Toward Autonomous Endoscopic Surgery: a Framework and Case Studies for Robotic Learning in Healthcare

Will Panitch

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2024-127
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2024/EECS-2024-127.html

May 17, 2024

Acknowledgement

This research was performed at the AUTOLAB at UC Berkeley in affiliation with the Berkeley AI Research (BAIR) Lab, and the CITRIS “People and Robots” (CPAR) Initiative. We thank Gary Guthart for his support. The da Vinci Research Kit is supported by the National Science Foundation, via the National Robotics Initiative (NRI), as part of the collaborative research project “Software Framework for Research in Semi-Autonomous Teleoperation” between The Johns Hopkins University (IIS1637789), Worcester Polytechnic Institute (IIS 1637759), and the University of Washington (IIS 1637444). Portions of this work were also performed in collaboration with the People, AI, and Robots groups at the University of Toronto.

Toward Autonomous Endoscopic Surgery: a Framework and Case Studies for Robotic Learning in Healthcare

by

William Chung-Ho Panitch

A thesis submitted in partial satisfaction of the

requirements for the degree of

Master of Science

in

Electrical Engineering and Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Ken Goldberg, Chair
Trevor Darrell

Spring 2024

Toward Autonomous Endoscopic Surgery: a Framework and Case Studies for Robotic Learning in Healthcare

Abstract

Toward Autonomous Endoscopic Surgery: a Framework and Case Studies for Robotic Learning in Healthcare

by

William Chung-Ho Panitch

Master of Science in Electrical Engineering and Computer Science

University of California, Berkeley

Ken Goldberg, Chair

The introduction of robot surgical assistants (RSAs) like Intuitive Surgical's da Vinci system has equipped surgeons with an additional set of powerful tools for constrained, precise, and endoscopic manipulation. These robots enable medical professionals to perform an array of previously impossible minimally-invasive procedures that result in better medical outcomes, less scarring, and faster recovery. Additionally, RSAs have the potential to standardize procedures and reduce surgeon fatigue through supervised subtask automation. Certain oft-performed, repetitive subtasks, such as incision closure and debridement, could be autonomously performed under surgeon supervision, eliminating certain time-consuming and tedious tasks from the surgeon's workload. To advance this line of research, we propose a unified toolkit for surgical augmented dexterity, consisting of a U-Net-based visual localization module that is trained using autonomously collected subtask data, as well as adaptations of the aforementioned model for 3- or 6-D localization of different common surgical objects, and a set of learned servoing modules that enable critical fine motor control tasks in the surgical setting, even under unreliable proprioceptive feedback. We then apply this sensing-and-planning paradigm to two common surgical subtasks: suturing and vascular shunt insertion, and demonstrate that it enables state-of-the-art autonomous task performance. The augmented dexterity framework achieves an average of 2.93 consecutive completed suture throws using unmodified surgical grippers and needles (important for ensuring instrument sterility), and demonstrates a 75%–100% success rate on different vessel phantoms in the shunt insertion task. These results validate the utility of the framework, and help demonstrate a potential path towards increasing subtask autonomy for surgical settings.

Contents

List of Figures

List of Tables

Acknowledgments

This research was performed at the AUTOLAB at UC Berkeley in affiliation with the Berkeley AI Research (BAIR) Lab, and the CITRIS "People and Robots" (CPAR) Initiative. We thank Gary Guthart for his support. The da Vinci Research Kit is supported by the National Science Foundation, via the National Robotics Initiative (NRI), as part of the collaborative research project "Software Framework for Research in Semi-Autonomous Teleoperation" between The Johns Hopkins University (IIS1637789), Worcester Polytechnic Institute (IIS 1637759), and the University of Washington (IIS 1637444). Portions of this work were also performed in collaboration with the People, AI, and Robots groups at the University of Toronto.

None of this work would have been possible without the help of my incredible mentors, collaborators, and friends across three countries and two continents. In particular, my awesome co-authors on the conference and journal submissions that have culminated in this thesis: Karthik Dharmarajan, my longtime research partner-in-crime; Kishore Srinivas, Vincent Schorp, Kush Hari, and Hansoul Kim, who were instrumental leaders on the shunt insertion, thread tracing, robot control, and suturing projects, respectively; Raven Huang, Baiyu Shi, Lawrence Chen, Kaushik Shivakumar, Vainavi Viswanath, Justin Kerr, Yahav Avigal, Shreya Ganti, and Tara Sadjadpour, for their incredible mentorship and patience with me; Masoud Moghani, Qinxi Yu, Jingzhou Liu, Mayank Mittal, our research collaborators on the simulation works; and, of course, Danyal Fer, Lionel Ott, Thomas Low, Animesh Garg, and Ken Goldberg, who have been my incredible faculty mentors—thank you for guiding and teaching me, even when I was stubborn or shortsighted. Thank you all so much, and I wish each and every one of you the best.

Chapter 1

Background

Robot Surgical Assistants (RSAs) such as Intuitive's da Vinci series are used by surgeons to perform over 2 million procedures annually to facilitate minimally-invasive ("keyhole") surgery to reduce pain, blood loss, scarring, complications, and recovery time. These robot-assisted operations are now the gold standard for procedures involving the appendix, colon, gallbladder, prostate, and numerous other internal structures. Although these robots are extremely sophisticated, they must currently be controlled at all times by human surgeons, as surgery is extremely sensitive to errors. There are a vast number of rare and potentially dangerous edge conditions in medical settings, and the consequences of even a single failure can be fatal for patients. For this reason, it may be a very long time before fully autonomous robots are sufficiently safe and reliable for the operating room.

However, recent advances in AI are opening the door to augmenting surgeon skills when performing specific subtasks such as suturing, shunt insertion, debridement, and resection. In a 2023 article, Goldberg [30] proposes the term "Augmented Dexterity" to describe systems where surgical subtasks are performed by autonomous robot systems under the close supervision of a human surgeon, who is ready to take over at a moment's notice. Augmented Dexterity has the potential to elevate and standardize surgical operation, making surgery safer, faster, and more reliable.

Furthermore, despite the critical need for precision in surgical procedures, the demands of extensive surgeries in the highest-demand scenarios, such as battlefields or disaster zones, can lead to surgeon fatigue and disruptions. Moreover, the growing shortage of surgeons, projected to range between 15,800 and 30,200 by 2034 [18], has resulted in the closure of over 100 rural hospitals in the US from January 2013 to February 2020 [77], with an additional 600 at risk of closure [14]. Utilizing RSAs to assist in surgery through either teleoperation or supervised autonomy has the potential to mitigate the effects of the surgeon shortage, reduce medical workloads, and improve the consistency and efficacy of surgery.

In this work, we propose a set of tools for enabling surgical augmented dexterity for common subtasks, and provide two fully implemented case studies using these tools. We additionally perform experiments that evaluate the performance of this framework for each of the tasks in question, and validate the components using a combination of simulated and

real-world experiments using an Intuitive Surgical da Vinci Research Kit RSA [46].

Chapter 2

Related Work

Surgical Automation

Research in surgical robotics has a long history [69]. Robotic Surgical Assistants (RSAs) are surgical robots designed to help surgeons perform complex surgical procedures such as minimally invasive surgeries, and have been increasingly adopted among high-volume surgeons in the past two decades [15, 10]. They improve surgeon dexterity and visualization [36], and have potential to automate some surgical subtasks to reduce surgeon fatigue [119].

Currently, RSAs are mainly used in hospitals for teleoperation. There are many prior works that study various aspects of improving the surgeon performance and telesurgery experience, including methods to provide better haptic feedback [1], automate camera movement [85], rate surgeon performance [8], and improve network latency [64, 32].

Recently, many studies have also explored automating surgical subtasks, including tissue manipulation [101], hemostasis [84], debridement [47, 97], suturing and knot tying [98, 13], pattern cutting [71, 111], peg transfer [36, 37, 79], tumor localization, and resection [29, 67, 38]. Surgical robots, such as the da Vinci Research Kit [6], Raven [34], and SRI International's Taurus Robotic System [108], face a unique challenge for automation as they are driven by cables and can suffer from inaccurate motion and actuation due to backlash hysteresis [50]. Many prior works have proposed methods for calibration [33, 80, 37]. In this work, we consider the vascular shunt insertion task using both the human teleoperation and the autonomous robot.

Fully automated robot systems have been approved for hair restoration and external beam radiation [89, 49]. However, all endoscopic surgical procedures are performed 100% under human surgeon teleoperation [91]. Some research efforts have focused on autonomously performing specific sub-tasks such as debridement [71], vascular shunt insertion [22], and brain tumor resection [35].

Interactive Perception in Robotics

Goldberg and Bajcsy [31] investigate how a robot can use active perception to recognize the shape of an object by moving a touch sensor to trace its contours. Bajcsy [4] defines *active perception* as the search for models and control strategies for perception which can vary depending on the sensor and the task goal, such as adjusting camera parameters [5] or moving a tactile sensor in response to haptic input [31].

Similarly, *interactive perception*, as explored by Bohg et al. [7], utilizes robot interactions to enhance perception. Interactive perception has been used in robotic manipulation to extract kinematic and dynamic models from physical interactions with the environment [65] and to improve the understanding of a scene in the presence of occlusions and perception uncertainty [7, 19, 75]. Murali et al. [72] leverage feedback from visual and tactile sensors to estimate the pose of partially occluded objects in cluttered environments. Danielczuk et al. [19] propose the mechanical search problem, where a robot retrieves an occluded target object from a cluttered bin through a series of targeted parallel jaw grasps, suction grasps, and pushes. Novkovic et al. [75] use a robot to move a camera and interact with the environment in order to find a hidden target cube in a pile of cubes, while Shivakumar et al. [102] use interactive perception to reduce perception uncertainty when untangling long cables.

In this work we utilize interactive perception-based approaches to enable surgical needle grasping, vessel dilation, and surgical thread coordination.

Surgical Thread Detection and Tracing

Detecting surgical thread from an RGB image has been previously explored in a number of different settings. Early approaches relying on analytic curvilinear detectors [78, 40] work well when the thread is isolated and clearly visible, but fail in realistic scenes with shadows and occlusions. Similarly, Joglekar et al. [43] assume that thread detections can be obtained from color segmentation; however, this may fail due to light glare, sensor noise, materials covering the thread (e.g., blood), and varying lighting conditions. Learning-based approaches generalize better to different backgrounds and lighting conditions, but require manual collection of large datasets. Lu et al. [56] train a U-Net [86] using semi-supervised learning leveraging hand-labeled images for supervision which are time consuming to obtain. We use a self-supervised data collection method that extracts labels autonomously using UV light [110], allowing the system to collect 10 labeled images per minute. Lu et al. [56] propose using a 3D graph to represent the triangulated 3D candidate thread points. The method then computes a minimum energy path through the graph and uses it as the 3D model of the thread. Joglekar et al. [43] propose using a minimum variation spline to represent the suture. This results in a smooth reconstruction with less tight curvature and yields a confidence value along the spline model which is useful to chose a grasp point along the thread. Both methods mentioned above fully reconstruct the model on each frame, ignoring prior frames, making them more susceptible to one-off missing or false detections. Padoy and Hager [78]

assume the 3D spline has been initialized in advance, and focus on tracking the spline across frames. However, this work assumes that the length of the thread is constant, which limits its applicability to certain applications like tail shortening or knot tying. Jackson et al. [40] propose an approach to jointly trace and reconstruct a 3D spline from stereo images as well as a tracking method using pixel-space error minimization. However, their approach assumes a known initial tracing point, manually defined using a space mouse. In contrast, we leverage tracing of 2D splines to address missing or occluded parts of the thread and use an approach which does not rely on a user-defined seed point. Furthermore, our method tracks the spline across frames, increasing its robustness to noisy detections.

Learned Control in Surgical Automation

Visual servoing has been explored for surgical automation, with needle [12, 42] and gripper [57] pose estimation using learned keypoint tracking models like DeepLabCut, a method that uses transfer learning to perform keypoint annotation for pose estimation by Mathis et al. [66]. Another line of research focuses on learning end-to-end visual servoing policies which implicitly model the object state [52, 44]. Paradis et al. [79] and Wilcox et al. [117] have proposed intermittent visual servoing in the context of peg transfer and needle handover, respectively, both of which use a visual servoing policy trained with imitation learning in lieu of classical trajectory optimization where high precision is required. We propose a novel visual needle pose estimation approach using learned image segmentation models as well as known visual features and system dynamics. The estimated object poses are used to automate the surgical suturing sub-tasks of needle insertion, extraction, and handover using analytic control methods.

Surgical Robotics Simulation

Several robot learning efforts have introduced domain specific frameworks, often catering to specialized needs. Many prior frameworks [120, 41] using MuJoCo [113] or Bullet [16] focus mainly on rigid object manipulation tasks. On the other hand, frameworks for deformable bodies [2, 54] mainly employ Bullet [16] or FleX [76], which use particle-based dynamics for soft bodies and cloth simulation. Most of these physics engines are CPU-based, relying on CPU clusters for parallelization. However, limited tooling exists for unified frameworks to enable content development specific to domains in surgery. ORBIT-Surgical builds on recently released framework ORBIT, which provides a modular and unified simulation interface for robot learning. ORBIT-Surgical relies on GPU-accelerated physics engines, signed-distance field (SDF) collision checking, and stable solvers based on FEM for deformable body simulation [59, 60].

Prior state-of-the-art surgical robotic simulators have limited rendering capabilities or do not support deformable bodies, which are necessary for modelling the vessel phantom in shunt insertion environments. dVRL [83] provides rigid body environments with the dVRK for training RL algorithms, but does not support deformables. LapGym [93] is a suite of RL

environments built on top of Simulation Open Framework Architecture (SOFA), but does not have photo-realistic rendering and utilizes the CPU for physics execution, which limits the potential speedup via parallelism. NVIDIA Isaac ORBIT [68] is a recently proposed simulation framework for robotics and robot learning powered by NVIDIA Isaac Sim. It features fast and accurate rigid- and soft-body simulation using GPUs and offers a modular design to create robotic environments with photo-realistic scenes.

To address the lack of fast and realistic simulation environments for surgical subtasks, we release a simulation environment for numerous surgical tasks built on top of NVIDIA Isaac ORBIT for ease of further study for the community. The simulation environment consists of a dVRK robot, a spring clamp holding a vessel phantom, and a shunt. It embodies realistic joint articulations, controllers, and collision properties to enable rich interaction with the surrounding environment. ORBIT provides APIs for object definition (i.e. rigid body, FEM-based deformable body, and well as particle based simulation). This will allow the model to capture high dimensional state and complex physical interactions between the dVRK grippers and deformable vessel phantoms.

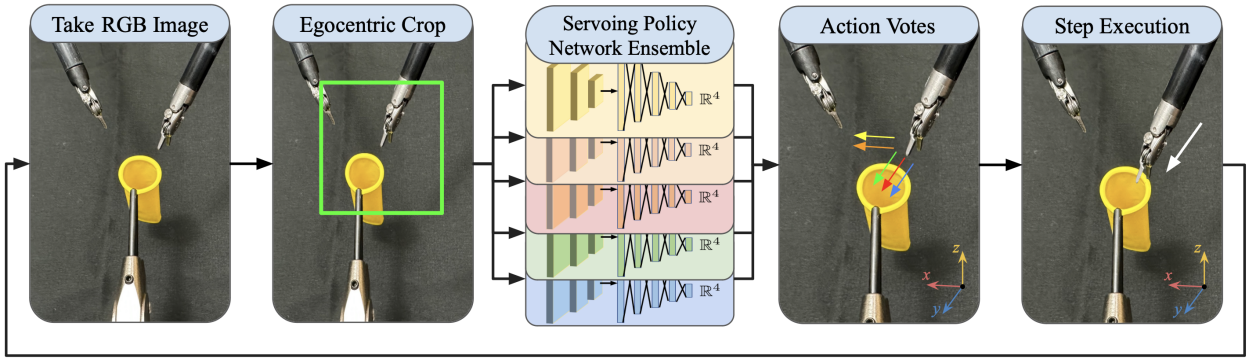

Figure 2.1: **Visual Servoing and Grasping Module**: At each step of the visual servoing module, an RGB image is captured by the camera and passed into the pipeline. Using the camera-to-robot transform and the forward kinematics of the robot, this image is cropped to a 180×180 square centered about the end effector. The crop is passed into an ensemble of convolutional neural networks which each output a direction $d \in \{+x, -x, +y, -y\}$ of motion. These outputs are collated through voting to determine the direction of motion for this step.

Chapter 3

Sensing and Localization

3.1 Overview

As the first step towards autonomous interaction with surgical environments, we need to develop a robust scene understanding system to properly localize objects of interest in our scene. In order to do so, we assume access to a depth-inclusive image representation of our scene, either through a depth camera or a stereo camera setup. Using the 2D RGB image representation, we train a U-Net [87], an image-based segmentation model, to detect the needle in image space. The training data is labeled using the *Labels from UV* technique proposed by [110]. The segmentation mask produced by the U-Net is used to isolate the needle from the pointcloud generated by the 3D image representation.

Specifically, we design and mount both an ultraviolet and a visible illumination system around the robot workspace, and use a programmable power switch to alternately illuminate the scene in either UV or visible light. By painting elements in the scene with different colors of UV-fluorescent paints (which are translucent under visible light), we are able to collect labeled masks of different objects using thresholding in HSV space on the UV-illuminated images. By pairing these images with time-syynchronized visible light images, we are able to automatically obtain large labeled training datasets for any object in the scene. We can further automate the data collection process by using the dVRK arms to manipulate and move objects around the scene to capture a random or otherwise advantageous training distribution. This data collection technique can be extended to any object and robot manipulator.

The U-Net architecture used is an asymmetric U-Net [86, 88] to generate the segmentation masks (Fig. 3.1(c)) from RGB images (Fig. 3.1(a)) using LUV [109]. As in [109], we utilize a network architecture with a 4-tier contracting path and a 4-tier expansive path. We use layer depths of 128, 256, 512, and 1,024 channels for both paths and a pooling factor of 2×, and train the model using an Adam optimizer with learning rate $\alpha = 0.001$. However, we replace the final "up-convolution" level of the expansive path with an upsampling layer to reduce the network runtime and parameter count and enable real-time (30 Hz) inference on

our NVidia GeForce RTX 2080 GPU. The segmentation mask is then projected onto the point cloud generated by the RGBD camera to select out points in 3D space.

3.2 Gripper Tracking

We use a painted needle driver to collect 1500 randomly distributed labeled data pairs in various workspaces with both the left and right end effectors. Using this dataset, we train a U-Net head to predict a separate mask over both needle drivers. We skeletonize these mask predictions to obtain an estimate for the pixel-space position and orientation of each gripper, then use least squares to estimate a gripper tip position and orientation in 3 dimensions. Empirically, we find that camera noise in the depth direction makes predicting full 6-DOF pose challenging, but that 3-D position estimates performed using this method are more reliable than those using encoder input alone.

3.3 Vessel Rim Pose Estimation

The vessel rim segmentation pipeline takes as input an RGB image of the workspace and converts it to a segmentation mask marking the location of the rim of the vessel. We use ultraviolet fluorescent paint and a UV–Visible light system to collect 3200 UV/visible light image pairs for training across a variety of different vessel phantom materials and sizes, as well as under different gripper orientations and workspace layouts. A further 805 image pairs with corresponding 3D depth data were held out as a validation set to test the performance of our full localization pipeline. In addition, to ensure that our perception system did not become overly reliant on the color profile of our vessel phantoms (which could differ significantly from the conditions in an in-vivo operation), we additionally trained an otherwise-identical version of our perception pipeline using only grayscale input information for comparison.

We extract masks localized to the vessel rim from the ultraviolet images by applying color thresholding over UV labels. Since each set of images is specific to a particular vessel phantom, we transform and reuse data across phantoms when possible; for example, if a phantom with similar structure but different color had previously been seen in dataset collection, then we would utilize color transformation and synthetic data augmentation while preserving the masks. If a phantom used for evaluation has unique structure or more complex textures that cannot easily be converted, we collect additional data for that vessel phantom.

The estimated segmentation mask output is then projected onto the point cloud generated by the RGBD camera to select out the points on the vessel rim in 3D space. We then apply random sample consensus (RANSAC) [27] to estimate the 3D orientation of the vessel lip represented as a tuple (c_p, c_n, r), representing the location of the circle's center, the orientation of the circle normal vector, and the radius of the circle, respectively. At each iteration of the RANSAC algorithm, we sample 3 points from the point cloud, to which we fit a circle. The best fit circle seen so far is kept and returned at the end of the algorithm

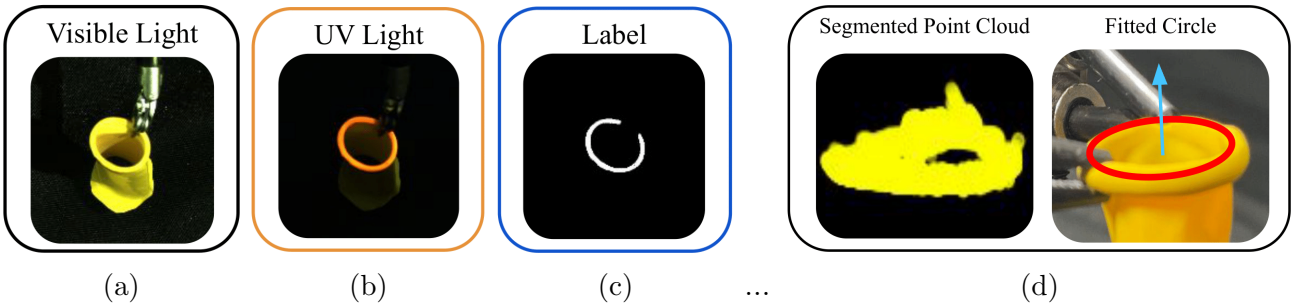

Figure 3.1: **Data collection and Vessel Segmentation.** We collect data for training our neural network, consisting of a) an image of the vessel phantom when exposed to visible light, b) an image of the phantom vessel when exposed to UV light, and c) the extracted segmentation mask. After training a network on these images, we d) use predicted masks to segment noisy point clouds deprojected from the RGBD images, then use RANSAC to fit a circle (in red) with its normal vector (in blue) and both elements on a side view of the same scene of the point cloud.

(Fig. 3.1 (d)). The system pipeline uses a RANSAC inlier radius $r_{inlier} = 1$ mm and runs until convergence or for a maximum of $n_{iter} = 1,000$ iterations to minimize the stochastic error between the 3D depth images and masking pipeline. The parameters r_{inlier} and n_{iter} were chosen qualitatively through hand-tuning.

3.4 Needle Tracking

Similarly to the vessel rim segmentation above, we utilize the semicircular structure of surgical needles to more easily parameterize the needle's position in space. Since we also need to accurately estimate the positions of the two endpoints on the needle and the orientation of the needle itself, we propose a novel 6D needle pose estimation module as seen in Figure 3.2. Based on Wilcox's work for needle handover (HOUSTON) [117], we propose an algorithm that determines the best-fitting 3D circle. A key distinction between our work and HOUSTON is that HOUSTON obtained a pointcloud from stereo point matching while this work uses RAFT-Stereo [55], a roboust stereopsis neural network. Given the 2-D image masks from our U-Net and the pointcloud from RAFT-Stereo, our pipeline begins by using RANSAC to estimate a 3D plane equation that fits the needle pointcloud. Then, we calculate the normal vector from the estimated 3D plane to ascertain the orientation of the needle. All inlier needle points are projected onto that plane. To achieve higher accuracy, we then project all inlier points on the plane to the xy-plane where we use RANSAC to estimate a 2D circle equation. Next, we derive the positions of the two endpoints of the needle on the 2D circle by finding the 2 most distant needle inliers in the pointcloud. After that, we determine the 3D circle and endpoint positions by back-projecting onto the aforementioned

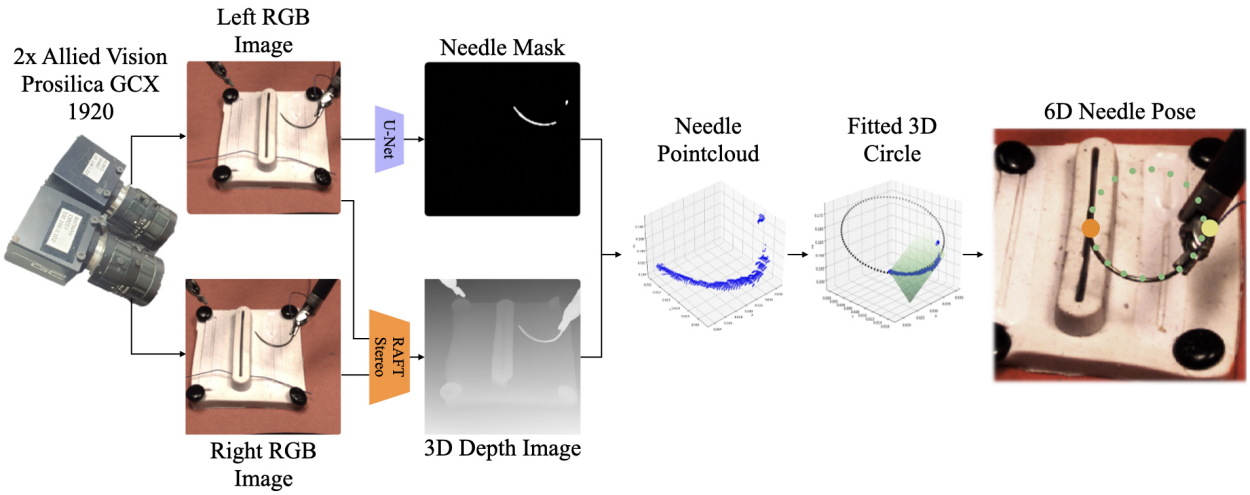

Figure 3.2: **6D Needle Pose Estimation Module.** The needle pose estimation starts with a pair of stereo left and right images. Using RAFT-Stereo, we generate a disparity image from the stereo pair [55]. Furthermore, we segment the needle in the left image with a U-Net to create a needle mask. From there, we apply the needle mask to the disparity image, and create the corresponding needle pointcloud. Using RANSAC, we find a best-fit plane to determine the normal vector of the 3D circle representing the needle (seen in green in the Fitted 3D Circle image). Then, we project all needle inliers from the RANSAC to the plane, and use RANSAC again to find the best fit circle (seen in black in the 3D Circle Fit image with the assumed fixed radius (12 mm for all experiments). Finally, we find the two farthest points on the needle pointcloud to determine the needle endpoints (seen as orange and yellow in the 6D Needle Pose image).
3D plane estimate.

3.5 Thread Tracing

We adapt the analytic cable tracing method from Shivakumar et al. [103] and developed in Schorp et al. [94] to trace the path segments from the 2D thread detection masks. However, instead of generating all possible global paths, this work leverages heuristic scoring rules similar to those proposed by Viswanath et al. [116] and Keipour, Bandari, and Schaal [48] to generate a single global trace. In contrast to the learning-based method proposed in [116], which detects and traces cables simultaneously, we propose an analytical method. The method proposed in [48] is similar in the sense that it uses scoring functions that prioritize traces which cover more of the cable and have lesser changes in angle. However, they model the thread as a chain of cylinders whereas we fit a 2D spline onto the traced detections to bridge gaps. The analytic thread tracer locally traces contiguous segments and greedily stitches them together, as described in Algorithm 1.

As in prior work [40], we model the suturing thread as a 3D NURBS parametric curve.

Algorithm 1 2D Surgical Thread Tracing Algorithm

Require: $D \leftarrow$ pixelwise thread detection
  mask $\leftarrow D >$ thresh$_\text{d}$
  mask $\leftarrow$ mask $-$ (conn components with area $<$ thresh$_\text{a}$)
  path_segs $\leftarrow$ []
  while sum(mask) $>$ thresh$_\text{s}$ **do**
    start_point $\leftarrow \arg\max$(D)
    paths $\leftarrow$ sgtm2tracer(mask, start_point).
    best_path $\leftarrow \arg\max_{\text{p}\in\text{paths}}$ score(path)
    On mask, set points along best_path to 0.
    Append best_path to path_segs.
  end while
  while length of path_segs > 1 **do**
    find i, j within path_segs with lowest matching cost
    new_seg $\leftarrow$ merge of path_segs[i] and path_segs[j]
    add new_seg to path_segs
  end while
    return path_segs[0]

Instead of jointly tracing and reconstructing the thread, we use a dedicated 2D tracer to compute the sequence of thread pixels in both images before reconstructing the 3D thread model. The spline parameter $t \in [0, 1]$ describes the normalized distance along the spline.

To start the 3D tracing method, a 2D NURBS spline defined by 32 control points is fitted to the traces in both images using a least squares approximation. The number of control points is chosen to allow a sufficient amount of flexibility to the spline so that it can approximate tight curves common in suturing thread.

Next, we triangulate these 2D splines into 3D to estimate the thread state. We propose the following stereo matching approach: The left trace spline point p_i^L is located at spline parameter t_i^L along the spline and has pixel coordinates $[u_i^L, v_i^L]$ for width and height respectively, starting from the top left corner. For each point along the left spline p_i^L, a corresponding point on the right spline $p_{j(i)}^R$ is found which minimizes the difference between spline parameters t_i^L and $t_{j(i)}^R$ and satisfies rectified stereo image properties. Specifically, the right image point should have the same vertical coordinate than the left image point except for a tolerance of up to $\alpha = 5$ pixels (condition a). $p_{j(i)}^R$ must be further left within the image than p_i^L (condition b). The right spline candidates must be further along the spline than the last matched right spline point (condition c). t_i^L and $t_{j(i)}^R$ must be within a distance $\beta = 0.05$ (condition d). For a given value of i, we seek to solve

$$j(i) = \arg\min_j |t_j^R - t_i^L|$$

such that a) $|v_i^L - v_j^R| \leq \alpha$, b) $u_j^R \leq u_i^L$, c) $t_j^R > t_{j(i-1)}^R \forall i$, d) $|t_j^R - t_i^L| \leq \beta$.

The matched points are then triangulated using the camera intrinsics to obtain their 3D position, and a 3D NURBS spline model is fitted to the triangulated points using least-squares optimization. The values for α, β and a rejection threshold for bad reconstructions were set empirically as a trade-off between reconstruction quality and number of discarded frames.

Inspired by Jackson et al. [40], we compute 200 correction vectors to update the coordinates of the 3D spline control points between frames. The number of correction vectors was set as a trade-off between tracking accuracy and computation speed. These parameters enable thread detection updates at 2.5 FPS, which we find to be sufficient for surgical thread manipulation tasks. Instead of an energy-minimization approach to compute correction vectors, we instead use the 2D splines fitted on the current stereo traces. The 2D correction vectors are then obtained as a sum of two vectors, c_{mask} and c_{trace}. c_{mask} is a vector in image space pointing towards the closest point on the prediction mask. c_{trace} matches the point of the 2D spline fitted on the 2D trace at parameter t with the point at parameter t of the projected 3D spline. The 2D correction vectors from both stereo images are triangulated to find 3D correction vectors, and the 3D correction vector terms are averaged to obtain the final set of correction vectors.

Using only the distance correction c_{mask}, the 3D spline tends to collapse as the segmentation mask of the thread does not constrain the 3D spline along the length of the thread.

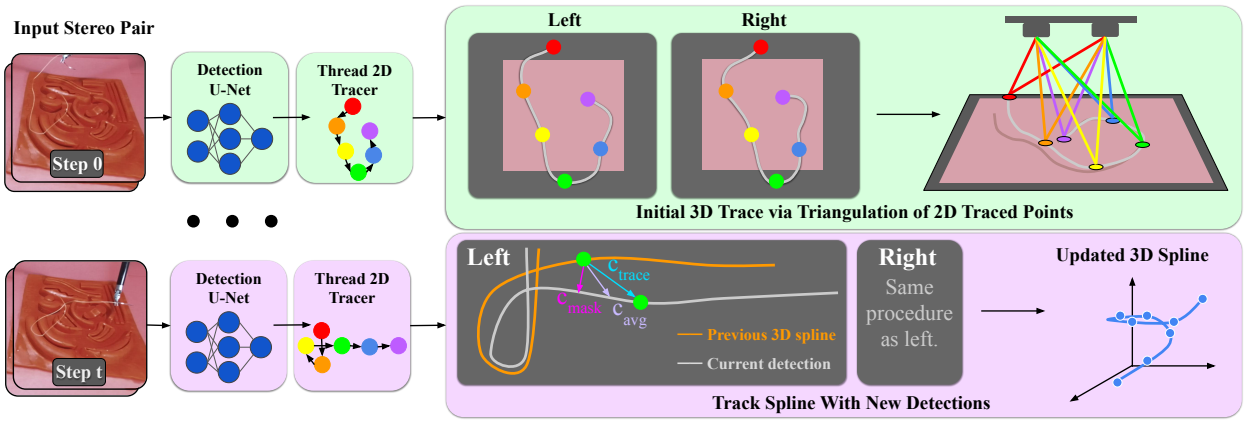

Figure 3.3: **Overview of the thread detection pipeline: 2D Surgical Thread Detection, 2D Tracing, 3D Tracing, and 3D Tracking**. *Left:* For every stereo pair of images, we predict thread segmentation masks, then run a 2D tracer to compute the sequence of pixels along the thread. *Top right:* To initialize the 3D spline of the thread, we match points meeting both stereo image and tracer topology constraints and triangulate their positions in 3D. We initialize the 3D trace by fitting a 3D spline to these points. *Bottom right:* To update the 3D spline with new frames, we compute correction vectors in 2D as an average of vectors which push the projected 3D spline onto the new detection and push each projected 3D point to its corresponding point on the 2D trace. We then triangulate the correction vectors across both images and apply them to the 3D spline to perform an update.

This is mitigated by the second correction vector, c_{trace}, which assigns a fully constrained pixel location to each point along the projected 3D spline. Given the correction vectors, an updated set of control points is computed using the least square control point update described by Jackson et al. [40].

Chapter 4

Control and Actuation in Surgical Robots

4.1 The Challenges of Encoder-based Feedback Control

In most robotic applications, the standard approach to closed-loop control has been encoder-based proprioceptive feedback. In this paradigm, an analytical or learned controller tracks joint positions and velocities using motor encoders. Modifications on this base setup have been extensively studied in the robotics literature, resulting in a wide variety of different controller frameworks. Many of the most popular algorithms are variations on Position-Integral-Derivative (PID) controllers, Linear-Quadratic Regulators (LQR), or Model Predictive Controllers (MPC), all of which are used in various modern robotic settings, including robot arms, UAVs, and autonomous vehicles.

However, in endoscopic surgical robots, robot geometries are highly constrained by their need to operate within the human body. The dVRK and Raven II surgical robots, for example, are designed to rotate about a single stationary point to allow for surgery using pinhole incisions, and must have minimal spatial and inertial bulk beyond this origin. These constraints force motors to be placed physically away from the joints that they actuate, requiring the use of cables, belts, or other force transfer mechanisms. Although this enables us to build more capable endoscopic robots, it also introduces mechanical errors, such as cable slip, stretch, and hysteresis.

Under human teleoperation, these challenges are relatively minor, as humans are equipped with powerful visual estimation capabilities to compensate for imperfect feedforward control. But robotic systems do not have this integration, and utilizing encoder feedback alone for control of RSAs can cause poor outcomes due to the inaccuracy of indirect joint encoder measurements. Encoder-only control of the dVRK sometimes results in errors of 1 cm or higher, an unacceptable tolerance for surgical applications. For this reason, one of the foremost challenges in medical automation has been how to use visual pose estimation systems

to build accurate feedback controllers for RSAs.

One simple approach is to adapt classical feedback controllers to use visual pose estimation systems as ground-truth input, rather than encoder readings. This eliminates errors that originate in the joint-motor mismatch that often plagues RSAs, in exchange for introducing sensor noise from the vision system. In particular, due to the imperfect nature of current 3D sensing techniques, noise in the direction normal to the camera plane is often amplified, requiring the use of multi-step averaging or kalman filtering to isolate signal from noise. Similarly, this setup is highly sensitive to noise in our learned gripper detection network (see Section 3.2), and therefore requires additional averaging. Empirically, we find that applying a combination of deep calibration and visual localization improves performance over pure proprioception, but does not provide sufficient precision for surgical operations. This motivates our exploration of the subsequent methods.

4.2 Coarse-Fine Visual Servoing

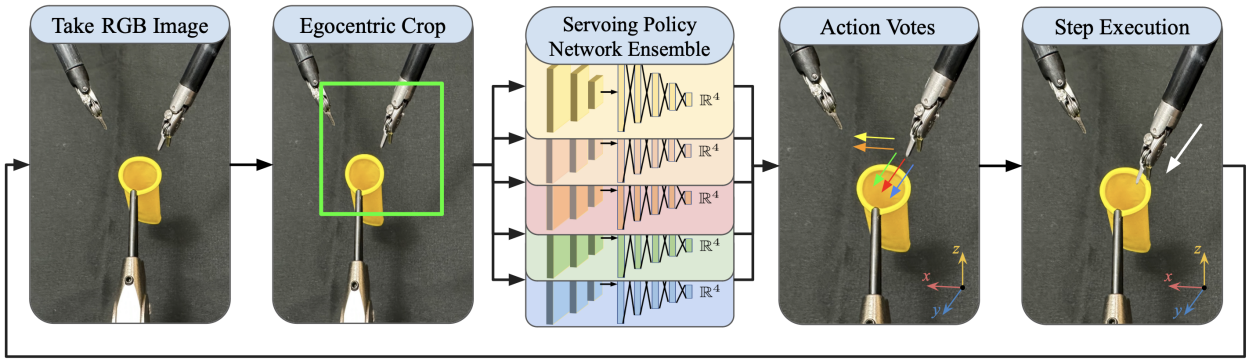

Figure 4.1: **Visual Servoing and Grasping Module**: At each step of the visual servoing module, an RGB image is captured by the camera and passed into the pipeline. Using the camera-to-robot transform and the forward kinematics of the robot, this image is cropped to a 180×180 square centered about the end effector. The crop is passed into an ensemble of convolutional neural networks which each output a direction $d \in \{+x, -x, +y, -y\}$ of motion. These outputs are collated through voting to determine the direction of motion for this step.

Visual servoing is a technique commonly used to make open-loop policies more robust, and has been used extensively in surgical automation for subtasks that require high precision, such as needle handover [117] and peg transfer [36]. In visual servoing, 2- or 3-dimensional scene images are used directly to calculate robot control inputs, without the use of an explicit intermediate pose estimation step. In our formulation, this is achieved by training goal-conditioned neural networks, which take as input an image of the scene and output a

distribution over robot actions. In particular, for the Surgical Augmented Dexterity framework, we formulate visual servoing as an imitation learning (IL) problem.

As shown in Fig. 4.2, our visual servoing pipeline utilizes two policies, π_{right} and π_{left}, to take corrective actions for the right and left arms respectively. Each policy takes in a 180×180 RGB image cropped around their respective grippers and outputs an action direction $d \in \{+x, -x, +y, -y\}$, with directions being with respect to the robot coordinate frame. The image cropping encourages the policy to learn about the relative positions of the grippers and the goal pose, reducing the overfitting against specific features from other parts of the workspace. The magnitude of the per-step motion begins at 0.8 mm, and decreases over the course of execution. The servoing terminates after 20 steps or once the magnitude of motion drops below 0.2 mm, whichever occurs first.

We represent the policy as a neural network ensemble consisting of 5 lightweight convolutional neural networks, each consisting of 3 convolutional layers and 5 fully connected layers. Ensemble learning methods have been shown to reduce bias and improve stability in many settings [92, 25, 24]. To train the policies, offline human demonstrations of 150 trajectories consisting of 15-30 actions were collected using a keyboard teleoperation interface, resulting in 4,303 image-action pairs. The networks were each trained using a cross-entropy classification loss on a disjoint subset of 20% of the collected data.

During execution time, we take the majority vote among the five ensemble networks as the direction for the robot to move. The magnitude of the first action is 0.8 mm. If the actions chosen at two consecutive timesteps are in opposite directions, the magnitude of subsequent actions is halved. When the action magnitude drops below a threshold of 0.2 mm, the servoing terminates. The policies do not need to explicitly learn a stopping action, as the decaying action magnitudes from consecutively moving in opposite directions causes convergence.

We find that using a combination of classical control for coarse actions and visual servoing for last-mile fine-motor control provides the best tradeoff between reliability and execution time.

4.3 Reinforcement Learning in Surgical Robotics

Fully learned control paradigms for robotics are broadly separated into two categories: IL and Reinforcement Learning (RL). In IL frameworks such as the Visual Servoing discussed above, a regression is performed from paired observation inputs to action outputs, usually using human-collected demonstrations. In contrast, RL paradigms largely eschew human-generated training data and attempt to learn action outputs that maximize some *reward* function that is calculated based on the state of the system. The difference in optimization objective between IL and RL means that well-trained RL algorithms often outperform IL algorithms at fine-motor tasks, which are extremely prevalent in surgical applications. However, RL algorithms often require large numbers of rollouts to learn effective policies, which can be difficult without access to a high-quality simulation environment.

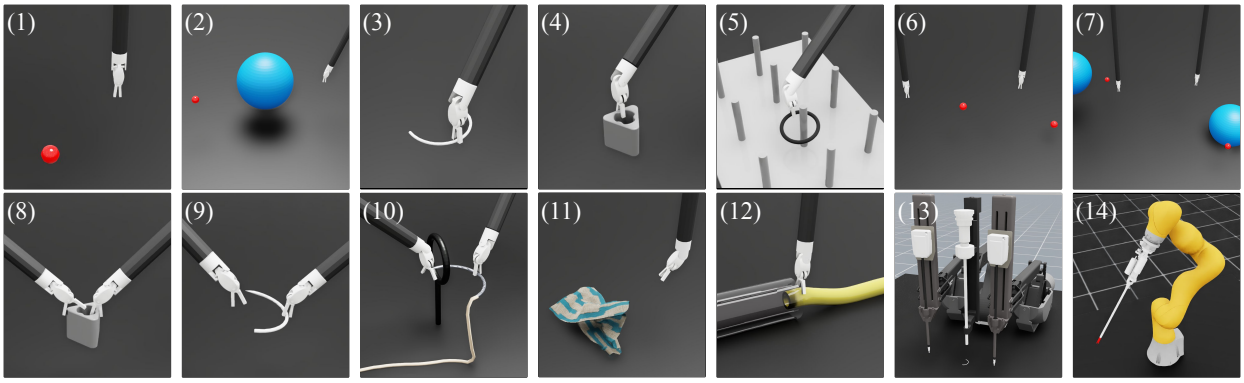

Figure 4.2: **Orbit Simulation Benchmark Tasks.** (1) `Reach`: dVRK Patient Side Manipulator (PSM) to reach a desired position (red sphere), (2) `Reach with Obstacles`: reach a desired position (red sphere) with randomly placed obstacles in the scene (blue sphere objects; object shape and size are customizable), (3) `Suture Needle Lift`: lift a suture needle to a desired position, (4) `Peg Block Lift`: lift a peg block to a desired position, (5) `Pick and Place`: pick and place a ring on a peg tower, (6) `Dual-arm Reach`: dual-arm reach to specific desired positions shown with red spheres, (7) `Dual-arm Reach with Obstacles`: dual-arm reach to specific desired positions (red spheres) with randomly placed obstacles in the scene, (8) `Pick and Transfer`: pick and transfer a peg block, (9) `Needle Handover`: handover and regrasp a suture needle, (10) `Threaded Needle Pass Ring`: handover a threaded suture needle through a ring pole, (11) `Gauze Cloth Pick`: retrieve a gauze and lift it to a desired location, (12) `Shunt Insertion`: insert a shunt (yellow tube) into a blood vessel phantom (clear tube), (13) `Multi-arm dVRK`: needle handover task with camera input from the dVRK Endoscopic Camera Manipulator (ECM), (14) `STAR Reach`: STAR arm to reach a desired position.

Robot learning for surgical tasks has been previously attempted for a number of subtasks with varying levels of autonomy [73, 26, 3] such as shape cutting [71, 111], suturing [98, 51], debridement [71], dissection, and tissue retraction [73]. However, the lack of easy to use, high quality simulation frameworks for surgical robotics has long limited learning at scale. State-of-the-art surgical robotics simulators have suffered from issues such as limited rendering capabilities [93], few available surgical environments [114, 107], and lack of native support for deformable objects [118]. Additionally, the lack of GPU-accelerated physics decreases the rollout wall-clock speed for many simulators [70]. Further, there are numerous proprietary systems such as SimNow [39] and Mimic & Simbionix [106]. These offer virtual environments primarily for teleoperated skill and procedural practice, while lacking programmatic access for learning based methods, and high-speed throughput. As a result, the lack of a surgical robotic framework that integrates precise physics, delivers high speed realistic rendering, and accommodates various robot learning tools continues to pose an ongoing challenge for RL in

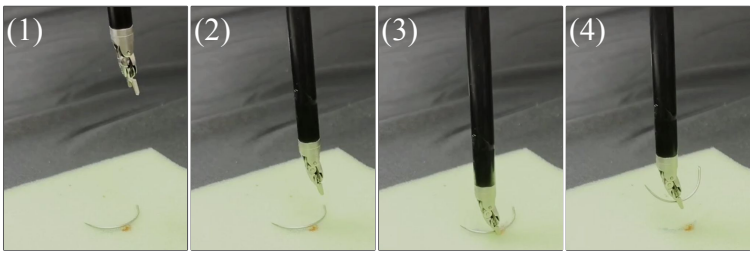

Figure 4.3: **Zero-shot `Suture Needle Lift` Policy Transfer.** An example of a Reinforcement Learning policy trained in ORBIT using RSL-rl PPO and deployed on the physical dVRK. (1) Starting position of the dVRK arm and suture needle, (2) arm moving towards the needle, (3) grasping the needle, (4) successfully lifting the needle. The policy was fully trained in ORBIT-Surgical.

surgical robotics.

To alleviate this, our earlier work develops ORBIT-Surgical. ORBIT-Surgical uses NVIDIA Isaac Sim and builds on the ORBIT framework to enable fast and accurate physics interactions, realistic scene rendering, and access to robot learning libraries geared toward surgical subtask learning. ORBIT-Surgical supports parallelized GPU simulation [62], contact-rich interactions (i.e. SDF-based collision meshes [74] and convex decomposition [63]), and APIs for object definition (i.e. rigid body, FEM-based deformable body, and particle-based simulation).

ORBIT-Surgical supports various reinforcement learning (RL) frameworks, including rl-games [61], RSL-rl [90], and stable-baselines-3 (SB3) [81]. For our training, we utilise the RSL-rl implementation of Proximal Policy Optimization [95], as it is optimized for vectorized simulation and GPU training, and build task-specific environments for shunt insertion and needle handover.

Chapter 5

Case Studies in Surgical Augmented Dexterity: Suturing

We now discuss the application of our Surgical Augmented Dexterity framework to the automation of two common surgical subtasks: Suturing and Vascular Shunt Insertion.

5.1 Background

Suturing is a surgical subtask that is common in almost all surgical disciplines and operations, used to secure body tissues and close the edges of wounds. Surgical suturing is typically performed to position tissue, provide support, or close wounds or incisions, but is a repetitive task which requires high precision and can result in significant surgeon fatigue.

In prior work, automated surgical suturing has been simulated in-vivo using IR markers and industrial robot arms (such as the KUKA LBR Med) for both the open-surgery setting [100] and the minimally invasive surgery setting [91]. Some recent research efforts have been directed towards automating surgical suturing in whole or in part using the dVRK RSA. While performing the complete surgical suturing task autonomously remains an elusive problem [99], many researchers have explored the automation of sub-tasks such as suture placement [45], needle handover [117, 11, 115], needle extraction [105], needle pick-up [17], and knot tying [112]. Though each of these methods individually have individually shown high success rates (90%), running them sequentially often results in a much lower success rate due to the independently multiplicative nature of their failure risks. To offset this, prior attempts to automate the complete suturing task have utilized hardware simplifications; Schwaner et al. [96] demonstrate impressive success rates with only a needle and no thread, avoiding the risk of the robot getting tangled in the thread, while Sen et al. [99] use colored needles and a special gripper mount to aid in needle detection and orientation.

In contrast to these works, the Augmented Dexterity framework allows us to demonstrate a surgical suturing pipeline using unmodified surgical needles and suture thread and without the use of any physical aides. This allows for the use of sterile surgical implements (such as

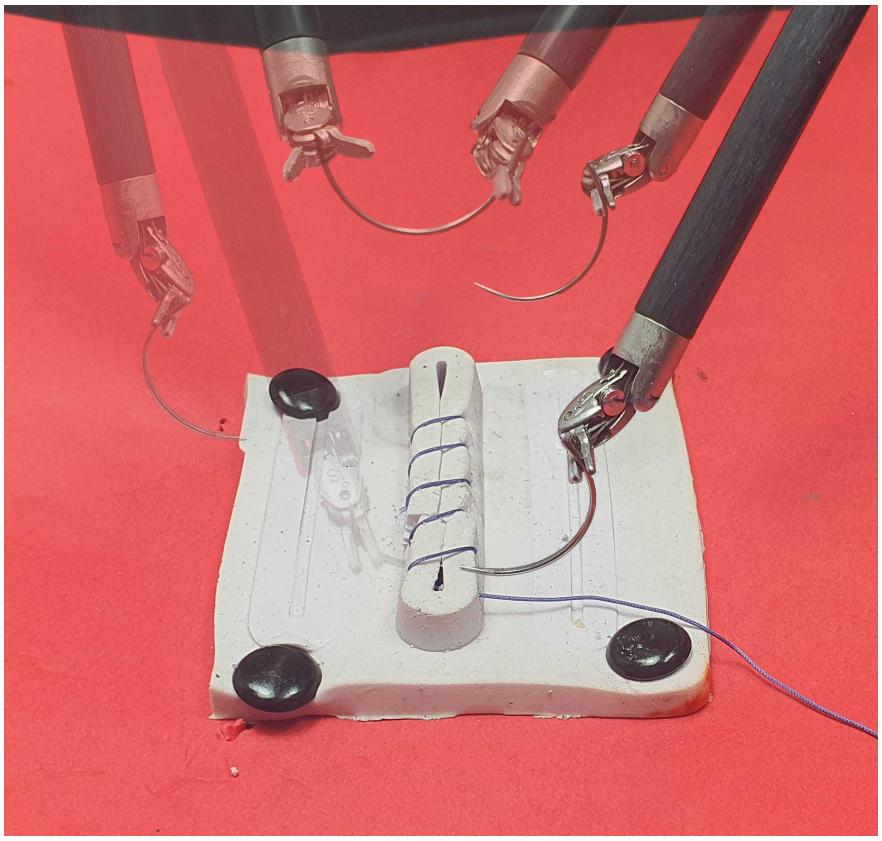

Figure 5.1: **Surgical Suturing Task**. Each suture throw consists of needle insertion, needle extraction, thread tensioning, and needle handover with pose correction. This process is repeated until failure or the wound is fully closed (6 successive sutures).

needles and end effectors), further improving the applicability of the proposed methodology.

5.2 Problem Statement

Overview

Given a linear wound, perform as many simple uninterrupted sutures as needed to close the wound.

Assumptions

We assume known needle shape and diameter, predetermined 3D points for needle insertion and extraction, a calibrated stereo camera pair, and the transformation between the robot

and camera coordinate frames. We also assume the wound is raised and orthogonal to the camera as shown in Fig. 6.2.

Objectives and Evaluation Metrics

We consider two evaluation metrics: number of successful consecutive sutures and completion time. We consider a suture throw to be successful if the robot is able to pass the needle through the wound, perform suture thread cinching (tightening), and return the needle to a neutral position outside the phantom with no tangles of the suture thread.

5.3 Methods

STITCH achieves augmented dexterity for surgical suturing using novel perception and control methods as shown in Figs. 3.2 and 5.2.

6D Needle Pose Estimation Module

The Allied Vision Prosilica GC 1290 cameras used for this paper capture a stereo pair of images with a resolution of 1280x960 at up to 33fps. With these images, we create a depth pointcloud of the scene using disparity images from RAFT-Stereo, a deep architecture for rectified two-view stereo that has been empirically shown to be more robust than standard stereopsis techniques and generalizes well to unseen real-world data [55]. Using this pointcloud, and the procedure outlined in Section 3.4, we are able to obtain an estimate of the pose of the needle in near-real time.

Augmented Dexterity Suturing Motion Controller

The surgical suturing task is composed of several distinct motions. The motion controller directs the robot motions in every state, as well as the state transitions and when they ought to be performed. Each suture is composed of the sequence of needle insertion, a thread sweeping motion to clear excess thread from the suture site, needle extraction with suture cinching, needle handover, needle pose correction, and failure recovery, as shown in Fig. 5.2. The needle motion inside the tissue is designed to reduce tissue damage while remaining withing the kinematic bounds of the gripper.

Needle Insertion

The robot inserts the needle into the tissue phantom with a circular twisting motion at the specified insertion point such that the needle tip exits the tissue at the specified extraction point.

The needle is inserted in the tissue phantom in two steps to minimize strain on the tissue. Both steps are performed open loop as a large part of the needle is occluded by the tissue

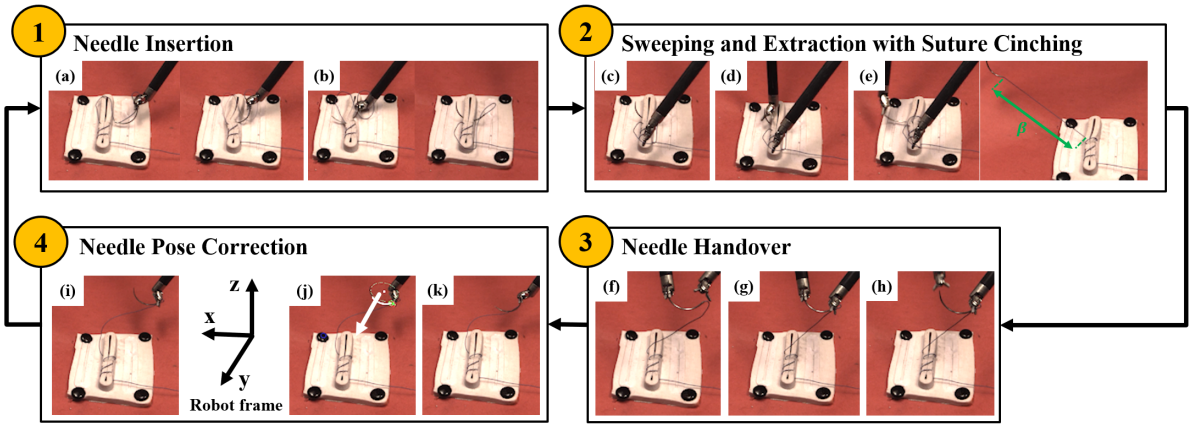

Figure 5.2: **The individual processes embedded within the STITCH motion controller** There are 4 parts to the state machine: 1. Needle insertion: (a) The right needle driver moves the needle to the initial insertion point at the proper orientation; (b) The right needle driver inserts the needle into the phantom with combined rotation and translation movements; 2. Sweeping and Needle Extraction with suture cinching: (c) The right needle driver follows the +y axis in the robot frame down the center of the wound to "sweep" any thread off the wound; (d) The left needle driver moves 1 centimeter behind the needle endpoint to prepare for extraction; (e) The left gripper grasps the needle and pulls it through until the length of the thread is at a desired β; 3. Needle Handover: (f) The right needle driver moves 1 centimeter behind the needle endpoint to prepare for handover; (g) The right gripper grasps the needle for handover; (h) The left gripper releases the needle; 4. Needle Pose Correction: (i) The right needle driver moves the needle to an optimal needle pose estimation region of the scene; (j) Based on the current pose of the needle, it is rotated such that the normal vector of the needle is aligned with the + y axis in robot frame; (k) The needle is rotated about the +y axis in robot frame so it is at the optimal orientation for the next insertion so the pipeline can be repeated again.

and the needle driver during insertion. First, the tip of the needle is pushed into the tissue phantom corresponding to the vector between the given insertion and extraction points (Fig. 5.2(a)). This allows for the straight tip of the needle to penetrate the tissue and exit at the needle extraction point. Second, the needle is rotated into the tissue by performing a 45° rotation around the estimated circle normal vector (Fig. 5.2(b)). This motion follows the curvature of the needle as it passes through the tissue, so that it passes along the hole made by the needle tip and does not stretch or tear the tissue.

Thread Sweeping

The thread sweeping motion is designed to prevent failure during the needle extraction step (Fig. 5.2(c)). Failures occurring during the needle extraction process can be broadly classified into two cases: (1) the thread passes in front of the needle, occluding it and leading

to detection errors; (2) when both the needle and the thread are accidentally grasped together during the re-grasping process, in spite of an accurate needle pose estimation. Pushing the thread out of the re-grasping site through a sweeping motion before the extraction step can prevent the failures described above.

We use the thread modelling method outlined in Section 3.5 to track the thread for the sweeping motion. In this work, a U-Net is trained to output a segmentation mask of the thread. Running an analytic tracer on that segmentation mask for the left and right images, a 3D NURBS spline is fitted for the thread.

The sweeping motion involves opening the gripper wide and passing it over the wound towards the camera so that the thread is caught and pushed ahead of the needle. With the thread in front of the needle, entanglement risks during extraction are significantly reduced.

Needle Extraction

The purpose of the needle extraction motion is to remove the needle from the tissue phantom and to pull the thread taut to close the wound. To successfully perform multiple sutures, it is necessary to pull the thread to an appropriate length prior to the next insertion motion. The needle extraction motion is performed in a closed-loop fashion using the needle pose estimator. The frequency of needle pose estimator for feedback was measured as 0.67 Hz. The re-grasp point and the axis around which to rotate and remove the needle are also determined through visual estimation.

The needle extraction motion is composed of two main components. At the start of the routine, the left needle driver is positioned at the upper left edge of the workspace, above the tissue phantom. The point closer to the left needle driver among the two needle endpoints, determined as described in 3.4, is defined as the re-grasp point. The left needle driver is moved to a point offset by 1 cm horizontally from the defined re-grasp point (Fig. 5.2(d). Then, the left needle driver is moved in the direction of the re-grasp point by 1.5 cm, and the gripper jaws are closed.

Once the needle has been grasped by the left needle driver, it is rotated by 80° about the estimated needle axis to extract the needle and to minimize tissue damage (similar to the "rotate in") motion in the needle insertion routine. At the end of this action, only a small bit of the needle remains inside the tissue so that the needle can be fully extracted by a linear motion (Fig. 5.2(e)).

Suture Cinching

To ensure each suture is properly tensioned, suture cinching (tightening) is performed after needle extraction. The length of the thread that needs to be pulled for suture cinching in the extraction motion can be defined as $\beta = l_{des}$ - $(i$ - $1)\times l_{each}$ depending on the number of sutures (Fig. 5.2(e)), where l_{des} represents the desired thread length for the final suture, i represents the number of sutures, and l_{each} is the length of thread used for a single suture.

Needle Handover

The goal of the needle handover motion is to transfer the extracted needle from the left needle driver to the right needle driver in preparation for the next insertion. Once the pose of the needle is estimated, we define the re-grasp point in handover as the endpoint further from the left needle driver, similar to the process in 5.3. The right needle driver opens its jaws and moves to a point offset by 1 cm horizontally from the defined re-grasp point (Fig. 5.2(f)). After moving by the offset in the direction of the re-grasp point, the right needle driver closes to grasp the needle, (Fig. 5.2(g)) and the left needle driver opens to release it (Fig. 5.2(h)). The entire sequence is done in a closed-loop fashion with visual feedback tracking the pose of the needle. If the right needle driver moves into position to grasp the needle and the detected orientation of the needle remains unchanged, we pull the driver back, add a small (< 0.5cm) random horizontal offset, and reattempt the grasp, up to 5 additional times before declaring the handover a failure.

Needle Pose Correction

Since the pose of needle right before insertion significantly affects the insertion, we use interactive perception to improve needle tracking so it can be actuated to the ideal pose for the subsequent insertion. We adjust the needle for the insertion using a needle pose correction algorithm, which consists of three steps. First, the needle is moved to an optimal detection location at the lower, back, right corner of the workspace for RAFT-Stereo reconstruction as shown in Fig. 5.2(i). The lower back right corner was chosen empirically for consistent needle pose estimates. An ablation study confirmed this choice, showing 90% success rate in this corner for left gripper-held needles and 70% for right gripper-held needles, compared to 50% success in random positions. Identifying both endpoints is crucial for insertion, so obtaining pose estimates in the chosen corner maximized successful suture throws. In the next step, we sample 10 measurements of the needle normal vector, and rotate the gripper such that the normal vector of the needle is identical to the positive y axis in robot frame (Fig. 5.2(j)). This step constrains the discrepancy in orientation to be about the positive y-axis, which makes it more repeatable. Because the needle configuration during handover is relatively similar for each throw, the final step is a 90 degree rotation about the y-axis as seen in Fig. 5.2k. At this point, the STITCH pipeline is ready for the next suture throw insertion.

Motion Failure Recovery

The STITCH algorithm includes recovery mechanisms for both extraction motion failures and handover motion failures. For the extraction motion, the algorithm compares the positions of the needle endpoint before and after extraction. If the difference is below 2 cm, the extraction motion is retried up to 5 times. For the handover motion, the algorithm compares the normal vector of the needle before and after moving the right needle driver when both

Table 5.1: Success Metric Comparison across Ablations for 15 Trials.

	Mean Sutures to Failure	Single-Suture Success Rate	Three Throw Success Rate	Full Wound Success Rate	Mean Time per Suture	Error types I	E	H	T	Mean Sutures to Intervention
Sensing Only	1.40	51.6%	0.0%	0.0%	106.8 sec	9	6	0	0	–
Thread Handling	1.80	55.9%	20.0%	0.0%	117.9 sec	6	5	2	2	–
STITCH	2.93	69.4%	73.3%	0.0%	159.3 sec	8	5	0	2	–
STITCH + Human	4.47	83.3%	100.0%	20.0%	141.9 sec	16	10	0	2	2.25

needle drivers have grasped the needle. If there is a nonzero change of the normal vector, then the handover motion is retried up to 5 times.

5.4 Physical Experiments

Experimental Setup

The experimental setup consists of a bi-manual dVRK robot, a soft tissue phantom consisting of a single wound from a 3-Dmed directional suture pad featuring parallel linear wounds, and a fixed RGB stereo camera pair. For our experiments, the stereo cameras are a pair of Allied Vision Prosilica GC 1290 industrial cameras, which each output images of size 1280x960 at 33 frames per second. The Edmund Optics lenses mounted on the sensors allow for precise adjustments to the focus and aperture based on our workspace depth and illumination. The cameras are angled at the phantom such that the full workspace of the robot is captured in the field of view. We define the workspace using a Cartesian (x, y, z) coordinate system. We use violet Polysorb^{TM} surgical suture thread from Covidien. The threads are of variable length between 10 and 40 cm, with 2-0 USP Size (0.35-0.399 mm in diameter) and are attached to a GS-21 half-circular surgical needle with a radius of 12 mm.

Baselines

We evaluate the full STITCH algorithm and two baselines, multi-throw suturing without thread manipulation or needle pose correction "Sensing Only") and multi-throw suturing thread management without needle pose correction ("Thread Management"). The sensing-only baseline performs the needle insertion, extraction, and handover steps, omitting the suture cinching, thread sweeping, and needle pose correction motions. The thread management baseline performs insertion, extraction (with thread sweeping), cinching, and handover while omitting the needle pose correction step. We also evaluate a human-supervised setting ("STITCH + Human"), in which the robot can request intervention up to 2 times from a supervising surgeon. At these points, the human supervisor performs a single recovery motion to return the workspace to a safe configuration, and then immediately returns control to the robot.

Trial Specifications and Error Classification

Each trial produces n successive running suture throws. A suture throw is considered complete when the needle has successfully been inserted and extracted through the raised edges of the wound and the thread has been sufficiently tensioned to close the wound between the insertion and extraction points. A trial ends when the robot either enters an unrecoverable state (such as a dropped needle or tangled thread preventing further sutures) or successfully closes the wound by throwing 6 consecutive sutures. For each experimental baseline, we perform 15 trials of 1–6 throws each, and report the following metrics in Table 5.1:

- *Mean sutures to failure*: the average number of successfully completed suture throws before the first unrecoverable error is encountered.
- *Single-suture success rate*: the percentage of successful suture throws out of the total number of suture throw attempts.
- *Three throw success rate*: the percentage of trials which terminate after at least three successful suture throws.
- *Full wound success rate*: the percentage of trials which terminate in wound closure.
- *Mean time per suture*: the average time elapsed per suture throw.
- *Error Types*: the number of insertion (I), extraction (E), handover (H), and thread management (T) errors encountered.
- *Mean sutures to intervention*: The average number of autonomously completed suture throws between human intervention requests.

We report trial-ending errors according to the portion of the pipeline during which the failure occurred. We use the following schema:

- *Insertion error*s occur when the robot fails to insert the needle through the raised edge of the wound, or the needle enters a non-wound region of the phantom, or the needle exits the wound through the top or bottom of the phantom.
- *Extraction error*s occur when the needle remains in the wound after extraction.
- *Handover error*s occur when the robot drops the needle during handover or enters an unrecoverable configuration during the handover process.
- *Thread management error*s occur when the robot fails to properly cinch the suture thread to close the wound or becomes dangerously entangled in the thread.

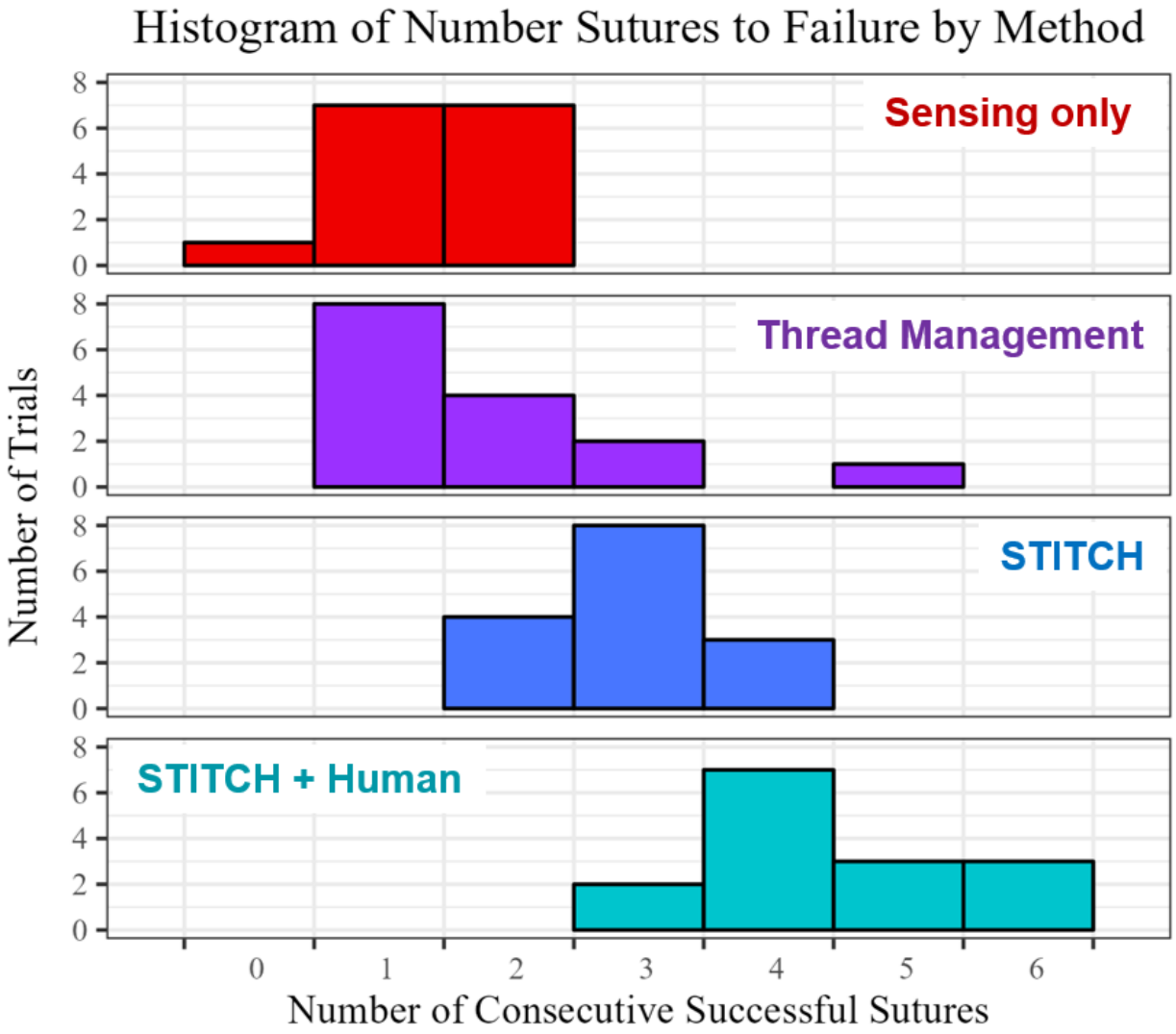

Figure 5.3: **Histogram of the Number of Sutures to Failure by Method.** The results in the histogram shown above is for 15 trials per each of the four methods.

Experimental Results

Table 5.1 and Fig. 5.3 report experimental results. Over 15 trials, STITCH achieves an average single-suture success rate of 69.39% and a mean sutures-to-failure of 2.93. When allowing the robot to request human intervention, the robot achieves a single-suture success rate of 83.33%, and a mean sutures-to-failure of 4.47.

Chapter 6

Case Studies in Surgical Augmented Dexterity: Vascular Shunt Insertion

Vascular shunt insertion is a surgical procedure that uses a hollow, flexible shunt tube to drain or divert fluid in the human body from one location to another [9]. Vascular shunt surgeries often take place in high-pressure clinical scenarios such as civilian and battlefield settings, where vascular injuries are common and a shunt can be utilized to bridge severed blood vessels [104, 82]. In most vascular shunt surgeries, the insertion is performed by a two-doctor team: a surgical assistant grasps the vessel at 2 points and orients it, while the surgeon grasps a third point on the vessel to dilate it and insert the shunt.

6.1 Background

There are 5 key subtasks where robotic assistance may be provided during a vascular shunt procedure:

1. The initial grasp of the blood vessel rim;
2. A secondary grasp of the blood vessel rim, with one grasping point already present;
3. A tertiary grasp of the blood vessel rim, with two grasping points already present;
4. Dilation of the blood vessel rim by pulling the third grasp point outward from the rim center;
5. Insertion of the shunt into the dilated vessel.

In prior work, Garcia et al. [28] present a semi-automated telerobotic surgical system where RSAs are remotely operated for phantom vascular shunt insertion operations. Recently, Dharmarajan et al. [23] propose an automated vascular shunt insertion solution with a da Vinci Research Kit (dVRK) where the RSA dilates a pre-grasped vessel and inserts the

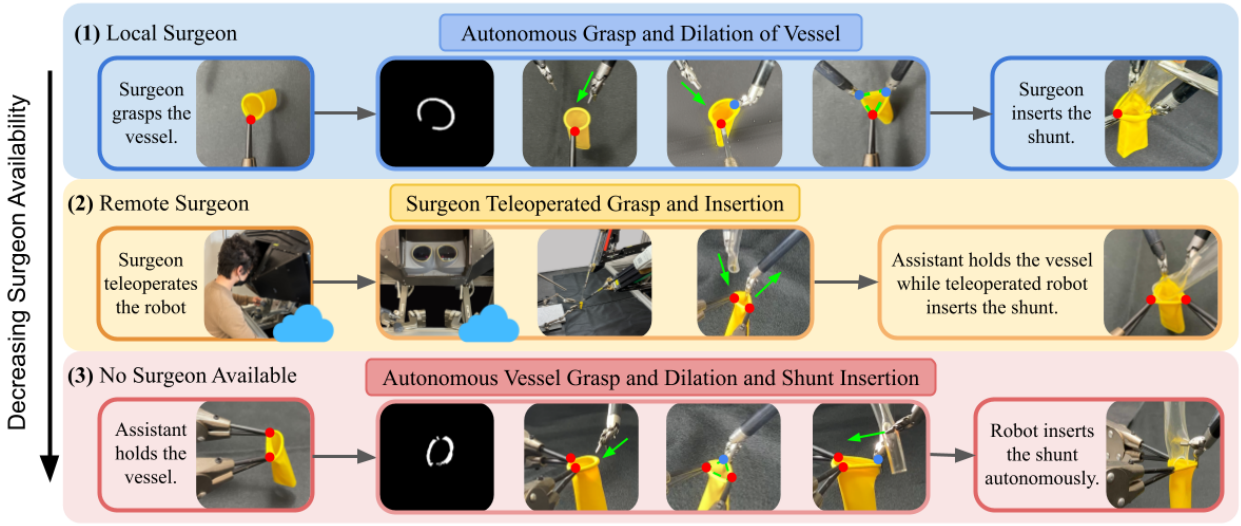

Figure 6.1: **Model Overview:** We consider three scenarios: (1) A local surgeon grasps one point (red) of the vessel (yellow balloon) while the robot autonomously grasps and stretches two points (blue) on the vessel with help of the vessel segmentation and visual servoing modules. The vessel's initial segmentation, steps of the sequential visual-servoed rim grasps, the vessel post autonomous grasps and dilation, and the surgeon-conducted shunt insertion are shown; (2) A remote surgeon (left) teleoperates the robot (middle) through the master tool manipulator (middle-left) to grasp and stretch the third point on the vessel rim and insert the shunt (transparent tube); (3) No surgeon is available. The robot automatically grasps the third point, and dilates and inserts the shunt into the vessel, utilizing the vessel segmentation, visual servoing, and shunt insertion modules. The vessel's initial segmentation, the visual-servoed rim grasp, the vessel post rim grasp and dilation, and the autonomous shunt insertion are shown; The surgeon-grasped point (red) in (1) and the two assistant-grasped points (red) in (2) and (3) are held by passive clamps.

shunt. This method's success rate varied from 50-80% depending on the size and orientation of the vessel phantom. In [21], we extend the role of the RSA from only performing shunt insertion to also executing bimanual vessel grasping and teleoperation in the form of a trimodal framework.

The work presented here also builds on our previous conference paper [21]. In [21], the vessel phantoms used for evaluation were yellow latex balloons with a 15mm inner diameter. However, their distinct rim, bright yellow color, and large size are unrealistic features for a blood vessel. Using the sensing and actuation techniques detailed above, we have improved the previous trimodal framework by introducing refined learning components, and we systematically evaluate this pipeline with additional vessel phantom types, including more realistic colors and materials. Specifically, we test the system on a smaller 9mm inner diameter red

vessel phantom for its smaller size and blood-like color, along with an 8mm inner diameter femoral artery phantom from LifeLike Biotissue [53] for its realistic material physics and size. These more realistic phantoms illustrate the robustness of the framework.

In addition, to facilitate safe robot learning and testing in future works, we develop and present a shunt insertion environment using the NVIDIA Isaac ORBIT-Surgical simulation framework. We collect a dataset of 1000 simulated trajectories using a state machine and empirically demonstrate the utility of this dataset by playing back the trajectories on a real dVRK.

6.2 Problem Statement

We consider three potential operational scenarios for robot-assisted vascular shunt insertion (local surgeon, local surgical assistant, and remote surgeon) and propose a paradigm for robotic assistance in each scenario. We provide a method for the execution of each paradigm and evaluate it based on success rate and completion time.

Assumptions

Initially, we assume that one or two passive grippers hold a vessel phantom, which resembles the role of a surgeon (Fig. 6.1(1)) or a surgical assistant (Fig. 6.1(2), Fig. 6.1(3)) in a surgical setting without robots respectively. We assume access to an inclined RGBD camera with known calibration with respect to robot arm coordinate frames. We also assume access to a stereo camera or endoscope that allows a human teleoperator to have a view of the workspace. In addition, if remote teleoperation is being performed, we assume that the network connection is stable and has low latency.

We assume that the size of the shunt used in the operation is known, and if the robot is manipulating it, the shunt is held at a known position. We assume the outer radius of the shunt is smaller than or equal to the inner radius of the vessel phantom.

Objectives and Evaluation Metrics

We consider three modes of operation (Fig. 6.1). For each, we evaluate success and completion time.

Mode (1): Local Surgeon. The surgeon is available at the operation site, and the bimanual surgical robot performs the role of a medical assistant, where it autonomously grasps two points on the vessel phantom given one point grasped by a fixed gripper. A trial is considered successful when both of the robot's grippers are grasping the vessel phantom.

Mode (2): Remote Surgeon. The bimanual surgical robot is teleoperated by a remote surgeon, and a human medical assistant is available locally to grasp the vessel. In particular, we assume two points on the vessel rim are held by fixed grippers, and the teleoperated robot

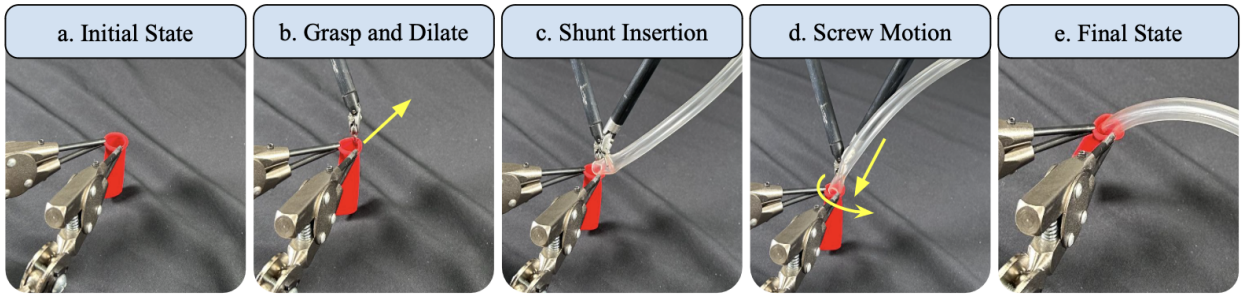

Figure 6.2: **Shunt Insertion Module** containing the Chamfer Tilt Shunt Insertion and the Screw Motion. (a) Starting from a surgical assistant holding two points on the vessel (red balloon), (b) the robot visual servos and grasps a third point on the vessel and dilates it to enlarge the opening. (c) It employs a chamfer tilt insertion motion to insert the shunt and (d) uses a screw motion to screw the shunt inside. (e) After the shunt is securely inserted, the robot releases the grasps of both grippers.

grasps a third point, dilates the vessel phantom, and inserts a shunt. A trial is successful if the shunt rim is fully enclosed within the vessel after both grippers release.

Mode (3): No Surgeon Available. This is an extended version of the case considered by Dharmarajan et al. [23] when there is no surgeon available but a human medical assistant is available locally to grasp the vessel, the bimanual surgical robot performs the teleoperated shunt inserter role, where it autonomously grasps the vessel rim on a third point, dilates, and inserts a shunt. A successful trial is defined in the same way as Mode (2).

6.3 Method

Overview

The system consists of 4 primary autonomous capabilities: a vessel phantom rim pose estimator, a grasping and visual servoing module, a shunt insertion module, and a teleoperation module. These components can be swapped out and reordered based on the required mode of operation, enabling the system to be effectively adapted to various on-the-ground situations.

Blood Vessel Rim Pose Estimation Module

Before performing any autonomous interaction with the vessel, such as grasping or dilation, the system must first detect and characterize the vessel state in space. We utilize a two-step process for 3D localization: segmentation mask generation and curve fitting. Using the visual detection framework described in Section **??** and expanded on in Section **??**, we use

a learning pipeline to obtain a 6D estimate of the position of the vessel rim in space, which we pass to the subsequent modules as input.

Servoing and Grasping Module

The servoing and grasping module initially presented in [21] uses the sensed 3D location and orientation of the vessel rim and attempts to actuate the robot to an intended grasping point on its rim. An open-loop policy calibrated using the method outlined in Seita et al. [97] is used to actuate to the intended 3D location for grasping. Once the gripper is within 2 mm horizontally of its target, control is transferred to a visual servoing policy, which determines the actuation. This policy is outlined broadly in Section 4.2.

Since there are two arms, each with a similar task that does not depend on the other, grasping can be executed either sequentially or concurrently. In sequential execution, one arm first servos and grasps the vessel rim, and then the second arm follows. In concurrent execution, both arms can be run simultaneously to reduce the time it takes to perform bimanual vessel grasping. In the concurrent version of servoing, the policies π_{right} and π_{left} each retrieve a corresponding cropped image from the camera, compute the desired actions, and execute the desired actions on separate threads. Once both grippers are finished servoing, they move downward and grasp the vessel contemporaneously. Once a successful grasp is performed, the grippers simultaneously move outward from the center to tension the vessel rim. We report experiments with both sequential and concurrent movements.

Shunt Insertion Module

The shunt insertion module starts at the state where the vessel phantom is grasped by two fixed grippers. The blood vessel rim pose estimation module (Section 6.3) outputs a 3D circle estimate of the vessel rim with RANSAC. The servoing and grasping module (Section 6.3) takes the circle estimate as input, actuates one gripper to move above a third computed point on the rim, runs visual servoing to line up the gripper with the rim, moves downward, and grasps the rim as shown in Fig. 6.2(b). Dilation is then performed away from the center of estimated circle rim.

After the dilation step, the rim of the vessel phantom is enlarged, as indicated by the yellow arrow in Fig. 6.2(b). The gripper and any fixed points used to tension the vessel now become obstacles that must be avoided during the insertion of the shunt, reducing the action space. The vessel phantom also cannot be further stretched due to the possibility of slip or tearing, further reducing available actions. To overcome the challenges associated with these low tolerances, the insertion module makes use of the chamfer tilt–screw motion insertion combination proposed in Dharmarajan et al. [23] to insert tightly fitting shunts. When the shunt is being inserted, no visual servoing is used, but instead the initial 3D vessel rim estimate from RANSAC is used. Once the chamfer tilt and screw motion are completed, both grippers release and return to their starting poses, leaving the shunt inside of the vessel.

Chamfer Tilt Insertion

The robot approaches the rim from above with the shunt held at an angle. The end effector is then actuated to a point slightly above the vessel phantom rim, and then moves downward, inserting the leading edge of the tilted shunt below the lip of the vessel. Once the shunt is partially seated below the rim, the end effector tilts to straighten the shunt, while at the same time, the arm dilating the vessel moves upward and inward to improve quality of the fit, as in Fig. 6.2(c).

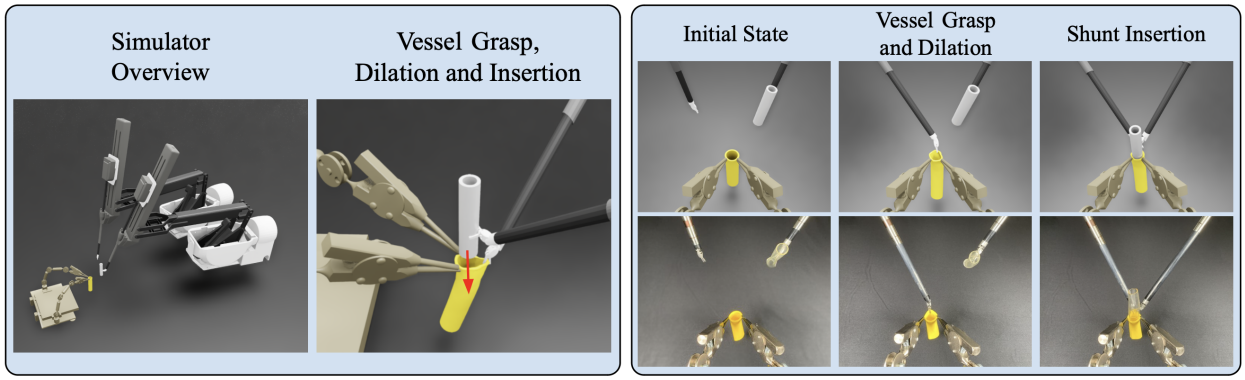

Figure 6.3: **Shunt Insertion Simulation Environment in ORBIT. Left:** The vascular shunt insertion setup with the dVRK and two mounted arms (left) and the vessel deformation post dilation and pre insertion (right) are shown. **Right:** Different stages of a vascular shunt insertion trajectory. Scenes pre and post vessel grasps and dilations and during shunt insertions in both the ORBIT simulator (top) and the real-world dVRK setup (bottom) are shown with a sim to real trajectory playback. See more details in Section 6.3.

Screw Motion

In some cases, after the chamfer tilt insertion motion is completed, a portion of the shunt remains outside the vessel rim. In this case, chamfer tilt alone is not sufficient to ensure that the shunt remains in place after both grippers release. To increase the probability that the entire shunt is situated within the rim of the vessel, the robot executes a screw motion, which is a counterclockwise rotation about the shunt axis combined with a concurrent downward translation, as shown by the yellow arrows in Fig. 6.2(d). This motion helps bring any portion of the shunt that was previously above or outside the rim inside. After the completion of this motion, both grippers release their grasps and retract to a home position away from the insertion site.

Teleoperation Module

In the teleoperation mode, a human teleoperator uses the dVRK controller with two tool manipulators and foot pedals to control both grippers of the surgical robot. One gripper first grasps and dilates the vessel, and then the other gripper, already holding a shunt, inserts the shunt into the vessel.

When performing remote teleoperation, the latency of the network must not be too high. In a study by Lum et al. [58], where the latency increased from 150 ms to 1000 ms, the teleoperator performed increasingly worse with greater mental fatigue on the task of surgical peg transfer [20]. Based upon a successful telesurgery procedure performed by Marescaux et al. [64], where the distance between the teleoperator and the robot was 24,000 km and the mean network latency was 155 ms. Based on these results, we set the network latency threshold to 155 ms. If the latency is less than or equal to the threshold, the teleoperation module will run; otherwise, the automated shunt insertion module from Mode 3 will run.

ORBIT Shunt Insertion Environment

We develop an open-access simulation environment built on top of NVIDIA Isaac Orbit [68] as shown in Fig. 6.3 for vascular shunt insertion task, which has the same setup as in the real environment. The simulation environment consists of a dVRK, two mounted arms mimicking a medical assistant holding a deformable phantom, and a rigid shunt as in the real setup. In addition, Orbit provides an interface for human teleoperation, allowing the data collection of human experts.

The physical dimensions of the mounted arms, vessel phantom, and shunt match the real world. We model the deformable vessel phantom with FEM and manually tune the parameters using videos of real world shunt insertion trials. We simulate the shunt using rigid material to closely mimic the stiff nature of the actual shunt. In addition, a rigid shunt provides a higher simulation speed as well as more stable simulated interactions with other objects including the vessel and the robot grippers.

We define a focus point inside the shunt, located above the center of the bottom of the shunt. This ensures that the distance between shunt focus point and the center of vessel phantom rim can only be minimized when the shunt is inserted into the vessel. The observation space of the environment includes the state of the dVRK in joint space (normalized by its range along each dimension), the position of the robot end effectors, positions of the center of shunt bottom, orientation of the shunt, positions of 8 points evenly distributed on the rim of the vessel phantom, and the position of the center of the rim of the vessel phantom. The action space of the environment includes the desired positions and orientations of the end effectors of the dVRK. All positions are represented in cartesian coordinates with respect to the workspace frame. The environment reward is a weighted combination of a term proportional to the inverse distance between the shunt focus point and the center of vessel top, and a term based on the extent to which the vessel is stretched. The episodes are terminated upon an environment timeout of 219 steps.

Parameter Tuning

To facilitate recreation of our simulation environment, we categorize the system and physics parameters of the simulation. To determine their values, we first set values that ensure simulation stability, then tune each individually to reduce the computation cost as much as possible while preserving stability. We list the parameters that are most important to tune for realism.

System Parameters:

1. Physics Simulation Frequency: We use 400 Hz, as this is the lowest frequency that does not induce jitter.
2. Solver Type: This determines the type of physics solver to use. We find that both available solvers, PGS (Projective Gauss-Seidel) and TGS (Truncated Gauss-Seidel) offer similar levels of stability and realism in simulation. We use TGS.

Physics Parameters:
I. Rigid Body - Shunt:

1. Contact Offset and Rest Offset: These parameters control contact generation. Contact points are generated when two objects get closer than the sum of their contact offsets. The rest offset quantifies how close an object gets to others at rest. An "autocompute" feature is provided by Nvidia IsaacSim. By default, the two offsets, re-meshing resolution, and triangle count can all be computed automatically by the simulation engine based on the objects' mesh geometry. For the shunt model, we use 0.0003 m for contact offset and 0.0002 m for the rest offset, which provide accurate and stable interactions in simulation.
2. Friction (μ_s and μ_k): We use 0.9 for both static (μ_s) and dynamic (μ_k) multipliers on base friction calculations, which best matched materials used.

II. Soft Body - Vessel:

1. Simulation Mesh Resolution: The simulation mesh resolution determines the resolution of soft simulation. We find this parameter is sensitive. We use a value of 10 for the vessel. Decreasing the parameter by a value greater than 3 can lead to a badly simulated collision mesh, a mismatch between visual and collision mesh, or insufficient deformation of the vessel mesh. The simulation mesh resolution needs to be carefully tuned.
2. Solver Position Iteration Count: Similarly to (I.1), we find this parameter is not sensitive above a certain threshold. Solver position iteration counts ranging from 1 to 128 provide similar simulation behavior. We use a `solver_position_iteration_count` of 16 for our current implementation, as this is the minimum value that ensures stability.

3. Contact Offset and Rest Offset: As explained in (I.2), offsets are used to compute contact. For the vessel, the autocomputed offsets lead to unstable simulation. Similarly to the shunt model, we instead use 0.0003 m for the contact offset and 0.0002 m for the rest offset. Increasing the rest offset can lead to a mismatch between visual and collision mesh.

4. Remeshing Resolution and Triangle Count: These parameters are used to compute the collision mesh. We use the autocomputed remeshing resolution and triangle count which gives a stable simulation.

Shunt Insertion Data Collection

To collect data of shunt insertion trajectories, we build a state machine to conduct the shunt insertion task. At each timestep, the state of this machine is determined based on the ground truth observations from the environment. Depending the current state, suitable actions for the robot are determined and conveyed to environment for execution. The state of the environment, including observations, rewards and terminals from the environment as well as the actions are recorded into the dataset. The actions designed in the state machine are based on absolute cartesian space positions. By utilizing the data buffer designed in ORBIT, we can read the joint state of the robot at each timestep. Therefore, we are able to efficiently collect trajectories with either cartesian-space actions or joint-space actions, enhancing the breadth and depth of the data collected for shunt insertion trajectories.

In total, we collect 1000 action trajectories of length 219 in Isaac ORBIT simulation environments. We collect 500 trajectories each for the phantom vessel with an inner diameter of 15 mm, and shunt outer diameters of 14 mm and 12.5 mm, using a policy trained with the environment and observations detailed in Section 6.3. All trajectories begin with the shunt grasped by the dVRK in the right gripper. The starting position of the shunt is randomized within a 4 cm sidelength axis-aligned cube, while the starting position of the tensioning arm is randomized within an axis-aligned rectangular prism measuring 8 cm in the x and y dimensions and 4 cm in the z direction. The clamp assembly holding the vessel phantom always starts at a fixed position.

The collected simulation dataset can be easily integrated into any existing learning library. This eases the community for future policy training either by supervised learning or offline reinforcement learning through the ORBIT environment interface.

6.4 Experiments

Experimental Setup

Once again, we perform experiments using the da Vinci Research Kit (dVRK) surgical robot with two cable-driven patient-side manipulator (PSM) arms. For autonomous roles, the robot captures RGBD images at a 1920x1200 resolution with 30 fps using an inclined Zivid

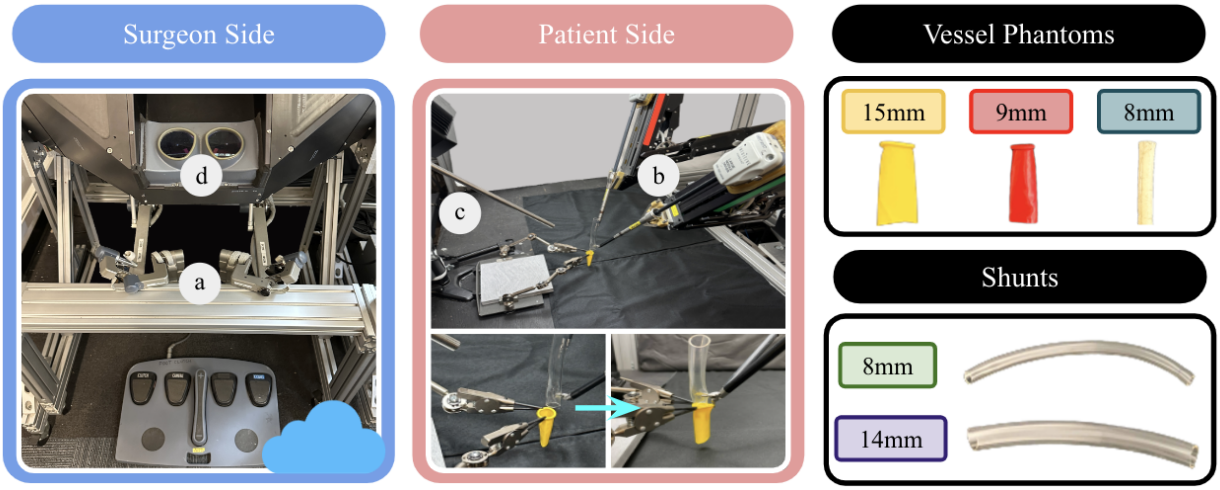

Figure 6.4: **Physical Experimental Setup.** The remote teleoperation setup includes: the two master tool manipulators (labeled "(a)"), which pass the motion commands from the remote surgeon to the the two patient side manipulators (labeled "(b)"), and an endoscope (labeled "(c)"), which captures binocular vision information and transmits it to the surgeon's console (labeled "(d)"). The goal of the teleoperation mode is to insert the shunt into the vessel phantom, as illustrated in the bottom right. The vessel phantoms used for evaluation are a 15mm inner diameter yellow balloon, a 9mm inner diameter red balloon, and an 8mm inner diameter femoral artery phantom (top right). The shunts used for evaluation are 8mm and 14mm plastic tubes.

One Plus S camera. The teleoperation interface consists of foot pedals, two master tool manipulators (MTMs), and a stereo viewer, as shown in Fig. 6.4. There are two mounted arms near the workspace to mimic a medical assistant holding the vessel. If the robot is performing shunt insertion, both are used to grasp the vessel, while only one is used if the robot is performing a bimanual grasp and dilation.

The vessel phantoms used for experiments include two yellow and red latex balloon stems, with inner diameters of 15 mm and 9mm respectively, and a femoral artery phantom with an inner diameter of 8 mm. The shunts used are two clear vinyl tubes with outer diameters of 8 mm and 14 mm as shown in Fig. 6.4.

Bimanual Vessel Grasping Metrics and Failure Modes

As described in Section 6.2, we consider a bimanual grasping trial successful if both of the robot grippers are grasping the vessel phantom. For this mode, we classify the failures into one of the following:

Vessel Phantom	Servoing	Execution Model	Success Rate	Avg trial time *(s)*	Failure Modes (O)	Failure Modes (T)
Yellow Balloon (15 mm ID)	N	Sequential	95%	13.2 ± 0.39	0	1
		Concurrent	65%	7.7 ± 0.37	7	0
	Y	Sequential	70%	17.4 ± 0.73	6	0
		Concurrent	100%	10.4 ± 0.42	0	0
Red Balloon (9 mm ID)	N	Sequential	15%	12.4 ± 0.24	17	0
		Concurrent	75%	7.3 ± 0.27	5	0
	Y	Sequential	65%	15.4 ± 1.07	7	0
		Concurrent	75%	9.7 ± 0.84	5	0

Table 6.1: **Mode 1: Bimanual Vessel Grasping Results:** Success rate and mean trial time for bimanual grasping with and without servoing, along with executing both arms' motions sequentially and concurrently. We track two failure modes: (O) One arm grasping failure and (T) Two arm grasping failure.

One-arm grasping failure (O)

Either the left or right arm attempts to grasp the vessel phantom, but misses. When the gripper is closed, there is no part of the vessel phantom inside it.

Two-arm grasping failure (T)

Both the left and right arms attempt to grasp the vessel phantom and miss.

Bimanual Vessel Grasping Results

We perform 20 trials of bimanual vessel grasping using the balloon phantom vessels, in which the dVRK autonomously grasps two points on the vessel rim, both sequentially and concurrently, as well as with and without servoing. In trials with visual servoing, an open-loop policy is utilized to actuate the gripper to the vicinity of the designated target location. This is then followed by the closed-loop visual servoing policy, which actuates the gripper to the precise target location. In trials without visual servoing, that same open-loop policy directly actuates the gripper to the target location. For all trials, a downward motion is subsequently executed to grasp the vessel's rim. For each set of 20 trials, the center of the vessel is placed at 4 different points for 5 trials each. The 4 points form a square with a side length of 2.54 cm. Results from these trials are reported in Table 6.1.

Yellow Balloon Vessel Phantom

For sequential bimanual grasping, we observe that the success rate declines from 95% to 70% after incorporating servoing modules, with 1 two-arm grasping failure (T) and 6 one-arm grasping failures (O) respectively. For concurrent bimanual grasping, we observe that the

success rate increases from 65% to 100% after incorporating servoing modules, and there are 7 one-arm grasping failures (O) and 0 failures respectively. We observe that on average, the duration of the concurrent execution of bimanual vessel grasping is less by 5.5 s in the no-servoing case, and 7.0 s in the servoing case.

For sequential execution, all of the one-arm failures (O) occurred on the second arm after the first arm had grasped the vessel. The initial single-arm grasp of the vessel results in the egocentric cropped image containing a tilted vessel rim with the gripper, which is out of distribution from the human demonstrations for servoing policy training and leads to incorrect servoing actions. This did not occur in the concurrent execution case where both arms servo simultaneously because the arms only move to grasp the vessel after they both finish servoing. These help explain why the addition of servoing module improves concurrent bimanual vessel grasping but hinders performance on its sequential counterpart.

Red Balloon Vessel Phantom

With the red balloon vessel phantom, the sequential execution model without servoing achieves a success rate of 15% with 17 single arm grasping failures (O). The large success rate drop off from the same yellow balloon phantom case occurred due to slight errors in the first grasp causing a larger overall change to the shape of the vessel phantom, since the red balloon's inner diameter is 6 mm smaller than the yellow balloon. When executing concurrently without servoing, the success rate of both grippers grasping the vessel phantom climbs to 75%, as the vessel phantom undergoes little shape change before both vessels are grasping the rim.

Bimanual grasping with servoing and a concurrent execution model achieved a 75% success rate, albeit with five one arm grasping failures. Notably, all five of those failures occurred in one of the four evaluation positions where the left gripper occluded the right gripper such that having cropped images around the right gripper made the policy execute actions as if it were in the left gripper's position (the right gripper consistently went too far to the right). Incorporating the visual servoing module in sequential executions increased the success rate by 50% contrasting the previous yellow balloon's case where servoing negatively impacts performance. This highlights the benefits of utilizing servoing in sequential executions under tight tolerances and non-circular rim shapes, despite its noisier and less robust servoing action outputs from out of distribution inputs. We believe performance can be improved in the future by augmenting our grasping dataset with demonstrations on single-arm grasped rims.

We observe that on average, the duration of the concurrent execution of bimanual vessel grasping is less by 5.1 s in the no servoing case, and 5.7 s in the servoing case.

Shunt Insertion Metrics and Failure Modes

We consider a shunt insertion trial, both for teleoperation as well as autonomous insertion, a success if one arm is able to dilate the vessel and the other arm is able to insert a shunt,

Vessel Phantom	Mode	Shunt Diameter	Success Rate	Avg trial time *(s)*	Failure Modes (D)	Failure Modes (S)
Yellow Balloon (15 mm ID)	2	8 mm	100%	13.6 ± 0.58	0	0
		14 mm	100%	20.3 ± 7.7	0	0
	3	8 mm	95%	14.5 ± 0.58	0	1
		14 mm	80%	14.4 ± 0.54	0	4
	Trajectory Replay	8 mm	100%	25.5	0	0
		14 mm	95%	22.0	0	1
Red Balloon (9 mm ID)	3	8 mm	95%	14.6 ± 0.18	0	1
Femoral Artery (8 mm ID)	3	8 mm	75%	14.7 ± 0.29	1	4

Table 6.2: **Modes 2 (Teleoperation) and 3 (Surgeon is not available): Shunt Insertion Results:** Success rate and mean trial time for shunt insertion with varying shunt outer diameters and insertion modes including replay of simulated trajectories. We track two failure modes: (D) dilation failure and (S) shunt insertion failure.

such that the rim of the shunt is fully enclosed when both grippers release. The elapsed time of each trial as well as the success or failure of that trial is noted. Failures can fall into two categories:

Dilation failure (D)

The robot, commanded with instructions from a teleoperator or through the autonomous pipeline, either attempts to grasp the vessel phantom and fails, or successfully grasps it but fails to dilate the vessel phantom rim outward.

Shunt insertion failure (S)

After both grippers release the vessel phantom and the shunt, if even a small portion of the shunt's rim is outside of the vessel phantom, it is considered a failure.

Teleoperation Results

For 20 trials of inserting the 8 mm and 14 mm outer diameter shunts into the yellow vessel phantom, one co-author (W. Panitch) served as the human teleoperator after 15 hours of experience. We report the results in Table 6.2. We observe that the human teleoperator has a 100% success rate for inserting both the 8 mm and 14 mm outer diameter shunts. The average trial time increased from 13.6 s to 20.3 s when the shunt outer diameter increased from 8 mm to 14 mm.

Autonomous Shunt Insertion Results

We perform 20 trials of autonomous shunt insertion when a surgeon is not available with both the 8 mm and 14 mm outer diameter shunts into the yellow vessel phantom, and report the results in Table 6.2.

We observe that the autonomous shunt insertion pipeline achieves a success rate of 95% with the 8 mm outer diameter shunt and 80% with the 14 mm outer diameter shunt. There were no dilation failures, but there were 1 and 4 shunt insertion failures respectively. The average time of each trial is 14.5 s and 14.4 s respectively. The decrease in the success rate of the larger 14 mm shunt can be attributed to the much tighter tolerance required for insertion when compared to the smaller 8 mm shunt.

The effect of the screw motion on the success rate of shunt insertion into the yellow vessel phantom is documented in [23]. When adding the screw motion to the result of the pipeline, the success rate of inserting the 14 mm shunt improved from 5% to 80%.

Red Balloon Vessel Phantom

We also perform 20 trials of autonomous shunt insertion into the red vessel phantom with only the 8 mm shunt, since the red vessel phantom has an inner diameter of 9 mm. We observe the success rate as 95% with the average trial time as 14.6 seconds.

Femoral Artery Vessel Phantom

Using the femoral artery vessel phantom, we perform autonomous shunt insertion with the 8 mm shunt. The success rate of the method is 75% with one dilation failure and four shunt insertion failures. The average trial time is 14.7 seconds.

The femoral artery phantom's rim is not as well defined as the balloon based phantoms, leading to a larger difficulty in rim perception. Specifically, the smaller rim has a lower pixel count in the blood vessel rim pose estimation module, leading to less points fed into the RANSAC algorithm. Noisier point clouds or slightly inaccurate segmentation masks can have a larger effect since not as many points may be on the rim. In the majority of cases, the masked depth points were sufficient for RANSAC to generate a reasonable estimation of the vessel phantom's rim. In the case where there was a dilation failure, the gripper missed the vessel rim, and in the shunt insertion failures, the shunt moved downward when it was completely outside of the vessel phantom. This suggests that with the small 8 mm inner diameter vessel phantom, accurate circle estimation can be an issue, leading to incorrect visual servoing action outputs and missed grasps and insertions.

To observe the effect of the shunt insertion method on a realistic vessel analogue, after performing all 20 trials, the femoral artery was visually inspected for damage. There were no tears in the vessel phantom. There were visible marks at the locations that were held by the two fixed grippers, but no such mark was present at the locations the dVRK gripper held. This suggests that the shape and clamping force of the fixed grippers contributed to the observed damage.

Simulated Trajectory Playback Results

To evaluate the utility of the simulated trajectories in the dataset, we randomly sample 40 trajectories out of 1000 and replay them on the real dVRK, with one example rollout shown on the right of Fig. 6.3. The vessel phantom used in all 40 trials was a yellow balloon with inner diameter of 15 mm. For half the trials, the real shunt size was 8 mm with a corresponding simulated shunt size of 12.5 mm, and for the other half, both the real and simulated shunt sizes were 14 mm. In each trial, we track the execution time in addition to the dilation (D) and shunt insertion (S) failure modes in Table 6.2.

Replaying the trajectories with the 8 mm shunt resulted in a 100% success rate with an average trial time of 25.5 seconds. Performing trajectory replay with the 14 mm shunt resulted in a 95% success rate, as the tighter tolerance led to one insertion failure. The mean trial time was 22.0 seconds. Notably, both cases had longer execution times compared to the other modes, as when following the trajectory, the robot ends with a zero velocity at each waypoint as the data contains positions but not velocities.

Chapter 7

Conclusion

In this paper, we present a unified framework for surgical automation using vision-based segmentation and control techniques and show how it can be adapted to two different surgical subtasks. Our results show that the proposed method achieves a high success rate across multiple tasks and settings varying in size, color, and material. These results suggest that autonomy-assisted surgery is a promising option for patients in situations where face-to-face surgical care is inaccessible or expensive. Furthermore, we discuss the development of a set of simulation environments designed in ORBIT to address the need for a surgical robotic simulator that accounts for deformable objects, and show that our environments enable realistic sim-to-real transfer of trained policies. With more development, the simulation has the potential to enable the use of modern Reinforcement Learning techniques to train autonomous surgical policies.

7.1 Limitations

Although our experiments demonstrate significant improvement over prior state-of-the-art methods in surgical task automation, there is still an incredible amount of work to be done before such systems will be of any use in clinical settings.

For example, we identify a number of common issues across both the suturing and vascular shunt insertion case studies:

Among the most common failure cases for both tasks involves high unexplained variance in the point clouds that are used for localization. Despite the two methods using different depth sensing techniques (one uses a projective depth camera, the other uses stereo vision), both pointclouds see poor resolution and high noise in the direction orthogonal to the camera sensor.

This is a particular challenge when correcting the needle orientation for the subsequent insertion in the suturing task. RAFT-Stereo struggles in high-disparity areas, so we move the needle to lower, back, right corner of the workspace as seen in Fig. 5.2(i). At this position, the perception algorithm can reliably detect the normal vector of the needle, allowing for the

first rotation correction step to align the normal vector with the positive y axis of the robot frame as seen in Fig. 5.2(j). However, the needle endpoint detection at this position still has a higher variance than desired. This means that we cannot perform the final rotation step based on needle endpoint feedback alone. In the future, we will investigate alternate stereo methods better tuned for small, reflective objects. With a more reliable stereopsis method and improved needle endpoint tracking, we hope to mitigate the insertion failure case by using the needle endpoint estimates to servo the needle to the optimal insertion orientation.

Similarly, using a fixed Zivid RGBD camera positioned above the workspace introduces issues in vascular shunt insertions involving small vessels with thin rims. In such cases, the low pixel count associated with thin rims leads to fewer data points available to accurately capture the rim geometry, leading the system to be less tolerant to the depth noise. This decrease in perception quality appears to consistently lead to a lower shunt insertion success rate, as evidenced by the case of the femoral artery phantom. Using more expressive 3D representations, such as multi-view reconstruction or NeRFs, might allow future researchers to reduce or eliminate the noise stemming from our imperfect 3-D representations.

Another particularly prevalent challenge is the lack of force feedback in the dVRK system. A common failure case in the suturing pipeline, for example, involves thread tensioning issues. Even with the thread-sweeping move, two potential thread failures are still present:

a. The sweeping move sometimes misses the thread and the extraction move will grab the thread with the endpoint causing system failure.

b. The later sutures run out of thread and fail because not enough thread was pulled during the initial suture throw extraction.

Thread tensioning and wound closure are inherently tactile tasks, involving a rich array of contact forces, and are therefore challenging to perform with visual information alone. Vessel tensioning, too, is performed with visual feedback only, which makes it difficult to directly consider the stress applied to the vessel during tensioning and could lead to tears in the vascular tissue. Some prior works have explored adding haptic feedback to human teleoperation or using motor voltages are a proxy for resistive force, but exploration of these techniques is reserved for future work.

Finally, this work largely assumes quasi-static scenes to allow for easier sensing and planning. However, to perform any in-vivo experiments, challenges like tissue deformation, visual changes due to blood and other bodily fluids, the actions of other surgeons or surgical assistants, and viewing angle variations must be addressed. We hope that future work will utilize the insights presented here to continue to build towards less expensive, more reliable surgical tools.

Bibliography

[1] Ahmad Abiri et al. "Multi-modal haptic feedback for grip force reduction in robotic surgery". In: *Scientific reports* 9.1 (2019), pp. 1–10.

[2] Rika Antonova et al. "Dynamic environments with deformable objects". In: *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*. 2021.

[3] Aleks Attanasio et al. "Autonomy in surgical robotics". In: *Annual Review of Control, Robotics, and Autonomous Systems* 4 (2021), pp. 651–679.

[4] Ruzena Bajcsy. "Active perception". In: *Proceedings of the IEEE* 76.8 (1988), pp. 966–1005.

[5] Ruzena Bajcsy, Yiannis Aloimonos, and John K Tsotsos. "Revisiting active perception". In: *Autonomous Robots* 42(2) (2018).

[6] Garth H Ballantyne and Fred Moll. "The da Vinci telerobotic surgical system: the virtual operative field and telepresence surgery". In: *Surgical Clinics* 83.6 (2003), pp. 1293–1304.

[7] Jeannette Bohg et al. "Interactive perception: Leveraging action in perception and perception in action". In: *IEEE T-RO* 33.6 (2017).

[8] Jeremy D Brown et al. "Using contact forces and robot arm accelerations to automatically rate surgeon skill at peg transfer". In: *IEEE Transactions on Biomedical Engineering* 64.9 (2016), pp. 2263–2275.

[9] Julia Cannon. *Shunt procedure*. Feb. 2018. URL: `https://www.hopkinsmedicine.org/neurology_neurosurgery/centers_clinics/cerebral-fluid/procedures/shunts.html`.

[10] Steven L Chang et al. In: *The Impact of Robotic Surgery on the Surgical Management of Prostate Cancer in the USA*. Vol. 115. BJU International, 2014.

[11] Zih-Yun Chiu et al. "Bimanual Regrasping for Suture Needles using Reinforcement Learning for Rapid Motion Planning". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021, pp. 7737–7743. DOI: `10.1109/ICRA48506.2021.9561673`.

[12] Zih-Yun Chiu et al. "Markerless suture needle 6d pose tracking with robust uncertainty estimation for autonomous minimally invasive robotic surgery". In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2022, pp. 5286–5292.

[13] Der-Lin Chow and Wyatt Newman. "Improved knot-tying methods for autonomous robot surgery". In: *2013 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE. 2013, pp. 461–465.

[14] CHQPR. *Rural Hospitals At Risk of Closing*. Oct. 2022. URL: https://ruralhospitals.chqpr.org/downloads/Rural_Hospitals_at_Risk_of_Closing.pdf.

[15] Bilal Chughtai et al. In: *National Trends and Cost of Minimally Invasive Surgery in Urology*. Vol. 2. Urology Practice, 2015.

[16] Erwin Coumans and Yunfei Bai. *PyBullet, a Python module for physics simulation for games, robotics and machine learning*. 2016–2021.

[17] Claudia D'Ettorre et al. "Automated pick-up of suturing needles for robotic surgical assistance". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 1370–1377.

[18] Tim Dall et al. *The Complexities of Physician Supply and Demand: Projections From 2019 to 2034*. June 2021. URL: https://www.aamc.org/media/54681/download.

[19] Michael Danielczuk et al. "Mechanical search: Multi-step retrieval of a target object occluded by clutter". In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 1614–1621.

[20] Anna M Derossis et al. In: *Development of a Model for Training and Evaluation of Laparoscopic Skills*. Vol. 175. 6. The American Journal of Surgery, 1998.

[21] Karthik Dharmarajan et al. "A Trimodal Framework for Robot-Assisted Vascular Shunt Insertion When a Supervising Surgeon is Local, Remote, or Unavailable". In: *International Symposium on Medical Robotics (ISMR)*. 2023.

[22] Karthik Dharmarajan et al. "Automating Vascular Shunt Insertion with the dVRK Surgical Robot". In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. 2023, pp. 6781–6788. DOI: 10.1109/ICRA48891.2023.10160966.

[23] Karthik Dharmarajan et al. "Automating Vascular Shunt Insertion with the dVRK Surgical Robot". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2023.

[24] Thomas G Dietterich et al. "Ensemble learning". In: *The handbook of brain theory and neural networks* 2.1 (2002), pp. 110–125.

[25] Xibin Dong et al. "A survey on ensemble learning". In: *Frontiers of Computer Science* 14 (2020), pp. 241–258.

[26] Fanny Ficuciello et al. "Autonomy in surgical robots and its meaningful human control". In: *Paladyn, Journal of Behavioral Robotics* 10.1 (2019), pp. 30–43.

[27] Martin A Fischler and Robert C Bolles. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". In: *Communications of the ACM* 24.6 (1981), pp. 381–395.

[28] Pablo Garcia et al. "Trauma Pod: a semi-automated telerobotic surgical system". In: *The International Journal of Medical Robotics and Computer Assisted Surgery* 5.2 (2009), pp. 136–146.

[29] Animesh Garg et al. "Tumor localization using automated palpation with gaussian process adaptive sampling". In: *2016 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE. 2016, pp. 194–200.

[30] K. Goldberg. "Augmented Dexterity: How Robots Can Enhance Surgeon Dexterity". In: *Science Robotics* (). Under Review (Summer 2023).

[31] Kenneth Y Goldberg and Ruzena Bajcsy. "Active touch and robot perception". In: *Cognition and Brain Theory* 7 (2) (1984).

[32] Glebys Gonzalez et al. "DESERTS: Delay-tolerant semi-autonomous robot teleoperation for surgery". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 12693–12700.

[33] Mohammad Haghighipanah et al. "Unscented kalman filter and 3d vision to improve cable driven surgical robot joint angle estimation". In: *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2016, pp. 4135–4142.

[34] Blake Hannaford et al. "Raven-II: an open platform for surgical robotics research". In: *IEEE Transactions on Biomedical Engineering* 60.4 (2012), pp. 954–959.

[35] Danying Hu et al. "Semi-autonomous image-guided brain tumour resection using an integrated robotic system: A bench-top study". In: *The International Journal of Medical Robotics and Computer Assisted Surgery* 14.1 (2018), e1872.

[36] Minho Hwang et al. "Applying depth-sensing to automated surgical manipulation with a da vinci robot". In: *2020 International Symposium on Medical Robotics (ISMR)*. IEEE. 2020, pp. 22–29.

[37] Minho Hwang et al. "Efficiently calibrating cable-driven surgical robots with RGBD fiducial sensing and recurrent neural networks". In: *IEEE Robotics and Automation Letters* 5.4 (2020), pp. 5937–5944.

[38] Minho Hwang et al. "Superhuman surgical peg transfer using depth-sensing and deep recurrent neural networks". In: *arXiv preprint arXiv:2012.12844* (2020).

[39] Intuitive Surgical. *SimNow Simulation Software*. URL: `https://www.intuitive.com/en-us/products-and-services/da-vinci/learning/simnow`.

[40] Russell C Jackson et al. "Real-time visual tracking of dynamic surgical suture threads". In: *IEEE Transactions on Automation science and Engineering* 15.3 (2017), pp. 1078–1090.

[41] Stephen James et al. "RLBench: The Robot Learning Benchmark & Learning Environment". In: *IEEE Robotics and Automation Letters* (2020).

[42] Yiwei Jiang, Haoying Zhou, and Gregory S. Fischer. "Markerless Suture Needle Tracking From A Robotic Endoscope Based On Deep Learning". In: *2023 International Symposium on Medical Robotics (ISMR)*. 2023, pp. 1–7. DOI: 10.1109/ISMR57123.2023.10130199.

[43] Neelay Joglekar et al. "Suture Thread Spline Reconstruction from Endoscopic Images for Robotic Surgery with Reliability-driven Keypoint Detection". In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2023, pp. 4747–4753.

[44] Dmitry Kalashnikov et al. "Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation". In: *arXiv preprint arXiv:1806.10293* (2018).

[45] Varun Kamat et al. "Automating 2D Suture Placement". In: *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*. IEEE. 2023.

[46] Peter Kazanzides et al. "An open-source research kit for the da Vinci® Surgical System". In: *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2014, pp. 6434–6439.

[47] Ben Kehoe et al. "Autonomous multilateral debridement with the raven surgical robot". In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2014, pp. 1432–1439.

[48] Azarakhsh Keipour, Maryam Bandari, and Stefan Schaal. "Deformable one-dimensional object detection for routing and manipulation". In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 4329–4336.

[49] W Kilby et al. "The CyberKnife® robotic radiosurgery system in 2010". In: *Technology in cancer research & treatment* 9.5 (2010), pp. 433–452.

[50] Hansoul Kim et al. In: *Effect of Backlash Hysteresis of Surgical Tool Bending Joints on Task Performance in Teleoperated Flexible Endoscopic Robot*. Vol. 16. The International Journal of Medical Robotics and Computer Assisted Surgery, 2020.

[51] Sanjay Krishnan et al. "Transition state clustering: Unsupervised surgical trajectory segmentation for robot learning". In: *International Journal of Robotics Research (IJRR)* (2017).

[52] Sergey Levine et al. "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection". In: *The International Journal of Robotics Research* 37.4-5 (2018), pp. 421–436.

[53] *LifeLike Biotissue*. https://lifelikebiotissue.com/about-us.

[54] Xingyu Lin et al. "SoftGym: Benchmarking Deep Reinforcement Learning for Deformable Object Manipulation". In: *Conference on Robot Learning (CoRL)*. PMLR, 2020.

[55] Lahav Lipson, Zachary Teed, and Jia Deng. "RAFT-Stereo: Multilevel Recurrent Field Transforms for Stereo Matching". In: *International Conference on 3D Vision (3DV)*. 2021.

[56] Bo Lu et al. "Toward image-guided automated suture grasping under complex environments: A learning-enabled and optimization-based holistic framework". In: *IEEE Transactions on Automation Science and Engineering* 19.4 (2021), pp. 3794–3808.

[57] Jingpei Lu, Florian Richter, and Michael C Yip. "Pose estimation for robot manipulators via keypoint optimization and sim-to-real transfer". In: *IEEE Robotics and Automation Letters* 7.2 (2022), pp. 4622–4629.

[58] Mitchell J.H. Lum et al. In: *TeleRobotic Fundamentals of Laparoscopic Surgery (FLS): Effects of Time Delay-Pilot Study*. International Conference of the IEEE Engineering in Medicine and Biology Society, 2008.

[59] Miles Macklin and Matthias Muller. "A Constraint-Based Formulation of Stable Neo-Hookean Materials". In: *ACM SIGGRAPH Conference on Motion, Interaction and Games*. New York, NY, USA, 2021.

[60] Miles Macklin et al. "Small Steps in Physics Simulation". In: *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 2019. DOI: 10.1145/3309486.3340247.

[61] Denys Makoviichuk and Viktor Makoviychuk. *rl-games: A High-performance Framework for Reinforcement Learning*. May 2021. URL: https://github.com/Denys88/rl_games.

[62] Viktor Makoviychuk et al. "Isaac Gym: High performance GPU-based physics simulation for robot learning". In: *arXiv preprint arXiv:2108.10470* (2021).

[63] Khaled Mamou, E Lengyel, and A Peters. "Volumetric hierarchical approximate convex decomposition". In: *Game Engine Gems 3*. CRC Press, 2016.

[64] Jacques Marescaux et al. "Transcontinental robot-assisted remote telesurgery: feasibility and potential applications". In: *Annals of surgery* 235.4 (2002), p. 487.

[65] Roberto Martín-Martín and Oliver Brock. "Coupled recursive estimation for online interactive perception of articulated objects". In: *IJRR* 41 (8) (2022).

[66] Alexander Mathis et al. "DeepLabCut: markerless pose estimation of user-defined body parts with deep learning". In: *Nature neuroscience* 21.9 (2018), pp. 1281–1289.

[67] Stephen McKinley et al. "An interchangeable surgical instrument system with application to supervised automation of multilateral tumor resection". In: *2016 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE. 2016, pp. 821–826.

[68] Mayank Mittal et al. "Orbit: A Unified Simulation Framework for Interactive Robot Learning Environments". In: *IEEE Robotics and Automation Letters* 8.6 (2023), pp. 3740–3747. DOI: 10.1109/LRA.2023.3270034.

[69] G.P. Moustris et al. In: *Evolution of Autonomous and Semi-Autonomous Robotic Surgical Systems: A Review of the Literature*. Vol. 7. International Journal of Medical Robotics and Computer Assisted Surgery, 2011.

[70] Adnan Munawar et al. "A real-time dynamic simulator and an associated front-end representation format for simulating complex robots and environments". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 1875–1882.

[71] Adithyavairavan Murali et al. "Learning by observation for surgical subtasks: Multilateral cutting of 3d viscoelastic and 2d orthotropic tissue phantoms". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2015, pp. 1202–1209.

[72] Prajval Kumar Murali et al. "Active visuo-tactile interactive robotic perception for accurate object pose estimation in dense clutter". In: *IEEE RAL* 7.2 (2022).

[73] Tamás Dániel Nagy and Tamás Haidegger. "Autonomous Surgical Robotics at Task and Subtask Levels". In: *Advanced Robotics and Intelligent Automation in Manufacturing*. IGI global, 2020, pp. 296–319.

[74] Yashraj Narang et al. "Factory: Fast contact for robotic assembly". In: *arXiv preprint arXiv:2205.03532* (2022).

[75] Tonci Novkovic et al. "Object finding in cluttered scenes using interactive perception". In: *ICRA*. IEEE. 2020.

[76] NVIDIA. *FleX: A particle-based simulation library*. May 2017. URL: `https://github.com/NVIDIAGameWorks/FleX`.

[77] Government Accountability Office. *Rural Hospital Closures: Affected Residents Had Reduced Access To Health Care Services*. Dec. 2020. URL: `https://www.gao.gov/products/gao-21-93`.

[78] Nicolas Padoy and Gregory D Hager. "3D thread tracking for robotic assistance in tele-surgery". In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2011, pp. 2102–2107.

[79] Samuel Paradis et al. "Intermittent visual servoing: Efficiently learning policies robust to instrument changes for high-precision surgical manipulation". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 7166–7173.

[80] Haonan Peng et al. "Real-time data driven precision estimator for raven-ii surgical robot end effector position". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 350–356.

[81] Antonin Raffin et al. "Stable-baselines3: Reliable reinforcement learning implementations". In: *The Journal of Machine Learning Research* (2021).

[82] Todd E Rasmussen et al. "The use of temporary vascular shunts as a damage control adjunct in the management of wartime vascular injury". In: *Journal of Trauma and Acute Care Surgery* 61.1 (2006), pp. 8–15.

[83] Florian Richter, Ryan K Orosco, and Michael C Yip. "Open-sourced reinforcement learning environments for surgical robotics". In: *arXiv preprint arXiv:1903.02090* (2019).

[84] Florian Richter et al. "Autonomous robotic suction to clear the surgical field for hemostasis using image-based blood flow detection". In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 1383–1390.

[85] Irene Rivas-Blanco et al. "Transferring know-how for an autonomous camera robotic assistant". In: *Electronics* 8.2 (2019), p. 224.

[86] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[87] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer. 2015, pp. 234–241.

[88] Sarahí Rosas Gonzalez et al. "Asymmetric Ensemble of Asymmetric U-Net Models for Brain Tumor Segmentation With Uncertainty Estimation". In: *Frontiers in Neurology* 12 (Sept. 2021), p. 609646. DOI: 10.3389/fneur.2021.609646.

[89] Paul T. Rose and Bernard Nusbaum. "Robotic Hair Restoration". In: *Dermatologic Clinics* 32.1 (2014). Advances in Cosmetic Dermatology, pp. 97–107. ISSN: 0733-8635. DOI: https://doi.org/10.1016/j.det.2013.09.008. URL: https://www.sciencedirect.com/science/article/pii/S0733863513000946.

[90] Nikita Rudin et al. "Learning to walk in minutes using massively parallel deep reinforcement learning". In: *Conference on Robot Learning*. PMLR. 2022, pp. 91–100.

[91] H. Saeidi et al. "Autonomous robotic laparoscopic surgery for intestinal anastomosis". In: *Science Robotics* 7 (62 2022), pp. 1–14. ISSN: 24709476. DOI: 10.1126/scirobotics.abj2908.

[92] Omer Sagi and Lior Rokach. "Ensemble learning: A survey". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8.4 (2018), e1249.

[93] Paul Maria Scheikl et al. "LapGym–An Open Source Framework for Reinforcement Learning in Robot-Assisted Laparoscopic Surgery". In: *arXiv preprint arXiv:2302.09606* (2023).

[94] Vincent Schorp et al. "Self-Supervised Learning for Interactive Perception of Surgical Thread for Autonomous Suture Tail-Shortening". In: *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*. IEEE. 2023.

[95] John Schulman et al. "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347* (2017).

[96] Kim L. Schwaner et al. "Autonomous Bi-Manual Surgical Suturing Based on Skills Learned from Demonstration". In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021, pp. 4017–4024. DOI: 10.1109/IROS51168.2021.9636432.

[97] Daniel Seita et al. "Fast and reliable autonomous surgical debridement with cable-driven robots using a two-phase calibration procedure". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 6651–6658.

[98] S. Sen et al. In: *Automating Multiple-Throw Multilateral Surgical Suturing with a Mechanical Needle Guide and Sequential Convex Optimization*. IEEE International Conference on Robotics and Automation (ICRA), 2016.

[99] Siddarth Sen et al. "Automating multi-throw multilateral surgical suturing with a mechanical needle guide and sequential convex optimization". In: *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2016, pp. 4178–4185.

[100] Azad Shademan et al. "Supervised autonomous robotic soft tissue surgery". In: *Science Translational Medicine* 8.337 (2016), 337ra64–337ra64.

[101] Changyeob Shin et al. "Autonomous tissue manipulation via surgical robot using learning based model predictive control". In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 3875–3881.

[102] Kaushik Shivakumar et al. "SGTM 2.0: Autonomously Untangling Long Cables using Interactive Perception". In: *arXiv preprint arXiv:2209.13706* (2022).

[103] Kaushik Shivakumar et al. "Sgtm 2.0: Autonomously untangling long cables using interactive perception". In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2023, pp. 5837–5843.

[104] Anuradha Subramanian et al. "A decade's experience with temporary intravascular shunts at a civilian level I trauma center". In: *Journal of Trauma and Acute Care Surgery* 65.2 (2008), pp. 316–326.

[105] Priya Sundaresan et al. "Automated extraction of surgical needles from tissue phantoms". In: *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*. IEEE. 2019, pp. 170–177.

[106] Surgical Science. *Mimic and Simbionix*. URL: https://surgicalscience.com/custom-made-simulation/robotic-surgery-simulation/.

[107] Eleonora Tagliabue et al. "Soft tissue simulation environment to learn manipulation tasks in autonomous robotic surgery". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 3261–3266.

[108] *Taurus: This small robot is reaching new heights and solving once-thought impossible challenges*. `https://medium.com/dish/taurus-this-small-robot-is-reaching-new-heights-and-solving-once-thought-impossible-challenges-e858fdbbb4ab`.

[109] Brijen Thananjeyan et al. "All You Need is LUV: Unsupervised Collection of Labeled Images using Invisible UV Fluorescent Indicators". In: *arXiv preprint arXiv:2203.04566* (2022).

[110] Brijen Thananjeyan et al. "All You Need is LUV: Unsupervised Collection of Labeled Images Using UV-Fluorescent Markings". In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2022, pp. 3241–3248.

[111] Brijen Thananjeyan et al. "Multilateral surgical pattern cutting in 2d orthotropic gauze with deep reinforcement learning policies for tensioning". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 2371–2378.

[112] Brijen Thananjeyan et al. "Safety augmented value estimation from demonstrations (saved): Safe deep model-based rl for sparse cost robotic tasks". In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 3612–3619.

[113] Emanuel Todorov, Tom Erez, and Yuval Tassa. "MuJoCo: A physics engine for model-based control". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2012, pp. 5026–5033. DOI: `10.1109/IROS.2012.6386109`.

[114] Vignesh Manoj Varier et al. "AMBF-RL: A real-time simulation based Reinforcement Learning toolkit for Medical Robotics". In: *2022 International Symposium on Medical Robotics (ISMR)*. IEEE. 2022, pp. 1–8.

[115] Vignesh Manoj Varier et al. "Collaborative Suturing: A Reinforcement Learning Approach to Automate Hand-off Task in Suturing for Surgical Robots". In: *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE. 2020, pp. 1380–1386.

[116] Vainavi Viswanath et al. "Learning to Trace and Untangle Semi-planar Knots (TUSK)". In: *arXiv preprint arXiv:2303.08975* (2023).

[117] Albert Wilcox et al. "Learning to localize, grasp, and hand over unmodified surgical needles". In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 9637–9643.

[118] Jiaqi Xu et al. "SurRol: An open-source reinforcement learning centered and dvrk compatible platform for surgical robot learning". In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021.

[119] Michael Yip and Nikhil Das. In: *Robot Autonomy for Surgery*. The Encyclopedia of Medical Robotics, 2017.

[120] Yuke Zhu et al. "robosuite: A Modular Simulation Framework and Benchmark for Robot Learning". In: *arXiv preprint arXiv:2009.12293*. 2020.