

Intermittent Visual Servoing: Efficiently Learning Policies Robust to Instrument Changes for High-precision Surgical Manipulation

*Samuel Paradis
Ken Goldberg
Joseph Gonzalez*



Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2021-104

<http://www2.eecs.berkeley.edu/Pubs/TechRpts/2021/EECS-2021-104.html>

May 14, 2021

Copyright © 2021, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

Acknowledgement

This work, as well as progress to this work, contains invaluable contributions from Minho Hwang, Brijen Thananjeyan, Jeffrey Ichnowski, Daniel Seita, and Danyal Fer.

Intermittent Visual Servoing: Efficiently Learning Policies
Robust to Instrument Changes for High-precision Surgical Manipulation

by

Samuel Paradis

A thesis submitted in partial satisfaction of the

requirements for the degree of

Master of Science

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Ken Goldberg, Chair

Professor Joseph E. Gonzalez

Spring 2021

The thesis of Samuel Paradis, titled Intermittent Visual Servoing: Efficiently Learning Policies

Robust to Instrument Changes for High-precision Surgical Manipulation, is approved:

Chair *Ken Goldberg*
Ken Goldberg (May 14, 2021 13:16 PDT)

Date 5/14/2021

Joseph E. Gonzalez

Date 5/13/2021

University of California, Berkeley

Intermittent Visual Servoing: Efficiently Learning Policies
Robust to Instrument Changes for High-precision Surgical Manipulation

Copyright 2021
by
Samuel Paradis

Abstract

Intermittent Visual Servoing: Efficiently Learning Policies
Robust to Instrument Changes for High-precision Surgical Manipulation

by

Samuel Paradis

Master of Science in Computer Science

University of California, Berkeley

Professor Ken Goldberg, Chair

Assisting surgeons with automation of surgical subtasks is challenging due to backlash, hysteresis, and variable tensioning in cable-driven robots. These issues are exacerbated as surgical instruments are changed during an operation. In this work, we propose a framework for automation of high-precision surgical subtasks by learning local, sample-efficient, accurate, closed-loop policies that use visual feedback instead of robot encoder estimates. This framework, which we call deep Intermittent Visual Servoing (IVS), switches to a learned visual servo policy for high-precision segments of repetitive surgical tasks while relying on a coarse open-loop policy for the segments where precision is not necessary. We train the policy using only 180 human demonstrations that are roughly 2 seconds each. Results on a da Vinci Research Kit suggest that combining the coarse policy with half a second of corrections from the learned policy during each high-precision segment improves the success rate on the Fundamentals of Laparoscopic Surgery peg transfer task from 72.9% to 99.2%, 31.3% to 99.2%, and 47.2% to 100.0% for 3 instruments with differing cable properties. In the contexts we studied, IVS attains the highest published success rates for automated surgical peg transfer and is significantly more reliable than previous techniques when instruments are changed.

Contents

Contents	i
List of Figures	ii
List of Tables	iv
1 IVS	1
1.1 Introduction	1
1.2 Related Work	4
1.3 Problem Definition	6
1.4 IVS: Method	8
1.5 Experiments	13
2 Conclusions & Future Work	18
2.1 Discussion of IVS	18
2.2 Extensions of IVS	18
2.3 Limitations of IVS	19
2.4 Reflections	19
3 Research Acknowledgements	20
Bibliography	21

List of Figures

1.1	Intermittent Visual Servoing (IVS). Visual Servoing Pipeline: To compensate for cable-related effects, a policy maps images of the workspace from a top-down RGB camera to corrective motions. Peg Transfer Example: The robot switches to a visual feedback policy for high-precision segments of the task (green), and uses a coarse policy to navigate between these segments (blue). Multiple Surgical Instruments: We experiment using 3 different large needle drivers, each with unique backlash, hysteresis, and cable tension properties. . . .	3
1.2	Robot Setup for FLS Peg Transfer. We use Intuitive Surgical’s da Vinci Research Kit robot. We use red 3D printed blocks and a red 3D printed peg board. We use a uniformly red setup to simulate the surgical environment, where color alone may not provide sufficient signal, as surgeons rely on minute differences in color, depth and texture to complete high-precision tasks. We use a top-down camera to generate open-loop trajectories (RGBD) and capture the input for visual servoing (RGB). The FLS peg transfer task involves transferring 6 blocks from the 6 left pegs to the 6 right pegs, and transferring them back from the right pegs to the left pegs. As each block’s opening has a 4.5 mm radius, and each peg’s cylindrical is 2.25 mm wide, the task requires high precision, making it a popular benchmark task for evaluating human surgeons.	6
1.3	Data Filtering. We preprocess the images in two ways: (1) crop a 150×150 image around the center of the target peg, and (2) color-crop out all red pixels outside of a block-sized radius from the center of the peg.	9
1.4	Supervision Extraction with Labeled Example. Top: The action a_t , indicated by the arrows, is a vector of length 1 mm (λ) in the direction $p_{t'} - p_t$ traveled by the expert trajectory, where p_t is the current waypoint and $p_{t'}$ is the next waypoint that is at least 1 mm (λ) distance from p_t . The extracted termination label, indicated by the color of the waypoint, is 1 if the distance to the final position in the trajectory p_T is less than 2 mm (ν), and 0 otherwise. Each waypoint corresponds to an image, and each image receives both a corrective action label and a termination signal label. Bottom: Preprocessed images (see Fig. 1.3) from a corrective place trajectory with labels extracted using method described above.	10

- 1.5 **IVS Example.** Example of IVS correcting positioning errors. **Top row (pick subtask):** IVS removes pickup error with 5 corrective updates in 0.6 seconds. We overlay small blue circles to highlight an optimal pick location on the block, defined as the graspable point furthest from the peg along the edge closest to the robot. The positioning of the robot is off due to the inaccuracy of the coarse policy, as the end-effector is not positioned over the block (Frame 1). We then switch to the learned policy, and visual servoing guides the end-effector over a pick point (Frames 2-5), and once determined a safe pickup is possible, picks the block successfully (Frame 6). **Bottom row (place subtask):** IVS removes placement error with 12 corrective updates in 1.2 seconds. We overlay small green circles to highlight the location of the pegs. The positioning of the robot is off due to the inaccuracy of the coarse policy, as the peg is not under the block (Frame 1). We then switch to the learned policy, and visual servoing guides the block over the peg (Frames 2-5), and once determined a safe situation to drop, places the block successfully (Frame 6).

List of Tables

1.1	Ablation Study Varying Number of Models in Ensemble. We compare IVS with differing number of models in the ensemble across 3 full trials of peg transfer (36 transfers). More models results in a more robust policy, but due to compute limitations, less frequent servoing. More frequent servoing results in higher precision, as both the corrective action and the termination signal are updated more frequently. The goal is to find a balance between maximizing robustness (many models) and maximizing update frequency (few models). We use an ensemble of 4 models.	12
1.2	Peg Transfer Baseline Comparison. Benchmark comparing performance of IVS to the baselines described in Section 1.5. IVS beats both baselines in terms of pick success rate, place success rate, and overall transfer success rate. Due to using RGB imaging, we are able to take many corrective steps per second without stopping, minimizing additions to the mean transfer time.	16
1.3	IVS Efficiency Benchmark. We analyze efficiency of IVS on 3 large needle driver instruments (see Fig. 1.1). On average, IVS moves the end-effector 4.1 mm, requiring an additional 1.2 seconds per transfer.	17
1.4	Instrument Transfer Comparison. Benchmark comparing performance on 10 full trials of peg transfer (120 transfers) across 3 different surgical instruments with unique cabling characteristics of the uncalibrated baseline, calibrated baseline trained on each instrument, and IVS trained on one instrument. IVS consistently beats both baselines, while remaining robust to instrument changes.	17

Acknowledgments

I extend a immense thank you to the countless people that believed me in, guided me, and allowed me to grow into the person graduating today. First, my advisor, Professor Ken Goldberg. Professor Goldberg took me in as a senior, and provided me with both the guidance to develop into a better researcher, and the freedom to explore my interests and discover surgical robotics. I will never forgot our fun Monday meetings, and the great food that accompanied them. Professor Goldberg taught me how to lead, how to set and maintain high expectations, and most importantly, how to always ask “so what’s next?” This work, as well as progress to this work, contains invaluable contributions from Minho Hwang, Brijen Thananjeyan, Jeffrey Ichnowski, Daniel Seita, and Danyal Fer. I greatly appreciate the time, effort, patience, and belief from each individual. This team gave me the confidence to ask new questions, and the unconditional support to answer them. I would also like to thank Professor Imke de Pater, Chris Moeckel, and Joshua Tollefson. Professor de Pater took a chance on me when I was first entering computer science as a sophomore; at the time, my abilities and knowledge were limited, but in time, I grew into the researcher with the confidence and experience to join Professors Goldberg’s automation laboratory. Professor de Pater, Chris Moeckel, and Joshua Tollefson conduct research with an infectious joy, and taught me that research is something to be enjoyed. I would not be standing here today without their guidance. Thank you all for giving me a chance. Lastly, thank you to my friends and family for their love and support. Thank you Ryan, Brian, Sid, Adam, Miles, Kyle, Steven, and Michael for your friendship. Thank you to my friend Garren for your company during the pandemic. Thank you to my parents, my brother Chris, and my girlfriend Jenette for a lifetime worth of love and support.

1. IVS

1.1 Introduction

Laparoscopic surgical robots such as the da Vinci Research Kit (dVRK) [20] are challenging to accurately control using open-loop techniques because of the hysteresis, cable-stretch, and complex dynamics of their cable-driven joints [13, 37, 18]. Furthermore, encoders are typically located at the motors, far from the joints they control, making accurate state estimation challenging. Prior work addresses these issues by learning a model of robot dynamics from data [18, 43, 58] for accurate open-loop control or by learning control policies that directly command the robot to perform tasks [59]. However, these approaches tend to require many training samples, which can take a long time to collect on a physical robot. Additionally, learning a model of the robot’s dynamics requires accurate state estimation, which requires motion capturing techniques using fiducials [18, 43]. Also, the learned dynamics models can overfit to the specific cabling properties of individual instruments (see Section 1.5). Because instrument changes are commonplace within and across surgeries, control strategies must be robust to these shifts in cabling properties.

We propose a framework called intermittent visual servoing (IVS), which combines coarse planning over a robot model with learning-based, visual feedback control at segments of the task that require high precision. We use RGBD sensing to construct open-loop trajectories to track with a coarse policy, but during intermittent servoing, we only use RGB sensing, as it can capture images at a much higher frequency. Further, depth sensing performs poorly when the scene is moving, so using RGB allows for continuous visual servoing while the robot is still in motion. Typical depth sensors operate at 3.5 frames per second (FPS), whereas commodity cameras can reach over 90 FPS. This, combined with not requiring the robot to fully stop to sense allows for 10.0 corrective updates per second during peg transfer experiments, compared with only 1.6 for RGBD servoing.

We use imitation learning (IL) to train a precise, visual feedback policy from expert demonstrations. Imitation learning is a popular approach for training control policies from

demonstrations provided by a human or an algorithmic supervisor, but may require significant amounts of data [2, 39], which is expensive in the case of a human supervisor [47]. To mitigate this requirement, the learning-based, visual feedback policy is only trained on segments where accuracy is necessary, while we rely on a coarse open-loop policy to navigate between these segments. As a result, training the local, visual feedback policy requires fewer demonstrations than a policy trained to perform the entire task. Because this policy is trained to directly output controls from images, it does not require explicit state estimation techniques as in prior work. While dVRK surgical instruments can have errors up to 6 mm in positioning [18], this is sufficient for low-precision segments such as transferring a block between pegs.

This work, as well as progress to this work, contains invaluable contributions from Minho Hwang, Brijen Thananjeyan, Jeffrey Ichnowski, Daniel Seita, and Danyal Fer. My contribution includes building the IVS architecture, leading its evaluation, creating the figures, and writing and revising sections of this text. Minho Hwang assisted with the setup of the dVRK, as well as the evaluation of IVS. Brijen Thananjeyan, Jeffrey Ichnowski, Daniel Seita, and Danyal Fer provided guidance, and assisted in writing and revising sections of this text.

This thesis makes the following contributions: (1) a novel deep learning framework, IVS, for automation of high-precision surgical tasks, (2) experiments on the FLS peg transfer task suggesting that IVS can match state-of-the-art calibration methods in terms of accuracy while requiring significantly less training data, (3) experiments suggesting that IVS is significantly more robust to instrument changes than prior methods, maintaining accuracy across 3 robot instruments having different cabling-related properties.

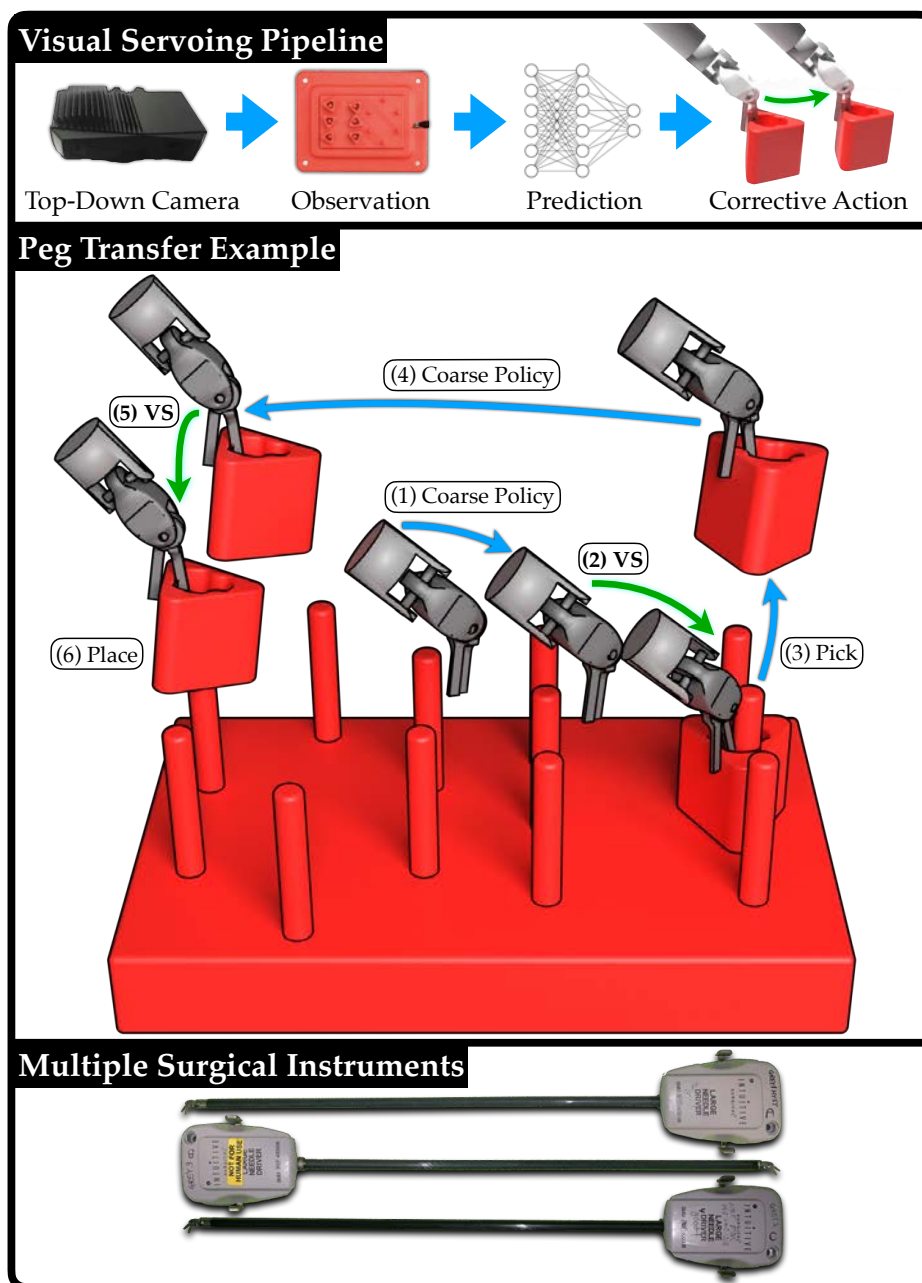


Figure 1.1: **Intermittent Visual Servoing (IVS)**. **Visual Servoing Pipeline:** To compensate for cable-related effects, a policy maps images of the workspace from a top-down RGB camera to corrective motions. **Peg Transfer Example:** The robot switches to a visual feedback policy for high-precision segments of the task (green), and uses a coarse policy to navigate between these segments (blue). **Multiple Surgical Instruments:** We experiment using 3 different large needle drivers, each with unique backlash, hysteresis, and cable tension properties.

1.2 Related Work

Robot assisted surgery has been widely adopted based on pure teleoperation, as exemplified by the popularity of curricula such as the Fundamentals of Laparoscopic Surgery (FLS) [11, 12]. Automating robot surgery has proven to be difficult, and so far all standard surgical procedures using robot surgery techniques use a trained human surgeon to teleoperate the robot arms to compensate for various forces and inaccuracies in the robot system [60]. Automating surgical robotics using cable-driven Robotic Surgical Assistants (RSAs) such as the da Vinci [20] or the Raven II [14] is known to be a difficult problem due to backlash, hysteresis, and other errors and inaccuracies in robot execution [33, 51, 43].

Automation of Surgical Robotics Subtasks

Automation of surgical robotics subtasks has a deep history in the robotics research literature. Key applications include cutting [57], debridement [21, 38, 51], hemostasis [15], suturing [52, 42, 48, 41], and more broadly manipulating and extracting needles [54, 61, 9, 8].

We focus on the Fundamentals of Laparoscopic Surgery (FLS) peg-transfer surgeon training task, in which the surgeon must transfer 6 blocks from a set of pegs to another set of pegs, and then transfer them back (Fig. 1.2). As each block's opening has a 4.5 mm radius, and each peg's cylindrical is 2.25 mm wide, the task requires high precision, making it a popular benchmark task for evaluating human surgeons [1, 5, 46, 40, 32, 44]. Prior work in automating peg transfer suggests that servoing based on encoder readings cannot reliably perform this task, as positioning errors lead to failure [18, 17]. As a result, sophisticated calibration techniques are used to correct for cabling effects of the surgical instrument during execution [18, 17]. These works, while producing effective and reliable results, result in systems are instrument-specific. In contrast, we focus on a system that does not require accurate calibration and transfers across a variety of surgical instruments with distinct cabling properties.

Switching Surgical Instruments

Switching surgical instruments during surgery is both necessary and common [35]. Depending on the type of procedure, up to four instruments may be exchanged on a single arm in rapid succession to perform a task, and this may occur multiple times over a given procedure [35]. These exchanges have been demonstrated to contribute to 10 to 30% of total

operative time, increasing patient exposure to anesthesia [36]. Additionally, each instrument is only permitted to be used for 10 operations regardless of the operation length due to potential instrument degradation and even within this permitted-use window instruments frequently fail [55, 31]. Moreover, between patients, instruments must undergo high pressure, high heat sterilization that further degrades the instrument [55]. Instrument collisions during a procedure are common and can alter the cabling properties of the instrument, necessitating re-calibration in the case of automated surgery. Sophisticated, instrument-specific calibration techniques require many long trajectories of data [18], which further increases the wear on the instrument, reduces its lifespan, and can require time during or before a surgical procedure to collect data. Therefore, developing policies that are efficiently transferable across instruments is critical to automation of surgical tasks, since instruments are exchanged frequently and instrument properties change over time with increased usage.

Visual Servoing for High Precision Tasks

Visual servoing has a rich history in robotics [16, 22]. Classical visual servoing mechanisms typically use domain-specific knowledge in the form of image features or system dynamics [6, 34]. In recent years, data-driven approaches to visual servoing have gained in popularity as a way to generalize from patterns in larger training datasets. For example, approaches such as Levine et al. [27] and Kalashnikov et al. [19] train visual servoing policies for grasping based on months of nonstop data collection across a suite of robot arms. Other approaches for learning visual servoing include Lee et al. [25], who use reinforcement learning and predictive dynamics for target following, Saxena et al. [50] for servoing of quadrotors, and Bateux et al. [4] for repositioning robots from target images.

In this work, to facilitate rapid instrument changes, it is infeasible to obtain massive datasets by running the da Vinci repeatedly for each new instrument, hence we prioritize obtaining high-quality demonstrations [2, 39] at the critical moments of when the robot inserts or removes blocks from pegs. This enables the system to rapidly learn a robust policy for the region of interest, while relying on a coarse, open-loop policy otherwise. Coarse-to-fine control architectures combining geometric planners with adaptive error correction strategies have a long history in robotics [29, 49, 45, 56, 28], with works such as Lozano-Pérez et al. [29] studying the combination of geometric task descriptions with sensing and error correction for compliant motions. Our work is similar to that of Lee et al. [26], who use a model-based planner for moving a robot arm in free space, and reinforcement learning for learning an insertion policy when the gripper is near the region of interest.

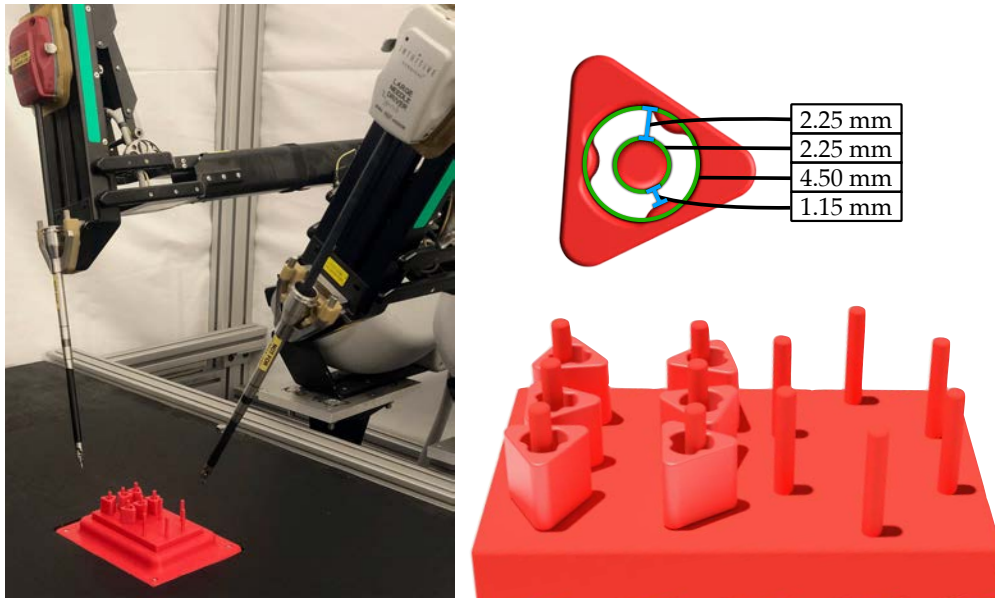


Figure 1.2: **Robot Setup for FLS Peg Transfer.** We use Intuitive Surgical’s da Vinci Research Kit robot. We use red 3D printed blocks and a red 3D printed peg board. We use a uniformly red setup to simulate the surgical environment, where color alone may not provide sufficient signal, as surgeons rely on minute differences in color, depth and texture to complete high-precision tasks. We use a top-down camera to generate open-loop trajectories (RGBD) and capture the input for visual servoing (RGB). The FLS peg transfer task involves transferring 6 blocks from the 6 left pegs to the 6 right pegs, and transferring them back from the right pegs to the left pegs. As each block’s opening has a 4.5 mm radius, and each peg’s cylindrical is 2.25 mm wide, the task requires high precision, making it a popular benchmark task for evaluating human surgeons.

1.3 Problem Definition

We focus on the FLS peg transfer task, using the setup in Hwang et al. [18], which uses red 3D printed blocks and a red 3D printed pegboard (see Fig. 1.2). In real surgical environments, blood is common, so surgeons rely on minute differences in color, depth, and texture to complete high-precision tasks. We use 3D printing to construct a uniformly red pegboard setup, so the environment more accurately reflects a surgical setting. The task involves transferring 6 blocks from the 6 left pegs to the 6 right pegs, and transferring them back from the right pegs to the left pegs (see Fig. 1.2). As in Hwang et al. [17, 18], we focus on the single-arm version of the task.

We define the peg transfer task as consisting of a series of smaller subtasks, with the

following success criteria:

Pick: the robot grasps a block and lifts it off the pegboard.

Place: the robot securely places a block over a target peg.

We define a **transfer** as a successful pick followed by a successful place. A **trajectory** consists of a single instance of any of the two subtasks in action. A single **trial** of the peg transfer task initially consists of 6 blocks starting on one side of the peg board, each with random configurations. A successful trial without failures consists of 6 transfers to move all 6 blocks to the other side of the board, and then 6 more transfers to move the blocks back to the original side of the peg board. A trial can have fewer than 12 transfers if failures occur during the process.

1.4 IVS: Method

Subtask Segmentation and Policy Design

Due to cabling effects, tracking an open-loop trajectory to pick or place targets using robot encoder estimates may result in positioning errors; we thus propose decomposing subtasks into 3 phases: (1) an open-loop *approach* phase, (2) a closed-loop visual-servoing *correction* phase, and (3) an open-loop *completion* phase. The open-loop phases are executed by a coarse policy π_c that tracks predefined trajectories using the robot’s odometry. The closed-loop phases are executed by a learned, visual feedback, fine motion policy π_f that corrects the robot’s position for the subsequent completion motion. At time t , the executed policy outputs an action vector a_t , as well as a termination signal $\phi_t \in \{0, 1\}$, which signals to the system to switch to the next segment.

Pick Subtask

The first segment uses an open-loop policy π_c^0 to execute a trajectory to a position above the target grasp (*approach*). After this motion, a visual feedback policy π_f^0 takes over to correct for positioning errors (*correction*). Once corrected, the robot again executes π_c^0 to perform a predefined grasping motion relative to its current pose (*completion*).

Place Subtask

Similar to block picking, an open-loop policy π_c^1 executes a trajectory to a position above the target placement (*approach*). After this motion, a visual feedback policy π_f^1 takes over to correct for positioning errors (*correction*). Once corrected, the robot opens its jaws, resulting in the block dropping onto the peg (*completion*).

Fine-Policy Data Collection

We collect demonstrations from a human teleoperator to generate a dataset to train a neural network for a fine policy. We collect 15 trajectories on each of the 12 pegs for both subtasks, resulting in 180 transfers, and 360 expert trajectories. Each trajectory consists of a small corrective motion, as the teleoperator navigates the end-effector from a starting position to the goal position. For picks, the goal position is directly above the optimal pick spot. For places, the goal position is such that the center of the block aligns with the center of the target peg. The starting position of each attempt is a random position within 5 mm of the goal position. Further, due to the size of the blocks, small segments of irrelevant blocks

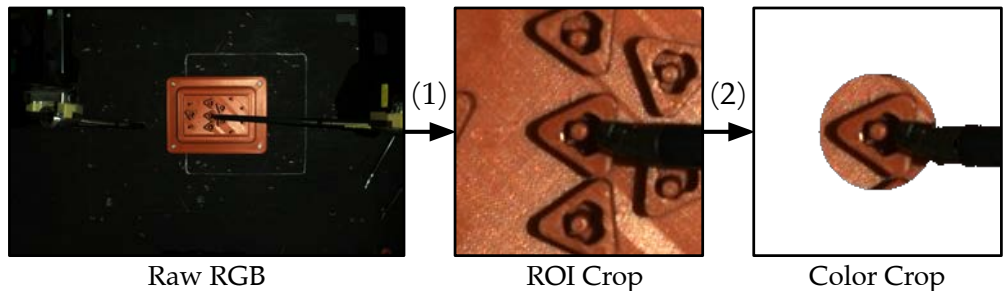


Figure 1.3: **Data Filtering.** We preprocess the images in two ways: (1) crop a 150×150 image around the center of the target peg, and (2) color-crop out all red pixels outside of a block-sized radius from the center of the peg.

may be visible after data preprocessing (see Sec. 1.4). To capture this data property, while collecting trajectories for a given target peg, we populate neighboring pegs with blocks. Then, prior to each attempt, we randomize the configurations of the neighboring blocks.

For each demonstration, we capture a top-down RGB image $I \in R^{1200 \times 1900 \times 3}$ and end-effector position $p \in R^2$ in the robot’s base frame estimated from encoder values at 5 Hz. We do not record the z coordinate, because the correction phase for both subtasks will be performed in an plane with a fixed z coordinate. While the recorded end-effector position has errors due to cabling properties [18, 43, 17], we demonstrate empirically that the high-frequency visual feedback policy trained from supervision extracted from these estimates is reliable (Sec. 1.5). Each demonstration is a raw trajectory: $T = \{(I_t, p_t)\}_{t=0}^T$. The pick dataset $\mathcal{D}_0 = \{\mathcal{T}_{0,i}\}_{i=1}^{N_0}$ consists of 2400 datapoints, and the place dataset $\mathcal{D}_1 = \{\mathcal{T}_{1,i}\}_{i=1}^{N_1}$ contains 1804 datapoints. The corrective pick trajectories are slightly longer than corrective place trajectories, resulting in 3.3 additional datapoints per demonstration.

Preprocessing of Visual Feedback Policy Training Data

Image Filtering

To train a single model on all 180 demonstrations, regardless of the target peg, we (1) crop images around the peg, and (2) color crop other blocks (see Fig. 1.3). For (1), we crop a 150×150 image centered on the target peg. For (2), we color-crop out all red pixels outside of a block-sized radius from the center of the target peg. This removes the other blocks from the input image as much as possible while keeping the instrument, target block, and target peg visible. Hereafter, we let I_t refer to the image after preprocessing the raw image (see

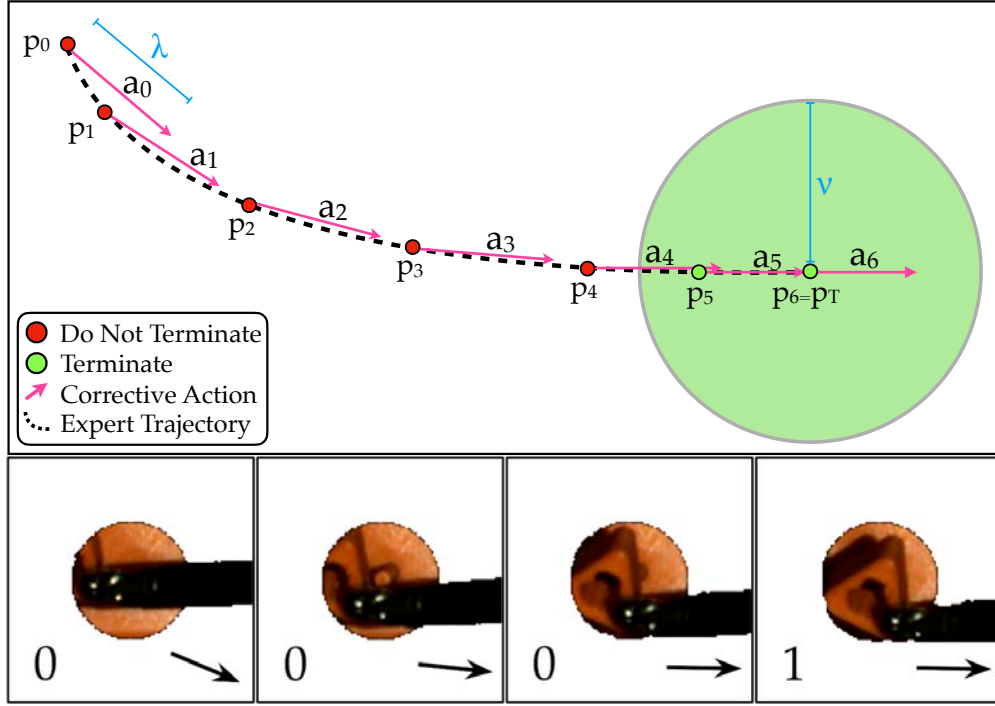


Figure 1.4: **Supervision Extraction with Labeled Example.** **Top:** The action a_t , indicated by the arrows, is a vector of length 1 mm (λ) in the direction $p_{t'} - p_t$ traveled by the expert trajectory, where p_t is the current waypoint and $p_{t'}$ is the next waypoint that is at least 1 mm (λ) distance from p_t . The extracted termination label, indicated by the color of the waypoint, is 1 if the distance to the final position in the trajectory p_T is less than 2 mm (ν), and 0 otherwise. Each waypoint corresponds to an image, and each image receives both a corrective action label and a termination signal label. **Bottom:** Preprocessed images (see Fig. 1.3) from a corrective place trajectory with labels extracted using method described above.

Fig. 1.3).

Supervision Extraction

We additionally process the datasets to extract supervision for subsequent learning. For an input raw trajectory \mathcal{T} , we transform it via mappings Π_a and Π_o that extract the action executed and a terminal condition, respectively (see Fig. 1.4).

Corrective Action Extraction: We transform \mathcal{T} to get the action-labeled dataset $\mathcal{T}_\rightarrow = \Pi_a(\mathcal{T}) = \{(I_t, a_t)\}_{t=1}^T$, where $a_t = \lambda \frac{p_{t'} - p_t}{\|(p_{t'} - p_t)\|_2}$

$s.t.t' = \min(\{t'' \mid t'' > t \wedge p_{t''} - p_t \geq \lambda\} \cup \{T\})$. The action a_t is a vector of length λ in the direction $p_{t'} - p_t$ traveled by the demonstration trajectory, where $p_{t'}$ is the next waypoint that is at least λ distance from p_t . In the case the distance to the final position in the trajectory p_T is less than λ , $p_{t'} = p_T$. In experiments, we set the λ hyperparameter to 1 mm, as 1 mm is an upper bound on the expected Cartesian distance traveled by the end-effector per corrective update.

Terminal Condition Extraction: We extract a binary completion label for each image. We transform \mathcal{T} to get the action-labeled dataset $\mathcal{T}_l = \Pi_o(\mathcal{T}) = \{(I_t, \phi_t)\}_{t=1}^T$, where $\phi_t = 1 \{\|p_T - p_t\|_2 \leq \nu\}$. The flag ϕ_t is 1 if the distance to the final position p_T is less than the hyperparameter ν . In experiments, we set ν to 2 mm, as 2 mm is strict enough to reliably confirm termination, and lenient enough to prevent label imbalance, as 30% of images are labeled positively.

Constructing the Visual Feedback Policy

We train the visual feedback policy π_f for each subtask from demonstrations using supervised learning. The policy takes in a top-down RGB image I_t as input and outputs (a_t, ϕ_t) , where a_t is an action vector and ϕ_t is a termination condition.

Training the Visual Feedback Policy

The policy consists of an ensemble of 4 Convolutional Neural Networks [23, 24], denoted by $f_{\theta_{[1..4]}}$. Each individual model f_{θ_i} consists of alternating convolution and max pooling layers, following by dense layers separated by Dropout [53]. We use an ensemble of k models to make the policy more robust, and we evaluate for $k \in \{1, 2, 4, 8\}$ in Table 1.1. We select k to be 4. Each model f_{θ_i} uses a processed $150 \times 150 \times 3$ image I_t as input and outputs estimates $a_{t,i}$ and $\phi_{t,i}$ of the supervisor action and terminal conditions respectively. Each model trains on 150 randomly sampled trajectories, with 30 for testing. We train each network by minimizing $l_{MSE} + \mu l_{CE}$ on sampled batches of its training data, where l_{MSE} is a Mean Square Error loss on the action prediction, l_{CE} is a cross-entropy loss for the terminal condition prediction, and μ is a relative weighting hyperparameter. Each model trains for 40 seconds on a K80 GPU. The weights in the convolutional layers are shared, as useful convolutional filters are likely similar across subtasks, while the weights in the dense layers are independent. Sharing layers provides more supervision when training the filters.

Table 1.1: **Ablation Study Varying Number of Models in Ensemble.** We compare IVS with differing number of models in the ensemble across 3 full trials of peg transfer (36 transfers). More models results in a more robust policy, but due to compute limitations, less frequent servoing. More frequent servoing results in higher precision, as both the corrective action and the termination signal are updated more frequently. The goal is to find a balance between maximizing robustness (many models) and maximizing update frequency (few models). We use an ensemble of 4 models.

Num Models	Update Frequency	Transfer Success Rate
1	15.6	97.2%
2	12.8	98.6%
4	10.0	100.0%
8	7.1	100.0%

Querying the Visual Feedback Policy

Once trained, we evaluate the ensemble of models in parallel with a filtered RGB image (Sec. 1.4). We let $\pi_f(I_t) = (a_t, \phi_t) = \left(\frac{1}{4} \sum_{i=1}^4 a_{t,i}, \sum_{i=1}^4 1 \{ \phi_{t,i} \geq \omega \} \geq \kappa \right)$,

where ω and κ are a hyperparameters set to 0.70 and 3 respectively. We hand-tune these hyperparameters to maximize speed and minimize false positives. The predicted action is the mean action across the ensemble, and the predicted termination condition checks if at least κ models predict termination with probability greater than ω .

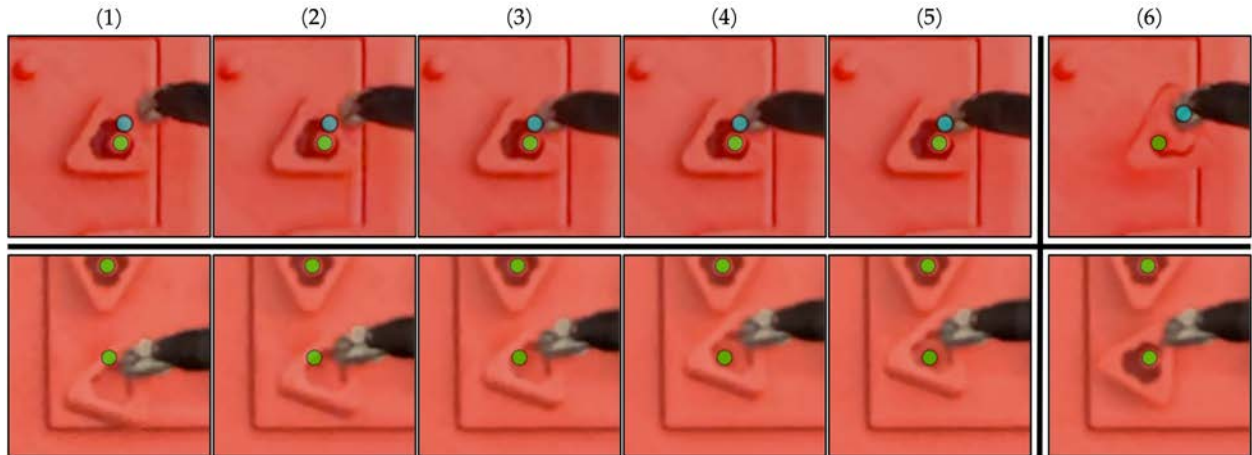


Figure 1.5: **IVS Example.** Example of IVS correcting positioning errors. **Top row (pick subtask):** IVS removes pickup error with 5 corrective updates in 0.6 seconds. We overlay small blue circles to highlight an optimal pick location on the block, defined as the graspable point furthest from the peg along the edge closest to the robot. The positioning of the robot is off due to the inaccuracy of the coarse policy, as the end-effector is not positioned over the block (Frame 1). We then switch to the learned policy, and visual servoing guides the end-effector over a pick point (Frames 2-5), and once determined a safe pickup is possible, picks the block successfully (Frame 6). **Bottom row (place subtask):** IVS removes placement error with 12 corrective updates in 1.2 seconds. We overlay small green circles to highlight the location of the pegs. The positioning of the robot is off due to the inaccuracy of the coarse policy, as the peg is not under the block (Frame 1). We then switch to the learned policy, and visual servoing guides the block over the peg (Frames 2-5), and once determined a safe situation to drop, places the block successfully (Frame 6).

1.5 Experiments

The experimental setup has a top-down RGBD camera and uses the teleoperation interface [20] to collect demonstrations on the dVRK. Training data are collected on a single instrument, but the system is tested with 3 different instruments that have unique dynamics due to differences in cabling properties. We use the position of the blocks and pegs estimated by an RGBD image to construct trajectories for picks and places, but only use RGB images for visual servoing.

We benchmark IVS against two baselines:

- **Uncalibrated Baseline (UNCAL):** This is a coarse open-loop policy, implemented using the default unmodified dVRK controller. The trajectories are tracked in closed-

loop with respect to the robot’s odometry, but open-loop with respect to vision.

- **Calibrated Baseline (CAL):** This is a calibrated open-loop policy [18] that is the current state-of-the-art method for automating peg transfer. To correct for backlash, hysteresis, and cable tension, the authors train a recurrent dynamics model to estimate the true position of the robot based on prior commands. Similar to the uncalibrated baseline, the robot tracks reference trajectories in closed-loop with respect to the position estimated by the recurrent model, but open-loop with respect to visual inputs. This method uses 1800 data points, however a full trajectory is executed between each sample. IVS collects 4204 data points, but at a much higher frequency, with consecutive data points less than a millimeter apart and each trajectory taking a few seconds to collect. As a result, both methods take around 30 minutes to collect data, but we observe in Section 1.5 that the data used for IVS can be reused for different instruments, unlike the data used by CAL.

Accuracy Results

Overall, IVS achieves a pick and place success rate of 99.2% and 100.0% respectively. IVS succeeds on 118 of 119 transfers, resulting in a 99.2% transfer success rate, exceeding the uncalibrated baseline by over 25%. See Table 1.2 for details and we illustrate an example of IVS correcting positioning errors in Figure 1.5.

Timing Results

The goal is to produce higher success rates, rather than to reduce the timing. However, we find that the proposed method is only marginally slower than the baselines (see Table 1.2). Due to fast image capture and continuous servoing via RGB imaging, we are able to both update the robot’s velocity and check for termination 10 times per second, minimizing additions to the mean transfer time. As a result, the mean transfer time is only 1.5 seconds slower than the uncalibrated baseline, and 0.7 seconds slower than the calibrated baseline.

We report IVS timing results in Table 1.3. On average, each pick requires 2.2 mm of correction, spanning 0.7 seconds and 7 corrective updates, and each place requires 1.9 mm of correction, spanning 0.5 seconds and 5 corrective updates.

Transferability Results

To conduct instrument transfer evaluation, we experiment using 3 large needle driver instruments: *A*, *B*, and *C* (Fig. 1.1 bottom). Each has inconsistent cabling characteristics, and we trained IVS with data only from instrument *A*.

We investigate whether models learned from data using one instrument can transfer to another instrument without modification. This is challenging, because different surgical instruments, even of the same type, have different cabling properties due to differences in wear and tear. However, the visual servoing algorithm does not rely on the cabling characteristics of any specific instrument, but rather only requires that the robot is able to roughly correct in the desired direction. We hypothesize that errors in executing the corrective motion can be mitigated over time by executing additional corrective motions, as long as the cumulative error is decreasing. However, the calibrated baseline uses an observer model that explicitly predicts the motion of the robot based on prior commands, which requires learning the dynamics of the specific instrument used in training which may not be sufficiently accurate on a new instrument. We report transferability results in Table 1.4 and observe that the calibrated baselines suffer significantly on different instruments, and the IVS model trained on Instrument *A* does not decrease in performance.

Table 1.2: **Peg Transfer Baseline Comparison.** Benchmark comparing performance of IVS to the baselines described in Section 1.5. IVS beats both baselines in terms of pick success rate, place success rate, and overall transfer success rate. Due to using RGB imaging, we are able to take many corrective steps per second without stopping, minimizing additions to the mean transfer time.

	Pick Success Rate	Place Success Rate	Mean Transfer Time (s)	Success / Attempts	Transfer Success Rate
Uncalibrated Baseline	96.3%	75.7%	8.7	77 / 106	72.6%
Calibrated Baseline	97.9%	99.6%	9.5	116 / 119	97.5%
IVS	99.2%	100%	10.2	118 / 119	99.2%

Table 1.3: **IVS Efficiency Benchmark.** We analyze efficiency of IVS on 3 large needle driver instruments (see Fig. 1.1). On average, IVS moves the end-effector 4.1 mm, requiring an additional 1.2 seconds per transfer.

Pick			
Instrument	Corrective Updates	Time (sec)	Distance (mm)
A	5.75 ± 2.10	0.60 ± 0.19	1.71 ± 0.65
B	7.73 ± 2.63	0.77 ± 0.22	2.52 ± 0.78
C	7.43 ± 3.03	0.74 ± 0.26	2.24 ± 0.85
Mean	7.01 ± 2.77	0.70 ± 0.24	2.17 ± 0.84

Place			
Instrument	Corrective Updates	Time (sec)	Distance (mm)
A	4.90 ± 4.82	0.52 ± 0.43	2.04 ± 1.77
B	5.63 ± 4.20	0.58 ± 0.37	2.12 ± 1.57
C	4.68 ± 4.21	0.49 ± 0.37	1.78 ± 1.43
Mean	4.97 ± 4.47	0.52 ± 0.39	1.93 ± 1.58

Table 1.4: **Instrument Transfer Comparison.** Benchmark comparing performance on 10 full trials of peg transfer (120 transfers) across 3 different surgical instruments with unique cabling characteristics of the uncalibrated baseline, calibrated baseline trained on each instrument, and IVS trained on one instrument. IVS consistently beats both baselines, while remaining robust to instrument changes.

Instrument	UNCAL	CAL _A	CAL _B	CAL _C	IVS _A
A	72.6%	97.5%	48.1%	55.2%	99.2%
B	31.3%	58.5%	98.3%	67.0%	99.2%
C	47.2%	27.8%	81.6%	97.4%	100.0%
Mean	50.5%	69.8%	77.3%	79.2%	99.4%

2. Conclusions & Future Work

2.1 Discussion of IVS

In this work, we presented intermittent visual servoing. IVS maintains performance across different instruments surprisingly well (see Table 1.4), and this transferability is critical to implementing automated surgical techniques where instruments are being exchanged frequently and an instrument’s properties are expected to change over time.

2.2 Extensions of IVS

We evaluate IVS using the FLS peg transfer task, a common training task for surgeons. IVS is an enticing method to further advance surgical robotics, as it allows us to develop efficient and effective policies that are robust to surgical instrument changes. In future work, we will investigate how to further optimize IVS, and apply it to surgical cutting [57], surgical suturing [52], and non-surgical applications such as assembly [28], to evaluate the ability of IVS to generalize. Further, in order to apply IVS to other tasks in the future, additional research into learning the switching model between the coarse policy and the visual servo policy should be conducted. In this work, we learn the switch from the visual servo policy to the coarse policy, but the switch from the coarse policy to the visual servo policy is hard-coded, which may not be possible in future tasks.

More specifically, surgical suturing is receiving a considerable amount of research attention in the medical robotics field recently. Lu et al. explore surgical knot tying [30], Barnoy et al. apply model-free reinforcement learning to surgical suturing in sim [3], Sen et al. present a specialized mechanical needle guide [52], D’Ettorre et al. and Chiu et al. investigate regrasping needles [10, 7], and Zhong et al. explore needle insertion with active tissue deformation [61]. Missing from above works is a fully autonomous, reliable system for suturing an artificial wound, due to the complex nature of the task, and the high precision maneuvering required from the dVRK. However, IVS may be a great candidate method

for reliably automating surgical suturing, as it decomposes well into short, high precision subtasks: aligning the needle for insertion, grasping the needle for extraction, and needle handover, with coarse motions between each. If each individual subtask can be automated, this work suggests the entire task may be reliably automated.

2.3 Limitations of IVS

While IVS shows promising results, its ability to adapt to new tasks is not fully evaluated in this work. Peg transfer corrections are rigid, and purely 2 dimensional. In real surgery, as well as more complex training tasks, such as surgical suturing, 3 dimensional corrections may be required. Further, the peg transfer environment contains simple and consistent visual signals: align the end effector over the block, and align the block over the peg. In the real surgical environment, visual signals are much more complex and inconsistent. Further, patients move during surgeries, typically due to breathing. However, IVS was developed with these issues in mind, as the method involves simplifying the task for efficient learning and dynamically adapting to errors, making it a promising technique in complex environments. Regardless, the ability of IVS to reliably output more complex corrections, as well as work reliably in more visually complex scenes, is an area future work will need to explore.

2.4 Reflections

Leading this project, among countless small lessons learned, there are two takeaways that stick out. For one, I experienced the tremendous amount of trial-and-error, experimentation, and overall meticulousness required to conduct valid and sound science. Science is an art, and its advancement relies on thorough analysis, abundant experimentation, and perhaps a weird idea or two. Second, I experienced first-hand how the solutions of yesterday help solve the problems of today. This work built upon a tremendous amount of research by many labs over many years, and I hope this work serves as a stepping stone for a problem of tomorrow.

3. Research Acknowledgements

This research was performed at the AUTOLAB at UC Berkeley in affiliation with the Berkeley AI Research (BAIR) Lab, Berkeley Deep Drive (BDD), the Real-Time Intelligent Secure Execution (RISE) Lab, the CITRIS “People and Robots” (CPAR) Initiative, and with UC Berkeley’s Center for Automation and Learning for Medical Robotics (Cal-MR). This work is supported in part by the Technology & Advanced Telemedicine Research Center (TATRC) project W81XWH-18-C-0096 under a medical Telerobotic Operative Network (TRON) project led by SRI International. The authors were supported in part by donations from Intuitive Surgical, Siemens, Google, Toyota Research Institute, Honda, and Intel. The da Vinci Research Kit was 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 (IIS 1637789), Worcester Polytechnic Institute (IIS 1637759), and the University of Washington (IIS 1637444).

Bibliography

- [1] Ahmad Abiri et al. “Multi-Modal Haptic Feedback for Grip Force Reduction in Robotic Surgery”. In: *Scientific Reports* 9.1 (2019), p. 5016.
- [2] Brenna D Argall et al. “A Survey of Robot Learning From Demonstration”. In: *Robotics and Autonomous Systems* 57 (2009).
- [3] Yotam Barnoy et al. “Robotic Surgery With Lean Reinforcement Learning”. In: (2021).
- [4] Quentin Bateux et al. “Training Deep Neural Networks for Visual Servoing”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2018.
- [5] 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*. 2017.
- [6] Francois Chaumette and Seth Hutchinson. “Visual Servo Control I. Basic Approaches”. In: *IEEE Robotics and Automation Magazine*. 2006.
- [7] Zih-Yun Chiu et al. “Bimanual Regrasping for Suture Needles using Reinforcement Learning for Rapid Motion Planning”. In: (2016).
- [8] Zih-Yun Chiu et al. “Bimanual Regrasping for Suture Needles using Reinforcement Learning for Rapid Motion Planning”. In: *arXiv preprint arXiv:2011.04813* (2020).
- [9] C D’Ettorre et al. “Automated Pick-up of Suturing Needles for Robotic Surgical Assistance”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2018.
- [10] C. D’Ettorre et al. “Automated pick-up of suturing needles for robotic surgical assistance”. In: (2021).
- [11] Anna M Derossis et al. “Development of a Model for Training and Evaluation of Laparoscopic Skills”. In: *The American Journal of Surgery* 175.6 (1998).
- [12] Gerald M. Fried et al. “Proving the Value of Simulation in Laparoscopic Surgery”. In: *Annals of Surgery* 240.3 (2004).

- [13] Mohammad Haghhighipناه et al. “Unscented Kalman Filter and 3D Vision to Improve Cable Driven Surgical Robot Joint Angle Estimation”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2016.
- [14] B. Hannaford et al. “Raven-II: An Open Platform for Surgical Robotics Research”. In: *IEEE Transactions on Biomedical Engineering*. 2013.
- [15] Jingbin Huang et al. “Model-Predictive Control of Blood Suction for Surgical Hemostasis using Differentiable Fluid Simulations”. In: *arXiv preprint arXiv:2102.01436* (2021).
- [16] Seth Hutchinson, Gregory Hager, and Peter Corke. “A Tutorial on Visual Servo Control”. In: *IEEE Transactions on Robotics and Automation*. 1996.
- [17] Minh Hwang et al. “Applying Depth-Sensing to Automated Surgical Manipulation with a da Vinci Robot”. In: *International Symposium on Medical Robotics (ISMR)*. 2020.
- [18] Minh Hwang et al. “Efficiently Calibrating Cable-Driven Surgical Robots With RGBD Fiducial Sensing and Recurrent Neural Networks”. In: *IEEE Robotics and Automation Letters (RA-L)*. 2020.
- [19] Dmitry Kalashnikov et al. “QT-Opt: Scalable Deep Reinforcement Learning for Vision-Based Robotic Manipulation”. In: *Conference on Robot Learning (CoRL)*. 2018.
- [20] P Kazanzides et al. “An Open-Source Research Kit for the da Vinci Surgical System”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2014.
- [21] B. Kehoe et al. “Autonomous Multilateral Debridement with the Raven Surgical Robot”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2014.
- [22] Danica Kragic and Henrik I Christensen. *Survey on Visual Servoing for Manipulation*. Tech. rep. Computational Vision and Active Perception Laboratory, 2002.
- [23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Neural Information Processing Systems (NeurIPS)*. 2012.
- [24] Yann Lecun et al. “Gradient-based Learning Applied to Document Recognition”. In: *Proceedings of the IEEE*. 1998, pp. 2278–2324.
- [25] Alex X Lee, Sergey Levine, and Pieter Abbeel. “Learning Visual Servoing with Deep Features and Fitted Q-Iteration”. In: *International Conference on Learning Representations (ICLR)*. 2017.

- [26] Michelle A. Lee et al. “Guided Uncertainty-Aware Policy Optimization: Combining Learning and Model-Based Strategies for Sample-Efficient Policy Learning”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2020.
- [27] S Levine et al. “Learning Hand-Eye Coordination for Robotic Grasping with Deep Learning and Large-Scale Data Collection”. In: *International Journal of Robotics Research (IJRR)*. 2017.
- [28] Tomas Lozano-Perez. “The design of a mechanical assembly system”. In: (1976).
- [29] Tomás Lozano-Pérez, Matthew T. Mason, and Russell H. Taylor. “Automatic Synthesis of Fine-Motion Strategies for Robots”. In: *IJRR* (1984).
- [30] Bo Lu et al. “A Learning-Driven Framework with Spatial Optimization For Surgical Suture Thread Reconstruction and Autonomous Grasping Under Multiple Topologies and Environmental Noises”. In: (2020).
- [31] Wesley W. Ludwig et al. “Instrument Life for Robot-assisted Laparoscopic Radical Prostatectomy and Partial Nephrectomy: Are Ten Lives for Most Instruments Justified?” In: *Urology* 86.5 (2015).
- [32] Naveen Madapana et al. “DESK: A Robotic Activity Dataset for Dexterous Surgical Skills Transfer to Medical Robots”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019.
- [33] J. Mahler et al. “Learning Accurate Kinematic Control of Cable-Driven Surgical Robots Using Data Cleaning and Gaussian Process Regression.” In: *IEEE Conference on Automation Science and Engineering (CASE)*. 2014.
- [34] Guillaume Caron Eric Marchand and El Mustapha Mouaddib. “Photometric Visual Servoing from Omnidirectional Cameras”. In: *Autonomous Robots*. 2013.
- [35] NY Mehta et al. “Sequence and task analysis of instrument use in common laparoscopic procedures”. In: *Surgical endoscopy* 16.2 (2002).
- [36] A Melzer. “Endoscopic instruments: conventional and intelligent”. In: *Toolii J, Gossot D, Hunter JG. eds. Endosurgery* (1996).
- [37] Muneaki Miyasaka et al. “Measurement of the Cable-Pulley Coulomb and Viscous Friction for a Cable-Driven Surgical Robotic System”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015.
- [38] Adithyavairavan Murali et al. “Learning by Observation for Surgical Subtasks: Multilateral Cutting of 3D Viscoelastic and 2D Orthotropic Tissue Phantoms”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2015.

- [39] Takayuki Osa et al. “An Algorithmic Perspective on Imitation Learning”. In: *Foundations and Trends in Robotics* 7 (2018).
- [40] Lucian Panait et al. “Do Laparoscopic Skills Transfer to Robotic Surgery?” In: *Journal of Surgical Research* (2014).
- [41] S. A. Pedram et al. “Autonomous Suturing Framework and Quantification Using a Cable-Driven Surgical Robot”. In: *IEEE Transactions on Robotics* (2020), pp. 1–14. DOI: 10.1109/TR0.2020.3031236.
- [42] Sahba Aghajani Pedram et al. “Autonomous Suturing Via Surgical Robot: An Algorithm for Optimal Selection of Needle Diameter, Shape, and Path”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2017.
- [43] Haonan Peng et al. “Real-time Data Driven Precision Estimator for RAVEN-II Surgical Robot End Effector Position”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2020.
- [44] Md Masudur Rahman et al. “Transferring Dexterous Surgical Skill Knowledge between Robots for Semi-autonomous Teleoperation”. In: *IEEE International Conference on Robot and Human Interactive Communication (Ro-Man)*. 2019.
- [45] Christian Rauch et al. “Learning-driven Coarse-to-Fine Articulated Robot Tracking”. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 6604–6610.
- [46] Irene Rivas-Blanco et al. “Transferring Know-How for an Autonomous Camera Robotic Assistant”. In: *Electronics: Cognitive Robotics and Control* (2019).
- [47] Stephane Ross, Geoffrey J Gordon, and J Andrew Bagnell. “A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning”. In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*. 2011.
- [48] H Saeidi et al. “Autonomous Laparoscopic Robotic Suturing with a Novel Actuated Suturing Tool and 3D Endoscope”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2019.
- [49] S. Salcudean and C. An. “On the control of redundant coarse-fine manipulators”. In: *1989 International Conference on Robotics and Automation (ICRA)*, 1834–1840 vol.3. DOI: 10.1109/ROBOT.1989.100240.
- [50] Aseem Saxena et al. “Exploring Convolutional Networks for End-to-End Visual Servoing”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2017.

- [51] Daniel Seita et al. “Fast and Reliable Autonomous Surgical Debridement with Cable-Driven Robots Using a Two-Phase Calibration Procedure”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2018.
- [52] S. Sen et al. “Automating Multiple-Throw Multilateral Surgical Suturing with a Mechanical Needle Guide and Sequential Convex Optimization”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2016.
- [53] Nitish Srivastava et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research (JMLR)* (2014).
- [54] Priya Sundaresan et al. “Automated Extraction of Surgical Needles from Tissue Phantoms”. In: *IEEE Conference on Automation Science and Engineering (CASE)*. 2019.
- [55] Da Vinci Surgical System. “INC DVSSIS”. In: *Endowrist Instrument and Accessory Catalog* (2013).
- [56] Russell Highsmith Taylor. “The synthesis of manipulator control programs from task-level specifications.” In: (1976).
- [57] Brijen Thananjeyan et al. “Multilateral Surgical Pattern Cutting in 2D Orthotropic Gauze with Deep Reinforcement Learning Policies for Tensioning”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2017.
- [58] 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 (RA-L)*. 2020.
- [59] Brijen Thananjeyan* et al. “Recovery RL: Safe Reinforcement Learning with Learned Recovery Zones”. In: *IEEE Robotics and Automation Letters (RA-L)*. 2021.
- [60] Michael Yip and Nikhil Das. “Robot Autonomy for Surgery”. In: *The Encyclopedia of Medical Robotics* (2017).
- [61] Fangxun Zhong et al. “Dual-Arm Robotic Needle Insertion With Active Tissue Deformation for Autonomous Suturing”. In: *IEEE Robotics and Automation Letters (RA-L)*. 2019.