sensitive to the number of demonstrations provided than the others.

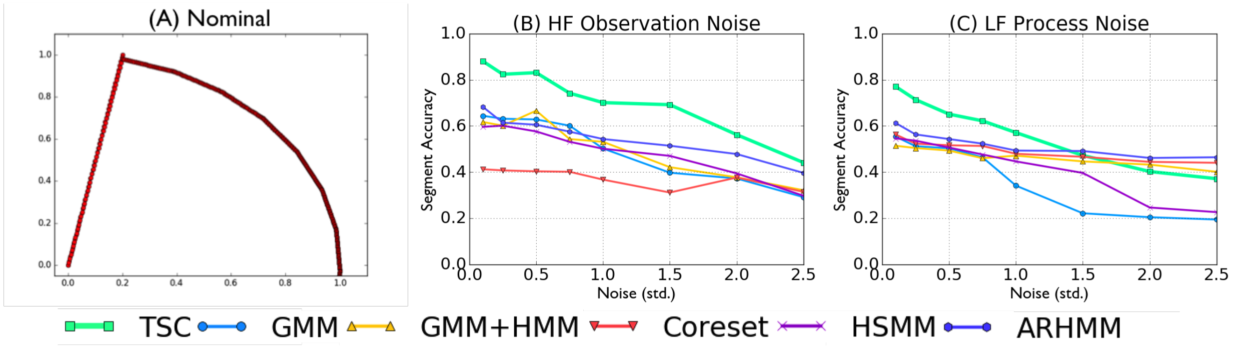


Figure 6.6: [Experiment 2] Non-Linear Trajectories: (A) illustrates a nominal trajectory of two linear dynamical motions. (B) TSC more accurately recovers the two segment ground truth than the alternatives under observation noise, (C) all of the techniques suffer in accuracy under process noise.

After establishing the correspondence between predictions and ground truth, we consider a true positive (a ground-truth segment is correctly identified) if the overlap (intersection-over-union) between the ground-truth segment and its corresponding predicted segments is more than a default threshold 60%. Then, we compute *Segment Accuracy* as the ratio of the ground-truth segments that are correctly detected. In [163], the authors use a 40% threshold but apply the metric to real data. Since this is a synthetic example, we increase this threshold to 60%, which we empirically found accounted for boundary effects especially in the Bayesian approaches (i.e., repeated transitions around segment endpoints).

Accuracy v.s. Noise

In our first experiment, we measured the segment accuracy for each of the algorithms. We also varied the amount of process and observation noise in the system. As Figure 6.4 illustrates, this is a very significant amount of noise in the data and successful techniques must exploit the structure

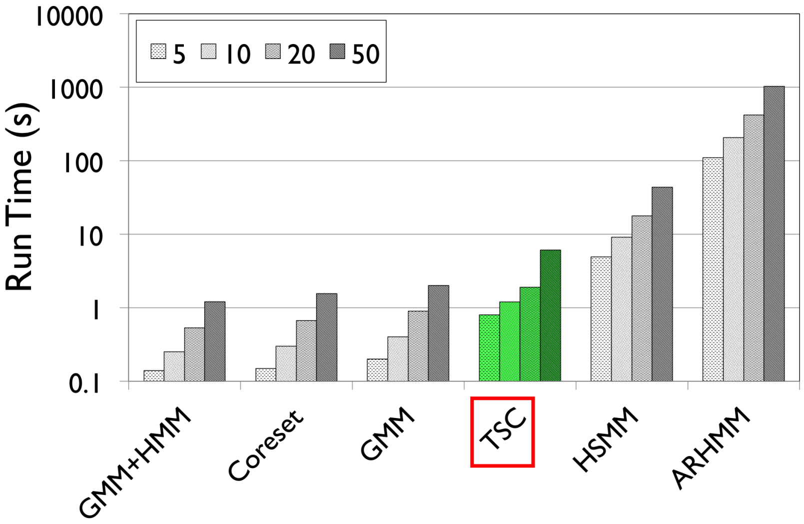


Figure 6.7: **Runtime Comparison** TSC is over 100x faster than the MCMC for the ARHMM model which is the highest performing alternate method. At the same time TSC is only about 6x slower than slower than using Coresets or the direct GMM approach, methods which do not perform so well on this task.

in multiple demonstrations. Figure 6.5a illustrates the performance of each of the techniques as a function of high-frequency observation noise. Results suggest that TSC is more robust to noise than the alternatives (nearly 20% more accurate for 2.5 std of noise). The Bayesian ARHMM approach is nearly identical to TSC when the noise is low but quickly loses accuracy as more noise is added. We attribute this robustness to the TSC's pruning step which ensures that only transition state clusters with sufficient coverage overall demonstrations are kept. These results are even more pronounced for low-frequency process noise (Figure 6.5(b)). TSC is 49% more accurate than all competitors for 2.5 std dev of noise added. We find that the Bayesian approaches are particularly susceptible to such noise. Furthermore, Figure 6.5(c) shows TSC requires no more data than the alternatives to achieve such robustness.

Another point to note is that TSC is solved much more efficiently than ARHMM or HSMM which require expensive MCMC samples. While parameter inference on these models can be solved more efficiently (but approximately) with Mean-Field Stochastic Variational Inference, we found that the results were not as accurate. TSC is about 6x slower than using Coresets or the direct GMM approach, but it is over 100x faster than the MCMC for the ARHMM model. Figure 6.7 compares the runtime of each of the algorithms as a function of the number of demonstrations.

Dynamical Trajectories

It is important to differentiate linear dynamical motions from linear trajectories. TSC models trajectories as linear dynamical systems and this allows for circular and spiral trajectories. Next, we evaluate TSC on an example with two linear dynamical systems. One system represents a straight line trajectory which transitions into a circular motion. Figure 6.6 illustrates the results. We find that this problem is substantially harder than the previous problem and all of the algorithms show reduced accuracy. TSC is still the most accurate.

TSC Hyper-Parameters

Next, we explored the dependence of the performance on the hyper-parameters for TSC. We focus on the window size and the pruning parameter. Figure 6.8a shows how varying the window size affects the performance curves. Larger window sizes can reject more low-frequency process noise. However, larger windows are also less efficient when the noise is low. Similarly, Figure 6.8b shows how increasing the pruning parameter affects the robustness to high-frequency observation noise. However, a larger pruning parameter is less efficient at low noise levels. Based on these curves, we selected $(w = 3, \rho = 0.3)$ in our synthetic experiments.

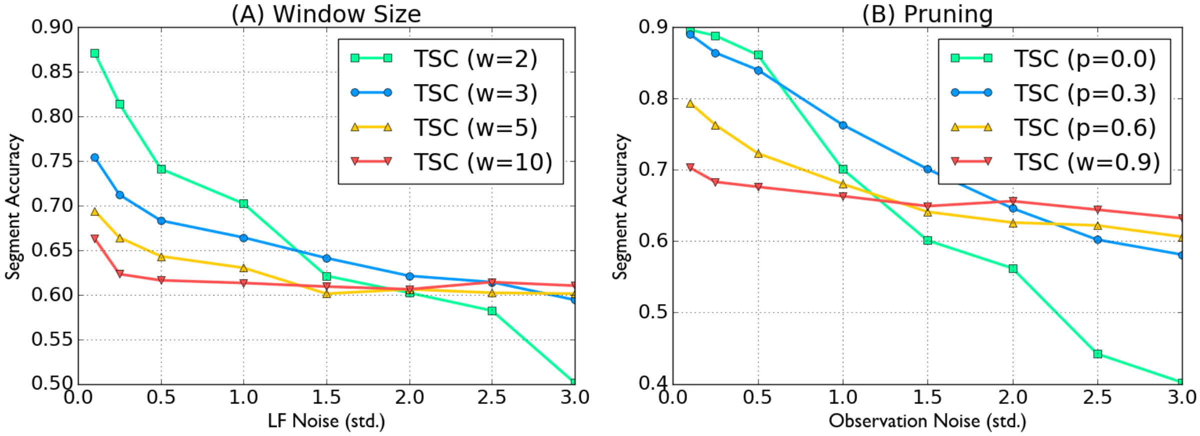


Figure 6.8: (A) shows the performance curves of different choices of windows as a function of the process noise. Larger windows can reject higher amounts of process noise but are less efficient at low noise levels. (B) the performance curves of different choices of the pruning threshold. Larger pruning thresholds are more robust to high amounts of observation noise but less accurate in the low noise setting. We selected $(w = 3, \rho = 0.3)$ in our synthetic experiments.

Loops

Finally, we evaluated 4 algorithms on how well they can detect and adjust for loops. TSC compacts adjacent motions that are overly similar, while HMM-based approaches correspond similar looking motions. An HMM grammar over segments is clearly more expressive than TSC's, and we explore whether it is necessary to learn a full transition structure to compensate for loops. We compare the accuracy of the different segmentation techniques in detecting that a loop is present (Figure 6.9). Figure 6.9a shows that TSC is competitive with the HMM approaches as we vary the observation noise; however, the results suggest that ARHMM provides the most accurate loop detection. On the hand, Figure 6.9b suggests that process noise has a very different effect.

TSC is actually more accurate than the HMM approaches when the process noise is high–even without learning a transition structure. This is an interesting property that we find is very useful in our experiments on real surgical data.

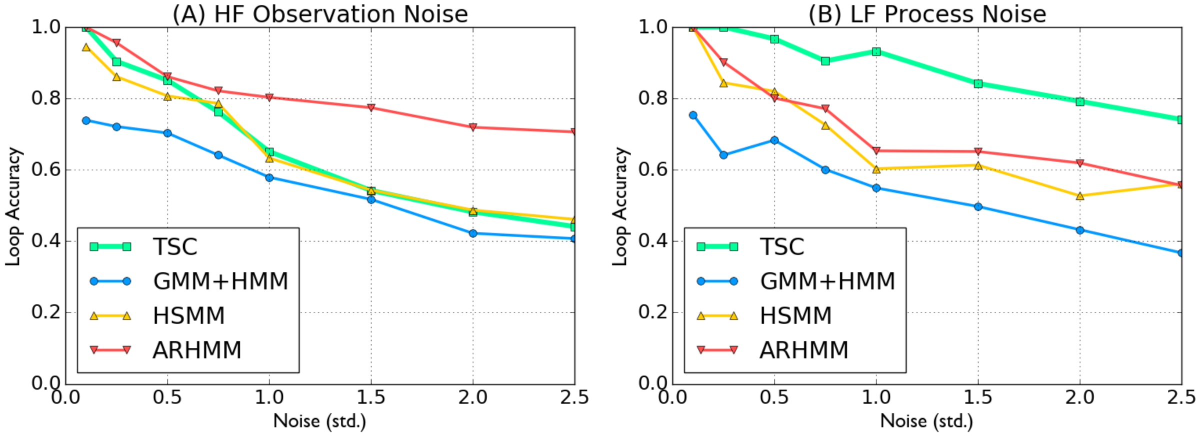


Figure 6.9: (A) illustrates the accuracy of TSC's compaction step as a function of observation noise. TSC is competitive with the HMM-based approaches without having to model the full transition matrix. (B) TSC is actually more robust to low-frequency process noise in the loops than the HMM-based approaches.

6.5.2 Evaluation on JIGSAWS Surgical Dataset

We describe the three tasks used in our evaluation and the corresponding manual segmentation (Figure 6.10). This will serve as ground truth when qualitatively evaluating our segmentation on real data. This set of experiments primarily evaluates the utility of segments learned by TSC. Our hypothesis is that even though TSC is unsupervised, it identifies segments that often align with manual annotations. In all of our experiments, the pruning parameter ρ is set to 80% and the compaction heuristic δ is to 1cm.

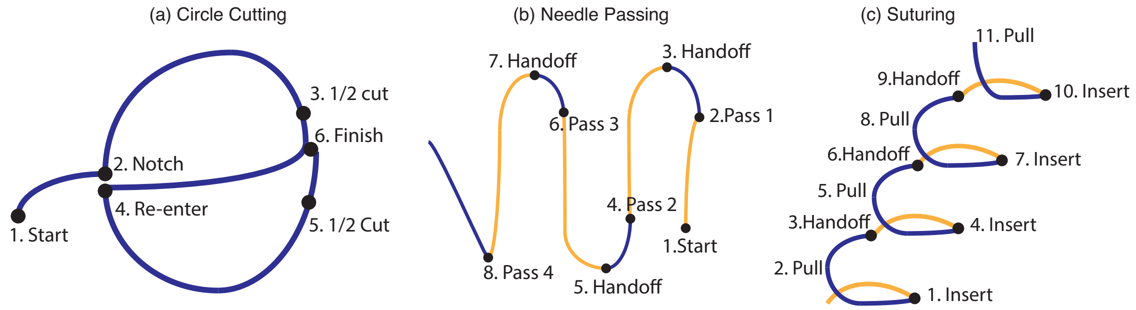


Figure 6.10: Hand annotations of the three tasks: (a) circle cutting, (b) needle passing, and (c) suturing. Right arm actions are listed in dark blue and left arm actions are listed in yellow.

Circle Cutting: A 5 cm diameter circle drawn on a piece of gauze. The first step is to cut a notch into the circle. The second step is to cut clockwise half-way around the circle. Next, the robot transitions to the other side cutting counter clockwise. Finally, the robot finishes the cut at the meeting point of the two cuts. As the left arm's only action is to maintain the gauze in tension, we exclude it from the analysis. In Figure 6.10a, we mark 6 manually identified transitions points for this task from [117]: (1) start, (2) notch, (3) finish 1st cut, (4) cross-over, (5) finish 2nd cut, and (6)

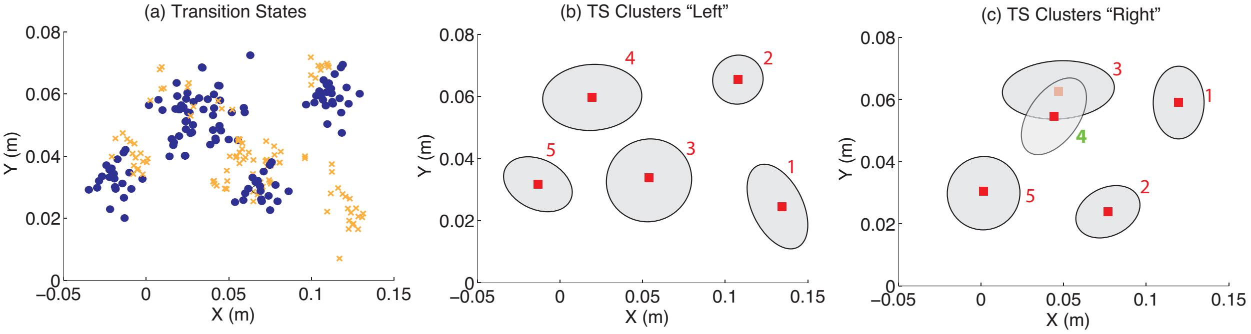


Figure 6.11: (a) The transition states for the task are marked in orange (left arm) and blue (right arm). (b-c) The TSC clusters, which are clusters of the transition states, are illustrated with their 75% confidence ellipsoid for both arms

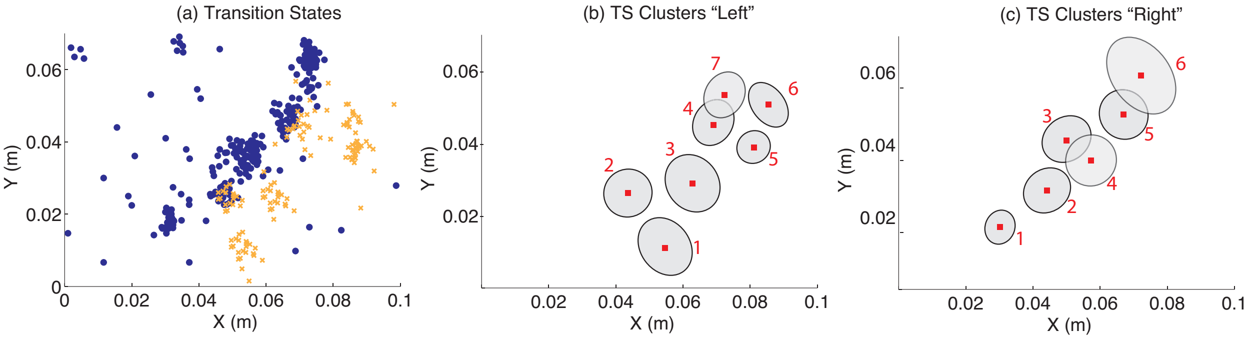


Figure 6.12: (a) The transition states for the task are marked in orange (left arm) and blue (right arm). (b-c) The clusters, which are clusters of the transition states, are illustrated with their 75% confidence ellipsoid for both arms

connect the two cuts. For the circle cutting task, we collected 10 demonstrations by non-experts familiar with operating the da Vinci Research Kit (dVRK).

We also perform experiments using the JIGSAWS dataset [20] consisting of surgical activity for human motion modeling. The dataset was captured using the da Vinci Surgical System from eight surgeons with different levels of skill performing five repetitions each of Needle Passing and Suturing.

Needle Passing: We applied TSC to 28 demonstrations of the needle passing task. The robot passes a needle through a hoop using its right arm, then its left arm to pull the needle through the hoop. Then, the robot hands the needle off from the left arm to the right arm. This is repeated four times as illustrated with a manual segmentation in Figure 6.10b.

Suturing: Next, we explored 39 examples of a 4 throw suturing task (Figure 6.10c). Using the right arm, the first step is to penetrate one of the points on the right side. The next step is to force the needle through the phantom to the other side. Using the left arm, the robot pulls the needle out of the phantom, and then hands it off to the right arm for the next point.

6.5.3 Results with Visual Features

TSC is compatible with visual features in addition to kinematic states. Our goal with these features was to illustrate that TSC applies to general state-spaces as well as spatial ones, and not to address the general perception problem. These features were constructed via manual annotation, where the Grasp and Needle Penetration were identified by reviewing the videos and marking the frames at which they occurred as described in Section 6.2.4. The use of generalized visual features from convolutional neural networks is also possible with the TSC algorithm and is studied in Chapter 7.

We evaluate TSC in this featurized state space that incorporates states derived from vision. We illustrate the transition states in Figure 6.13 with and without visual features on the circle cutting task. At each point where the model transitions, we mark the end-effector (x, y, z) location (ignoring the orientation). In particular, we show a region (red box) to highlight the benefits of these features. During the cross-over phase of the task, the robot has to re-enter the notch point and adjust to cut the other half of the circle. When only using the end-effector kinematic pose, the locations where this transition happens is unreliable as operators may approach the entry from slightly different angles. On the other hand, the use of a gripper contact binary feature clusters the transition states around the point at which the gripper is in position and ready to begin cutting again. In the subsequent experiments, we use the same two visual features.

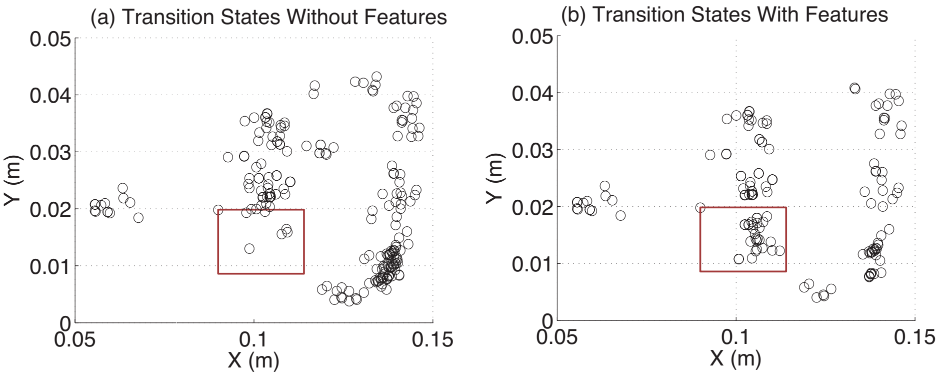


Figure 6.13: (a) We show the transition states without visual features, (b) and with visual features. Marked in the red box is a set of transitions that cannot always be detected from kinematics alone.

6.5.4 Results with Pruning and Compaction

In Figure 6.14, we highlight the benefit of pruning and compaction using the Suturing task as exemplar. First, we show the transition states without applying the compaction step to remove looping transition states (Figure 6.14a). We find that there are many more transition states at the "insert" step of the task. Compaction removes the segments that correspond to a loop of the insertions. Next, we show the all of the clusters found by DP-GMM. The centroids of these clusters are marked in Figure 6.14b. Many of these clusters are small containing only a few transition states. This is why we created the heuristic to prune clusters that do not have transition states from at least 80% of the demonstrations. In all, 11 clusters are pruned by this rule.

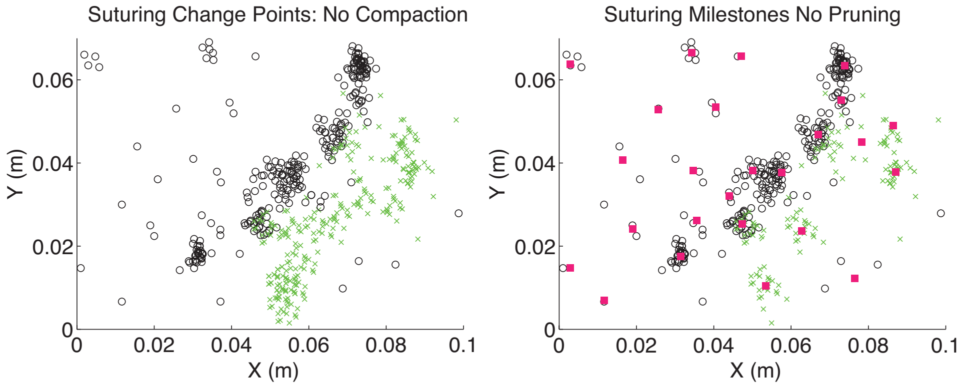


Figure 6.14: **Pruning and Compaction** We first show the transition states without compaction (in black and green), and then show the clusters without pruning (in red). Compaction sparsifies the transition states and pruning significantly reduces the number of clusters.

6.5.5 Results with JIGSAWS Surgical Data

Circle Cutting: Figure 6.15a shows the transition states obtained from our algorithm. And Figure 6.15b shows the TSC clusters learned (numbered by time interval midpoint). The algorithm found 8 clusters, one of which was pruned using our $\rho = 80\%$ threshold rule.

The remaining 7 clusters correspond well to the manually identified transition points. It is worth noting that there is one extra cluster (marked $2'$), that does not correspond to a transition in the manual segmentation. At $2'$, the operator finishes a notch and begins to cut. While at a logical level notching and cutting are both penetration actions, they correspond to two different linear transition regimes due to the positioning of the end-effector. Thus, TSC separates them into different clusters even though a human annotators did not. This illustrates why supervised segmentation is challenging. Human annotators segment trajectories on boundaries that are hard to characterize mathematically, e.g., is frame 34 or frame 37 the segment boundary. Supervisors may miss crucial motions that are useful for automation or learning.

Needle Passing: In Figure 6.11a, we plot the transition states in (x, y, z) end-effector space for both arms. We find that these transition states correspond well to the logical segments of the task (Figure 6.10b). These demonstrations are noisier than the circle cutting demonstrations and there are more outliers. The subsequent clustering finds 9 (2 pruned). Next, Figures 6.11b-c illustrate the TSC clusters. We find that again TSC learns a small parametrization for the task structure with the clusters corresponding well to the manual segments. However, in this case, the noise does lead to a spurious cluster (4 marked in green). One possible explanation is that the demonstrations contain many adjustments to avoid colliding with the needle hoop and the other arm while passing the needle through leading to numerous transition states in that location.

Suturing: In Figure 6.12, we show the transition states and clusters for the suturing task. As before, we mark the left arm in orange and the right arm in blue. This task was far more challenging than the previous tasks as the demonstrations were inconsistent. These inconsistencies were in the way the suture is pulled after insertion (some pull to the left, some to the right, etc.), leading to transition states all over the state space. Furthermore, there were numerous demonstrations with

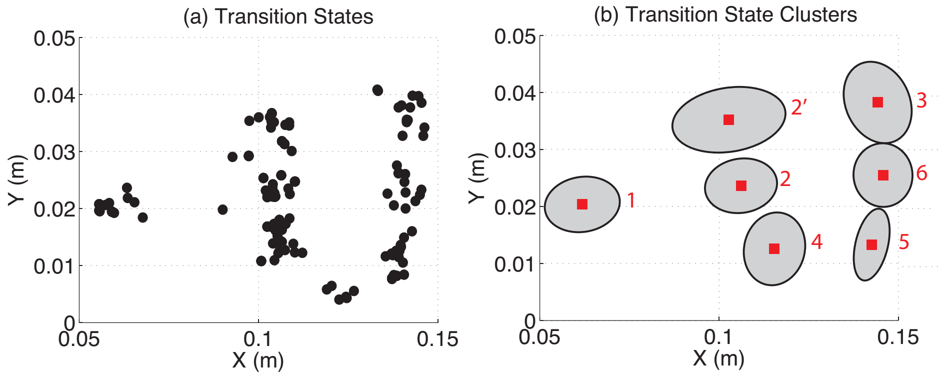


Figure 6.15: (a) The transition states for the circle cutting task are marked in black. (b) The TSC clusters, which are clusters of the transition states, are illustrated with their 75% confidence ellipsoid.

looping behaviors for the left arm. In fact, the DP-GMM method gives us 23 clusters, 11 of which represent less than 80% of the demonstrations and thus are pruned (we illustrate the effect of the pruning in the next section). In the early stages of the task, the clusters clearly correspond to the manually segmented transitions. As the task progresses, we see that some of the later clusters do not.

6.5.6 Comparison to Surgemes

Surgical demonstrations have an established set of primitives called surgemes, and we evaluate if segments discovered by our approach correspond to surgemes. In Table 6.1, we compare the number of TSC segments for needle passing and suturing to the number of annotated surgeme segments. A key difference between our segmentation and number of annotated surgemes is our compaction and pruning steps. To account for this, we first select a set of surgemes that are expressed in most demonstrations (i.e., simulating pruning), and we also apply a compaction step to the surgeme segments. When surgemes appear consecutively, we only keep the one instance of each. We explore two metrics: **TSC-Surgeme** the fraction of TSC clusters with only one surgeme switch (averaged over all demonstrations), and **Surgeme-TSC** the fraction of surgeme switches that fall inside exactly one TSC cluster. We found that the transitions learned by TSC often aligned with the surgemes. 83% and 73% of transition clusters for needle passing and suturing respectively contained exactly one surgeme transition (**TSC-Surgeme** metric). These results suggest that TSC aligns with surgemes without any explicit supervision.

Table 6.1: This table compares transitions learned by TSC and transitions identified by manual annotators in the JIGSAWS dataset. We found that the transitions mostly aligned. 83% and 73% of transition clusters for needle passing and suturing respectively contained exactly one surgeme transition. These results suggest that TSC aligns with surgemes without any explicit supervision.

	No. of Surgeme Segments	No. of Segments + C/P	No. of TSC	TSC-Surgeme	Surgeme-TSC
Needle Passing	19.3 ± 3.2	14.4 ± 2.57	11	83%	74%
Suturing	20.3 ± 3.5	15.9 ± 3.11	13	73%	66%

6.6 Discussion

Limitations The current methods although more robust to alternate algorithms are often not completely accurate. This is in part because of the linear dynamics assumption. Furthermore, the current method can only handle looping behavior up to a small number of contiguous retrials. However, failure amidst a task with restarts at a last known good state can result in going back an arbitrary number of steps in the task. This situation is not handled in TSC and would perhaps require a more sophisticated compaction procedure.

Chapter Summary We presented Transition State Clustering (TSC), which leverages the consistent structure of repeated demonstrations robustly learn segmentation criteria. To learn these clusters, TSC uses a hierarchical Dirichlet Process Gaussian Mixture Model (DP-GMM) with a series of merging and pruning steps. Our results on a synthetic example suggest that this approach is more robust than 5 other segmentation algorithms. We further applied our algorithm to three surgical datasets and found that the transition state clusters correspond well to manual annotations and transitions with respect to motions from a pre-defined surgical motion dictionary (surgemes).

Chapter 7

Transition State Clustering with Deep Learning: Unsupervised Surgical Trajectory Segmentation with videos

Overview

Inspired by the recent success of deep neural networks in reinforcement learning [164–167] and supervised learning with neural networks in robotics [168], we explore how visual features extracted from CNNs can be used for task segmentation. We are motivated by examples in robot-assisted surgery, where there are a growing number of datasets with kinematic and video recordings of surgical procedures [20, 169, 170]. While these datasets have the potential to facilitate learning and autonomy, the variability of surgical data poses a unique challenge.

Extracting common segments shared across multiple demonstrations of the same surgical task is an important pre-processing step before using this data [15, 137]. Segmentation of trajectories into locally-similar contiguous sections can facilitate subtask learning from expert demonstrations, and salvaging useful segments from otherwise inconsistent demonstrations. In the previous chapter, we introduced the algorithm Transition State Clustering for unsupervised segmentation of robotic trajectories consisting of kinematic state information along with carefully crafted visual features.

In this chapter, we extend TSC algorithm with Deep Learning that leverages unannotated video along with kinematic data for task-level segmentation, and finds regions of the visual feature space that correlate with transition events. It uses features constructed from layers of pre-trained image classification Deep Convolutional Neural Networks (CNNs).

There are several recent proposals to learn segmentation criteria with minimal supervision (i.e., no dictionaries or labels) [114, 137]. Inherently, the success of these approaches depends on the state representation, which is particularly challenging for visual features. Visual perception pipelines often require hand-coding of essential features (e.g., object tracking and pose estimation), and thus, have to be modified for each new task. The recent results in Deep Learning, especially

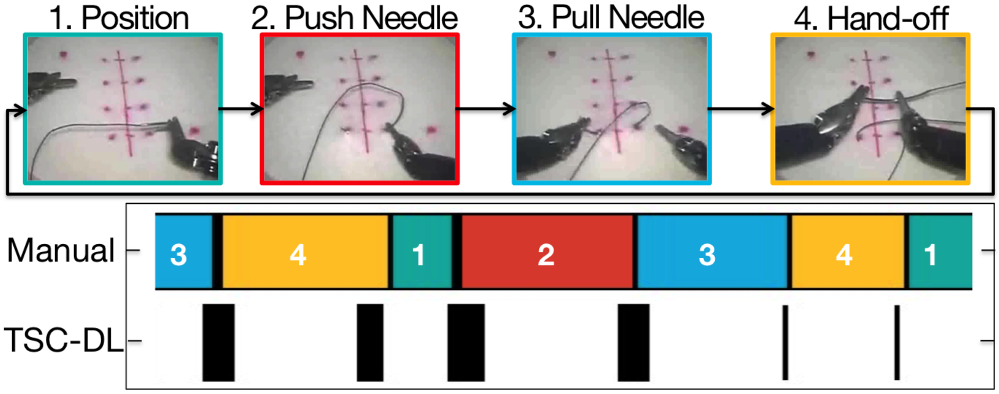


Figure 7.1: Illustrative TSC-DL result for a 4-throw suturing task. TSC-DL extracts a segmentation that closely aligns with the manual annotation without supervision. The width of the black segments illustrates a confidence interval on the predicted segment endpoint.

with Convolutional Neural Networks (CNNs), show that it is possible to use pre-trained CNNs to extract task-agnostic features [171]. These features have been shown empirically to perform well in recent work in robot visual perception [168, 172].

We propose Transition State Clustering with Deep Learning (TSC-DL), which extends the TSC algorithm (see Chapter 6) with automatically constructed visual features from pre-trained CNNs (i.e., trained on large libraries of images [173]). The key insight here is that pre-trained CNNs are effective for extracting relevant features from videos for Transition State Clustering.

In this chapter the primary contributions are: (1) exploring the effectiveness of Deep Learning methods to extract visual features for segmentation, (2) a hierarchical multi-modal clustering algorithm combining visual and kinematic trajectory data, and (3) a resampling-based estimator to predict segmentation times with confidence intervals. We report results on three datasets, two Deep Learning architectures (AlexNet and VGG), different convolutional layers, and varying dimensionality reduction techniques, to study the performance when compared with standard implementations of Scale Invariant Feature Transforms (SIFT). Comparing the performance of a pre-trained Deep Neural Network against SIFT on extracting visual features for segmentation into a sequence of segments with distinct linear dynamical system parameters, the former produced a significant (up to 30.4%) improvement in Silhouette Score (a standard measure of cluster tightness). We also compare TSC-DL with manual annotations when available using Normalized Mutual Information (NMI, a measure of sequence alignment). On real surgical datasets from JHU JIGSAWS, we find that TSC-DL matches the manual annotation with up to 0.806 NMI. Our results also suggest that applying TSC-DL to both kinematic and visual states results in increases of up to 0.215 NMI over just using the kinematics alone.

7.1 Background and Related Work

7.1.1 Deep Features in Robotics

Neural networks have demonstrated empirical success in end-to-end robotic control problems, where robots learn policies directly from images [168, 172]. The success of convolutional features in learning control policies, suggests that these features may also have other properties related to the underlying dynamical system. In this chatper, we presented methodologies for leveraging deep features in segmentation in combination with the Transition State model which is motivated by dynamical system theory. We believe that segmentation is an important first step in many robot learning applications, and the appropriate choice of visual features is key to accurate segmentation.

7.1.2 Visual Gesture Recognition

A highly relevant line of work is visual activity recognition, and many recent works attempt to segment human motion primitives from videos [163, 174–178]. There are a few unsupervised models for segmentation of human actions: Jones and Shao [177], Yang et al. [176], Wu and Shao [178], and Wu et al. [163]. TSC-DL studies a broader problem of robot task segmentation where states may be represented by kinematics, vision, or both. Jones and Shao [177] studied the problem of segmentation with two temporally aligned views of the same action, and they proposed an algorithm called Dual Assignment k-Means (DAKM) to relate the segments in the two views. It is not clear how this would support multiple demonstrations (>2) with temporal inconsistencies. Other algorithms derived from k-means have also been popular. Yang et al. [176] and Wu et al. [163] use k-means to learn a dictionary of primitive motions, however, in prior work, we found that transition state clustering outperforms a standard k-means segmentation approach. In fact, the model that we propose is complementary to these works and could provide a robust drop-in-replacement for the k-means dictionary learning step [15].

7.1.3 Learning From Videos in Surgical Robotics

In addition to the segmentation review in Chapter 6, few other studies analyze videos along with kinematic data for trajectory segmentation. One study from Zappella et al. [141] use features from both the videos and kinematic data to classify surgical motions with manually segmented videos as input. Similarly, Quellec et al. [142] use manually segmented examples as training for segmentation and recognition of surgical tasks based on archived cataract surgery videos.

The methods in this chapter do not assume prior knowledge of motion primitives and focuses on general visual features used across domains without manual annotations or feature construction.

7.2 Problem Statement

The goal of Transition State Clustering is to learn set of transition clusters $\mathcal{C}$ from a set of demonstrations of a task. There are two sub-problems related to this goal: (1) learning the parameters of the model $\mathcal{C}$ from all demonstrations, and (2) for each demonstration d, identifying states that most align with the segments defined by $\mathcal{C}$. This is a similar problem setup as described in the graphical model for multi-modal data in Figure 6.2 in Chapter 6.

Problem 1. Task Segmentation

A set of demonstrations is consistent if there exists a clustering model $\mathcal{C}$ that respects the partial order of every demonstration (see Section 6.2 for a precise definition). Given a consistent set of demonstrations, the problem is to find a sequence of transition state clusters $\mathcal{C}$ reached by at least a fraction ρ of the demonstrations.

Problem 2. Temporal Segmentation

The set of clusters $\mathcal{C}$ define regions of the state-space and times where transitions occur common to multiple demonstrations of a task. For each demonstration d, we would like to know which states are transitions that correspond to the clusters in $\mathcal{C}$. Due to the pruning, there may be transitions that are present in some demonstrations but not in others. Furthermore, a demonstration may have multiple transitions within the same cluster. Hence, we also need a measure of confidence on when the transition occurs.

Given $\mathcal{C}$, the problem is to find a set of *predicted* transitions *for each demonstration* d_i. For every d_i, there will be some subset of transition state clusters $C^i \subseteq \mathcal{C}$ that are relevant to the individual demonstration. For each $c \in C^i$, we would like to identify the time t_c of the transition event in d_i.

7.2.1 Evaluation Metrics

It is important to note that TSC-DL is an unsupervised algorithm that does not use labeling. Therefore, we evaluate TSC-DL both intrinsically (without labels) and extrinsically (against human annotations).

Intrinsic metric: The goal of the intrinsic metric is to compare the performance of different featurization techniques, encodings, and dimensionality reduction within TSC-DL without reference to external labels. This score is not meant to be an absolute metric of performance but rather a relative measure. This measures "tightness" of the transition state clusters. This metric is meaningful since we require that each transition state cluster contains transitions from a fraction of at least ρ of the demonstrations. The tightness of the clusters measures how well TSC-DL discovers regions of the state space where transitions are grouped together. This is measured with the mean *Silhouette Score* (denoted by ss), which is defined as follows for each transition state i:

$$\mathsf{ss}(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}}, \quad \mathsf{ss}(i) \in [-1, 1].$$

If transition state i is in cluster C_j, $a(i)$ is defined the average dissimilarity of point i to all points in C_j, and $b(i)$ is the dissimilarity with the closest cluster measured as the minimum mean dissimilarity of point i to cluster C_k, $k \neq j$. We use the L_2-norm as the dissimilarity metric and rescale `ss` $\in [0,1]$ for ease of comparison.

Extrinsic metric: For every time t, we will have a TSC-DL prediction τ_t and a manual annotation l_t. To calculate a measure of similarity between τ and l we use the *Normalized Mutual Information* (NMI), which measures the alignment between two label assignments irrespective of index choice. NMI is equal to the KL-divergence between the joint distribution and the product distribution of the marginals; intuitively quantifying the distance from pairwise statistical independence. The NMI score lies in [0, 1], where 0 indicates independence while 1 corresponds to a perfect matching. It is defined as:

$$NMI(\tau,l) = \frac{I(\tau,l)}{\sqrt{H(\tau)H(l)}}, \quad NMI(\tau,l) \in [0,1].$$

7.3 TSC-DL for Visual State Space

Suppose there was only one regime, then following from the Gaussian assumption, we obtain a linear regression problem:

$$\arg\min_{A} \|AX_t - X_{t+1}\| ,$$

where $X_t = [\mathbf{x}(1),\ldots,\mathbf{x}(T)] \in \mathbb{R}^{n\times T}$ with each column as the state at time t: $\mathbf{x}(t) \in \mathbb{R}^n$. Generalizing to multiple regimes, Moldovan et al. [145] showed that fitting a Jointly Gaussian model to $n(t) = \binom{\mathbf{x}(t+1)}{\mathbf{x}(t)}$ is equivalent to Bayesian Linear Regression–and thus fitting a GMM finds locally linear regimes.

Over all of the demonstrations, TSC-DL clusters the states at which these transitions occur. The key challenge is that we have a state-space composed of multiple sensing modalities such as kinematics and visual state. Such states may not be directly comparable due to differences in cardinality (many more visual states than kinematics states), in semantics (distances between kinematic states may be more significant), and in stochasticity (kinematic measurements are likely less noisy than visual ones). We address this problem by constructing a hierarchy of GMM clusters, where each hierarchy only clusters over a single sensing modality.

7.3.1 Visual Features

Transition State Clustering with Deep Learning (TSC-DL) utilizes domain independent visual features from pre-trained CNNs. CNNs are increasingly popular for image classification and with existing models trained on millions of natural images. Intuitively, CNNs classify based on aggregations (pools) of hierarchical convolutions of the pixels. Yosinski et al. noted that CNNs trained on natural images exhibit roughly the same Gabor filters and color blobs on the first layer for various datasets [171]. They established that earlier layers in the hierarchy learn more general features while later layers learn more specific ones. Hence, removing the aggregations and the classification layers results in convolutional filters which can be used to derive generic features across datasets.

Algorithm 4: TSC-DL: Transition Learning

```
Data: Set of demonstrations: $\mathcal{D}$
foreach $d_i \in \mathcal{D}$ do
    // concatenate kinematic & visual features
    $x_i(t) \leftarrow \left[\binom{\mathbf{k_i}(t-1)}{\mathbf{z_i}(t-1)}, \binom{\mathbf{k_i}(t)}{\mathbf{z_i}(t)}, \binom{\mathbf{k_i}(t+1)}{\mathbf{z_i}(t+1)}\right]^T \ \forall t \in \{1,\ldots,T_i\}$
    foreach $t \in \{1,\ldots,T_i\}$ do $\mathbf{X} \leftarrow \mathbf{X} \cup x_i(t)$
// $C_i(t)$ is Index of cluster containing $x_i(t)$
$\{C_i(t), \forall\, x_i(t) \in \mathbf{X}\} \leftarrow$ DP-GMM(X)
$\Theta \leftarrow \emptyset$                         // $\Theta$: set of all transition states in $\mathcal{D}$
foreach $d_i \in \mathcal{D}$ do
    $\Theta \leftarrow \Theta \cup x_i(t)$, $\forall t$, s.t. $C_i(t) \neq C_i(t+1)$
Result: The set of transitions $\Theta$
```

We use layers from a pre-trained Convolutional Neural Network (CNNs) to derive the features frame-by-frame. In particular, we explore two architectures designed for image classification task on natural images: (a) **AlexNet:** Krizhevsky et al. proposed multilayer (5 in all) a CNN architecture [173], and (b) **VGG:** Simoyan et al. proposed an alternative architecture termed VGG (acronym for Visual Geometry Group) which increased the number of convolutional layers significantly (16 in all) [179]. In our experiments, we explore the level of generality of features required for segmentation. We also compare these features to other visual featurization techniques such as SIFT for the purpose of task segmentation using TSC-DL.

Visual Feature Encoding and Dimensionality Reduction

1. *Feature Encoding* After constructing these features, the next step is encoding the results of the convolutional filter into a vector $z(t)$. We explore three encoding techniques: (1) Raw values, (2) Vector of Locally Aggregated Descriptors (VLAD) [180], and (3) Latent Concept Descriptors (LCD) [181].

2. *Dimensionality Reduction* After encoding, we feed the CNN features $z(t)$, often in more than 50K dimensions, through a dimensionality reduction process to boost computational efficiency. This also balances the visual feature space with a relatively small dimension of kinematic features (< 50). Moreover, GMM-based clustering algorithms usually converge to a local minimum and very high dimensional feature spaces can lead to numerical instability or inconsistent behavior. We explore multiple dimensionality reduction techniques to find desirable properties of the dimensionality reduction that may improve segmentation performance. In particular, we analyze Gaussian Random Projections (GRP), Principal Component Analysis (PCA) and Canonical Correlation Analysis (CCA) in Table 7.1. GRP serves as a baseline while PCA is used based on its wide application in computer vision [181]. We also explore CCA as it finds a projection that maximizes the correlation between the visual features and the kinematics features.

Algorithm 5: TSC-DL: Task Segmentation Learning

```
Data: The set of transitions $\Theta$, data $\mathbf{X}$                // line #3 Alg 4
// Cluster over Visual Features of Transitions
$\mathcal{C}^z : \{z_i(t), \forall\, x_i(t) \in \Theta\} \leftarrow$ DP-GMM($\Theta$)      $z_i(t)$:cluster index
foreach $z_i(t) \in \mathcal{C}^z$ do
    $\Theta^z \leftarrow \{x_i(t) \in \Theta$, s.t. $\hat{z}_i(t) = z_i(t)\}$      $\hat{z}_i(t)$: index of $x_i(t)$
    // Cluster over Kinematic Features of Transitions
    $\mathcal{C}^z_k : \{k_i(t), \forall\, x_i(t) \in \Theta^z\} \leftarrow$ DP-GMM($\Theta^z$)
    foreach $k_i(t) \in \mathcal{C}^z_k$ do
        $\Theta^z_k \leftarrow \{x_i(t)$ s.t. $x_i(t) \in \Theta^z, \hat{k}_i(t) = k_i(t)\}$
        if $\sum_{d_i} \mathbf{1}\Big(\sum_{t \in T_i} \mathbf{1}(x_i(t) \in \Theta^z_k) \geq 1\Big) \leq \rho|\mathcal{D}|$ then
            $\mathcal{C}^z_k \leftarrow \mathcal{C}^z_k \setminus \{k_i(t)\}$               // Cluster Pruning
Result: The set of transitions $\Theta^z_k, \forall z, k\}$
```

7.3.2 Algorithm Overview

We define an augmented state space $\mathbf{x}(t) = \binom{k(t)}{z(t)}$, where $k(t) \in \mathbb{R}^k$ are the kinematic features and $z(t) \in \mathbb{R}^v$ are the visual features. The augmented state for each demonstration $d_i \in \mathcal{D}$ is collected in a state vector $\mathbf{X}$. GMM clustering over the sequence of states in $\mathbf{X}$, results in the identification of the set of transitions Θ, or switching events where $A(t) \neq A(t+1)$ as outlined in Algorithm 4.

Subsequent hierarchical clustering uses state representations only at transitions in set Θ. Intuitively, the Transition Learning results in an over-segmentation of the trajectory in state space, while subsequent clustering steps retain only a small subset of transition states that are consistent across the data set. After that, we cluster in sub-spaces of each of the modalities – perception and kinematics. We start with clustering over subspace of visual feature to obtain a set clusters:Θ^z, indexed by z_i. Within each visual feature space cluster (Θ^z), we model the kinematics change points to be drawn from a GMM: $k \sim N(\mu_i, \sigma_i)$, and fit a GMM to the kinematic subspace of the transition states in Θ^z_k as outlined in Algorithm 5.

Similarly, time can also be modeled as a separate sensing modality. Without consideration of time, the transitions may be ambiguous. For example, in a "Figure 8" trajectory, the robot may pass over a point twice in the same task. Within a state cluster, we model the times at which change points occur as drawn from a GMM: $t \sim N(\mu_i, \sigma_i)$. This groups together events that happen at similar times during the demonstrations. The result is clusters of states and times. Thus, a transition state m_k defines a GMM over the state-space and a time interval.

Skill-Weighted Pruning: After the second stage of clustering, we perform a consistency check in recovered transition state clusters by pruning clusters which do not have change points from at least a ρ-fraction of the dataset. This accounts for outliers and identifies inconsistent demonstrations.

However, demonstrators may have varying skill levels leading to increased outliers, and so we extend our outlier pruning to include weights. Let, w_i be the weight for each demonstration $d_i \in \mathcal{D}$,

Algorithm 6: TSC-DL: Temporal Segmentation

```
Data: Set of demonstrations:$\mathcal{D}$
foreach $d_i \in \mathcal{D}$ do
    $\Theta_{[i]} \leftarrow$ Transition-Learning($\mathcal{D}'$) for $\mathcal{D}' = \mathcal{D} \setminus d_i$
    $\{\ddot{\Theta}^z_k, \forall z, k\} \leftarrow$ Task-Seg-Learning($\Theta_{[i]}, \mathbf{X}_{[i]}$)
    foreach $c \in \mathcal{C}_{[i]}$ do
        foreach $d_i \in \mathcal{D}'$ do
            $T_i \leftarrow T_i \cup \{t : \hat{k}_i(t) = k_i(t), x_i(t) \in d_i\}$
    $\mathtt{T}_j \leftarrow \mathtt{T}_j \cup T_j^{(i)}$, $\{\forall j : d_j \in \mathcal{D}'\}$        // $T_j^{(i)}$:  ith iteration
// Cluster over time to predict Transition Windows
foreach $d_i \in \mathcal{D}$ do
    $(\mathcal{T}_i, \sigma_i) \leftarrow DPGMM(\mathtt{T}_i, \alpha_4)$
Result: Set of Predicted Transitions Times $\mathcal{T}_i \pm \sigma_i$,  $\forall d_i \in \mathcal{D}$
```

such that $w_i \in [0,1]$ and $\hat{w}_i = \frac{w_i}{\sum w_i}$. Then a cluster $k_i(t)$ is pruned if it does not contain transitions Θ^z_k from at least a ρ fraction of demonstrations:

$$\sum_{d_i} \hat{w}_i \mathbf{1}\Big(\sum_{t \in T_i} \mathbf{1}(x_i(t) \in \Theta^z_k) \geq 1\Big) \leq \rho.$$

This criterion enforces that the task segmentation contains transition states from highly weighted demonstrations even if the data set is unbalanced, i.e., it contains many more noisy data points than good ones. In our experiment, the choose the weights as inversely proportional to average time of each example: $\hat{w}_i = 1/T_i$.

State Memory To better capture transitions that are not instantaneous, we use rolling window states where each state $\mathbf{x}_{(t)}$ is a concatenation of T consecutive states starting at t. We varied the length of temporal history T and evaluated the performance of the TSC-DL algorithm for the suturing task using a metric defined in Section 7.2.1. We empirically found a sliding window of size 3, i.e., $\mathbf{x}_t = \left[\binom{\mathbf{k_i}(t-1)}{\mathbf{z_i}(t-1)}, \binom{\mathbf{k_i}(t)}{\mathbf{z_i}(t)}, \binom{\mathbf{k_i}(t+1)}{\mathbf{z_i}(t+1)}\right]^T$, as the state representation that led to improved segmentation accuracy while balancing computational effort.

Temporal Segmentation of Each Demonstration Once we have learned the model parameters for the entire task, the next step is to identify which states in each demonstration correspond to transition events. In a single demonstration, we may have missing transitions and transitions with multiple candidate states, and so there is some ambiguity about which state best represents a particular transition cluster. Our criteria for disambiguating the assignments is *robustness*, where we want to identify those assignments that are most likely to persist even if the rest of the demonstrations are slightly different.

We iteratively hold out one of the N demonstrations and apply TSC-DL to the remaining demonstrations. For each demonstration $d_i \in \mathcal{D}$, there are $N-1$ predictions in each of the runs where d_i is in the sample. We aggregate the predictions using another clustering step, and output cluster means ($\mathcal{T}_i$) and variances (σ_i) as temporal segment predictions with standard deviations as outlined in Algorithm 6. This style of estimation has been well studied in non-parametric statistics (e.g., Bootstrapping, and Jackknife estimators).

7.4 Evaluation: Synthetic and Real Data

7.4.1 Pre-processing

Once the images were pre-processed, we applied the convolutional filters from the pre-trained neural networks frame by frame. To reduce variance due to extraneous objects and lighting changes, we crop each video to capture only the relevant workspace where robot manipulation occurs. Then, the videos are rescaled to 640x480 along with down-sampling to 10 frames per second for computational efficiency. All frames in the videos are normalized to a *zero* mean in each RGB-channel. individually [173, 179]. All pre-processing was performed with the open source `ffmpeg` library.

Sensitivity Analysis There are two hyperparameters for TSC-DL which we set empirically: sliding window size (T = 3), and the number of PCA dimensions (k = 100). In Figure 7.2, we show a sensitivity plot with the ss as a function of the parameter. We calculated the ss using the same subset of the suturing dataset as above and with the VGG conv5_3 CNN. We found that T = 3 gave the best performance. We also found that PCA with k = 1000 dimensions was only marginally better than k = 100 yet required >30 mins to run. For computational reasons, we selected k = 100.

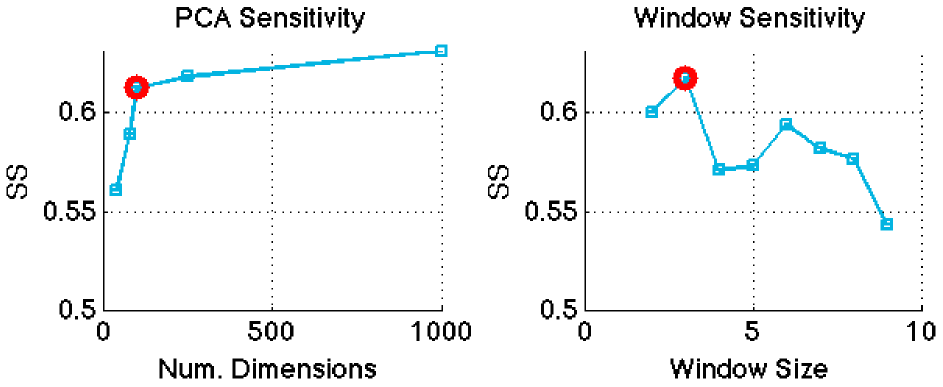


Figure 7.2: We evaluate the sensitivity of two hyperparameters set in advance: number of PCA dimensions and sliding window size. The selected value is shown in red double circles.

7.4.2 Evaluation of Visual Featurization

In our first experiment, we explore different visual featurization, encoding, and dimensionality reduction techniques. We applied TSC-DL to our suturing experimental dataset and measured

Table 7.1: The Table lists the silhouette scores for each of the techniques and dimensionality reduction schemes on a subset of suturing demonstrations (5 expert examples). We found that PCA (100 dims) applied to VGG conv5_3 maximizes silhouette score

	GRP	PCA	CCA
Non-Network Features			
SIFT		0.443±0.008	
CNN Features			
AlexNet conv3	0.559±0.018	0.600±0.012	0.494±0.006
AlexNet conv4	0.568±0.007	0.607±0.004	0.488±0.005
AlexNet pool5	0.565±0.008	0.599±0.005	0.486±0.012
VGG conv5_3	0.571±0.005	**0.637±0.009**	0.494±0.013
VGG LCD-VLAD	0.506±0.001	0.534±0.011	0.523±0.010
AlexNet LCD-VLAD	0.517±0.001	0.469±0.027	0.534±0.018

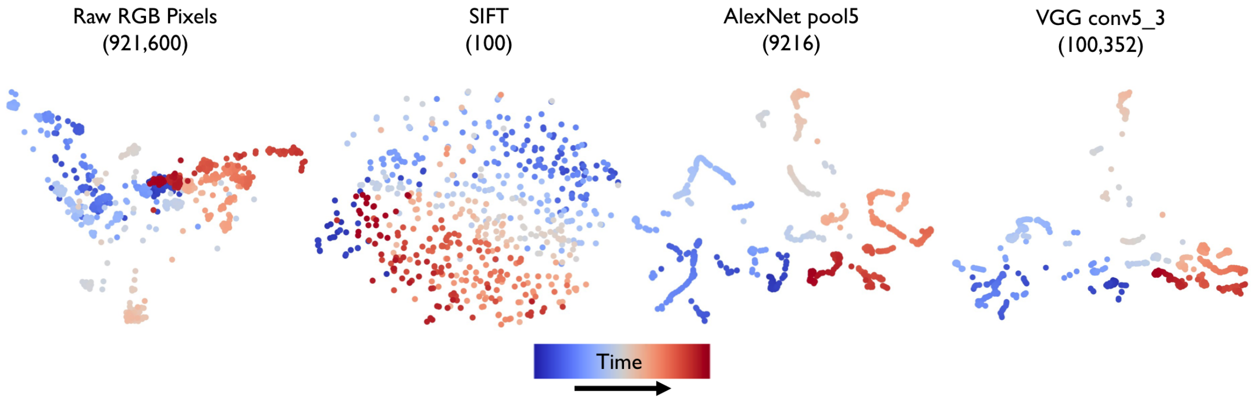


Figure 7.3: Each data point in the figure corresponds to a t-SNE visualization of features of a single frame in the video. (a) RGB pixel values of original image (b) shallow SIFT features (c) CNN features from AlexNet pool5 (d) CNN features from VGG Conv5_3. CNN features result in a better clustering performance than SIFT features as supported by Table 7.1

the silhouette score of the resulting transition state clusters. Table 7.1 describes the featurization techniques on the vertical axis and dimensionality reduction techniques on the horizontal axis.

On this dataset, our results suggest that features extracted from the pre-trained CNNs resulted in tighter transition state clusters compared to SIFT features with a 3% lower ss than the worst CNN result. Next, we found that features extracted with the VGG architecture resulted in the highest ss with a 3% higher ss than the best AlexNet result. We also found that PCA for dimensionality reduction achieved a ss performance of 7% higher than the best GRP result and 10% higher than best CCA result. Because CCA finds projections of high correlation between the kinematics and video, we believe that CCA discards informative features resulting in reduced clustering performance. We note that neither of the encoding schemes, VLAD or LCD significantly improves the ss.

t-SNE visualization of visual features One of the main insights of this study is that features from pre-trained CNNs exhibit locally-linear behavior which allows application of a switching linear dynamical system model. We experimentally tested this by applying dimensionality reduction to trajectories of features from different video featurization techniques. Figure 7.3 shows t-SNE embeddings of visual features extracted for a single demonstration of suturing. The deep features display clear locally-linear properties and can be more easily clustered than SIFT features extracted for the corresponding frames. We speculate that SIFT breaks up trajectory structure due to its natural scale and location invariance properties. We also compared to using the raw RGB image pixel values and discovered that the deep features result in more well-formed locally linear trajectories. However, it is important to note that unlike spatial trajectories there are discrete jumps in the convolutional trajectories. We hope to explore this problem in more detail in future work.

7.4.3 End-to-End Evaluation

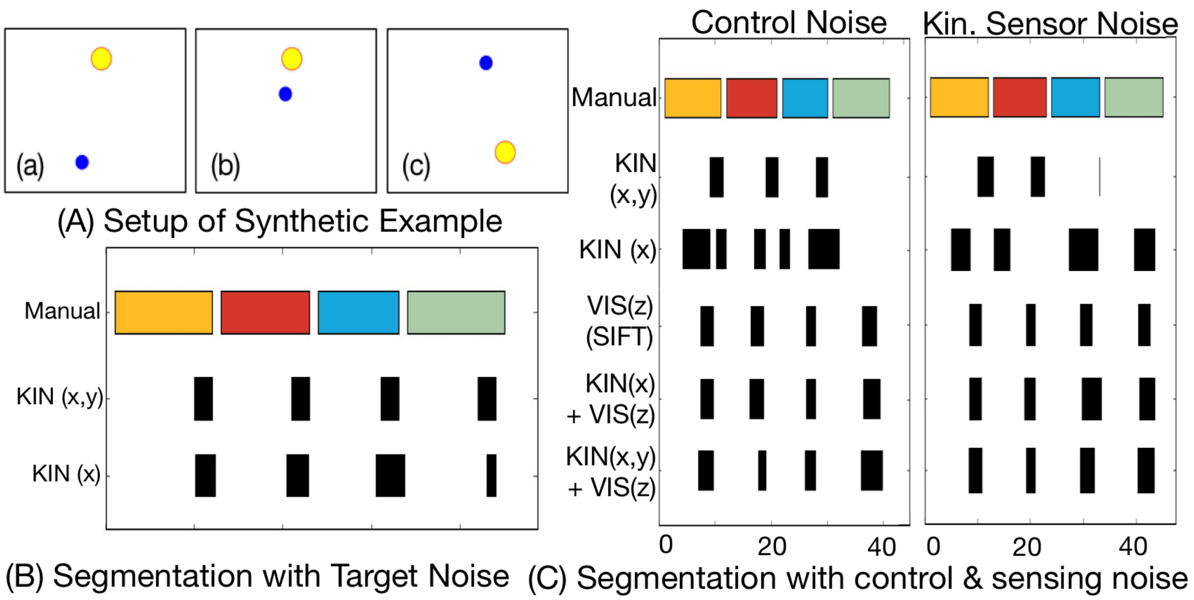
For all subsequent experiments on real data, we used a pre-trained VGG CNN conv5_3 and encoded with PCA with 100 dimensions.

1. Synthetic Example: We first evaluate TSC-DL on a synthetic example consisting of 4 linear segments (Figure Figure 7.4). A point robot on a plane moves towards a target in a straight line. Once it reaches the target, the target moves to a new location. This process is repeated four times. We use the simulation to generate image data and kinematics data. Figure 7.4 (b) shows the results of unsupervised segmentation using only kinematics component of the data ($\binom{x(t)}{y(t)}$). When the state is fully observed (i.e., we have both x and y positions), we accurately recover four segments with kinematics alone. If one of these dimensions is unobserved, we find that we can still recover the four segments. In this example, when there is no noise on the kinematics, one dimension alone is enough to learn the segmentation.

Next, in Figure 7.4, we make this scenario more complex by introducing control noise: $x(t+1) = x(t) + u(t) + \nu$, with $\nu \sim \mathcal{N}(0, d_1)$ where $d_1 = 0.25$ We find that when there is control noise, partial observed kinematics can lead to erroneous segments even in this synthetic example. We use this example to demonstrate the importance of visual features. If we add visual features (using SIFT since these are not natural images), we find that we can mitigate the problems caused by noise and partial observability. Finally, we repeat the above experiment for kinematic sensor noise in the system $\hat{x}(t) = x(t) + \nu$, where $\nu \sim \mathcal{N}(0, d_2)$ with $d_2 = 0.25$. We note that only the kinematics is corrupted with noise while the vision sees a straight trajectory.

2. Suturing:

We apply our method to a subset of the JIGSAWS dataset [20] consisting of surgical task demonstrations under teleoperation using the da Vinci surgical system. The dataset was captured from eight surgeons with different levels of skill, performing five repetitions each of suturing and needle passing. Table 7.2 lists quantitative results for both needle passing and suturing with both SS and NMI agreement with the human labels. Demonstrations from the JIGSAWS dataset were annotated with the skill-level of the demonstrators (Expert (E), Intermediate (I), and Novice (I)). For the suturing dataset, we find that using both kinematics and video gives up to 30.1% improve-



(A) Setup of Synthetic Example

(B) Segmentation with Target Noise

(C) Segmentation with control & sensing noise

Figure 7.4: (A) The figure shows a 2D synthetic example with a moving point in blue and target in yellow. The robot moves to the target in a straight line in discrete steps, and a new target appears. (B) Segmentation results for repeated demonstrations with variance in target position. (C) Segmentation under Control noise, Sensor noise, and Partial observation.

ment in ss and 52.3% improvement in NMI over using kinematics alone. Not surprisingly, we also find that the expert demonstrations, which are usually smoother and faster, lead to improved segmentation performance when using only the kinematic data. However, when we incorporate the visual data, the trend is not as clear. We speculate this has to do with the tradeoff between collecting more data (denser clusters and more accurate modeling) versus inconsistencies due to novice errors, and this tradeoff is evident in higher dimensional data.

We visualize the results of the segmentation on one representative trajectory (Figure 7.5). With combined kinematics and vision, TSC-DL learns many of the important segments identified by annotation in [20]. Upon further investigation of the false positives, we found that they corresponded to meaningful actions missed by human annotators. TSC-DL discovers that a repositioning step where many demonstrators penetrate and push through the needle in two different motions. While this is largely anecdotal evidence, we were able to find some explanations for some of the false positives found by TSC-DL.

3. Needle Passing: Next, we applied TSC-DL to 28 demonstrations of the needle passing task. These demonstrations were annotated in [20]. In this task, the robot passes a needle through a loop using its right arm, then its left arm to pull the needle through the loop. Then, the robot hands the needle off from the left arm to the right arm. This is repeated four times. Similar to the suturing dataset, we find that the combination of the features gives the best results. For the needle passing dataset, we find that using both kinematics and video gives up to 22.2% improvement in ss and 49.7% improvement in NMI over using the best of either kinematics or vision alone.

We found that the learned segments for the needle passing task were less accurate than those

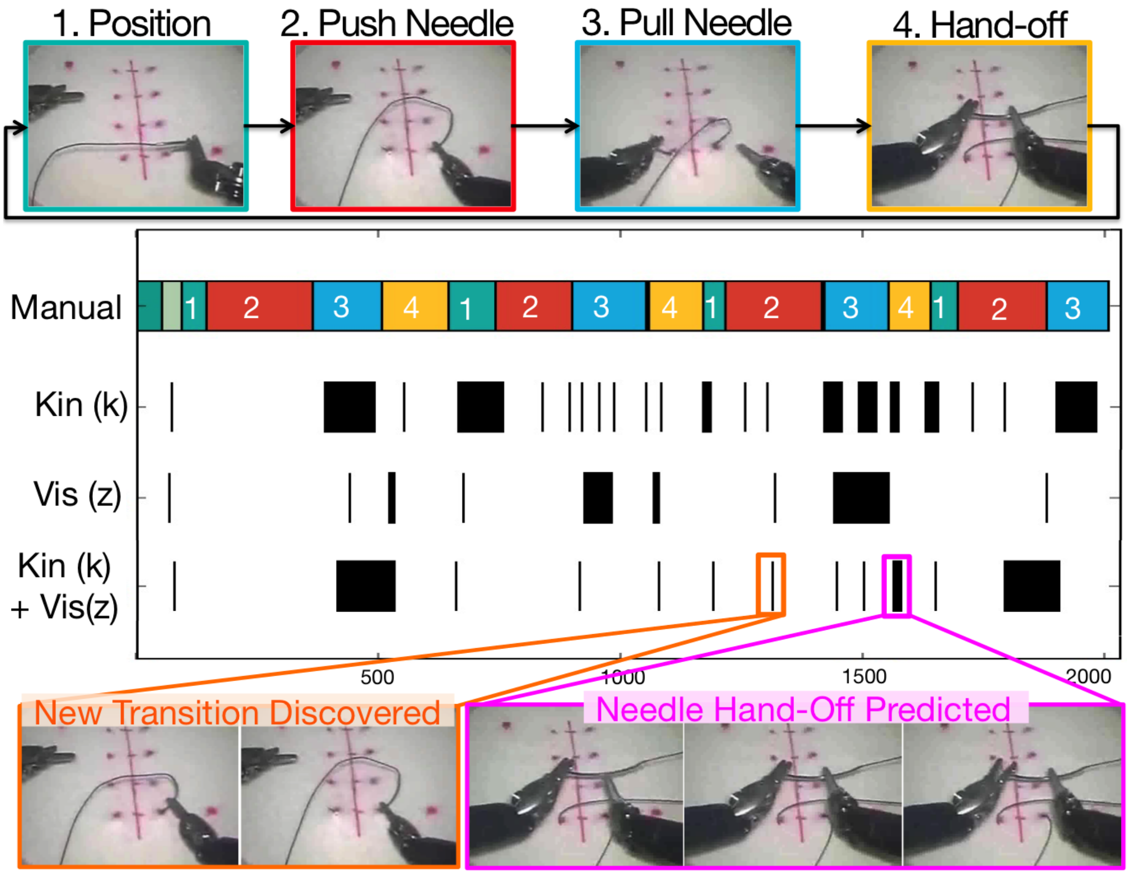


Figure 7.5: The first row shows a manual segmentation of the suturing task in 4 steps: (1) Needle Positioning, (2) Needle Pushing, (3) Pulling Needle, (4) Hand-off. TSC-DL extracts many of the important transitions without labels and also discovers un-labled transition events.

		Kin	Vid	Kin+Vid
Silhouette Score – Intrinsic Evaluation				
Suturing	E+I+N	0.518±0.008	0.576±0.018	0.733±0.056
	E+I	0.550±0.014	0.548±0.015	0.716±0.046
	E	0.630±0.014	0.515±0.021	0.654±0.065
Needle Passing	E+I+N	0.513±0.007	0.552±0.011	0.557±0.010
	E+I	0.521±0.006	0.536±0.013	0.666±0.067
	E	0.524±0.004	0.609±0.010	0.716±0.097
NMI Score – Extrinsic evaluation against manual labels				
Suturing	E+I+N	0.307 ± 0.045	0.157 ± 0.022	0.625 ± 0.034
	E+I	0.427 ± 0.053	0.166 ± 0.057	0.646 ± 0.039
	E	0.516 ± 0.026	0.266 ± 0.025	0.597 ± 0.096
Needle Passing	E+I+N	0.272 ± 0.035	0.186 ± 0.034	0.385 ± 0.092
	E+I	0.285 ± 0.051	0.150 ± 0.048	0.471 ± 0.023
	E	0.287 ± 0.043	0.222 ± 0.029	0.565 ± 0.037

Table 7.2: Comparison of TSC-DL performance on Suturing and Needle Passing Tasks. We compare the prediction performance by incrementally adding demonstrations from Experts (E), Intermediates (I), and Novices (N) respectively to the dataset.

learned for the suturing task. We speculate that this is due to the multilateral nature of this task. This task uses both arms more than the suturing task, and as a result, there are many visual occlusions for a fixed camera. Important features such as the needle pose and the thread may be obscured at different points during the task. Furthermore, we constructed the state-space using the states of both arms. For such a task, it may be better to segment each of the arms independently.

7.5 Discussion

This chapter explored how task segmentation can be learned from visual state representations extracted from *deep* convolutional neural networks (CNNs) with a new algorithm called TSC-DL. It is surprising to observe that "off-the-shelf" visual filters derived from Deep Learning CNNs trained on non-surgical images can yield valuable features for clustering and segmentation.

However, this required several novel contributions including hierarchical clustering, dimensionality reduction, and temporal clustering. On real datasets, we find that TSC-DL matches the manual annotation with up to 0.806 NMI, and our results also suggest that including kinematics and vision results in increases of up to 0.215 NMI over kinematics alone.

Chapter 8

Conclusion: Review and Open Questions

8.1 Review of Contributions

The main contributions of this dissertation are algorithms and hardware designs that demonstrate that the performance of autonomous systems can be improved by leveraging the interaction between optimization based algorithms and the design of hardware systems. We present five case studies towards **High Dose-Rate Brachytherapy (HDR-BT)** treatment delivery for cancer and Subtask automation in **Robot-assisted minimally invasive surgery (RMIS)**.

In Chapter 2, we present a new approach for High-Dose-Rate Intracavitary Brachytherapy with the use of 3D-printed patient-specific implants. We model the problem of designing channels in the implant to guide radiation sources to the tumor location as a motion planning problem and have proposed the Channel Layout Algorithm based on rapidly-expanding randomized trees (RRT). We evaluate this approach on a simulated case of OB/GYN cervical and vaginal cancer and compare it with standardized ring implant (current practice) and customized implant with linear channels. We find that using a two-parameter coverage quality, customized implants with curved channels can offer an improvement over clinical alternatives.

In Chapter 3, we present a first systematic integration of optimization-based needle and dose planning algorithm with the robot-guided needle implants. The use of robot-assisted implants allows skew line needle configurations that can minimize trauma to sensitive structures such as the penile bulb. Chapter 3 also introduces a new concept of 3D printed customized needle guides and presents the algorithmic method to create these guides. We show that both the robot-assisted implants and customized needle guides can achieve clinical dose requirements without puncturing healthy organs at risk and fare on par with expert clinician performance in controlled experiments.

Chapter 4 quantifies the hardness of a particular prostate cancer case. We model the needle planning problem as a reachability calculation problem with convex polyhedral representations of entry zone, avoidance volume and target volume. This can be formulated as an LP, hence resulting in an exact polynomial time check for existence and calculation of the occluded set in the target volume.

In Chapter 5, we present supervised automation of Multi-Throw Suturing (MTS), one of the tasks from Fundamental Skills of Robotic Surgery (FSRS). Our method involves a combination of sequential convex programming, for curvature constrained motion planning, and new hardware concept, Surgical Needle Angular Positioner (SNAP), designed to align and hold the needle in a known orientation. Evaluation of the system with closed loop needle tracking resulted in a 86.3% completion rate for individual suture throws performed at 30% of the average manual speed. Further, we also observe that SNAPimproves repeatability in needle grasping by 10× and enables suturing automation with low-fidelity feedback.

Chapter 6 addresses the limitation of task-specific supervised automation with finite state machines. This dissertation develops novel unsupervised segmentation algorithm, Transition State Clustering (TSC) to recover latent task-structure. We model each input demonstration trajectory as a realization of a switched linear dynamical system, and then cluster them into spatially and temporally similar transition events (i.e. switches in linear regime). TSC uses a hierarchical nonparametric Bayesian model to identify the transition events without specifying the number of segments *a priori*. We evaluate TSC on synthetic for comparison with five alternate trajectory algorithms. Our experimental results suggest that TSC recovers this ground truth with greater accuracy than the alternatives, especially under low-frequency process noise. Further, on the real surgical dataset, TSC finds 83% of the needle passing transitions and 73% of the suturing transitions found by human experts.

Chapter 7 extends the TSC algorithm by with Deep Learning that leverages unannotated video along with kinematic data for task-level segmentation, and finds regions of the visual feature space that correlate with transition events. We use features constructed from layers of image classification Deep Convolutional Neural Networks (CNNs). We observe that TSC-DL output matches human annotations with up to 0.8 Normalized Mutual Information score and the use of both kinematics and visual features results in an increase of up to 0.215 NMI over just kinematics features.

8.2 Open Questions

This dissertation has demonstrated feasibility new design concepts and algorithms on real robots and datasets, for both brachytherapy delivery and robot-assisted surgery. Along the way to development of this dissertation, we have also discovered exciting new questions for further exploration: new applications of presented methods, and algorithmic improvements for open problems.

Intracavitary Brachytherapy

Customized implants for intracavitary brachytherapy treatment allows (a) placement of radioactive sources closer to tumors, (b) more options for dose planning optimization that can reduce the dose of healthy organs-at-risk and (c) dose conformity to tumor volumes.

In addition to exploring placing interstitial catheters in implants as discussed in Chapter 2 (Section 2.5), the application of the proposed concept can be extended beyond gynecological and oral tumors to other anatomical intracavitary locations and various treatment modalities. One example is the hard-to-access Nasopharyngeal carcinoma (NPC) [182]. Due to the anatomical

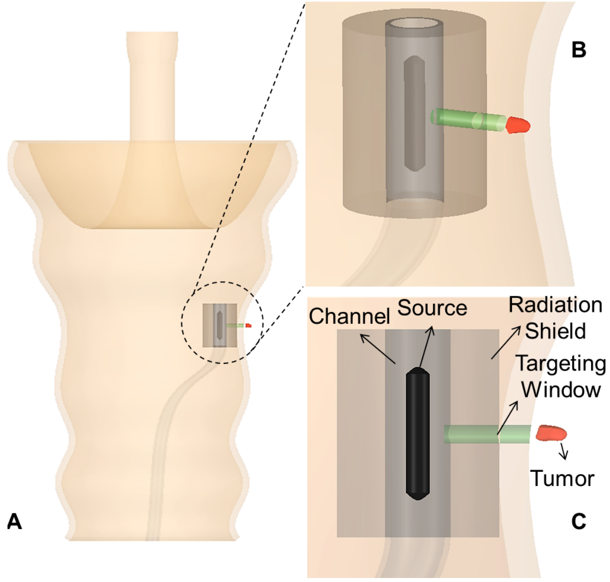


Figure 8.1: Conceptual illustration of how lead shielding could be incorporated into the implant as it is now possible to include multiple materials during 3D printing fabrication. (A) illustrates a channel proximal to a small tumor shown in red. (B,C) are close-up views of the co-axial source, channel, and lead shielding, the latter with a small cylindrical void that serves as a "targeting window" to allow radiation to be emitted toward the tumor while shielding nearby healthy tissue.

position of NPC and its tendency to present with cervical lymph node metastases, it is not amenable to surgery for local control. During localized radiation therapy in NPC, eyes, brain, brain stem, spinal cord, and upper soft-palate need shielding as much as possible. One exciting extension can be facilitated by innovations in 3D printing. As illustrated in Figure 8.1, radiation shielding materials like lead and tungsten could be printed along with the implant. This has the potential to shield healthy tissue and direct radiation to small tumor targets. The research questions here are the feasibility of fabrication of implants with shielding and optimization models for potential gains in dosimetry.

Interstitial Brachytherapy

We noted in Chapter 3 that the skew line needle configuration implants with robotic-assistance and customized needle guides are feasible. However, further system integration is needed to reduce registration errors and clinical evaluations

Similarly, Chapter 4 discussed reachability analysis in 2D. In future work, we will perform additional experiments with more complex anatomy, for example, enlarged prostates where it may be difficult to avoid pubic arch interference and to treat cancers in other organs.

Finally, the exploration of enhancement of planning algorithms, such as NPIP and IPIP, with higher-resolution sampling, and deployment in cloud computing may make it feasible to compute plans that are more robust to uncertainty in anatomy and needle motion.

Suturing Automation in Robot-Assisted Surgery

One of the current limitations of the system is the unpredictability of suture thread. Trajectory planning with collision avoidance and suture thread tracking would improve performance significantly. The thread used in our experiments is difficult to track with regular visual feedback methods and also has high friction with tissue phantoms. We will explore the use anatomically accurate tissue phantoms such as animal tissue to better approximate clinical settings. In fact, animal tissue has natural lubrication might lower friction and consequently ease manipulation at the cost of making sensing and registration harder.

Furthermore, the lack of force sensing in the gripper hinders the ability to pull the suture thread to correct tension without tissue damage. Six-axis force sensing while manipulation is yet to be achieved. Such force feedback would also assist in active thread management. We have made progress towards adding uni-axial tactile feedback to enable tumor localization in soft tissue. We have designed a low-cost palpation probe for Da Vinci manipulators [183] and demonstrated that it can be used to perform tumor retraction, a common multi-step surgical procedure to locate, expose, and debride a subcutaneous tumor and seal the resulting wound with adhesive [133].

Aside from the traditional robotic platforms, there is a growing emergence of technological innovations that offer all the advantages of the robotic degrees of freedom (EndoWrist) and manipulation but using, instead, a standard laparoscopic approach and instrumentation but at a much-reduced capital outlay (e.g., Dexterite Surgical, Transentrix). These systems can be allied with 3D laparoscopic technology, which is widely available in many hospital operating rooms Oŕeilly [184]. It would be of great import to study how lessons learned from automation of subtasks on the da Vinci system can be translated to new systems with different kinematics and sensing modalities.

Unsupervised Trajectory Segmentation

The Transition State Clustering algorithm presented in Chapter 6 is an efficient method for discovering regime changes under the linear dynamics assumption. However, it is at times not accurate because of the transitions output by TSC may not align with semantically meaningful segments.

Additionally, an exploration of deep learning models such as Autoencoders and Recurrent Networks can be used to segment data without linearity assumptions. For these experiments in Chapter 7, we used “off-the-shelf" pre-trained deep learning architectures trained on large image libraries that do not include surgical images. We intend to investigate if the performance improves when we train the CNNs with surgical images.

We will also explore extraction of consistent structure across inconsistent demonstrations. We find that some surgical demonstrations have loops, i.e., repetitive motions where the surgeon repeats a subtask until success. If the input data contains sub-optimal demonstrations with failures and resets, the algorithm can have difficulty in identifying the correct latent structure. Consolidating these motions into a single primitive is an important priority for robot learning.

Segmentation is the first step in a broader robot learning pipeline, and we are actively exploring using segmentation to construct rewards for Reinforcement Learning. We have attempted to address some of these limitations in an extension of this work in [185, 186]. The new proposed framework, Sequential Windowed Inverse Reinforcement Learning (SWIRL) is a new formalism that builds on the TSC algorithm. SWIRL is a three phase algorithm to learn sequential robot tasks, where a task is modeled by an MDP with a sequence of reward functions. SWIRL relaxes the local linearity assumption using kernelization and introduces a state-space augmentation to enforce sequential dependencies using binary indicators of the previously completed segments.

Let D be a set of demonstration trajectories $\{d_1, ..., d_N\}$ of a task with a delayed reward. Given a sequence for which we require a policy, SWIRL can be described in terms of three sub-algorithms:
Inputs: Demonstrations D, Dynamics (Optional) P

1. *Sequence Learning:* Given D, SWIRL segments the task into k sub-tasks whose start and end are defined by arrival at the sub-goals $G = [\rho_1, ..., \rho_k]$.
2. *Reward Learning:* Given G and D, SWIRL associates a local reward function with the segment resulting in a sequence of rewards $\mathbf{R}_{seq}$.
3. *Policy Learning:* Given $\mathbf{R}_{seq}$ and G, SWIRL applies reinforcement learning for I iterations to learn a policy for the task π.

Please refer to Krishnan et al. [186] for more details on SWIRL. SWIRL is a step towards the larger problem of learning from demonstrations. And in the future, we should explore the incorporation of more complex transition conditions and allow for sub-optimal demonstrations. Another avenue for future work is modeling complex tasks as hierarchies of MDPs, namely, tasks composed of multiple MDPs that switch upon certain states with switching dynamics modeled as another MDP. We are also interested in exploring the connections between TSC and other time-series models such as Derivative Dynamic Time Warping which aligns the derivative of two signals.

Final Words

This dissertation is a step towards bringing automation through new design and optimization to two healthcare applications: radiation therapy and surgical robotics. These results have introduced new paradigm of automation underlining the importance of interdependence of optimization based algorithms and the design of hardware systems in performance of an autonomous system. While this is only one of the ways to analyze these problems, interesting questions arise when machine learning methods are jointly used to optimize for design and planning. And I hope that this dissertation drives application of this paradigm by researchers in robotics and automation to larger open problems in robot autonomy and healthcare applications such as surgery and cancer treatment.

Bibliography

[1] Sherry L Murphy, Jiaquan Xu, and Kenneth D Kochanek. “Deaths: final data for 2010”. In: *National vital statistics reports* 61.4 (2013) (cit. on p. 1).

[2] American Medical Association et al. *Physician Characteristics and Distribution in the US 2013*. American Medical Association, 2013 (cit. on p. 1).

[3] Jack Valentin et al. “Prevention of High-dose-rate Brachytherapy Accidents”. In: *Annals of the ICRP* 35.2 (2005), p. 1 (cit. on pp. 1, 9).

[4] Animesh Garg, Sachin Patil, Timmy Siauw, J Adam M Cunha, I Hsu, Pieter Abbeel, Jean Pouliot, and Ken Goldberg. “An Algorithm for Computing Customized 3D Printed Implants with Curvature Constrained Channels for Enhancing Intracavitary Brachytherapy Radiation Delivery”. In: *Proc. IEEE Int. Conf. Automation Science and Engg. (CASE)*. Aug. 2013, pp. 466–473 (cit. on pp. 2, 4).

[5] Animesh Garg, Timmy Siauw, Dmitry Berenson, J Adam M Cunha, I-C Hsu, Jean Pouliot, Dan Stoianovici, and Ken Goldberg. “Robot-Guided Open-Loop Insertion of Skew-Line Needle Arrangements for High Dose Rate Brachytherapy”. In: *IEEE Trans. on Automation Science and Engineering (TASE)* 10.4 (Oct. 2013), pp. 948–956 (cit. on pp. 2, 5, 41).

[6] Intuitive Surgical. “Annual report 2015”. In: *URL* `http://phx.corporate-ir.net/phoenix.zhtml?c=122359&p=irol-irhome` (2015) (cit. on pp. 2, 55).

[7] Ted A Skolarus, Yun Zhang, and Brent K Hollenbeck. “Robotic surgery in urologic oncology: gathering the evidence”. In: *Expert review of pharmacoeconomics & outcomes research* 10.4 (2010), pp. 421–432 (cit. on pp. 3, 55).

[8] Ruben Veldkamp, Esther Kuhry, WC Hop, J Jeekel, G Kazemier, H Jaap Bonjer, Eva Haglind, L Pahlman, Miguel A Cuesta, Simon Msika, et al. “Laparoscopic surgery versus open surgery for colon cancer: short-term outcomes of a randomised trial”. In: *Lancet Oncol* 6.7 (2005), pp. 477–484 (cit. on pp. 3, 55).

[9] S. A. Darzi and Y. Munz. “The Impact of Minimally Invasive Surgical Techniques”. In: *Annu Rev Med.* Vol. 55. 2004, pp. 223–237 (cit. on pp. 3, 55, 56).

[10] Brian M Benway, Sam B Bhayani, Craig G Rogers, Lori M Dulabon, Manish N Patel, Michael Lipkin, Agnes J Wang, and Michael D Stifelman. “Robot assisted partial nephrectomy versus laparoscopic partial nephrectomy for renal tumors: a multi-institutional analysis of perioperative outcomes”. In: *The Journal of urology* 182.3 (2009), pp. 866–873 (cit. on p. 3).

[11] Mustapha Daouadi, Amer H Zureikat, Mazen S Zenati, Haroon Choudry, Alan Tsung, David L Bartlett, Steven J Hughes, Ken K Lee, A James Moser, and Herbert J Zeh. “Robot-assisted minimally invasive distal pancreatectomy is superior to the laparoscopic technique”. In: *Annals of surgery* 257.1 (2013), pp. 128–132 (cit. on p. 3).

[12] John F Boggess, Paola A Gehrig, Leigh Cantrell, Aaron Shafer, Mildred Ridgway, Elizabeth N Skinner, and Wesley C Fowler. “A comparative study of 3 surgical methods for hysterectomy with staging for endometrial cancer: robotic assistance, laparoscopy, laparotomy”. In: *American journal of obstetrics and gynecology* 199.4 (2008), 360–e1 (cit. on pp. 3, 55).

[13] Ehab E Barakat, Mohamed A Bedaiwy, Stephen Zimberg, Benjamin Nutter, Mohsen Nosseir, and Tommaso Falcone. “Robotic-assisted, laparoscopic, and abdominal myomectomy: a comparison of surgical outcomes”. In: *Obstetrics & Gynecology* 117.2, Part 1 (2011), pp. 256–266 (cit. on p. 3).

[14] Siddarth Sen*, Animesh Garg*, David Gealy, Stephen McKinley, Yiming Jen, and Ken Goldberg (*denotes equal contribution). “Autonomous Multiple-Throw Multilateral Surgical Suturing with a Mechanical Needle Guide and Optimization based Needle Planning”. In: *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*. 2016 (cit. on pp. 4, 6).

[15] Sanjay Krishnan*, Animesh Garg*, Sachin Patil, Colin Lea, Gregory Hager, Pieter Abbeel, and Ken Goldberg (*denotes equal contribution). “Transition State Clustering: Unsupervised Surgical Trajectory Segmentation For Robot Learning”. In: *Int. Symp. on Robotics Research (ISRR)*. Springer STAR. 2015 (cit. on pp. 4, 7, 57, 94, 96).

[16] Katherine Mellis, Timmy Siauw, Atchar Sudhyadhom, Rajni Sethi, I-Chow Hsu, Jean Pouliot, Animesh Garg, and Ken Goldberg. “Material Evaluation of PC-ISO for Customized, 3D Printed, Gynecologic 192Ir HDR Brachytherapy Applicators”. In: *Journal of Applied Clinical Medical Physics (JACMP)* (2014) (cit. on p. 5).

[17] Animesh Garg, Timmy Siauw, Dimity Berenson, Adam Cunha, I-Chow Hsu, Jean Pouliot, Dan Stoianovici, and Ken Goldberg. “Initial experiments toward automated robotic implantation of skew-line needle arrangements for HDR brachytherapy”. In: *Proc. IEEE Int. Conf. Automation Science and Engg. (CASE)*. Aug. 2012, pp. 26–33 (cit. on pp. 5, 25).

[18] Timmy Siauw, J. Adam M. Cunha, Animesh Garg, Ken Goldberg, I Hsu, and Jean Pouliot. “Customized Needle Guides for Inserting Non-Parallel Needle Arrangements in Prostate HDR Brachytherapy: A Phantom Study”. In: *Brachytherapy* 13 (2014) (cit. on pp. 5, 41).

[19] Animesh Garg, Timmy Siauw, Guang Yang, Sachin Patil, J Adam M Cunha, I-Chow Hsu, Jean Pouliot, Alper Atamtürk, and Ken Goldberg. “Exact Reachability Analysis for Planning Skew-Line Needle Arrangements for Automated Brachytherapy”. In: *Proc. IEEE Int. Conf. Automation Science and Engg. (CASE)*. 2014 (cit. on p. 6).

[20] Yixin Gao, S Swaroop Vedula, Carol E Reiley, Narges Ahmidi, Balakrishnan Varadarajan, Henry C Lin, Lingling Tao, Luca Zappella, Benjamın Béjar, David D Yuh, Chi Chen, Rene Vidal, Sanjeev Khudanpur, and Greg D. Hager. “The JHU-ISI Gesture and Skill Assessment Dataset (JIGSAWS): A Surgical Activity Working Set for Human Motion Modeling”. In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. 2014 (cit. on pp. 6, 56, 61, 66–69, 71, 89, 94, 104, 105).

[21] Animesh Garg*, Sanjay Krishnan*, Adithyavairavan Murali, Trevor Darrell, Pieter Abbeel, and Ken Goldberg (*denotes equal contribution). "On Visual Feature Representations for Transition State Learning in Robotic Task Demonstrations". In: *NIPS 2015 Workshop on Feature Extraction* (cit. on p. 7).

[22] Adithyavairavan Murali*, Animesh Garg*, Sanjay Krishnan*, Florian Pokorny, Pieter Abbeel, Trevor Darrell, and Ken Goldberg (*denotes equal contribution). "TSC-DL: Unsupervised Trajectory Segmentation of Multi-Modal Surgical Demonstrations with Deep Learning". In: *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*. 2016 (cit. on p. 7).

[23] K Wallner, J C Blasko, and M Datolli. *Prostate Brachytherapy Made Complicated*. Seattle, WA: Smart Medicine Press, 2001 (cit. on p. 9).

[24] Nicolas Magné, Cyrus Chargari, Nicholas SanFilippo, Taha Messai, Alain Gerbaulet, and Christine Haie-Meder. "Technical aspects and perspectives of the vaginal mold applicator for brachytherapy of gynecologic malignancies". In: *Brachytherapy* 9.3 (2010), pp. 274–277 (cit. on pp. 9–11, 18).

[25] Luis Delclos, Gilbert H Fletcher, Edwin Bailey Moore, and Vincent A Sampiere. "Minicolpostats, dome cylinders, other additions and improvements of the Fletcher-Suit afterloadable system: Indications and limitations of their use". In: *International Journal of Radiation Oncology* Biology* Physics* 6.9 (1980), pp. 1195–1206 (cit. on p. 10).

[26] M Bernstein, K J Mehta, R Yaparpalvi, H Kuo, and S Kalnicki. "Results of the Hybrid Interstitial-Intracavitary Utrecht Aapplicator for cervical cancer in an Outpatient setting". In: *Radiotherapy and Oncology* 103 (2012), S116 (cit. on p. 10).

[27] Johannes C A Dimopoulos, Christian Kirisits, Primoz Petric, Petra Georg, Stefan Lang, Daniel Berger, and Richard Pötter. "The Vienna applicator for combined intracavitary and interstitial brachytherapy of cervical cancer: Clinical feasibility and preliminary results". In: *International Journal of Radiation Oncology*Biology*Physics* 66.1 (2006), pp. 83–90 (cit. on p. 10).

[28] Paul F. Jacobs and David T. Reid. *Rapid prototyping and manufacturing: Fundamentals of stereolithography*. Society of Manufacturing Engineers in cooperation with the Computer and Automated Systems Association of SME, 1992 (cit. on p. 11).

[29] Hod Lipson and Melba Kurman. *Fabricated: The New World of 3D Printing*. Wiley, 2013 (cit. on p. 11).

[30] Ferry P W Melchels, Jan Feijen, and Dirk W Grijpma. "A review on stereolithography and its applications in biomedical engineering." In: *Biomaterials* 31.24 (Aug. 2010), pp. 6121–30 (cit. on p. 11).

[31] Vincent Duindam, Ron Alterovitz, and Ken Goldberg. "Motion planning for steerable needles in 3D environments with obstacles using rapidly-exploring Random Trees and backchaining". In: *IEEE International Conference on Automation Science and Engineering*. Aug. 2008, pp. 41–46 (cit. on p. 11).

[32] Sachin Patil and Ron Alterovitz. "Interactive Motion Planning for Steerable Needles in 3D Environments with Obstacles." In: *IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics* (Jan. 2010), pp. 893–899 (cit. on pp. 11, 12, 14–16).

[33] Noah J. Cowan, Ken Goldberg, Gregory S. Chirikjian, Gabor Fichtinger, Ron Alterovitz, Kyle B. Reed, Vinutha Kallem, Wooram Park, Sarthak Misra, and Allison M. Okamura. "Robotic Needle Steering: Design, Modeling, Planning, and Image Guidance". In: *Surgical Robotics: System Applications and Visions*. Springer, 2011. Chap. 23, pp. 557–582 (cit. on pp. 11, 12).

[34] Myung Hwangbo, James Kuffner, and Takeo Kanade. "Efficient Two-phase 3D Motion Planning for Small Fixed-wing UAVs". In: *IEEE International Conference on Robotics and Automation*. Apr. 2007, pp. 1035–1041 (cit. on p. 11).

[35] J Borg and DWO Rogers. "Monte Carlo calculations of photon spectra in air from 192Ir sources". In: *National Research Council Report PIRS-629r, Ontario, Canada* (1999) (cit. on pp. 11, 12).

[36] S. M. LaValle. *Planning Algorithms*. Available at http://planning.cs.uiuc.edu. Cambridge, U.K.: Cambridge University Press, 2006 (cit. on pp. 12, 14).

[37] Esther M Arkin, Sándor P Fekete, Joondong Kim, Joseph SB Mitchell, Girishkumar R Sabhnani, and Jingyu Zou. "The Pencil Packing Problem". In: (2009) (cit. on p. 13).

[38] G.J.A. van den Bergen. *Collision detection in interactive 3D environments*. Morgan Kaufmann, 2004 (cit. on p. 16).

[39] K. Wallner, J.C. Blasko, and M. Dattoli. *Prostate brachytherapy made complicated*. SmartMedicine Press, 2001 (cit. on p. 22).

[40] S.K. Kang, R.H. Chou, R.K. Dodge, R.W. Clough, H.S.L. Kang, M.G. Bowen, B.A. Steffey, S.K. Das, S.M. Zhou, A.W. Whitehurst, et al. "Acute urinary toxicity following transperineal prostate brachytherapy using a modified Quimby loading method". In: *International Journal of Radiation Oncology* Biology* Physics* 50.4 (2001), pp. 937–945 (cit. on p. 22).

[41] Libni Eapen, Cathy Kayser, Yves Deshaies, Gad Perry, Christopher Morash, Joanna E Cygler, David Wilkins, Simone Dahrouge, et al. "Correlating the degree of needle trauma during prostate brachytherapy and the development of acute urinary toxicity". In: *International Journal of Radiation Oncology* Biology* Physics* 59.5 (2004), pp. 1392–1394 (cit. on p. 22).

[42] C. Vargas, M. Ghilezan, M. Hollander, G. Gustafson, H. Korman, J. Gonzalez, and A. Martinez. "A new model using number of needles and androgen deprivation to predict chronic urinary toxicity for high or low dose rate prostate brachytherapy". In: *The Journal of Urology* 174.3 (2005) (cit. on p. 22).

[43] R.M. Munarriz, Q.R. Yan, A. Nehra, D. Udelson, and I. Goldstein. "Blunt trauma: the pathophysiology of hemodynamic injury leading to erectile dysfunction". In: *The Journal of Urology* 153.6 (1995), pp. 1831–1840 (cit. on p. 22).

[44] B.M. Fisch, B. Pickett, V. Weinberg, and M. Roach. "Dose of radiation received by the bulb of the penis correlates with risk of impotence after three-dimensional conformal radiotherapy for prostate cancer". In: *Urology* 57.5 (2001), pp. 955–959 (cit. on p. 22).

[45] G.S. Merrick, K. Wallner, W.M. Butler, R.W. Galbreath, J.H. Lief, and M.L. Benson. "A comparison of radiation dose to the bulb of the penis in men with and without prostate brachytherapy-induced erectile dysfunction". In: *International Journal of Radiation Oncology* Biology* Physics* 50.3 (2001), pp. 597–604 (cit. on p. 22).

[46] JAM Cunha, IC Hsu, and J. Pouliot. “Dosimetric equivalence of nonstandard HDR brachytherapy catheter patterns”. In: *Medical Physics* 36 (2009), p. 233 (cit. on pp. 22, 43).

[47] Ashutosh Tewari. *Prostate Cancer: A Comprehensive Perspective: A Comprehensive Perspective*. Springer, 2013 (cit. on pp. 23, 43).

[48] A Graham Macdonald, Mira Keyes, Alexandra Kruk, Graeme Duncan, Veronika Moravan, and W James Morris. “Predictive factors for erectile dysfunction in men with prostate cancer after brachytherapy: Is dose to the penile bulb important?” In: *International Journal of Radiation Oncology* Biology* Physics* 63.1 (2005), pp. 155–163 (cit. on pp. 23, 43).

[49] Y. Kim, IC Hsu, and J. Pouliot. “Measurement of craniocaudal catheter displacement between fractions in computed tomography-based high dose rate brachytherapy of prostate cancer”. In: *J Appl Clin Med Phys* 8.4 (2007), pp. 2415–2415 (cit. on pp. 23, 31, 34, 44).

[50] D Stoianovici, K Cleary, A Patriciu, D Mazilu, A Stanimir, N Craciunoiu, V Watson, and L Kavoussi. “AcuBot: a robot for radiological interventions”. In: *IEEE Transactions on Robotics and Automation* 19.5 (2003), pp. 927–930 (cit. on pp. 23, 25, 27).

[51] Timmy Siauw, Adam Cunha, Alper Atamtürk, I-Chow Hsu, Jean Pouliot, and Ken Goldberg. “IPIP: A new approach to inverse planning for HDR brachytherapy by directly optimizing dosimetric indices”. In: *Medical Physics* 38.7 (2011), pp. 4045–4051 (cit. on pp. 23, 24, 26, 27, 36, 41, 44, 49).

[52] T. Siauw, A. Cunha, D. Berenson, A. Atamtürk, I.C. Hsu, K. Goldberg, and J. Pouliot. “NPIP: A skew line needle configuration optimization system for HDR brachytherapy”. In: *Medical physics* 39.7 (2012), p. 4339 (cit. on pp. 23, 26, 36, 41, 43, 44, 53).

[53] Carl Salembier and Peter Hoskin. “Prostate Brachytherapy: High Dose Rate”. English. In: *Prostate Cancer: A Comprehensive Perspective*. Ed. by Ashutosh Tewari. Springer London, 2013, pp. 739–748 (cit. on pp. 23, 24, 43, 44).

[54] Y. Yu, JBY Zhang, R.A. Brasacchio, P.G. Okunieff, D.J. Rubens, J.G. Strang, A. Soni, and E.M. Messing. “Automated treatment planning engine for prostate seed implant brachytherapy”. In: *International Journal of Radiation Oncology* Biology* Physics* 43.3 (1999), pp. 647–652 (cit. on p. 24).

[55] M. Lahanas, D. Baltas, and N. Zamboglou. “Anatomy-based three-dimensional dose optimization in brachytherapy using multiobjective genetic algorithms”. In: *Medical Physics* 26 (1999), p. 1904 (cit. on p. 24).

[56] Etienne Lessard and Jean Pouliot. “Inverse planning anatomy-based dose optimization for HDR-brachytherapy of the prostate using fast simulated annealing algorithm and dedicated objective function”. In: *Medical Physics* 28 (2001), p. 773 (cit. on p. 24).

[57] R. Alterovitz, E. Lessard, J. Pouliot, I.C.J. Hsu, J.F. O’Brien, and K. Goldberg. “Optimization of HDR brachytherapy dose distributions using linear programming with penalty costs”. In: *Medical physics* 33 (2006), p. 4012 (cit. on p. 24).

[58] A. Karabis, P. Belotti, and D. Baltas. “Optimization of catheter position and dwell time in prostate hdr brachytherapy using hipo and linear programming”. In: *World Congress on Medical Physics and Biomedical Engineering, September 7-12, 2009, Munich, Germany*. Springer. 2009, pp. 612–615 (cit. on p. 24).

[59] A Majewicz, S Marra, M van Vledder, M Lin, M Choti, D Song, and A Okamura. “Behavior of Tip-Steerable Needles in ex vivo and in vivo Tissue.” In: *IEEE Transactions on bio-medical engineering* (2012) (cit. on p. 24).

[60] Ron Alterovitz, Andrew Lim, Ken Goldberg, Gregory S Chirikjian, and Allison M Okamura. “Steering flexible needles under Markov motion uncertainty”. In: *IEEE/RSJ International Conference on-Intelligent Robots and Systems, 2005.(IROS 2005)*. IEEE. 2005, pp. 1570–1575 (cit. on p. 24).

[61] Vincent Duindam, Jijie Xu, Ron Alterovitz, Shankar Sastry, and Ken Goldberg. “Three-dimensional motion planning algorithms for steerable needles using inverse kinematics”. In: *The International Journal of Robotics Research* 29.7 (2010), pp. 789–800 (cit. on p. 24).

[62] Jur Van Den Berg, Sachin Patil, Ron Alterovitz, Pieter Abbeel, and Ken Goldberg. “LQG-based planning, sensing, and control of steerable needles”. In: *Algorithmic Foundations of Robotics IX*. Springer, 2011, pp. 373–389 (cit. on p. 24).

[63] Jijie Xu, Vincent Duindam, R. Alterovitz, J. Pouliot, J.A.M. Cunha, I-Chow Hsu, and K. Goldberg. “Planning fireworks trajectories for steerable medical needles to reduce patient trauma”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009. (IROS)*. 2009, pp. 4517–4522 (cit. on pp. 24, 44).

[64] Ron Alterovitz and Ken Goldberg. “Motion Planning in Medicine: Optimization and Simulation Algorithms for Image-Guided Procedures”. In: *Springer Tracts in Advanced Robotics* 50 (2008) (cit. on p. 24).

[65] K.B. Reed, A. Majewicz, V. Kallem, R. Alterovitz, K. Goldberg, N.J. Cowan, and A.M. Okamura. “Robot-Assisted Needle Steering”. In: *Robotics Automation Magazine, IEEE* 18 (2011), pp. 35–46 (cit. on p. 24).

[66] C.M. Schneider, A.M. Okamura, and G. Fichtinger. “A robotic system for transrectal needle insertion into the prostate with integrated ultrasound”. In: *IEEE International Conference on Robotics and Automation, 2004*. Vol. 1. IEEE. 2004, pp. 365–370 (cit. on p. 25).

[67] L. Phee, D. Xiao, J. Yuen, C.F. Chan, H. Ho, C.H. Thng, C. Cheng, and W.S. Ng. “Ultrasound guided robotic system for transperineal biopsy of the prostate”. In: *IEEE International Conference on Robotics and Automation, 2005*. IEEE. 2005, pp. 1315–1320 (cit. on p. 25).

[68] G. Fichtinger, E.C. Burdette, A. Tanacs, A. Patriciu, D. Mazilu, L.L. Whitcomb, and D. Stoianovici. “Robotically assisted prostate brachytherapy with transrectal ultrasound guidance-Phantom experiments”. In: *Brachytherapy* 5.1 (2006), pp. 14–26 (cit. on pp. 25, 28).

[69] N. Hungr, J. Troccaz, N. Zemiti, and N. Tripodi. “Design of an ultrasound-guided robotic brachytherapy needle-insertion system”. In: *IEEE International Conference on Engineering in Medicine and Biology Society, 2009*. IEEE. 2009, pp. 250–253 (cit. on p. 25).

[70] Y. Yu, TK Podder, YD Zhang, WS Ng, V. Misic, J. Sherman, D. Fuller, DJ Rubens, JG Strang, RA Brasacchio, et al. “Robotic system for prostate brachytherapy”. In: *Computer Aided Surgery* 12.6 (2007), pp. 366–370 (cit. on p. 25).

[71] A. Patriciu, D. Petrisor, M. Muntener, D. Mazilu, M. Schär, and D. Stoianovici. “Automatic brachytherapy seed placement under MRI guidance”. In: *IEEE Transactions on bio-medical engineering* 54.8 (2007), p. 1499 (cit. on p. 25).

[72] A.L. Trejos, R. Patel, and R. Malthaner. “A device for robot-assisted minimally-invasive lung brachytherapy”. In: *IEEE International Conference on Robotics and Automation, 2006*. IEEE. 2006, pp. 4187–4192 (cit. on p. 25).

[73] S.E. Salcudean, T.D. Prananta, W.J. Morris, and I. Spadinger. “A robotic needle guide for prostate brachytherapy”. In: *IEEE International Conference on Robotics and Automation, 2008*. IEEE. 2008, pp. 2975–2981 (cit. on p. 25).

[74] G. Fichtinger, J.P. Fiene, C.W. Kennedy, G. Kronreif, I. Iordachita, D.Y. Song, E.C. Burdette, and P. Kazanzides. “Robotic assistance for ultrasound-guided prostate brachytherapy”. In: *Medical image analysis* 12.5 (2008), pp. 535–545 (cit. on p. 25).

[75] Jitendra N. Roy, Kent E. Wallner, Lowell L. Anderson, and C.Clifton Ling. “CT-based optimized planning for transperineal prostate implant with customized template”. In: *International Journal of Radiation Oncology*Biology*Physics* 21.2 (July 1991), pp. 483–489 (cit. on p. 25).

[76] R. Pollock, P. Mozer, T.J. Guzzo, J. Marx, B. Matlaga, D. Petrisor, B. Vigaru, S. Badaan, D. Stoianovici, and M.E. Allaf. “Prospects in percutaneous ablative targeting: comparison of a computer-assisted navigation system and the AcuBot robotic system”. In: *Journal of Endourology* 24.8 (2010), pp. 1269–1272 (cit. on p. 25).

[77] S. Badaan, D. Petrisor, C. Kim, P. Mozer, D. Mazilu, L. Gruionu, A. Patriciu, K. Cleary, and D. Stoianovici. “Does needle rotation improve lesion targeting?” In: *The International Journal of Medical Robotics and Computer Assisted Surgery* 7.2 (2011), pp. 138–147 (cit. on p. 25).

[78] J.A. Long, N. Hungr, M. Baumann, J.L. Descotes, M. Bolla, J.Y. Giraud, J.J. Rambeaud, and J. Troccaz. “Development of a novel robot for transperineal needle based interventions: focal therapy, brachytherapy and prostate biopsies”. In: *The Journal of Urology* (2012) (cit. on pp. 25, 40).

[79] I. Hsu, K. Bae, K. Shinohara, J. Pouliot, J. Purdy, G. Ibbott, J. Speight, E. Vigneault, R. Ivker, H. Sandler, et al. “Phase II trial of combined high-dose-rate brachytherapy and external beam radiotherapy for adenocarcinoma of the prostate: preliminary results of RTOG 0321”. In: *International Journal of Radiation Oncology* Biology* Physics* 78.3 (2010), pp. 751–758 (cit. on p. 25).

[80] K.S. Arun, T.S. Huang, and S.D. Blostein. “Least-squares fitting of two 3-D point sets”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 5 (1987), pp. 698–700 (cit. on p. 26).

[81] Andriy Myronenko and Xubo Song. “Point set registration: Coherent point drift”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32.12 (2010), pp. 2262–2275 (cit. on pp. 38, 63).

[82] Martin Uecker, Shuo Zhang, Dirk Voit, Alexander Karaus, Klaus-Dietmar Merboldt, and Jens Frahm. “Real-time MRI at a resolution of 20 ms”. In: *NMR in Biomedicine* 23.8 (2010), pp. 986–994 (cit. on p. 40).

[83] Shuo Zhang, Kai Tobias Block, and Jens Frahm. “Magnetic resonance imaging in real time: Advances using radial FLASH”. In: *Journal of Magnetic Resonance Imaging* 31.1 (2010), pp. 101–109 (cit. on p. 40).

[84] Saúl Tovar-Arriaga, Ralf Tita, Jesús Carlos Pedraza-Ortega, Efren Gorrostieta, and Willi A Kalender. “Development of a robotic FD-CT-guided navigation system for needle placement & preliminary accuracy tests”. In: *The International Journal of Medical Robotics and Computer Assisted Surgery* 7.2 (2011), pp. 225–236 (cit. on p. 40).

[85] Wenzhi Ji. "Reconfigurable Fiducial-Integrated Modular Needle Driver for MRI-Guided Percutaneous Interventions". MA thesis. Worcester Polytechnic Institute, 2013 (cit. on p. 40).

[86] Yong-Lae Park, S. Elayaperumal, B. Daniel, Seok Chang Ryu, Mihye Shin, J. Savall, R.J. Black, B. Moslehi, and M.R. Cutkosky. "Real-Time Estimation of 3-D Needle Shape and Deflection for MRI-Guided Interventions". In: *Mechatronics, IEEE/ASME Transactions on* 15.6 (2010), pp. 906–915 (cit. on p. 40).

[87] Reza Seifabadi, Nathan B. J. Cho, Sang-Eun Song, Junichi Tokuda, Nobuhiko Hata, Clare M. Tempany, Gabor Fichtinger, and Iulian Iordachita. "Accuracy study of a robotic system for MRI-guided prostate needle placement". In: *The International Journal of Medical Robotics and Computer Assisted Surgery* (2012) (cit. on p. 40).

[88] N Howlader, AM Noone, M Krapcho, et al. "SEER Cancer Statistics Review, 1975-2010.[Based on the November 2012 SEER data submission, posted to the SEER website, April 2013]". In: *Bethesda, MD: National Cancer Institute* (2013) (cit. on p. 41).

[89] Daniel Scherzer, Michael Wimmer, and Werner Purgathofer. "A Survey of Real-Time Hard Shadow Mapping Methods". In: *Computer Graphics Forum*. Vol. 30. 1. Wiley Online Library. 2011, pp. 169–186 (cit. on p. 43).

[90] CGAL, *Computational Geometry Algorithms Library* (cit. on p. 43).

[91] Erik Sintorn, Ola Olsson, and Ulf Assarsson. "An Efficient Alias-free Shadow Algorithm for Opaque and Transparent Objects Using Per-triangle Shadow Volumes". In: *Proceedings of the 2011 SIGGRAPH Asia Conference*. SA11. Hong Kong, China: ACM, 2011 (cit. on p. 43).

[92] Pascal Lecocq, Pascal Gautron, Jean-Eudes Marvie, and Gael Sourimant. "Sub-pixel shadow mapping". In: *ACM SIGGRAPH 2013 Talks*. ACM. 2013, p. 19 (cit. on p. 43).

[93] E. Fogel and D. Halperin. "Polyhedral Assembly Partitioning With Infinite Translations or The Importance of Being Exact". In: *Automation Science and Engineering, IEEE Transactions on* 10.2 (Apr. 2013), pp. 227–241 (cit. on p. 43).

[94] Jose Luis Alarcon-Herrera, Xiang Chen, and Xuebo Zhang. "Viewpoint Selection for Vision Systems in Industrial Inspection". In: *IEEE Int'l Conference on Robotics and Automation (ICRA)*. 2014 (cit. on p. 43).

[95] Narendra Karmarkar. "A new polynomial-time algorithm for linear programming". In: *Proceedings of the sixteenth annual ACM symposium on Theory of computing*. ACM. 1984, pp. 302–311 (cit. on p. 48).

[96] Hans Raj Tiwary. "On the hardness of computing intersection, union and minkowski sum of polytopes". In: *Discrete & Computational Geometry* 40.3 (2008), pp. 469–479 (cit. on p. 48).

[97] Komei Fukuda, Thomas M Liebling, and Christine Lütolf. "Extended convex hull". In: *Computational Geometry* 20.1 (2001), pp. 13–23 (cit. on p. 48).

[98] M. Herceg, M. Kvasnica, C.N. Jones, and M. Morari. "Multi-Parametric Toolbox 3.0". In: *Proc. of the European Control Conference*. Zürich, Switzerland, July 2013, pp. 502–510 (cit. on p. 49).

[99] Andrew P Stegemann, Kamran Ahmed, Johar R Syed, Shabnam Rehman, Khurshid Ghani, Ricardo Autorino, Mohamed Sharif, Amrith Rao, Yi Shi, Gregory E Wilding, et al. “Fundamental skills of robotic surgery: a multi-institutional randomized controlled trial for validation of a simulation-based curriculum”. In: *Urology* 81.4 (2013), pp. 767–774 (cit. on p. 56).

[100] Julio Hochberg, Kathleen M Meyer, and Michael D Marion. “Suture choice and other methods of skin closure”. In: *Surgical Clinics of North America* 89.3 (2009), pp. 627–641 (cit. on p. 56).

[101] Jon M Burch, Reginald J Franciose, Ernest E Moore, Walter L Biffl, and Patrick J Offner. “Single-layer continuous versus two-layer interrupted intestinal anastomosis: a prospective randomized trial”. In: *Annals of surgery* (2000) (cit. on p. 56).

[102] Colon Cancer Laparoscopic or Open Resection Study Group et al. “Laparoscopic surgery versus open surgery for colon cancer: short-term outcomes of a randomised trial”. In: *The lancet oncology* (2005) (cit. on p. 56).

[103] G.P. Moustris, S.C. Hiridis, K.M. Deliparaschos, and K.M. Konstantinidis. “Evolution of Autonomous and Semi-Autonomous Robotic Surgical Systems: A Review of the Literature”. In: *Int. Journal of Medical Robotics and Computer Assisted Surgery* 4 (2011), pp. 375–392 (cit. on p. 56).

[104] Michael Kranzfelder, Christoph Staub, Adam Fiolka, Armin Schneider, Sonja Gillen, Dirk Wilhelm, Helmut Friess, Alois Knoll, and Hubertus Feussner. “Toward increased autonomy in the surgical OR: needs, requests, and expectations”. In: *Surgical endoscopy* 27.5 (2013), pp. 1681–1688 (cit. on p. 56).

[105] H. Kang and J.T. Wen. “Autonomous Suturing using Minimally Invasive Surgical Robots”. In: *IEEE Int. Conf. on Control Applications*. 2000, pp. 742–747 (cit. on p. 57).

[106] C. Staub, T. Osa, A. Knoll, and R. Bauernschmitt. “Automation of tissue piercing using circular needles and vision guidance for computer aided laparoscopic surgery”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2010 (cit. on p. 57).

[107] M. Cenk Jackson Russell C.and Cavusoglu. “Modeling of needle-tissue interaction forces during surgical suturing”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2012 (cit. on p. 57).

[108] Russell C. Jackson and M. Cenk Cavusoglu. “Needle path planning for autonomous robotic surgical suturing”. In: *IEEE Int. Conf. on Robotics and Automation (ICRA)*. 2013 (cit. on pp. 57, 58).

[109] Hermann Georg Mayer, Faustino J Gomez, Daan Wierstra, Istvan Nagy, Alois Knoll, and Jürgen Schmidhuber. “A System for Robotic Heart Surgery that Learns to Tie Knots Using Recurrent Neural Networks.” In: *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)* (2006) (cit. on p. 57).

[110] J. Van Den Berg, S. Miller, D. Duckworth, H. Hu, A. Wan, X. Fu, K. Goldberg, and P. Abbeel. “Superhuman Performance of Surgical Tasks by Robots using Iterative Learning from Human-Guided Demonstrations”. In: *IEEE Int. Conf. on Robotics and Automation (ICRA)*. 2010, pp. 2074–2081 (cit. on p. 57).

[111] John Schulman, Ankush Gupta, Sibi Venkatesan, Mallory Tayson-Frederick, and Pieter Abbeel. “A Case Study of Trajectory Transfer through Non-Rigid Registration for a Simplified Suturing Scenario”. In: *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*. 2013, pp. 4111–4117 (cit. on p. 57).

[112] N. Padoy and G.D. Hager. “Human-Machine Collaborative Surgery using Learned Models”. In: *IEEE Int. Conf. on Robotics and Automation (ICRA)*. 2011, pp. 5285–5292 (cit. on p. 57).

[113] Carol E Reiley and Gregory D Hager. “Task versus Subtask Surgical Skill Evaluation of Robotic Minimally Invasive Surgery”. In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. 2009, pp. 435–442 (cit. on p. 57).

[114] Scott Niekum, Sarah Osentoski, George Konidaris, Sachin Chitta, Bhaskara Marthi, and Andrew G Barto. “Learning Grounded Finite-State Representations from Unstructured Demonstrations”. In: *Int. Journal of Robotics Research* 34.2 (2015), pp. 131–157 (cit. on pp. 57, 94).

[115] Colin Lea, Gregory D Hager, and René Vidal. “An Improved Model for Segmentation and Recognition of Fine-Grained Activities with Application to Surgical Training Tasks.” In: *IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2015, pp. 1123–1129 (cit. on pp. 57, 69, 71).

[116] B. Kehoe, G. Kahn, J. Mahler, J. Kim, A. Lee, A. Lee, K. Nakagawa, S. Patil, W.D. Boyd, P. Abbeel, and K. Goldberg. “Autonomous Multilateral Debridement with the Raven Surgical Robot”. In: *IEEE Int. Conf. on Robotics and Automation (ICRA)*. 2014, pp. 1432–1439 (cit. on pp. 57, 69).

[117] Adithyavairavan Murali, Siddarth Sen, Ben Kehoe, Animesh Garg, Seth McFarland, Sachin Patil, W Douglas Boyd, Susan Lim, Pieter Abbeel, and Ken Goldberg. “Learning by Observation for Surgical Subtasks: Multilateral Cutting of 3D Viscoelastic and 2D Orthotropic Tissue Phantoms”. In: *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*. 2015, pp. 1202–1209 (cit. on pp. 57, 65, 69, 88).

[118] Stephen McKinley, Siddarth Sen, Animesh Garg, Yiming Jen, David Gealy, Pieter Abbeel, and Ken Goldberg. *Autonomous Tumor Localization and Extraction: Palpation, Incision, Debridement and Adhesive Closure with the da Vinci Research Kit*. Hamlyn Surgical Robotics Conference, London. 2015 (cit. on p. 57).

[119] John Schulman, Yan Duan, Jonathan Ho, Alex Lee, Ibrahim Awwal, Henry Bradlow, Jia Pan, Sachin Patil, Ken Goldberg, and Pieter Abbeel. “Motion planning with sequential convex optimization and convex collision checking”. In: *The International Journal of Robotics Research* (2014) (cit. on p. 57).

[120] Yan Duan, Sachin Patil, John Schulman, Ken Goldberg, and Pieter Abbeel. “Planning locally optimal, curvature-constrained trajectories in 3D using sequential convex optimization”. In: *IEEE International Conf. on Robotics and Automation (ICRA)*. 2014 (cit. on pp. 57, 61).

[121] Sachin Patil, Yan Duan, John Schulman, Ken Goldberg, and Pieter Abbeel. “Gaussian belief space planning with discontinuities in sensing domains”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2014, pp. 6483–6490 (cit. on p. 57).

[122] Azad Shademan, Ryan S Decker, Justin D Opfermann, Simon Leonard, Axel Krieger, and Peter CW Kim. “Supervised autonomous robotic soft tissue surgery”. In: *Science translational medicine* 8.337 (2016), 337ra64–337ra64 (cit. on p. 57).

[123] Florent Nageotte, Christophe Doignon, Michel de Mathelin, Philippe Zanne, and Luc Soler. “Circular needle and needle-holder localization for computer-aided suturing in laparoscopic surgery”. In: vol. 5744, pp. 87–98 (cit. on p. 58).

[124] S Speidel, A Kroehnert, S Bodenstedt, H Kenngott, B Mueller-Stich, and R Dillmann. “Image-based tracking of the suturing needle during laparoscopic interventions”. In: *SPIE Medical Imaging*. 2015 (cit. on p. 58).

[125] Russell C Jackson, Rick Yuan, Der-Lin Chow, Wyatt Newman, and M Cenk Cavusoglu. “Automatic initialization and dynamic tracking of surgical suture threads”. In: *IEEE Int. Conf. Robotics and Automation*. 2015 (cit. on p. 58).

[126] D T Martin, J A Woodard, C J Shurtleff, and A C Yoo. *Articulating needle driver*. US Patent 9,113,861. 2015 (cit. on pp. 58, 62).

[127] S.U. Qureshi, K.M. Rupp, and B. Thompson. *Needle holder with suture filament grasping abilities*. US Patent 5,951,587. Sept. 1999 (cit. on pp. 58, 62).

[128] Fernando De Goes, Siome Goldenstein, and Luiz Velho. “A hierarchical segmentation of articulated bodies”. In: *Computer Graphics Forum*. 2008 (cit. on p. 60).

[129] John Schulman, Jonathan Ho, Alex Lee, Henry Bradlow, Ibrahim Awwal, and Pieter Abbeel. “Finding Locally Optimal, Collision-Free Trajectories with Sequential Convex Optimization”. In: *Robotics:Systems and Science*. 2013 (cit. on p. 61).

[130] 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: *IEEE Int. Conf. on Robotics and Automation (ICRA)*. 2014 (cit. on p. 65).

[131] Carol E Reiley, Erion Plaku, and Gregory D Hager. “Motion generation of robotic surgical tasks: Learning from expert demonstrations”. In: *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*. IEEE. 2010, pp. 967–970 (cit. on p. 69).

[132] Jeffrey Mahler, Sanjay Krishnan, Michael Laskey, Siddarth Sen, Adithyavairavan Murali, Ben Kehoe, Sachin Patil, Jiannan Wang, Mike Franklin, Pieter Abbeel, and Kenneth Y. Goldberg. “Learning accurate kinematic control of cable-driven surgical robots using data cleaning and Gaussian Process Regression”. In: *IEEE International Conference on Automation Science and Engineering, CASE*. 2014, pp. 532–539 (cit. on p. 69).

[133] Stephen McKinley, Animesh Garg, Siddarth Sen, David V. Gealy, Jonathan P. McKinley, Yiming Jen, Menglung Luo, Doug Boyd, and Ken Goldberg. “An Interchangeable Surgical Instrument System with Application to Supervised Automation of Multilateral Tumor Resection.” In: *IEEE Conf. on Automation Science and Engineering (CASE)*. 2016 (cit. on pp. 69, 111).

[134] Henry C. Lin, Izhak Shafran, Todd E. Murphy, Allison M. Okamura, David D. Yuh, and Gregory D. Hager. “Automatic Detection and Segmentation of Robot-Assisted Surgical Motions”. In: *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2005, 8th International Conference, Palm Springs, CA, USA, October 26-29, 2005, Proceedings, Part I*. 2005, pp. 802–810 (cit. on pp. 69, 71).

[135] Balakrishnan Varadarajan, Carol Reiley, Henry Lin, Sanjeev Khudanpur, and Gregory Hager. “Data-derived Models for Segmentation with Application to Surgical Assessment and Training”. In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer, 2009, pp. 426–434 (cit. on pp. 69, 71).

[136] Lingling Tao, Luca Zappella, Gregory D Hager, and René Vidal. “Surgical gesture segmentation and recognition”. In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2013*. Springer, 2013, pp. 339–346 (cit. on pp. 69, 71).

[137] Sylvain Calinon, Florent D’halluin, Eric L Sauser, Darwin G Caldwell, and Aude G Billard. “Learning and reproduction of gestures by imitation”. In: *Robotics & Automation Magazine, IEEE* 17.2 (2010), pp. 44–54 (cit. on pp. 70, 94).

[138] Sang Hyoung Lee, Il Hong Suh, Sylvain Calinon, and Rolf Johansson. “Autonomous framework for segmenting robot trajectories of manipulation task”. In: *Autonomous Robots* 38.2 (2015), pp. 107–141 (cit. on pp. 70, 72, 84).

[139] Volker Krüger, Vadim Tikhanoff, Lorenzo Natale, and Giulio Sandini. “Imitation learning of non-linear point-to-point robot motions using Dirichlet processes”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2012, pp. 2029–2034 (cit. on pp. 70, 72).

[140] Scott Niekum, Sarah Osentoski, George Konidaris, and Andrew G. Barto. “Learning and generalization of complex tasks from unstructured demonstrations”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*. 2012, pp. 5239–5246 (cit. on pp. 70, 72, 84).

[141] Luca Zappella, Benjamin Béjar Haro, Gregory D. Hager, and René Vidal. “Surgical gesture classification from video and kinematic data”. In: *Medical Image Analysis* 17.7 (2013), pp. 732–745 (cit. on pp. 71, 96).

[142] Gwénolé Quellec, Mathieu Lamard, Béatrice Cochener, and Guy Cazuguel. “Real-Time Segmentation and Recognition of Surgical Tasks in Cataract Surgery Videos”. In: *IEEE Trans. Med. Imaging* 33.12 (2014), pp. 2352–2360 (cit. on pp. 71, 96).

[143] Zoubin Ghahramani and Michael I. Jordan. “Supervised learning from incomplete data via an EM approach”. In: *Advances in Neural Information Processing Systems 6, [7th NIPS Conference, Denver, Colorado, USA, 1993]*. 1993, pp. 120–127 (cit. on pp. 72, 84).

[144] Mohammad Khansari-Zadeh and Aude Billard. “Learning stable nonlinear dynamical systems with gaussian mixture models”. In: *Robotics, IEEE Transactions on* 27.5 (2011), pp. 943–957 (cit. on pp. 72, 79).

[145] Teodor Mihai Moldovan, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. “Optimism-driven exploration for nonlinear systems”. In: *IEEE International Conference on Robotics and Automation, ICRA 2015, Seattle, WA, USA, 26-30 May, 2015*. 2015, pp. 3239–3246 (cit. on pp. 72, 79, 98).

[146] Tamim Asfour, Pedram Azad, Florian Gyarfas, and Rüdiger Dillmann. “Imitation Learning of Dual-Arm Manipulation Tasks in Humanoid Robots”. In: *I. J. Humanoid Robotics* 5.2 (2008), pp. 183–202 (cit. on pp. 72, 84).

[147] Sylvain Calinon and Aude Billard. “Stochastic gesture production and recognition model for a humanoid robot”. In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai, Japan, September 28 - October 2, 2004*. 2004, pp. 2769–2774 (cit. on pp. 72, 84).

[148] Volker Kruger, Dennis Herzog, Sanmohan Baby, Ales Ude, and Danica Kragic. “Learning actions from observations”. In: *Robotics & Automation Magazine, IEEE* 17.2 (2010), pp. 30–43 (cit. on pp. 72, 79, 84).

[149] Aleksandar Vakanski, Iraj Mantegh, Andrew Irish, and Farrokh Janabi-Sharifi. “Trajectory Learning for Robot Programming by Demonstration Using Hidden Markov Model and Dynamic Time Warping”. In: *IEEE Trans. Systems, Man, and Cybernetics, Part B* 42.4 (2012), pp. 1039–1052 (cit. on pp. 72, 84).

[150] Ajay Kumar Tanwani and Sylvain Calinon. “Learning Robot Manipulation Tasks with Task-Parameterized Semi-Tied Hidden Semi-Markov Model”. In: (2015) (cit. on p. 72).

[151] Nicolas Padoy, Diana Mateus, Daniel Weinland, M-O Berger, and Nassir Navab. “Workflow monitoring based on 3d motion features”. In: *IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops)*. IEEE. 2009, pp. 585–592 (cit. on p. 72).

[152] Hao Tang, Mark Hasegawa-Johnson, and Thomas S Huang. “Toward robust learning of the Gaussian mixture state emission densities for hidden Markov models”. In: *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*. IEEE. 2010, pp. 5242–5245 (cit. on p. 72).

[153] Dana Kulić and Yoshihiko Nakamura. “Scaffolding on-line segmentation of full body human motion patterns”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems IROS*. IEEE. 2008, pp. 2860–2866 (cit. on p. 72).

[154] Daniel H Grollman and Odest Chadwicke Jenkins. “Incremental learning of subtasks from unsegmented demonstration”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2010, pp. 261–266 (cit. on p. 72).

[155] San Mohan, Volker Krüger, Danica Kragic, and Hedvig Kjellström. “Primitive-Based Action Representation and Recognition”. In: *Advanced Robotics* 25.6-7 (2011), pp. 871–891 (cit. on p. 72).

[156] Emily Fox, Michael I Jordan, Erik B Sudderth, and Alan S Willsky. “Sharing features among dynamical systems with beta processes”. In: *Advances in Neural Information Processing Systems*. 2009, pp. 549–557 (cit. on pp. 72, 84).

[157] Ardavan Saeedi, Matthew Hoffman, Matthew Johnson, and Ryan Adams. “The Segmented iHMM: A Simple, Efficient Hierarchical Infinite HMM”. In: *arXiv preprint arXiv:1602.06349* (2016) (cit. on p. 73).

[158] Guy Rosman, Mikhail Volkov, Dan Feldman, John W Fisher III, and Daniela Rus. “Coresets for k-segmentation of streaming data”. In: *Advances in Neural Information Processing Systems*. 2014, pp. 559–567 (cit. on p. 73).

[159] Cynthia Sung, Dan Feldman, and Daniela Rus. “Trajectory clustering for motion prediction”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2012, pp. 1547–1552 (cit. on pp. 73, 84).

[160] Mikhail Volkov, Guy Rosman, Dan Feldman, John W Fisher, and Daniela Rus. “Coresets for visual summarization with applications to loop closure”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2015, pp. 3638–3645 (cit. on pp. 73, 84).

[161] Brian Kulis and Michael I. Jordan. “Revisiting k-means: New Algorithms via Bayesian Nonparametrics”. In: *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. 2012 (cit. on p. 79).

[162] Sylvain Calinon, Danilo Bruno, and Darwin G Caldwell. “A task-parameterized probabilistic model with minimal intervention control”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2014, pp. 3339–3344 (cit. on p. 79).

[163] Chenxia Wu, Jiemi Zhang, Silvio Savarese, and Ashutosh Saxena. "Watch-n-patch: Unsupervised understanding of actions and relations". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 4362–4370 (cit. on pp. 84, 85, 96).

[164] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. "Human-level control through deep reinforcement learning". In: *Nature* 518.7540 (2015), pp. 529–533 (cit. on p. 94).

[165] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. "Continuous control with deep reinforcement learning". In: *arXiv preprint arXiv:1509.02971* (2015) (cit. on p. 94).

[166] Sergey Levine and Pieter Abbeel. "Learning neural network policies with guided policy search under unknown dynamics". In: *Advances in Neural Information Processing Systems*. 2014, pp. 1071–1079 (cit. on p. 94).

[167] John Schulman, Sergey Levine, Philipp Moritz, Michael I Jordan, and Pieter Abbeel. "Trust region policy optimization". In: *CoRR, abs/1502.05477* (2015) (cit. on p. 94).

[168] Ian Lenz, Honglak Lee, and Ashutosh Saxena. "Deep learning for detecting robotic grasps". In: *The International Journal of Robotics Research* (2015) (cit. on pp. 94–96).

[169] Seung-kook Jun, Madusudanan Sathia Narayanan, Pankaj Singhal, Sudha Garimella, and Venkat Krovi. "Evaluation of robotic minimally invasive surgical skills using motion studies". In: *Journal of robotic surgery* 7.3 (2013), pp. 241–249 (cit. on p. 94).

[170] Stamatia Giannarou, Danail Stoyanov, David Noonan, George Mylonas, Jim Clark, Marco Visentini-Scarzanella, Pete Mountney, and Guang-Zhong Yang. *Hamlyn Centre Laparoscopic / Endoscopic Video Datasets* `http://hamlyn.doc.ic.ac.uk/vision/`. 2015 (cit. on p. 94).

[171] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. "How transferable are features in deep neural networks?" In: *Advances in Neural Information Processing Systems*. 2014 (cit. on pp. 95, 98).

[172] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. "End-to-End Training of Deep Visuomotor Policies". In: *arXiv preprint arXiv:1504.00702* (2015) (cit. on pp. 95, 96).

[173] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105 (cit. on pp. 95, 99, 102).

[174] Minh Hoai, Zhen-Zhong Lan, and Fernando De la Torre. "Joint segmentation and classification of human actions in video". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2011 (cit. on p. 96).

[175] Kevin Tang, Li Fei-Fei, and Daphne Koller. "Learning latent temporal structure for complex event detection". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2012 (cit. on p. 96).

[176] Yang Yang, Imran Saleemi, and Mubarak Shah. "Discovering motion primitives for unsupervised grouping and one-shot learning of human actions, gestures, and expressions". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35.7 (2013), pp. 1635–1648 (cit. on p. 96).

[177] Simon Jones and Ling Shao. "Unsupervised spectral dual assignment clustering of human actions in context". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2014 (cit. on p. 96).

[178] Di Wu and Ling Shao. "Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014 (cit. on p. 96).

[179] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv:1409.1556* (2014) (cit. on pp. 99, 102).

[180] Relja Arandjelovic and Andrew Zisserman. "All about VLAD". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* IEEE. 2013, pp. 1578–1585 (cit. on p. 99).

[181] Zhongwen Xu, Yi Yang, and Alexander G Hauptmann. "A discriminative CNN video representation for event detection". In: *arXiv:1411.4006* (2014) (cit. on p. 99).

[182] Barbara A Murphy and Jill Gilbert. "Dysphagia in head and neck cancer patients treated with radiation: assessment, sequelae, and rehabilitation". In: *Seminars in radiation oncology*. Vol. 19. 1. Elsevier. 2009, pp. 35–42 (cit. on p. 109).

[183] Stephen McKinley, Animesh Garg, Siddarth Sen, Rishi Kapadia, Adithyavairavan Murali, Kirk Nichols, Susan Lim, Sachin Patil, Pieter Abbeel, Allison M Okamura, and Ken Goldeber. "A Disposable Haptic Palpation Probe for Locating Subcutaneous Blood Vessels in Robot-Assisted Minimally Invasive Surgery". In: *Proc. IEEE Int. Conf. Automation Science and Engg. (CASE)*. 2015 (cit. on p. 111).

[184] Barry A Oŕeilly. "Patents running out: time to take stock of robotic surgery". In: *International urogynecology journal* 25.6 (2014), p. 711 (cit. on p. 111).

[185] Sanjay Krishnan, Animesh Garg, Richard Liaw, Lauren Miller, Florian T Pokorny, and Ken Goldberg. "HIRL: Hierarchical Inverse Reinforcement Learning for Long-Horizon Tasks with Delayed Rewards". In: *arXiv preprint arXiv:1604.06508* (2016) (cit. on p. 111).

[186] Sanjay Krishnan, Animesh Garg, Richard Liaw, Brijen Thananjeyan, Lauren Miller, Florian T Pokorny, and Ken Goldberg. "SWIRL: A Sequential Windowed Inverse Reinforcement Learning Algorithm for Robot Tasks With Delayed Rewards". In: *under review at Workshop on Algorithmic Foundations of Robotics (WAFR)*. 2016 (cit. on pp. 111, 112).