Vision Based Surgical Tool Tracking and Force Estimation with Robot Kinematics Prior

Yun-Hsuan Su

A dissertation
submitted in partial fulfillment
of the requirements for

Doctor of Philosophy

University of Washington

2020

Reading Committee:

Blake Hannaford, Chair

Samuel Burden

Jenq-Neng Hwang

Program Authorized to Offer Degree:
Electrical and Computer Engineering

University of Washington

Abstract

Vision Based Surgical Tool Tracking and Force Estimation with Robot Kinematics Prior

Yun-Hsuan Su

Chair of the Supervisory Committee:
Professor Blake Hannaford
Electrical and Computer Engineering

Robot assisted minimally invasive surgery combines the skill and techniques of highly-trained surgeons with the robustness and precision of machines. Through a teleoperation scheme, surgeons can execute high-level surgical tasks by commanding instruments controlled by precise robotic devices. Several advantages arise. To name a few: (1) achieved precision is beyond that of human dexterity alone (2) a greater number of kinematic degrees of freedom are possible at the surgical tool tip (3) surgeons are able to operate remotely, i.e. agnostic of patient location given a suitable communication line. Despite the numerous advantages over traditional key-hole or laparoscopic surgery, the lack of realistic and real-time force feedback is a major drawback — discerning tool-tissue interactions can be unintuitive and can ultimately result in unintentional tissue damage. Directly sensing forces at the tool-tissue interface is theoretically possible using tool tip mounted force sensors, but this approach is not amenable to required sterilization procedures. Thus, a vision based force estimation method is proposed to infer the applied force based on real-time analysis of tissue deformation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

Ever since the first documented use of a robot-assisted surgical procedure back in 1985 [1] , great progress has been done towards advancing the technology of robot manipulated surgical practice. Yet, obtaining accurate force feedback during such kinds of surgical operations is a heated topic. In fact, no current commercialized surgical robots provide information about the applied force, and thus this field of research still holds great potential.

1.1 Background

In minimally invasive surgeries (MIS), surgeons rely on endoscope images to operate on patients. Recently, surgical robots have been employed in MIS. If they control the instruments via haptic devices, force feedback can be sent as a warning when the surgical tool tip approaches important arteries or nerves, which could promote a higher level of safety [40]. However, force feedback can be hard to obtain. Due to size, cost, constraints and difficulties in sterility maintenance, directly applying force sensors on the tip of surgical tools is often impractical. An alternative is to visually measure the tissue indentation due to surgical contacts in endoscope images, then infer force accordingly from the level of deformation.

1.2 Overview

The proposed method for achieving visual force estimation can be divided into three stages as shown in Figure 1.1. The first stage encompasses generating a 3D model of the surgical scene

[1]The robot, PUMA 200 (Westinghouse Electric, Pittsburgh, PA), was used for needle placement in a CT-guided brain biopsy [109].

with binary labels of surgical tool versus background tissue. The second stage is to measure tissue deformation at the tool-tissue contact point. Meanwhile, this stage also correlates tissue deformation with applied force. Lastly, the third stage realizes feedback of force to the human operator of the Raven-II surgical robot platform using a haptic input device. Below is a more in-depth description for each of the three stages.

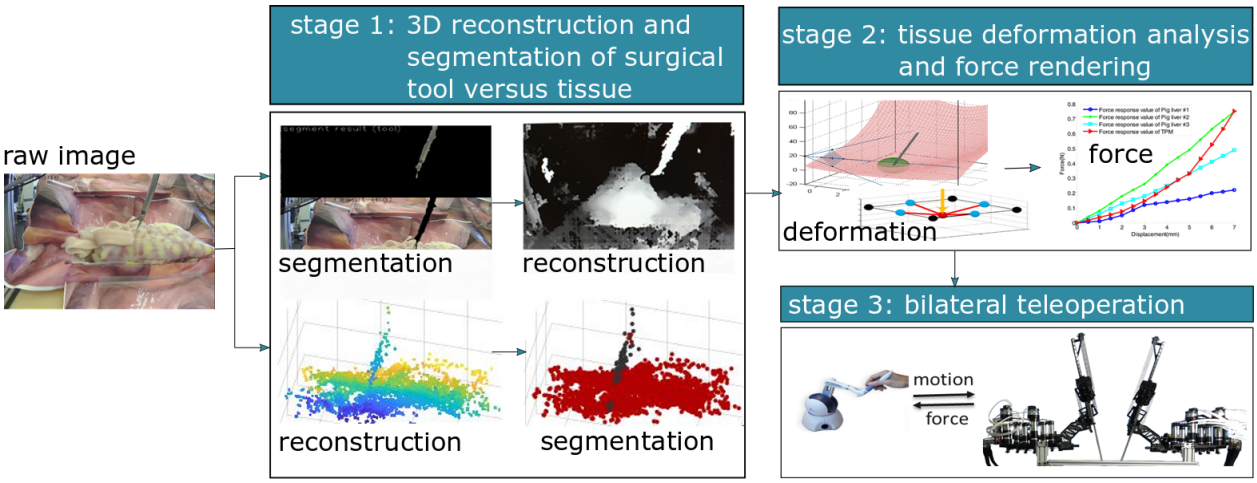


Figure 1.1: Three stages involved in the proposed vision-based force estimation method. Most of the completed work is focusing on stage 1.

1.2.1 3D Reconstruction and Segmentation of Surgical Tool versus Tissue

In this research, two strategies for surgical tool segmentation and surgical scene 3D reconstruction are considered [91]. The core difference between the two strategies is the order in which tool segmentation and 3D reconstruction take place.

As shown in Figure 1.2, Strategy I first carries out 2D tool segmentation with robot kinematics prior, then proceeds with 3D reconstruction near the tool-tissue contact region. On the other hand, Strategy II starts with 3D reconstruction of the entire surgical scene from multiple 2D endoscopic images then later followed by 3D tool segmentation. It was hypothesized that the performance of the two methods would depend on several salient task parameters, e.g. known vs unknown camera pose, static vs dynamic scene, or with vs

without camera motion. To that end, a hypothesis matrix of various parameters and datasets was synthesized and tested with both strategies. In fact, the difference between Strategy I and Strategy II lies in the order of (a) image segmentation and (b) 3D reconstruction. In particular, while Strategy I executes in the sequence of (a) then (b), Strategy II executes (b) followed by (a). Below is a brief description of the two components (a) and (b).

Image Segmentation

Surgical instrument tracking from endoscopic images is a prerequisite for many medical robotics research. In vision-based force estimation in MIS, the goal is to identifying the tool-tissue contact point and analyze the nearby tissue deformation, so instead of merely creating a bounding box that tracks the tool location, pixel-wise segmentation is necessary. Some challenges in surgical tool segmentation include motion blur, partial occlusion, specular reflections [51] on wet tissue surface, lighting changes from the coaxial light source, and the metallic surgical tool shafts that often reflect tissue colors.

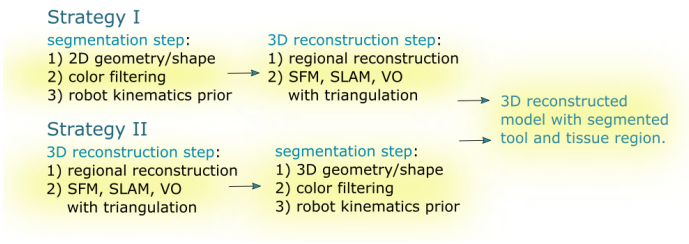


Figure 1.2: Two strategies for surgical scene 3D reconstruction and tool segmentation.

Considering the above factors, Figure 1.3 shows a broad spectrum of different ways previous researchers have dealt with the task of image-based surgical instrument detection. According to a survey paper [54] in 2016, there are two main approaches - marker and marker-less, where a marker can be further classified into either a visual marker or a non-visual marker. In terms of visual markers, topology markers stand out in instrument detection because distortions of the tags still yield a positive detection, unless they change the topology. Since most surgical tool shafts have a cylindrical shape, when putting a marker on the tool shaft, it is often not fixed on a flat surface, distortion is thus inevitable. Yet in general, visual markers suffer from lack of robustness to occlusion, which can easily happen

during surgical operations when the tool is stained with blood. Contrarily, non-visual markers are not subject to occlusion, some of which include RFID sensors [161], acoustic [132] or electromagnetic trackers [135] etc. Some concerns about using those markers are that preferably no external electronics should be placed inside patients' body, and so it is not desirable to place those sensors or trackers on the surgical tool tip, besides some of them are relatively expensive, and others have receivers that are undesirably big in size.

Despite the effectiveness of using markers, since not many clinically used surgical tools have external sensors or markers attached, for better generalizability, marker-less surgical instrument segmentation is considered. My first attempt of marker-less surgical tool segmentation with traditional computer vision was proposed [53].

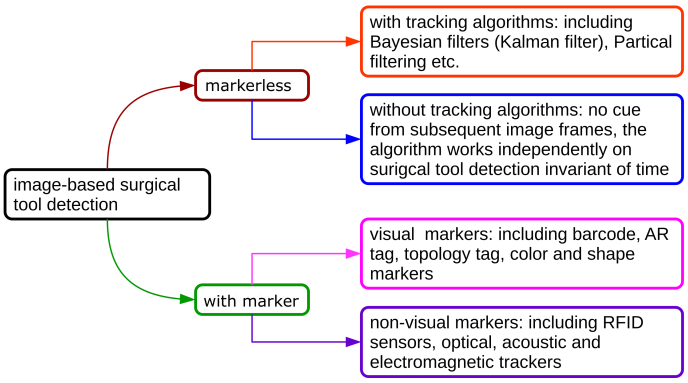


Figure 1.3: Different approaches for image-based surgical instrument segmentation.

Later, I discovered that Raven-II [17] joint angles and end-effector position are available and could be helpful in tool segmentation. Given the camera extrinsic matrix with respect to the Raven base frame, robot kinematics provides a shape prior indicating where the surgical instrument may appear on the image frame by projecting the 3D surgical tool pose onto the 2D image. Afterwards, color filtering can be applied to modify the kinematics derived shape prior using the shape matching algorithm in the frequency domain using DFT. Finally, with a modified shape prior mask, a color mask is applied further polish the segmentation result. In summary, the proposed surgical tool segmentation algorithm [204] fuses robot kinematics shape prior with color filtering in the Opponent color space.

In [170], the idea of using robot kinematics in tool segmentation was further integrated in a data-driven approach using convolution neural network (CNN) and proven to improve segmentation accuracy and robustness especially with longer runtime.

3D Reconstruction: Depth Lattice Rendering

The goal of stage 1 is to generate a 3D model of a local tissue patch centered at the surgical instrument contact point, so that a temporal deformation map due to tool-tissue contact can be generated and analyzed. In Strategy I, since image segmentation is already done, the likelihood map of the surgical tool on the image plane can be utilized in verifying local region where 3D reconstruction is needed. On the contrary, Strategy II 3D reconstructs the entire image because there is no prior knowledge provided by the image segmentation step.

Preliminary experiments were conducted using a stereo camera system with known camera poses. Figure 1.4 demonstrates the workflow that achieves tissue deformation analysis through tool segmentation and 3D reconstruction from stereo vision. With robot kinematics, a spherical volume of interest centered at the surgical tool tip can be projected onto the image frame forming a 2D region of interest illustrated as a green tinted circle. In fact, since the sphere volume of interest is fixed sized, the circle radius on the image is inversely proportional to distance from surgical tool to camera.

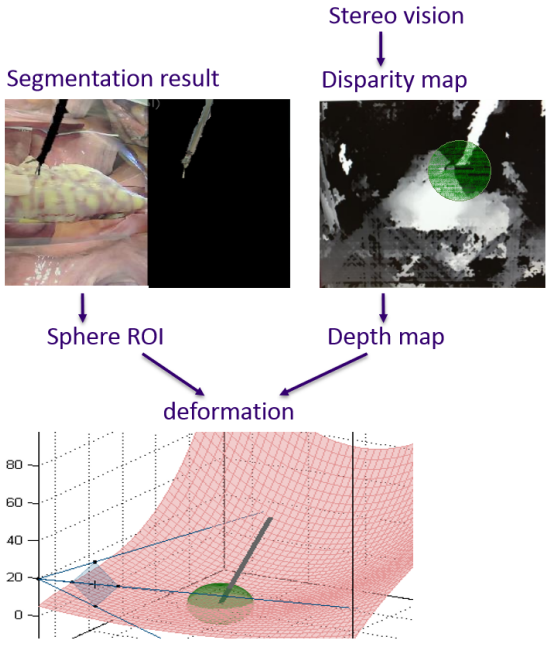


Figure 1.4: The workflow of stage 1 using a stereo camera system.

To improve the computational cost, 3D reconstruction is done only within the green tinted circle. Meanwhile, the tool segmentation result is used to identify tissue pixels on the depth map. Then, a custom feature enhancement algorithm pads the relatively featureless tool region with designated pattern so the accuracy of the resultant 3D model is ensured. Afterwards, one can get the local tissue topology near the tool-tissue interaction point. Figure 1.5 shows the overview of stage 1 in the research plan.

In [207], three different methods of 3D reconstruction from stereo laparoscopic image streams including localization and mapping (SLAM), structure from motion (SFM) and visual odometry (VO) followed by feature points triangulation are compared. A novel dataset was created for testing, and a ground truth model was acquired via high fidelity 3D scanning. The dataset was generated using a pre-calibrated stereo camera viewing a realisitc phantom surgical cavity with Raven-II surgical end effector.

The comparative experiments show that 3D reconstruction can be accomplished via motion of a single camera, yet in MIS constant camera motion can be disorienting and distracting. Alternatively, dense surgical scene reconstruction can be pursued from multiple cameras from different viewpoints. Previous work has demonstrated that multiple viewpoint autostereoscopic display (AD) technology maintains stable surgeon perception of the scene while allowing for camera repositioning [211]. It allows all cameras to remain relatively stationary while collectively streaming multiple view points. Multiple cameras are particularly amenable to dynamic scenes, not unlike the human body, e.g. caused by respiration and heartbeat. This method does not necessitate additional incision ports as cameras can be attached to the interior of the abdomen and provide multiple views once insufflated [190].

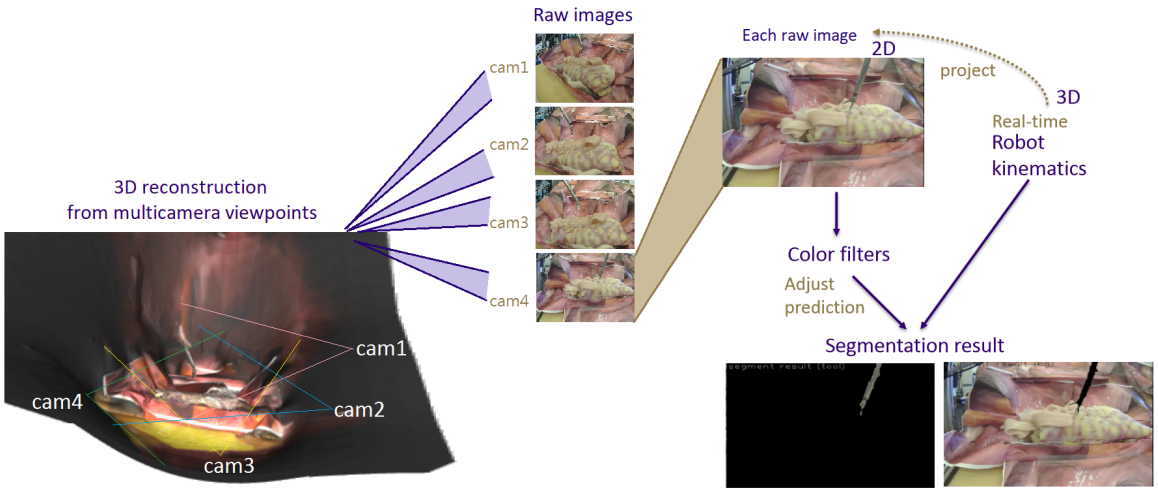


Figure 1.5: This is an overview of stage 1. The raw surgical images (middle) are sent to the tool segmentation (right) and multicamera 3D reconstruction (left) units.

COSLAM is a related work that achieves visual reconstruction using multiple independent cameras in dynamic environments [241]. However, COSLAM applications are different from MIS settings in several key aspects: (1) Camera pose is unknown in COSLAM. However, camera pose can potentially serve as a prior and lead to better reconstruction accuracy. (2) COSLAM has been implemented on room-scale environments. Surgical cavities are much smaller. (3) Camera motion in COSLAM algorithms do not deviate much from straight-line trajectories. Cameras presented here roughly orbit around the volume of interest (VOI) within the surgical cavity. These observations highlight the need to develop a specialized surgical cavity 3D reconstruction algorithm using multiple independently moving RGB cameras with known poses, which later resulted in a series of three research articles [203, 205, 206].

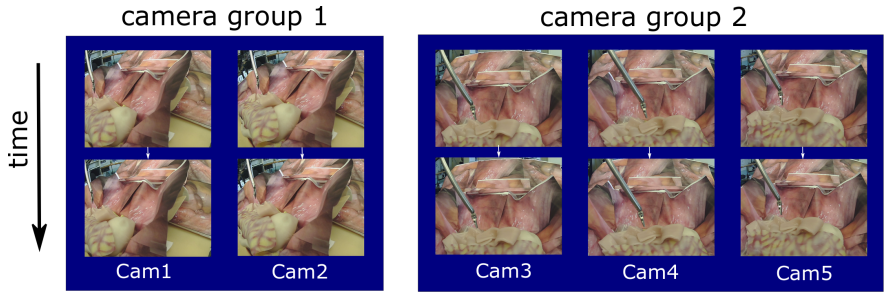


Figure 1.6: Camera grouping and (intra/inter) camera matching in a multicamera system.

[205] focuses on the topic of camera grouping and pair sequencing in a surgical environment, where I define a camera group to be a subset of cameras that share relatively similar views. These camera groups are automatically formed based on individual camera poses using a graph based algorithm. Then, pairs of images within the same camera group undergo 3D trangulation to create a 3D map. Finally, pair sequencing is an algorithm that decides the order to merge the 3D maps rendered from selected image pairs.

The subsequent two articles [203,206] in the three part series concern the temporal change in the 3D tissue model; namely, tissue deformation analysis and camera motion to achieve optimal tissue model tracking. The details will be covered in the next section.

1.2.2 Tissue Deformation Analysis and Force Rendering

Stage 2 aims to render a deformation map centered at the surgical tool interaction point. Then estimate contact force based on the deformation and a known tissue dynamics model [40]. In order to get the deformation map, non-rigid registration for consecutive 3D tissue maps is necessary. That being said, [203] extends previous findings in [205] and formulated an energy minimization function that performs non-rigid point cloud registration over time and classifies local surface patches into either static, shifting, or deforming. Again, a robot-assisted MIS setup with multiple cameras considered, as depicted in Figure 1.7.

The energy function in non-rigid registration oftentimes contain both data and regularization terms. Quadratic data terms used in [80, 116, 234] implicitly assume positional errors with Gaussian distributions and ensure corresponding points align after registration process. On the other hand, regularization terms in [116, 221] preserve smoothness by created deformation fields to fit data, affording the optimization procedure robustness to noise and outliers.

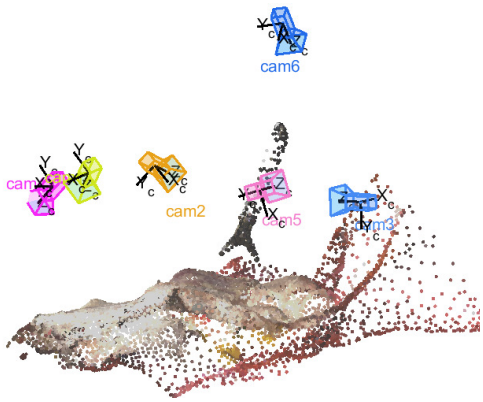


Figure 1.7: Multiple independently moving cameras from different views of the surgical cavity. Calculated geometries are represented as a point cloud.

However, soft tissue deformation resulting from natural breathing or heartbeat are large, piece-wise smooth signals residing on 3D surfaces. On the other hand, indentations due to tool-tissue interactions usually result in larger positional errors close to the incision point, with smaller errors for the remaining surfaces. Therefore, this indicates that the positional errors are sparse, and are thus better suited for and modeled by a heavy-tailed distribution instead of a Gaussian one. Such a model was incorporated in the registration method presented in [117], for which surfaces were assumed piece-wise smooth, and substantial changes in transformations could occur only in relatively local areas.

Viewpoint selection is another important factor in many robotics tasks, and may influence navigation, object recognition, environment reconstruction, camera placement and mesh simplification for polygonal models [19]. Salient relevant work include next best view planning (NBV) [1, 179] and swarm-based mapping [43, 68, 209] as they relate to configuring vision sensors for optimal view coverage. While both NBV planning and swarm-based mapping present aspects of optimal vision sensor placement, the application domains of large-scale, openly navigable spaces with rigid objects are not directly extendable to RMIS. Surgical environments, in contrast, are oftentimes spatially constrained, highly dynamic and deforming, and contain reflective surfaces.

Since manual camera positioning in robotic minimally invasive surgery is suboptimal and error-prone [158], I am interested instead in autonomous solutions. Unlike other tool-tracking focused autonomous camera positioning research [192, 223, 238], this work [206] presents a novel context-aware autonomous multicamera viewpoint adjustment pipeline from the perspective of simultaneously maintaining the surgical tool within view and providing better point coverage for real-time 3D surgical cavity reconstruction.

To render force from tissue deformation, there are roughly two main routes. In a mechanical or bio-medical approach, nonlinear viscoelastic dynamic models of tissue are determined either by offline tissue identification or directly treated as prior knowledge from a medical database, then force is calculated from the model [243]. Data scientist, however, use machine learning techniques to bridge the gap between tissue deformation and the applied force. Recurrent Neural networks (RNN) often adopted as it handles time variant features well. In these cases, they either apply a relatively simple mass-spring model to simulate tissue dynamics behavior or go entirely model-free [3]. In this research, I infer force directly by looking up a medical database for a given deformation level and tissue type. Later, I approached the vision-based force estimation idea from security stand point in robotic teleoperation, and elaborated my vision in [208].

1.2.3 Bilateral Teleoperation

In stage 3, the estimated force is sent back to a haptic device used to remotely control the surgical robot arm. In other words, a master-slave system will be setup and surgeons can perform surgical operations via a haptic device while getting real-time force feedback as though they were operating in-person. In a typical bilateral teleoperation system shown in Figure 1.8, both the visual and force sensors are provided, and yet the time-varying latency in sensor measurements as well as the various sampling rate for different sensors can cause challenges. In fact, while a visual measurements update at every 30-40 ms, force information updates at 0.5-1 ms, which is the frequency requirements of touch/tactile senses in human perception [4]. Contrary to a typical bilateral teleoperation system, the force feedback in our application is not obtained from a force/torque sensor, but instead it is derived from vision, which means that the force information is significantly sparser than if a force sensor is used. Due to computational limits with image processing in stage 1 and 2, the update rate of the deformation map is further slowed down to roughly 6-7 Hz, significantly lower than the 1000Hz force feedback update rate requirement.

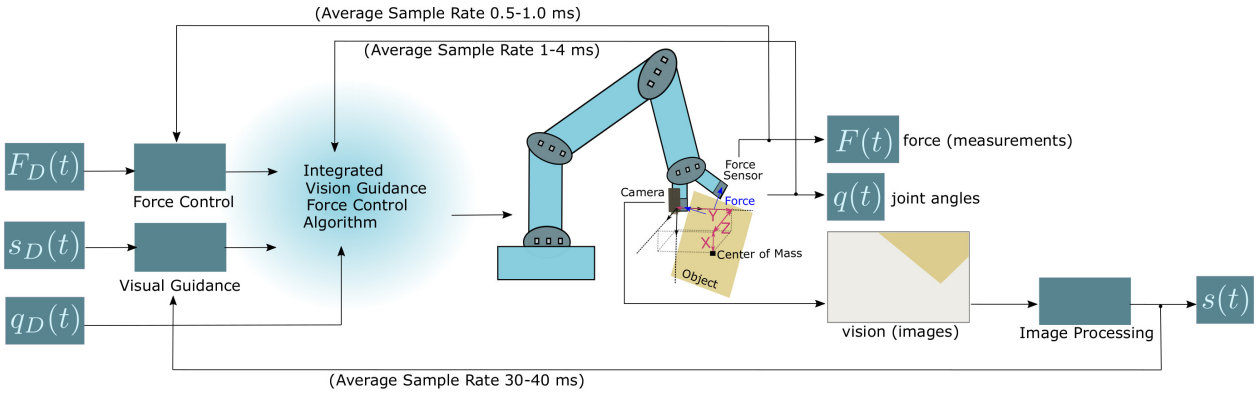


Figure 1.8: Visual guidance and force control in a robotic system.

In a fully teleoperated robotic surgery, interpolation or prediction algorithms like Kalman filtering [25] can be used to improve the force feedback quality, and surgerons can close the loop if any unrealistic haptic sensation occur. Yet, when a robot is programmed to perform

part of a surgical procedures autonomously, the synchronization and quality of visual and force feedback become very essential [152]. With the awareness that this will be an important issue in my work and that only a literature review [72] on Spanish research works were conducted, I have decided to do a survey myself [119] for existing methods in dealing with robot manipulation tasks with vision and force information.

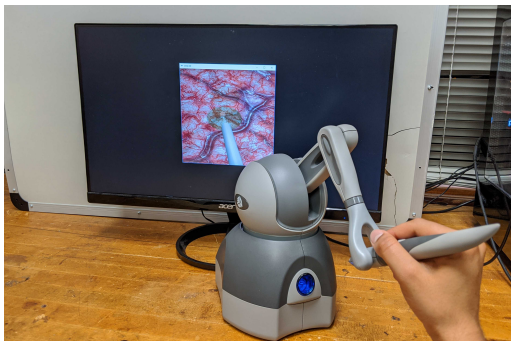


Figure 1.9: The experimental setup in [42]. Motion commands are sent and a deviated force feedback is received.

Also, instead of artificially apply an intensive interpolation on the deformation data, which will not provide a realistic haptic sensation given the current state, I decided to define it as future work, after the computation speed on the vision side is further improved through code efficiency optimization. In the meantime, I joined a collaborative research project that empirically evaluates the degree to which haptic feedback may deviate from ground truth yet result in acceptable teleoperated performance in a simulated RMIS-based palpation task [42]. In Figure 1.9, a preliminary user-study is conducted to verify the utility of the simulation platform, and the results provide implications in haptic feedback for RMIS and inform guidelines for vision-based force estimation. Adaptive thresholding is used to collect the minimum and maximum tolerable errors in force orientation and magnitude of presented haptic feedback to maintain sufficient performance.

Later, I got involved in another collaborative work [89] that investigates the possibility of reflecting additional information like robot joint limit warning in haptic feedback. Specifically, a locally sampled joint limit surface is generated and represented as a point cloud. This local point cloud is then used to provide 3 DOF haptic feedback to the operator as an indication that a joint limit has been reached, and provides kinesthetic force feedback to efficiently remove the operator from that joint limit. This work can potentially be applied to robotic surgery for surgeons to naturally intuit robot joint limits and avoid confusion.

1.3 Submitted Scholarship

From the overview, this task consists of three stages - 3D reconstruction and segmentation of surgical tool versus tissue (Chapter 2,3,4), tissue deformation analysis and force rendering (Chapter 5), and bilateral teleoperation (Chapter 6). Future work is covered in Chapter 7.

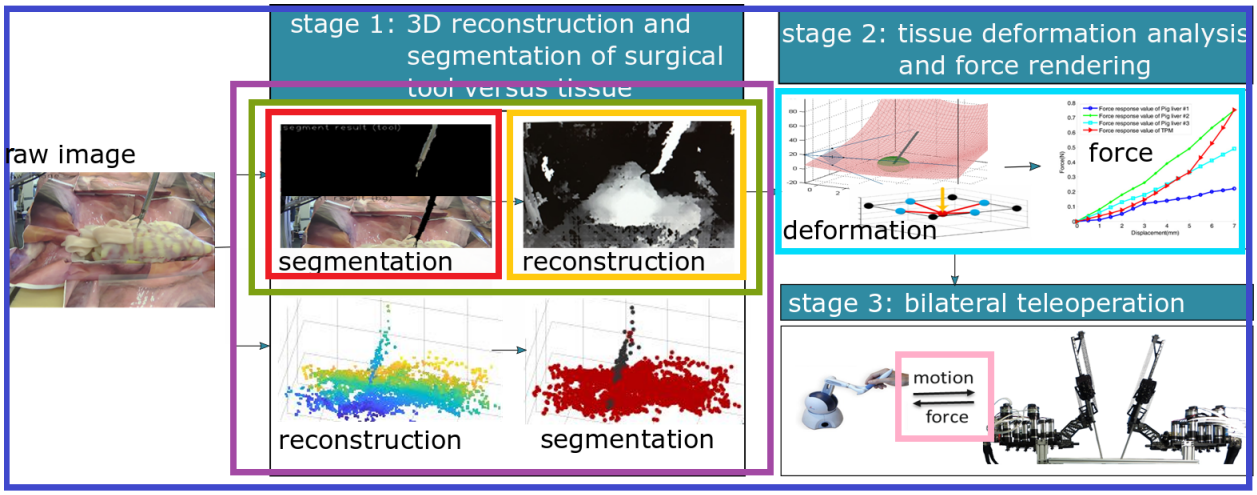


Figure 1.10: Illustration of the three stages and how the submitted scholarships are related to the entire research plan.

Several scholarships were submitted based on certain part of this research plan. Below is a summary of each of the scholarships which are marked with colored boxes in Figure 1.10.

1.3.1 Vision-Based Force Estimation Framework (blue box)

At Grace Hopper Celebration (GHC) 2018, a session talk titled "Vision-Based Surgical Tool Tracking and Force Estimation With Kinematics Prior" was proposed and accepted, where I shared the three stages of my research and preliminary results.

1.3.2 2D Surgical Tool Segmentation (red box [53] [170] and green box [204])

Surgical tool segmentation is a process that performs pixel-wise classification of 2D surgical images into two categories - surgical instrument and background tissue. It is a prerequisite

for many medical robotics applications including vision-based force estimation. Three approaches for surgical tool segmentation were implemented over the course of my PhD. At MICCAI 2017, I was involved in the Surgical Instrument Segmentation Challenge on behalf of the University of Washington, where a pure computer vision method was proposed and later submitted as a journal paper to MedIA [53]. At ISMR 2018, a real-time vision based surgical tool segmentation framework with robot kinematics prior was presented [204], and it covers Strategy I of stage 1 in the research plan. The novelty of this work is the utilization of robot kinematics as a prior information for visual processing, which significantly speeds up the computation. In ICRA 2019 [170], we further proved that the kinematics prior mask not only fused well with traditional computer vision results, but is also beneficial when combined with data-driven method using CNN. See Chapter 2 for details.

1.3.3 A Comparison of 3D Surgical Tool Segmentation Strategies (purple box [91])

Previously in section *1.3.2*, the framework was designed to perform 2D image segmentation of the surgical tool followed by 3D reconstruction of the background tissue. However, I was inspired by a feedback during my Qualifying exam to swap the order of segmentation and 3D reconstruction. So, I named the default sequence Strategy I and the reversed version Strategy II, and started a study to comparatively evaluate the performances. This work was later published at IROS 2018 in a paper titled, "Comparison of 3D Surgical Tool Segmentation Procedures with Robot Kinematics Prior" [91]. More results can be found in Chapter 3.

1.3.4 Multicamera Dynamic Surgical Cavity 3D Reconstruction (yellow box [205, 207])

Real-time surgical scene 3D reconstruction from multiple camera viewpoints is a crucial step that concludes stage 1 in the research plan. Since numerous related research exist, I started with an experimental comparison of common 3D reconstruction algorithms including SLAM, Visual Odometry (VO) and Structure from Motion (SfM) in robot-assisted MIS and evaluated their respective performances in an SII 2020 paper titled "A Comparison of Surgical Cavity

3D Reconstruction Methods" [207]. Through the review, I identified the need to develop a specialized 3D reconstruction framework tailored for robot-assisted MIS which adapts well with specular reflections, the highly deforming nature of soft tissue and the spherical spatial distribution of the camera swarm along the inner abdominal wall. The proposed framework rooted on the novel concept of camera grouping and pair sequencing was later implemented and accepted by ISMR 2019 [205]. More results will be provided in Chapter 4.

1.3.5 Tissue Deformation Analysis and Force Estimation (cyan box [203, 206, 208])

Up to this point, a 3D model of the local tissue patch can be derived from a set of surgical images from various viewpoints at any time instance. Yet, vision-based force estimation requires tissue deformation analysis; namely, information about the temporal change in the 3D tissue model. To that end, I presented an energy minimization based non-rigid registration and point classification algorithm at IROS 2019 [203] that is suitable for a piece-wise smooth surgical scene with highly dynamic motion close to the tool tissue contact point and near stationary tissue patches further away. During my work in multicamera 3D reconstruction, I received multiple comments about the minimum number of cameras needed for a sufficiently accurate 3D model. I believe this question boils down to two factors - (a) the topological complexity of the tissue patch, and (b) derivation of an optimal camera motion algorithm for a good 3D reconstruction result so that fewer cameras are needed. Although (a) is out of our control, I implemented an autonomous camera viewpoint adjustment algorithm to address problem (b) and submitted the work to ISMR 2020 [206]. Finally, I wrapped up stage 2 of the research plan by feeding the deformation data into a tissue model that outputs the 3D contact force vector given a particular tissue type. I later approached the visual force estimation concept from a cyber security perspective and submitted the work to IRC 2020 in a paper titled "Securing Robot-assisted Minimally Invasive Surgery through Perception Complementarities" [208]. Find Chapter 5 for more details.

1.3.6 Haptic Feedback in Robotic Surgery (pink box [42, 89, 119])

In order to realize bilateral teleoperation in stage 3 of the research plan, a survey of state-of-the-art combined vision and force control methods in robotic manipulation is conducted, which I am preparing to submit as a journal paper titled "A Review on Integration of Vision and Force in Robot Manipulations." to Advanced Robotics (AD) [119]. I outlined current use-cases and landscape for combined vision and force control approaches and envision to provide a guideline for researchers to pick the ideal visual, force, or combined control scheme in their particular robotic application. Meanwhile, the vision derived force is bound to include some extent of error. To investigate the tolerable inaccuracy in the force feedback during teleoperated robotic surgery, I was involved in a collaborative user study that required subjects to repeatedly palpate a tissue membrane while receiving haptic feedback that was deviated either in magnitude or orientation. The experimental results were analyzed and accepted to EMBC 2020 [42]. In another ICARM 2019 paper [89], I collaboratively worked on a research project that studied the possibility to incorporate contact force information and robot joint limit warning in tandem into haptic feedback during robot teleoperation. More details will be provided in Chapter 6.

Chapter 2

2D SURGICAL TOOL SEGMENTATION

In vision-based force estimation for robot-assisted minimally invasive surgeries, the force information is derived from the perceived tissue deformation due to the tool-tissue contact in the laparoscopic images. That being said, being able to automatically identify the surgical tool region in the surgical image sequence in real-time is a prerequisite. While object recognition with natural features only (no barcode) can be a challenging task itself in the general case, in our application, since both the camera and the surgical tools are mounted on a robot that provides kinematics information, the extrinsic parameters of the camera as well as the kinematics information from the surgical robot can be integrated and provide a helpful cue to estimate the projected surgical tool on the image plane, which in our implementation, is called the shape prior mask U. With the help of the shape prior mask, some color filtering scheme tailored for surgical images and a transformation to the frequency domain to speed up the computation, the surgical tool can be segmented under pixel level in real-time. The detailed methodologies and algorithms [204] are presented in the following.

Apart from this work, two other surgical tool segmentation approaches were explored. The first one examines the possibility of using pure computer vision alone [53] and the second one investigated the combined result from vision based machine learning and robot kinematics information [170]. Both studies are mentioned in **section 2.5**.

2.1 Methods

2.1.1 Camera Pose Estimation

In order to use robot kinematics to ascertain the surgical instrument's location within the camera image plane, defining the camera frame with respect to the robot base frame is required. In what is often referred to as the perspective-n-point (PNP) problem, the aim is to estimate the pose of an object given n 3D points on the object and their corresponding 2D projections onto the image plane. This process also requires the camera intrinsic parameters. Instead, we require the converse, that is to determine camera pose with respect to the object.

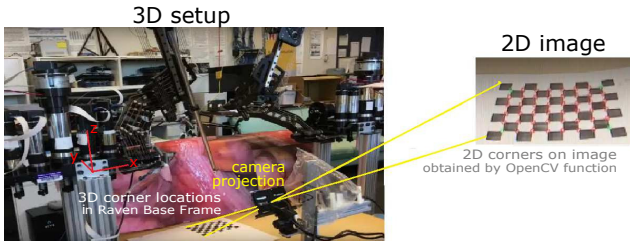


Figure 2.1: The 2D and 3D coordinate input of checkerboard corners to the PNP algorithm. From this, the transformation from Raven II base frame to camera frame is obtained.

First, the 2D (x, y) projections were generated from function `cvFindChessboardCorners` [22] in OpenCV to detect checkerboard corners in the image frame, as illustrated in Figure 2.1. The 3D corner locations of the 48 checkerboard corners were obtained by manually measuring the corner positions with respect to the Raven II base frame. Combined with camera intrinsic parameters, determining the transformation matrix between robot and camera frame is trivial. Figure 2.1 illustrates the two coordinate frames.

2.1.2 Kinematics Shape Prior Mask

Given robot joint states, forward kinematics, and camera pose, a raw projection of joint locations onto the camera image plane is straightforward. Then, from the physical thickness of each robot link, the perceived thickness on either end of a robot link in the image is inferred respectively. Suppose an object point with known width W is distance D from camera with focal length F. Then the apparent width in camera pixels, P, is defined as

$$P = {}^{FW}/_{D} \tag{2.1}$$

The overall shape of the projected robot tool can be obtained with simple trigonometry. The union of these pixels forms the initial shape prior mask, u. This is more computationally efficient than projecting all points on the tool surface.

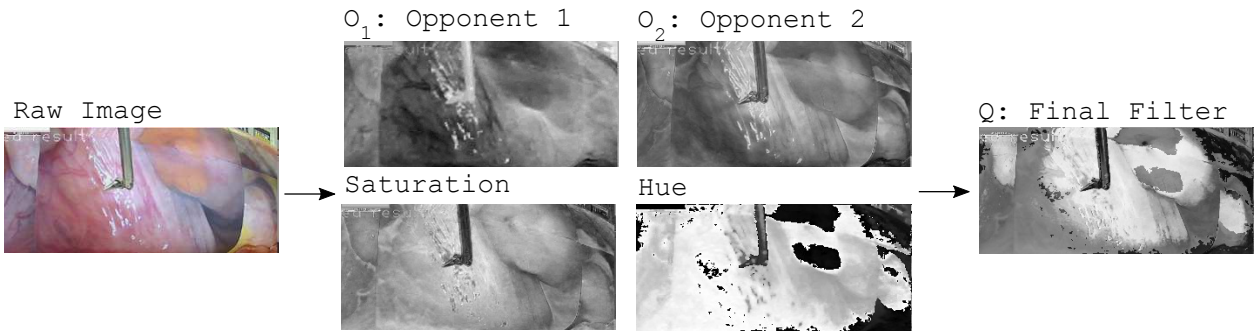


Figure 2.2: Colorspace components used to determine color mask. The likelihood map Q is defined by a weighted sum of these components: hue, saturation, O_1 and O_2.

2.1.3 Log-likelihood Color Mask

Once the robot kinematics shape prior u is generated, color filtering across the entire image further refines the prior estimate. The color filtering scheme adapted for this work was based upon work by Van De Sande et al., which claimed that hue and saturation in the HSV colorspace and Opponent_1 and Opponent_2 (denoted O_1 and O_2) are colorspace components providing the most discriminative power to separate surgical tool pixels from background pixels [95], where:

$$\begin{aligned} O_1 &= G - R \\ O_2 &= B - Y = B - (G + R) \end{aligned}$$

The log-likelihood mask (Q) is then defined as

$$Q = w_1 H + w_2 S + w_3 O_1 + w_4 O_2$$

where H and S are the hue and saturation components respectively and the weights w_1, w_2, w_3, w_4 were heuristically tuned. Because HSV is a non-Euclidean colorspace, the coneHSV

colorspace [167] was adopted for use during color comparisons, where (H, S, V) values are transformed into $(V, S\cos(H), S\sin(H))$. The used colorspace components are shown in Figure 2.2. An ideal post filtering image will appear bright for the tissue pixels and significantly darker for surgical tool pixels.

2.1.4 Frequency Domain Shape Matching

The two masks, u and Q, provide two estimates of the surgical tool shape within the image frame. The robot kinematics shape prior mask, u, was derived by projecting surgical tool configurations and thickness onto the image plane. Meanwhile, the log-likelihood mask, Q, was generated via a linear combination of four colorspace components.

In mask u, pixels corresponding to surgical tools are white (255 in 8-bit gray), while the remaining pixels are black (0 in 8-bit gray). The converse is true for Q, that is the determined tool pixels approach 0 (black), and the rest of the image approaches 255 (white). Therefore, ideally the black pixels in u should correspond to white pixels in Q and vice versa. Multiplying ideal masks pixel-wise should result in all zeros.

However, u may not align well with Q, as shown by comparing Figure 2.3-b and Figure 2.3-c This can be due to inaccuracies in camera extrinsic parameters, robot kinematics and joint sensors, or timing mismatches between robot pose and image frame. An objective function defined as the sum of pixel-wise multiplication between u and Q can be interpreted as the energy, E, to be minimized for optimal alignment. Suppose there are N_R rows and N_C columns in both images A, B. Then E is defined as

$$E(A, B) = \sum_{y=1}^{N_R} \sum_{x=1}^{N_C} A(x, y)\, B(x, y) \tag{2.2}$$

The mask matching procedure aims to modify shape prior mask u to best match Q and thus minimize E, and is achieved in two optimization steps:

- Finding the optimal translation for u to match Q, generating the shifted mask U.
- Finding the optimal rotation and scaling for U to match Q, generating mask $\mathbb{U}$.

Translation Suppose a translational error exists between shape prior u and actual surgical tool image location. To counteract this error, a translational offset which minimizes E is sought. Let $\vec{t} = (t_x, t_y)$, and then define $u_{\vec{t}}$ as the resultant mask of u translated by $\vec{t}$. The optimal translated mask is denoted U

$$U = \arg\min_{u_{\vec{t}}} E(u_{\vec{t}}, Q) \tag{2.3}$$

The solution to this optimization is achieved efficiently in the frequency domain using duality between spatial and frequency domains [191], namely

$$u \circledast Q = \mathbb{F}^{-1}(\mathbb{F}_Q \mathbb{F}_u^*) \tag{2.4}$$

where $\circledast$ denotes spatial convolution and $\mathbb{F}$ the Discrete Fourier Transform (DFT). Consider pixel (t_x, t_y) of the spatial convolution (origin is center of image):

$$u \circledast Q(t_x, t_y) = \sum_{y=1}^{N_R} \sum_{x=1}^{N_C} u\left(x - t_x, y - t_y\right) Q\left(x, y\right)$$

which is precisely $E(u_{\vec{t}}, Q)$. Thus the optimal offset is determined by the minimum pixel of $u \circledast Q = \mathbb{F}^{-1}(\mathbb{F}_Q \mathbb{F}_u^*)$. The time complexity reduces from $\mathbb{O}\left(N^4\right)$ to $\mathbb{O}\left(N^2 \log N\right)$ using the DFT. Figure 2.3 outlines the procedure.

Consider Figure 2.4f, where Q, u, U are respectively marked with red, blue, and green. The green is a translated version of blue that better matches red.

Rotation and Scale The two masks U and Q may also misalign in rotation and scale. To account for this, the masks U and Q were first transformed to log-polar coordinates and zero-padded, forming U' and Q'. A Cartesian coordinate (x, y) is represented in log-polar

coordinates as (a, b) where

$$\begin{aligned} a &= \log \sqrt{x^2 + y^2} \\ b &= \text{atan2}(y, x) \end{aligned}$$

a and b correspond to scale and rotation respectively. Finding the minimum pixel of $U' \circledast Q'$ thus determined the optimal scale and rotation of U to best match Q. The scaled and rotated version of U is denoted $\mathbb{U}$.

(a) (b) (c) (d)

Figure 2.3: Translation matching. (a) raw image frame (b) initial shape prior mask, u (c) color filtering mask, Q (d) two masks convolved, minimum value gives optimal translational offset to generate mask U.

Consider Figure 2.4l, where Q, U, $\mathbb{U}$ are respectively marked with red, blue, and green. The green is a scaled and rotated version of blue that better matches red. Theoretically, due to nonlinear coupling of the two steps, the global optimum is achieved by interchangeably applying translation and rotation/scale adjustments until convergence, but only one iteration is applied in this work, under the assumption that the kinematics data is of high accuracy, to improve efficiency.

2.2 Experimental Design

Figure 2.5 shows the experimental system setup using the Raven II platform. A 40mm baseline stereo camera with 640×480 pixel resolution was fixed to the Raven II base frame to acquire image data. Realistic tissue images were placed in the background.

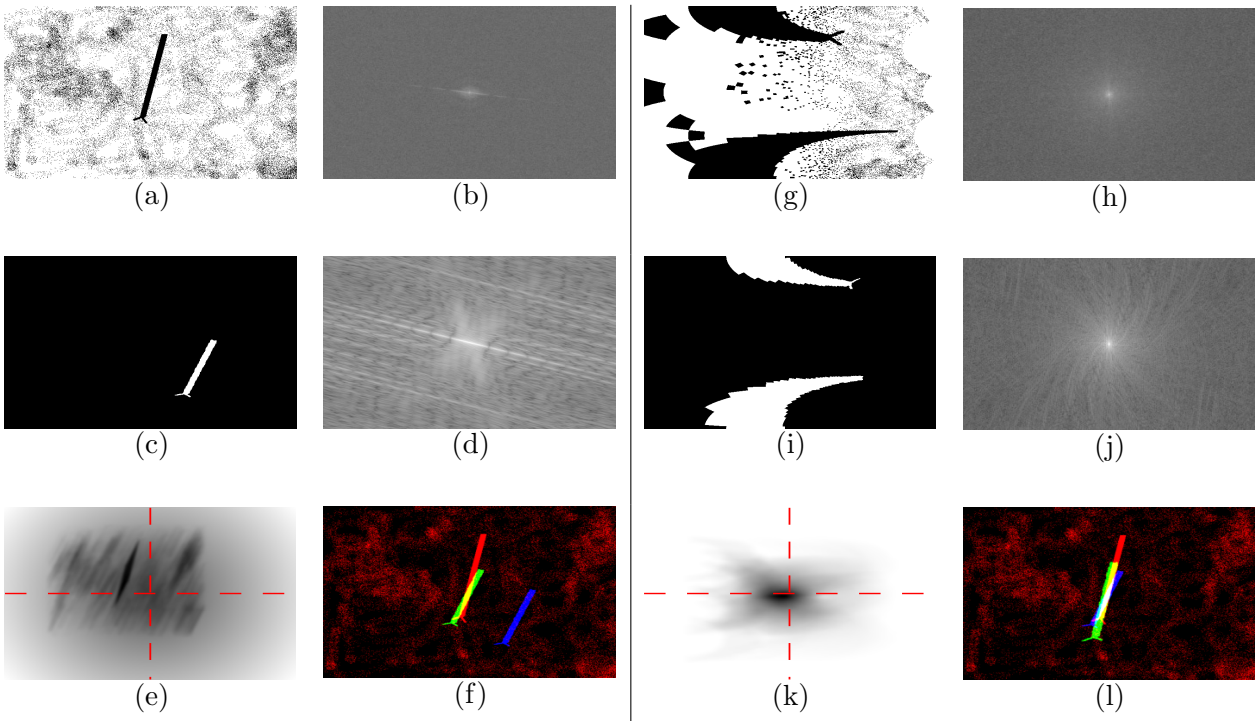


Figure 2.4: DFT shape matching, (a)-(f) illustrate translation matching to find U, (g)-(l) rotation and scale matching to find $\mathbb{U}$. (a) color filter mask Q (b) Fourier transform $\mathbb{F}_Q$ (c) shape prior mask u (d) Fourier transform $\mathbb{F}_u$ (e) convolution $u \circledast Q$ (f) red - Q, blue - u, green - U (g) log-polar color mask Q' (h) Fourier transform $\mathbb{F}_{Q'}$ (i) log-polar shape mask U' (j) Fourier transform $\mathbb{F}_{U'}$ (k) convolution $U' \circledast Q'$ (l) red - Q, blue - U, green - $\mathbb{U}$.

2.2.1 Robot Kinematics Shape Prior

The joint locations of the Raven-II platform were obtained from encoder readings and forward kinematics. The positions can be projected onto the camera image plane and shape can be determined via Eq.2.1. With raw position data, a static positioning error was observed. This was compensated with a

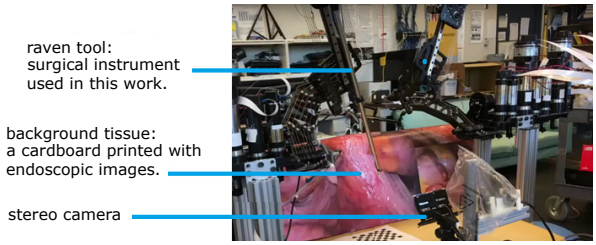


Figure 2.5: Experimental setup, includes Raven II and stereo camera hardware.

static offset added to the initial robot pose estimate as illustrated in Figure 2.6.

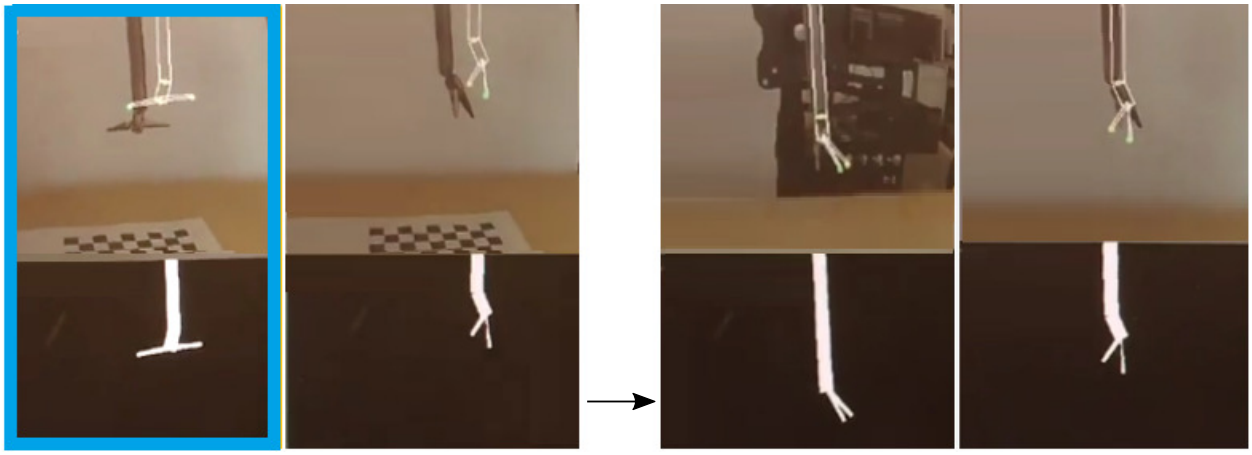


Figure 2.6: The left shows raw projection of robot joints and initial shape prior of two poses without static offset. The right shows with static offset.

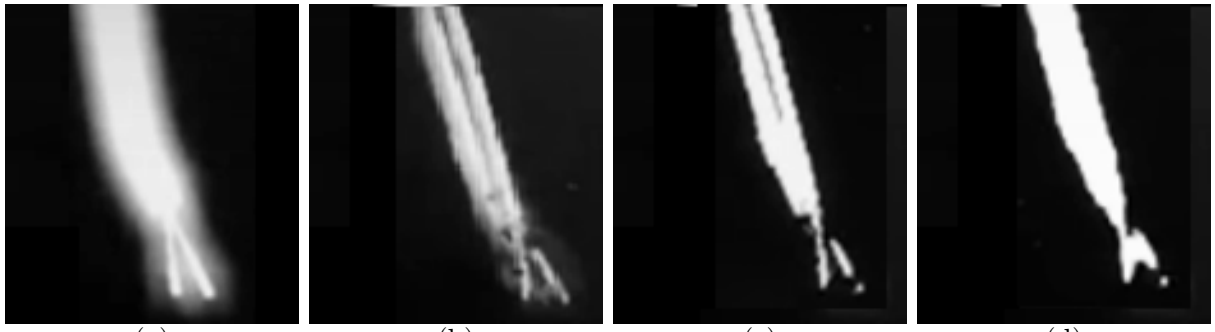
2.2.2 Frequency Domain Shape Matching

The main source of misalignment between color mask and shape prior mask arises from latency between image stream and robot kinematic information. That is, the robot pose is sampled slightly prior to the image frame with some variance. The methods described in Section 2.1.4 were used to determine optimal mask shift, scaling and rotation. In practice, to avoid mismatching different tools in view, a 2D Gaussian distributed penalty map centered at the origin is fused with $u \circledast Q$ to bias the optimal translational solution towards smaller magnitude, under the assumption that initial shape prior u is close to true tool projection,.

2.2.3 Color Mask Post Processing

A final color mask was used to account for the effects of partial occlusion from real tissue and trivial ambiguity near segmentation boundaries. As illustrated in Figure 2.7, there are four steps to turn the nicely aligned shape prior $\mathbb{U}$ into the actual binary segmentation result.

First, mask borders were expanded outward to tolerate trivial edge misalignment using



(a) (b) (c) (d)

Figure 2.7: Final color mask procedure. (a) dilated and blurred edges (b) log-likelihood color mask (c) binary threshold (d) morphological operations, resulting in final mask.

OpenCV functions `dilate` and `blur` [22] (Figure 2.7a). Then, the same log-likelihood color filter for generating Q was applied. This helps to eliminate tool pixels partially occluded by real tissue (Figure 2.7b). Observe that this step incorrectly removed some tool pixels, due to the reflective nature of the tool. Next, a simple binary threshold classifies pixels as either tool or non-tool (Figure 2.7c). Finally, to account for the misclassified pixels due to the tool reflection, the OpenCV function `morphologyEX` was used to dilate boundaries and eliminate noise [22]. This results in the final segmentation mask as shown in Figure 2.7d.

2.3 Results and Discussion

2.3.1 Raven II Tool Segmentation

Figure 2.8 illustrates the final results of the real-time image-based surgical tool segmentation with robot kinematics shape prior. This was performed with the Raven II surgical robot platform, and the final mask was achieved using the techniques and workflow described in sections 2.1 and 2.2. Overlaying the final mask with the raw image allows for segmentation of foreground (surgical tool tip) and background. The results shown here were achieved at a refresh rate of approximately 6 Hz using a commodity workstation.

Figure 2.8: Raven II tool segmentation. (a) raw image (b) final shape prior mask (c) segmented foreground tool (d) segmented background tissue.

From the robot kinematics and information about static pose estimation offset, an initial raw shape prior mask, u, is first generated. A log-likelihood color mask, Q, is created from raw image data. These two masks are then convolved (using duality property and DFT) to estimate optimal translation to match the shape prior to color mask, generating translated shape prior, U. Masks U, Q were then converted to log-polar coordinates, where they were again convolved to estimate optimal scale and rotation of the shape prior mask to match the color mask, generating mask $\mathbb{U}$. A post process color mask produces the final shape prior.

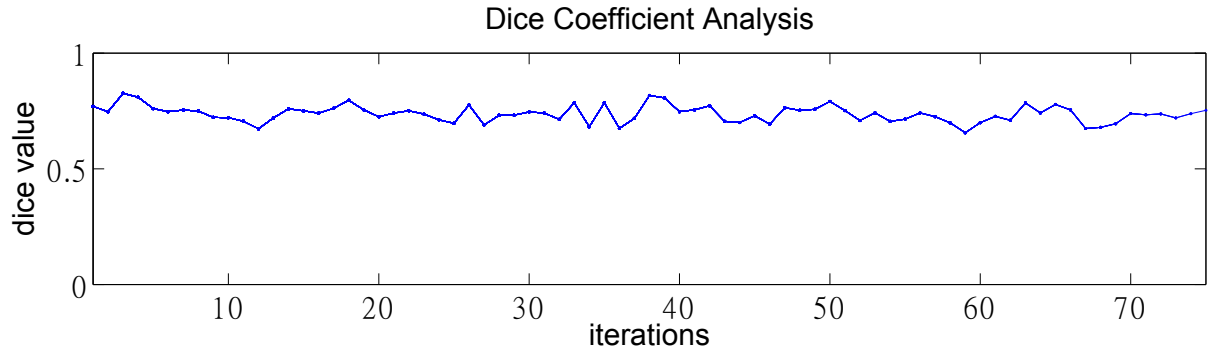


Figure 2.9: Sørensen-Dice indices for 75 analyzed frames. Manually labeled mask pixels were compared to real-time generated mask pixels.

2.3.2 Sørensen-Dice Index Analysis

The Sørensen-Dice index was used to measure the accuracy of the automatic real-time surgical tool segmentation. For evaluation, data was collected by actuating the Raven II tool

along a trajectory traversing a wide variety of joint configurations while staying within the image frame. The maximum displacement and rotational speed of motion are $10cm/s$ and $30°/s$ respectively, which meets standard surgical operation requirements [228]. Image frames (640×480 pixels) were captured and processed in real-time as described in sections 2.1 and 2.2. 75 of the 2000 frames were randomly selected and manually labeled offline to classify tool from background. These manually labeled masks formed the ground truth X against which the real-time, automatically generated masks Y were evaluated.

The Sørensen-Dice index is a measure of similarity between two datasets defined as

$$QS = \frac{2|X \cap Y|}{|X| + |Y|} \tag{2.5}$$

When datasets X, Y are identical, $QS = 1$, while disjoint X, Y result in $QS = 0$. For each of the 75 images, the ground truth dataset included pixel locations of the manually labeled surgical tool. The experimental set included segmented tool pixel locations generated by the proposed method. An average dice coefficient of 0.7372 is achieved, which compares well to state-of-the-art graphics-accelerated methods [112]. Sørensen-Dice indices for each individual frame over time are depicted in Figure 2.9.

A broad selection of tool poses were captured for this analysis, and Figure 2.10 demonstrates the very slight dependence that tool configuration bears on the Sørensen-Dice index. This suggests that the method is robust to varying tool configurations.

2.3.3 Color Spectrum Stochastic Modeling

In this work, the log likelihood map Q was a weighted sum of the Opponent, RGB and HSV color components, and was essential for the shape matching of kinematics prior mask $\mathbb{U}$. While computationally efficient, a more discriminative color filtering scheme is possible through statistical analysis. To that end, a large number of surgical operation images were manually labeled and analyzed for color space components. From this, the probability distri-

butions of tool pixels and non-tool pixels along each color space component can be generated. This statistical representation in color space, as shown in Figure 2.11, can provide the means for an advanced color filtering scheme.

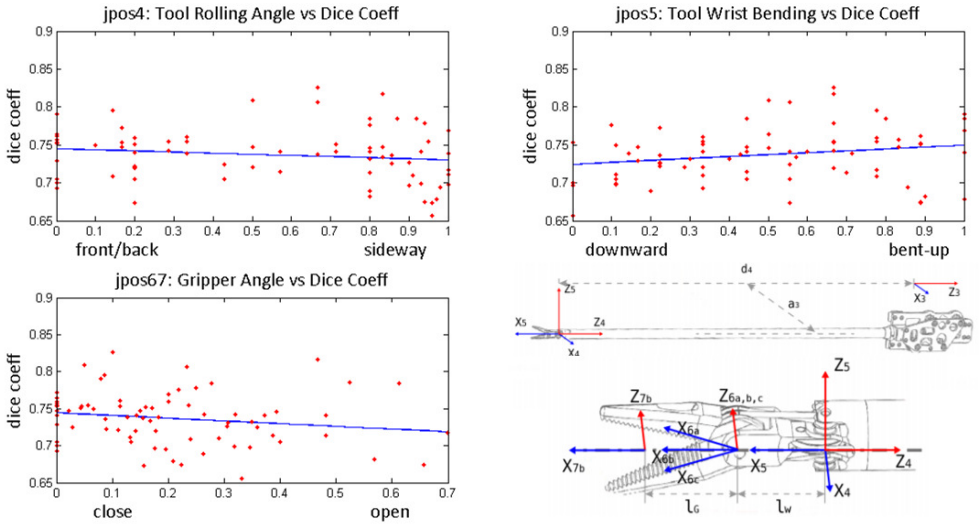


Figure 2.10: The correlation between each tool joint and dice coefficient output.

2.4 Conclusion

This chapter describes a method for real-time vision-based surgical instrument segmentation with kinematic prior [204]. The method affords a notable combination of attributes, including

- results validating use with Raven II tools.
- low computational complexity.
- 6Hz execution rate without GPU acceleration.
- average Sørensen-Dice index > 0.73.
- robustness to partial occlusion by fusing robot kinematics with color filtering.

The technique was evaluated on the Raven II surgical platform, and segmentation results were compared with manually segmented images. The results were encouraging with high Sørensen-Dice index that is robust to tool configuration. Thus, the method is promising

towards the use of kinematic prior and color masking for real-time tool segmentation in a robot-assisted minimally invasive surgical setting.

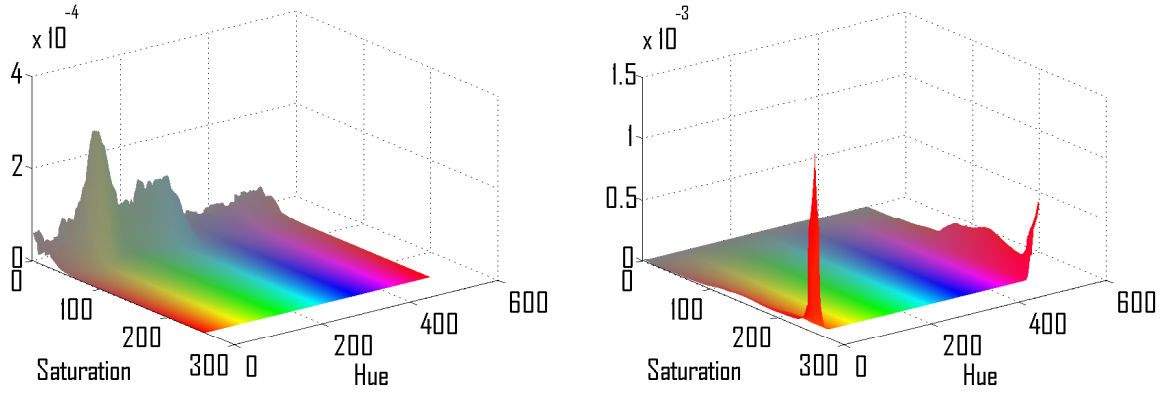


Figure 2.11: Color statistics. (a) probability distribution of tool pixels (b) probability distribution of non-tool pixels.

Future improvements to the proposed image segmentation method include stochastic modeling of surgical tool and tissue pixels, as described in section 2.3.3. This can greatly improve the generation of log likelihood color mask Q. Furthermore, static offset correction of estimated robot pose can be automated through Kalman filtering. Validating the method in various lighting conditions and with a reduced baseline stereo camera (or endoscopic) setup will further promote this method towards clinical issue. A natural extension of this work includes exploring the remaining subtasks of the vision-based force estimation as illustrated in Figure 1.10 while integrating the segmentation method described here.

2.5 Further Study

Other than the proposed surgical tool segmentation approach [204], I also collaboratively worked on two side projects on the same topic. The first project was a MICCAI surgical tool segmentation challenge with pure computer vision methods [53], and the second project was a data-drive approach to tool segmentation fused with kinematics prior [170].

2.5.1 Surgical Tool Segmentation with Pure Computer Vision

In Summer 2017, Niveditha Kalavakonda and I teamed up and participated in the 2017 MICCAI Endoscopic Vision Challenge - surgical instrument segmentation hosted by Intuitive Surgical Inc. The dataset was made up of 10 sequences of abdominal porcine procedures recorded using Da Vinci systems. From each of the 8 surgical procedures, 225 frames (down-sampled to 1Hz) were selected from the stereo video sequences as training data and the last 75 frames were kept as test data. Our team was among the top 10 of all the contestants and a joint paper describing the method from each team is submitted [53].

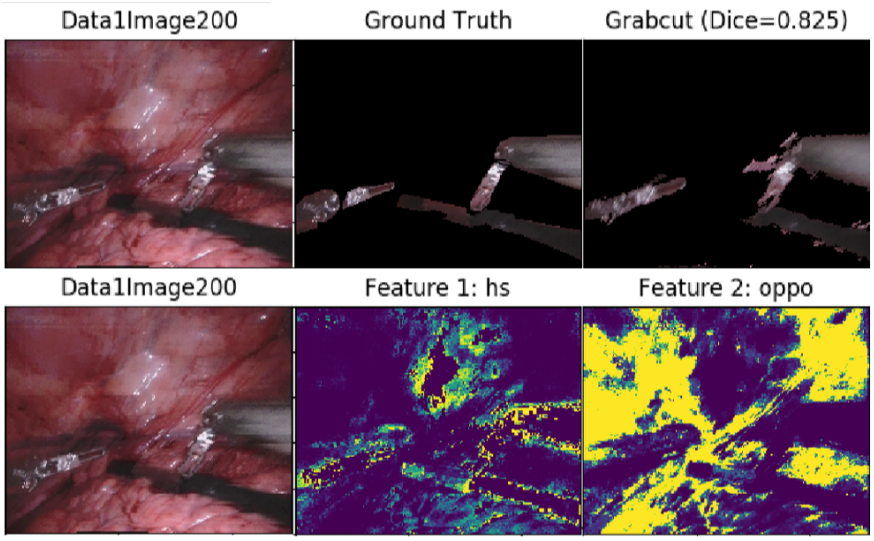


Figure 2.12: A sample segmentation result using our method in the EndoVis Challenge.

Unlike the solutions from many other teams, we were interested in developing a surgical tool segmentation method without machine learning, and see how far traditional computer vision approaches could go in this matter. The motivation comes from the lack of massive pre-labeled surgical image dataset in general situations [105]. In fact, as shown in our previous work in [204], the robot-held surgical tools can be segmented in real-time with an average dice coefficient [112] of 0.7372 by including additional information of robot kinematics data and camera extrinsic parameters.

Many image features including shape masking, edge constraints, border constraints, disparity discontinuity and color filtering have potential impact on the segmentation result. If a weighted sum of the features is computed, one may be able to determine whether each pixel belongs to a tool or tissue. Depending on the blurriness of the image, the weights for the features vary. For example, edge constraints and disparity information are less reliable in blurry images, and can be misleading in the case of interlacing [193]. Every time a new image comes in, a high level classification of image blurriness is conducted. Then, corresponding weighting factors are applied based on its blurriness score. Finally, a probability mask is generated from the weighted sum of the features. Figure 2.12 shows one of the well performed sample image processed using this method.

Few advantages for this algorithm are that no training is required, and efficient on-line execution. Thus, this work can be considered a preliminary study on the image segmentation part of stage 1 in the research plan, which is marked with a red box in Figure 1.10.

2.5.2 A Data-Driven Surgical Tool Segmentation Approach with Robot Kinematic Prior

This is another side project I worked jointly with Fangbo Qin, a visiting PhD student from China. Inspired by the other contestants in the EndoVis Challenge, we decided to focus on improving one of the best-performing convolution neural network in surgical tool segmentation - the ToolNet. The proposed CNN model ToolNet-C has the capability of learning features from numerous unlabeled images and learning segmentation from few labeled images, making the application more convenient. The kinematic pose based silhouette projection is implemented leveraging the prior knowledge of instrument 3D shape. However, problems exist with both solutions alone. Specifically, reflection issue occurs in surgical instrument segmentation from endoscopic vision which causes false negatives and holes in the segmented surgical tool, at the same time there is the drifting effect with robot kinematics solution. So, we proposed data fusion of CNN prediction and kinematic pose, where the data fusion is realized by the particle filter to refine the kinematic pose. The weight suppression and shape

matching likelihood are proposed to effect the resampling and weighting of particles, respectively. The experiments showed the proposed ToolNet-C model could be learned with only 30 labeled images without using data augmentation, and the proposed data fusion method could increase the segmentation performance significantly. In the future, the time cost of the silhouette projection will be further reduced, so that more particles can be employed in real time to enhance the particle filter.

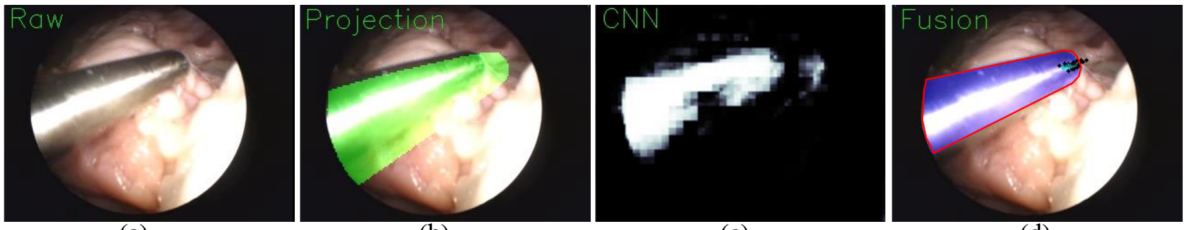


Figure 2.13: Segmentation results with data fusion. (a) Raw image, (b) robot kinematics projection, (c) prediction of ToolNet, (d) fusion of kinematics and machine learning methods.

In this work, Fangbo was in charge of experimenting with implementing the Convolution Neural Network (CNN) algorithm, revising the loss function and exploring feature extraction methods, while I gave advice on integrating robot kinematics information with the machine learning results. By combining the machine-learning-based surgical tool segmentation approaches with robot kinematics prior information, as shown in Figure 2.13, it is proven that the segmentation accuracy is further improved. The work was later submitted to [170].

Chapter 3

A COMPARISON OF 3D SURGICAL TOOL SEGMENTATION STRATEGIES

In stage 1 of the research plan shown in Figure 1.1, 3D reconstruction and segmentation of the surgical scene is required. Although image segmentation and 3D reconstruction is respectively discussed in previous chapters 2 and 3 and implemented chronically, an interesting discussion arose during my qualifying exam of the potential effects and performances if the order of segmentation and reconstruction is reversed. To find out the result, two 3D segmentation and reconstruction strategies were defined and then a series of experiments were conducted with the two strategies on the same surgical dataset. The result was accepted to IROS 2018 [91]. Below is a summary of this study.

3.1 Background

3D reconstruction and surgical tool segmentation are necessary for several advanced tasks in robot-assisted laparoscopic surgery. These tasks include vision-based force estimation, surgical guidance, and medical image registration where pre-operative data (CT or MRI scan image slices) are overlaid on patient anatomy in real-time during surgery [123] to name a few. In this work, two main strategies were considered: (1) initialize with surgical tool segmentation from 2D images, then proceed to local 3D reconstruction near the tool-tissue interaction region by projecting the segmented result into 3D space, and (2) initialize with 3D reconstruction of the entire surgical task space, followed by surgical tool segmentation from within the 3D reconstructed model. Both methods were implemented on the Raven-II surgical robot system, and accuracy and time complexity for both methods were compara-

tively analyzed while considering various task parameters. Finally, based on the results of this work, guidelines for selecting reconstruction and segmentation strategies and procedure for particular situations are outlined.

3.1.1 Related Work

Surgical Scene 3D Reconstruction

High level classifications of surgical scene 3D reconstruction algorithms have been formulated based on camera motion and scene type. In particular, methods without dependencies on camera motion utilize different visual cues, including stereo [195] [175], actively projected spatial patterns [166] [82], and shading and shadows [122] [140] to achieve reconstruction.

Typical laparoscopic cameras move frequently during operation, and thus SFM can be leveraged to recover 3D structure. For rigid scenes, SLAM methods that simultaneously estimate 3D structure and camera motion exist [73] [13] [216] [120]. With regard to feature points extraction, detectors specific to surgical scenes have shown better results than SURF or SIFT alone [124].

Kinematics-Based Tool Segmentation

Several segmentation methods with robot kinematics prior exist which do not rely on visual markers [7] [8] [52]. The segmentation method used in this work determines the optimal alignment between robot kinematics prior and color mask by DFT shape matching, which was shown to achieve segmentation with average Sørensen-Dice index greater than 0.73 at 6Hz without GPU acceleration. Furthermore, this method is amenable to Raven II tools [204].

3.1.2 Contribution

To the best of the author's knowledge, this work is the first to *analyze and compare strategies for surgical tool 3D segmentation by interchanging the order of 3D reconstruction and segmentation across several parameters.* This study informs a preferable methodology for stage 1 in the vision based force estimation proposal and serves as a guideline for future and related ongoing work within the parameters of the hypothesis matrix define below.

Hypothesis Matrix

Several task parameters are considered when comparing the accuracy and time complexity between the two proposed methods of 3D reconstruction and segmentation. In particular, motion and pose of both the cameras and surgical tool are considered, resulting in four separate parameters. This is shown in Figure 3.1.

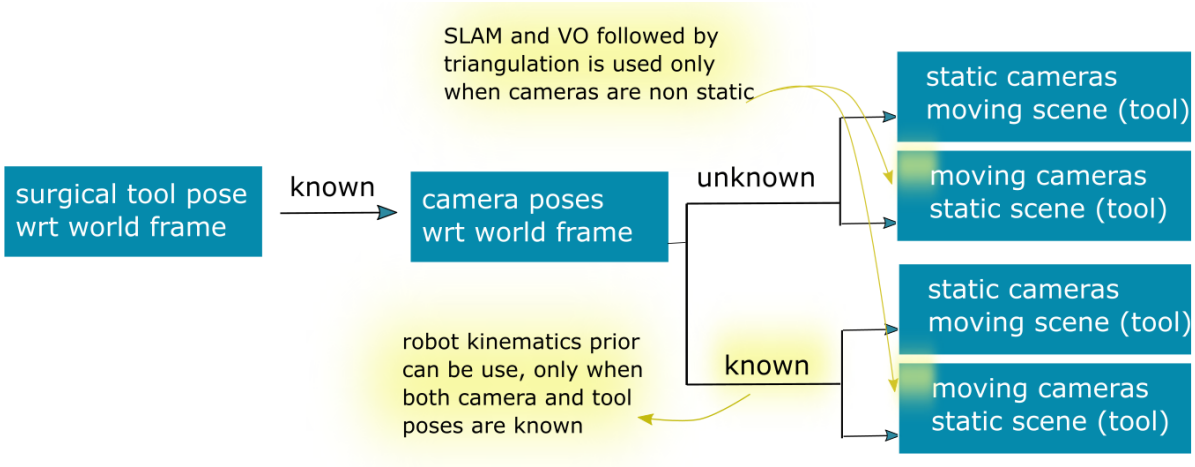


Figure 3.1: Parameters used to populate hypothesis matrix.

Several considerations are made with regard to test data (the method by which data were collected is described in detail in a later section). In most cases, surgical robots are utilized with a closed loop control scheme – tool pose is always known. With segmentation algorithms, the robot kinematics shape prior is applicable only when both camera and tool

poses are known. However, camera pose is not always tracked. The data was collected with a pre-calibrated stereo camera, and can be broadly classified into two sets: MS Dataset – static camera moving scene, and 577 Dataset – moving camera static scene. MS Dataset was collected with a static camera but with dynamic tool motion, while 577 Dataset contains recordings of a static surgical scene from various viewpoints of a moving camera. Both datasets contain information of both the surgical tool and camera poses. However, by omitting camera pose information in each of the two datasets, an additional two test cases arise. This results in four total test sets, called A,B,C,D.

The four test conditions combined with the four variations of test parameters form the hypothesis matrix shown in Table 3.1, by which the two reconstruction and segmentation strategies, Strategy I and Strategy II, will be evaluated. Note that there is a parameter in

Table 3.1: Hypothesis Matrix

Dataset	MS		577	
Scenario	Static Cam Moving Scene		Moving Cam Static Scene	
Parameter \ Test ID	A	B	C	D
Known camera pose	○	⊗	○	⊗
Known tool pose	○	○	○	○
Static scene	⊗	⊗	○	○
Static camera	○	○	⊗	⊗
Ground truth	⊗	⊗	○	○

○ indicates that the parameter assumption is true for that test condition, while ⊗ indicates false. In general, the more true parameters, or ○, the less challenging the test condition.

the hypothesis matrix of whether the camera pose is known. In fact, an unknown camera pose can arise from the fact that not all endoscopes are machine tracked, instead, they are sometimes hand-held by a nurse or assistant during the surgical operation [108]. Camera pose can be unknown even with machine held cameras, for example when the endoscope holding system is separated from the surgical robot system. In such a case, the coordinate

transform between the two systems needs to be calibrated prior to the surgery [102]. Even so, over time there is inevitable accrued error, so camera pose accuracy decreases overtime. Moreover, in many applications, it is the camera pose with respect to specific region of interest on a particular organ that matters. In this case, as the human body is not static, obtaining the transformation matrix between the task and camera frame is a challenge.

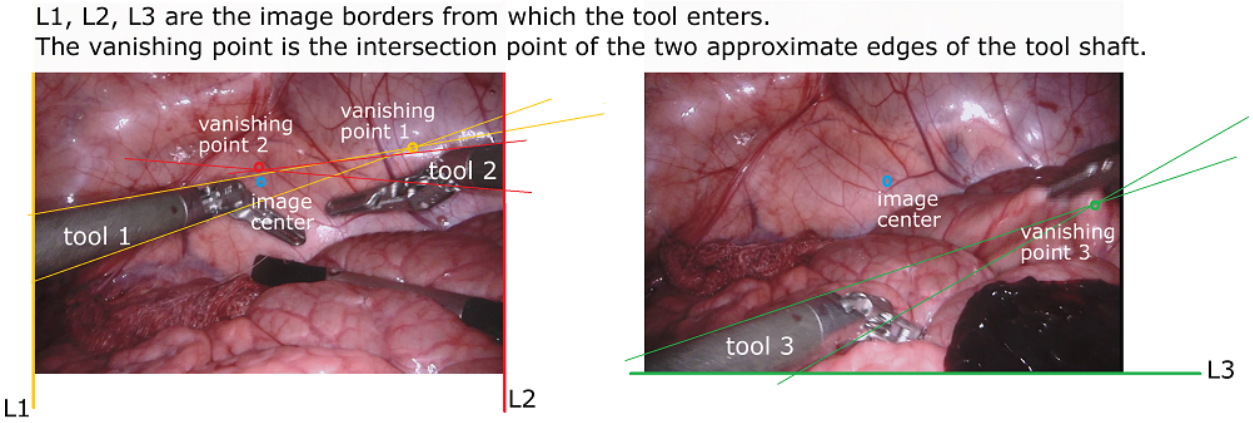


Figure 3.2: Vanishing point constraint for surgical tool segmentation.

3.2 Methods

The two strategies outlined in Figure 1.2, are tested and evaluated on the four test conditions, A-D, described in Table 3.1. The first strategy conducts segmentation prior to 3D reconstruction and the second strategy interchanges the order of these operations. The performance is evaluated against a ground truth 3D model captured via the Space Spider 3D White Light Scanner (0.1mm precision). The reconstruction and segmentation methods were carried out on the test conditions A-D. Note:

- SLAM and VO followed by triangulation are 3D reconstruction algorithms applicable to moving cameras with unknown trajectories. Therefore, for static camera test conditions (A,B) only SFM was implemented.
- In moving scene conditions (A,B), movement is isolated to the surgical tool, i.e. the background tissue is static.

- In Strategy I, tool segmentation is executed from 2D images, while Strategy II employs a 3D segmentation method.
- 3D reconstruction in Strategy I is localized to tool tip and adjacent tissue. In contrast, Strategy II involves reconstruction of the entire scene.

3.2.1 Strategy I: Segmentation → Reconstruction

2D Tool Segmentation

Known Camera Pose (A,C) This portion of Strategy I relies heavily of prior work [204] and is outlined briefly here. With camera extrinsic information, 3D surgical tool pose is projected back to the 2D image frame, forming robot kinematics shape prior u. A log-likelihood color mask Q is generated by a linear combination with heuristically tuned weights w_1, w_2, w_3, w_4 of hue H, saturation S in the HSV colorspace and O_1, O_2 in the Opponent colorspace, shown in Eq.3.1.

$$Q = w_1 H + w_2 S + w_3 O_1 + w_4 O_2 \tag{3.1}$$

where $O_1 = G - R$ and $O_2 = B - (G + R)$, with R, G, B being red, green and blue components in RGB colorspace. This color mask formation (which was also described in Section 2.1.3) provides the most discriminative power over separating surgical tool pixels from background tissue pixels [95]. Optimal translational alignment between u and Q is determined via DFT shape matching and minimizing the energy defined as the sum of pixel-wise multiplication of u and the shifted Q by (t_x, t_y):

$$\begin{aligned} \operatorname{argmin} E(t_x, t_y) &= \operatorname{argmin} \left(u \circledast Q(t_x, t_y)\right) && (3.2) \\ &= \operatorname{argmin} \mathbb{F}^{-1}(\mathbb{F}_Q \mathbb{F}_u^*) && (3.3) \end{aligned}$$

where $\circledast$ denotes spatial convolution and $\mathbb{F}$ is the Discrete Fourier Transform (DFT).

Unknown Camera Pose (B,D) Without camera extrinsic information, the robot kinematics shape prior u cannot be determined, and thus only the log-likelihood color mask Q was applied. To eliminate false positives, geometric constraints inherent to the specific surgical tool segmentation application were implemented. To name a few: (1) the tool always emerged from the image borders, so any island blobs were disregarded, (2) the tool shaft is cylindrical, so two near-parallel lines were found on either tool edge, (3) generally the tools orient away from the camera near the image center region, resulting in a vanishing point where the two lines intersect [29]. This vanishing point should reside on the image side of the border from which the tool emerges, as shown in Figure 3.2. Pure color filtering methods are hypothesized to produce less accurate segmentation results than those incorporating known camera pose.

Local 3D Reconstruction

In Strategy I, the 3D reconstruction step is performed after segmentation. This order of operations presents a few distinct advantages. Firstly, the segmentation result can serve as a prior indicator of the tool-tissue interaction region, reducing the 3D reconstruction region. Thus, only local 3D reconstruction around a bounding box centered at the tool location with width= 20 times the perceived surgical tool width is performed, drastically reducing the time complexity. Secondly, segmentation results can help enhance features of tool regions. The surgical tool shaft often affords few distinct features, resulting in sparse reprojection in 3D reconstruction. By first obtaining a segmentation result, pixels corresponding to surgical tool are known, and thus a feature-filled pattern can be padded on the tool region starting from the left edge of the tool shaft. The reconstructed model can then be enhanced, shown in Figure 3.3.

Three 3D reconstruction algorithms were considered in this work: SFM, SLAM and VO followed by feature points triangulation. Again, only SFM was used if the camera poses are known (A,C). These methods are described below.

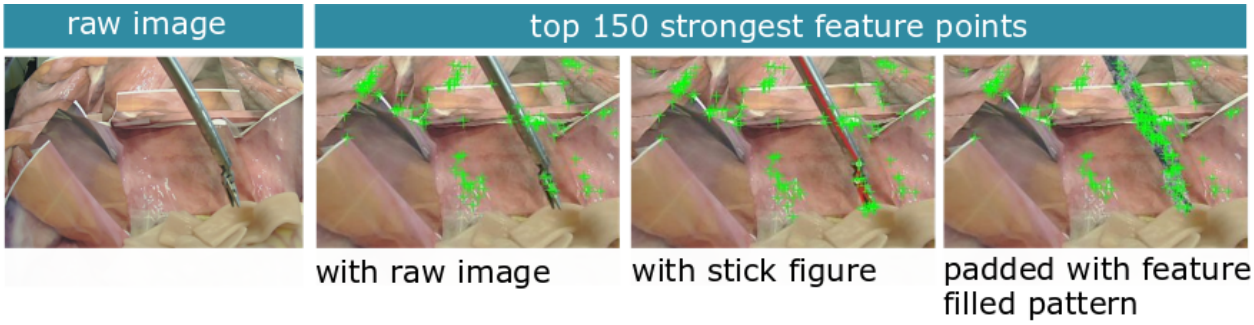


Figure 3.3: Feature points detected before and after tool region feature enhancement. The number of feature points increases at the tool tip when augmented to robot pose. The tool shaft contains even more features when padded with a feature filled pattern.

SFM Based Reconstruction SFM estimates 3D structure given a set of ordered 2D images. Without time constraints, SFM generates very accurate reconstruction. With the advancement of recent SFM technique, a reduction from $O(N^4)$ to $O(N)$ is achieved through bundle adjustments [229]. As shown in Figure 4.2, a 3D model was generated from two subsequent images. Then point cloud alignment merged the point clouds. The SFM algorithm can be realized in four steps - (1) feature point detection, (2) camera matrix estimation (3) relative camera motion derivation and (4) point cloud generation. In the case of known camera poses (A,C), camera extrinsic values directly replace the first 3 steps. Also, step (1) is performed twice with different *'MinQuality'* values. The first operation helps estimate camera pose changes while the second aids in point cloud generation.

SLAM Based Reconstruction SLAM operates in real-time with ordered sequence of images, and is relatively time-efficient compared to SFM. Several off-the-shelf SLAM frameworks were investigated prior to selecting the method for this work [74] [149] [115] [145] [111]. Because of its fast operation, broad open-source development community and wide range of supported imaging setups, ORB SLAM2 was implemented. The default ORB_SLAM2 Bag of Words (BoW) vocabulary for feature detection was not amenable to surgical scenes. A visual vocabulary relevant to surgical cavity imagery was generated via open-source DBoW2.

An identical vocabulary structure as ORB_SLAM2's default implementation was created and populated with images from the surgical workspace.

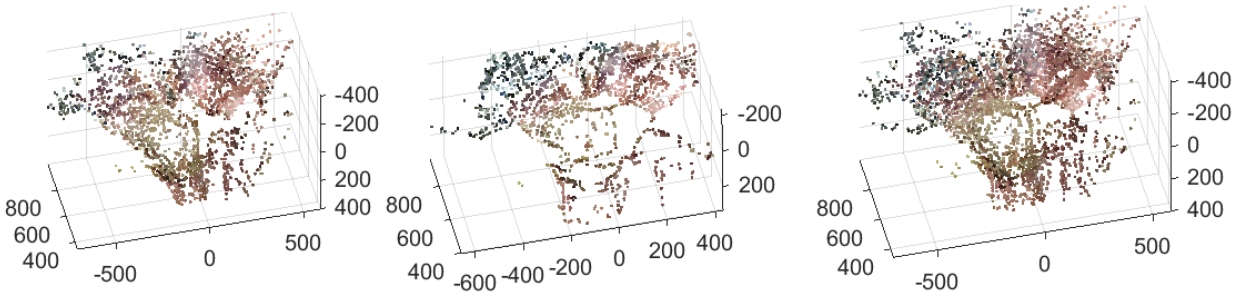


Figure 3.4: Point cloud alignment and stitching.

VO Based Reconstruction VO estimates the relative camera pose of two image frames from disparate viewpoints. Image points are then reprojected back into 3D space based on estimated camera trajectory. Finally, by feature point matching, the 3D model is generated.

3.2.2 Strategy II: Reconstruction → Segmentation

Global 3D Reconstruction

3D reconstruction for Strategy II was identical to that in Section 3.2.1 with two distinctions:

1. Lacking prior tool segmentation, 3D reconstruction was performed on the entire scene, increasing runtime.
2. Tool region feature enhancement cannot be performed without prior tool segmentation. Since the surgical tool is relatively thin with large depth discontinuities at tool edges, tool regions tend to reproject improperly.

Below are drawbacks of Strategy II, conducting 3D reconstruction prior to segmentation.

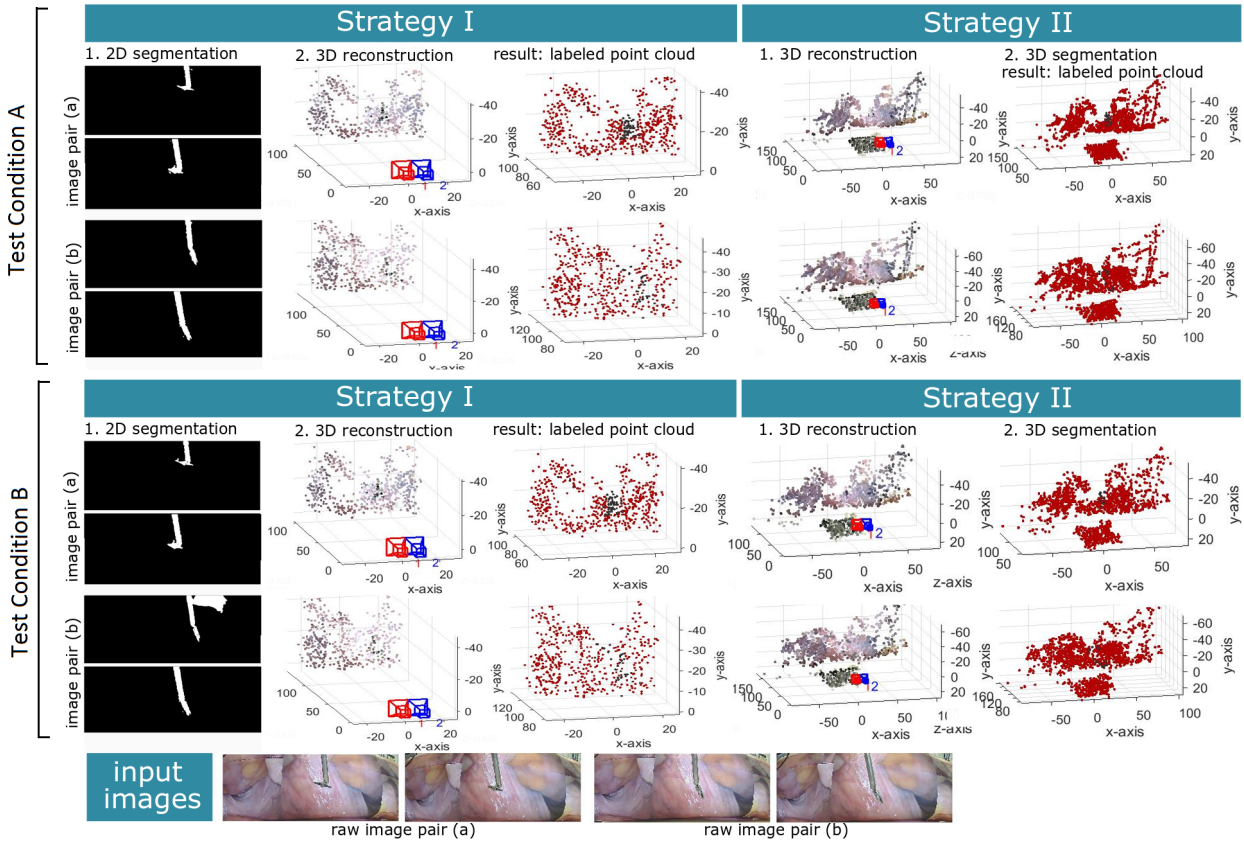


Figure 3.5: Results of Test Condition A: MS Dataset with Known Camera Pose (Top); Test Condition B: MS Dataset with Unknown Camera Pose (Bottom).

3D Segmentation

The 3D segmentation method follows closely to the 2D segmentation case in Section 3.2.1. The log likelihood color filtering scheme was directly applied to the 3D scene. With known camera pose (A,C), the 3D model was directly aligned with the world frame. Voxels corresponding to surgical tool volume were derived from robot kinematics and mapped to regions of the 3D model for segmentation. However, with unknown camera pose (B,D), color filtering was executed in tandem with a cylinder finding function, *pcfitcylinder*, which supports point cloud objects. This aided in detection of tool shaft of known physical size and shape.

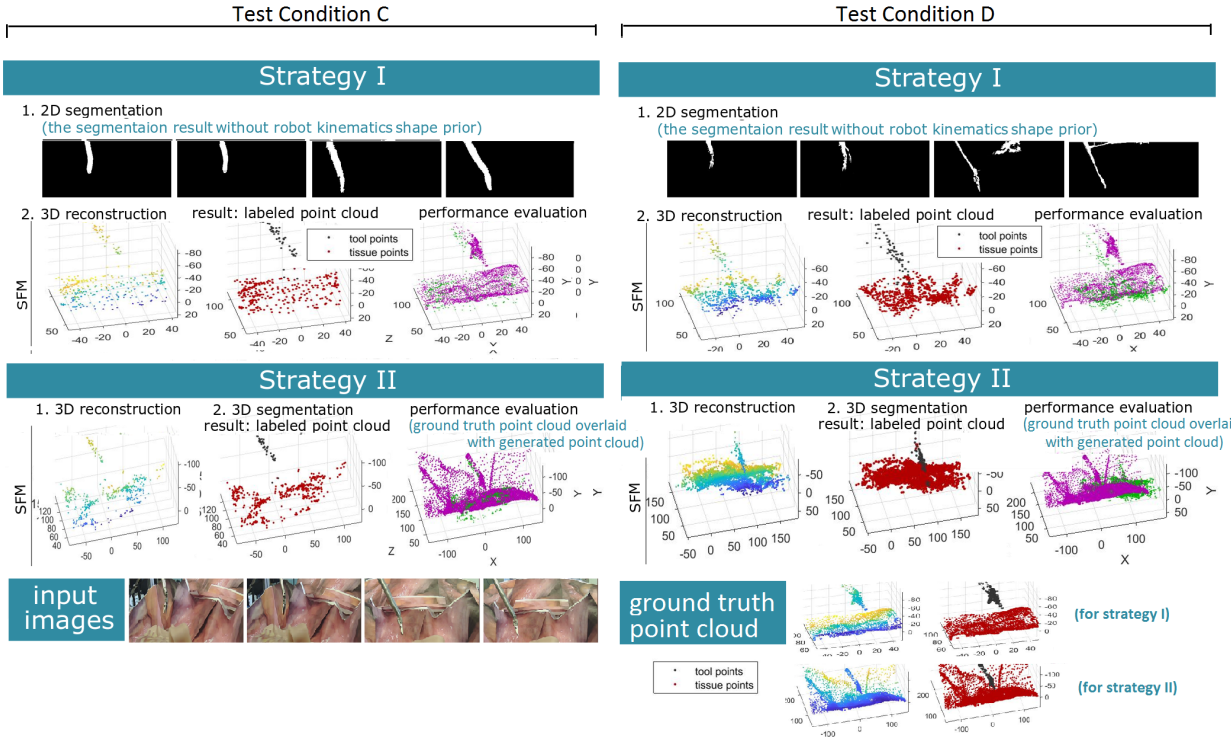


Figure 3.6: Results of Test Condition C: 577 Dataset with Known Camera Pose (Left); Test Condition D: 577 Dataset with Unknown Camera Pose (Right).

3.2.3 Data Collection

All data were collected from tissue phantoms with the Raven II surgical robot platform. The arrangement of tissue phantom were consistent within each dataset.

Test Data

Two data sets were collected for analysis: MS Dataset – involving static camera and moving scene/robot, and 577 Dataset – involving moving camera and static scene/robot.

MS Dataset were collected from static stereo cameras. A Raven II tool was continually actuated with random trajectories while maintaining end effector within both left and right image frames. The maximum translational and rotational speed of motion were 10 $^{cm}/_{s}$ and

30 $^{deg}/_{s}$ respectively, consistent with typical surgical operation speeds [228]. Image frames (640×480 pixels), kinematic information as well as camera poses were captured in real-time.

577 Dataset consists of stereo recordings of static Raven II tool and surgical scene. The recording procedure entailed two passes of stereo cameras orbiting 180 degrees about the tool interaction region at a radius between 10-20 cm at rough roughly 3 revolutions per minute, a reasonable trajectory for realistic constrained endoscope motion [9].

Ground Truth

The Space Spider, an in-house white light 3D scanner, was used to obtain ground truth 3D reconstruction of the surgical scene used in 577 Dataset. The 3D resolution of the scanner is 0.1mm with point accuracy at the 0.05mm scale. 11 scans were required to completely record the geometry of the surgical scene, which each scan containing about 700 image frames. Post processing was performed in Artec Studio, generating the 3D ground truth model.

3.2.4 Performance Analysis

Segmentation Accuracy

The Sørensen-Dice index measures the similarity between two sets, was used in this work to measure segmentation accuracy and is defined in Eq. 2.5. In this study, a small preprocessing step was required for 3D segmentation Sørensen-Dice index calculation: down sampling of generated result to match spatial resolution of ground truth data.

Reconstruction Accuracy

Two metrics were used to evaluate 3D reconstruction as shown in Eq.3.4 and Eq.3.5: root mean squared error (RMSE) and relative reprojection error (R_{Err}). Let X denote ground

truth point cloud, and Y the reconstructed point cloud. Then

$$\text{RMSE} = \frac{1}{M}\sum_{k=1}^{M}\sqrt[2]{\frac{\sum_{i=1}^{N_X}|X_i^k - Y_i^k|^2}{N_X}} \tag{3.4}$$

$$\text{R}_{\text{Err}} = \frac{1}{M}\sum_{k=1}^{M}\frac{|X^k - Y^k|_F}{|X^k|_F} \tag{3.5}$$

where $||_F$ is the Frobenius Norm, M is the number of frames, N_X is the cardinality of X, X_i^k is the i^{th} point in frame k. Furthermore, Y_i^k is the nearest (in the Euclidean sense) point to X_i^k after alignment.

3.3 Experimental Design

Reconstruction error and operation time were measured between Strategy I and Strategy II on test conditions A-D. Results help to characterize methodology choices in real-time 3D tool segmentation approaches given problem parameters.

3.3.1 MS Dataset

Table 3.1 outlines test conditions A and B, and the four relevant parameters therein. Both Strategy I and Strategy II were tested on MS Dataset via these test conditions. Recall that MS Dataset involves static cameras, and therefore only the modified SFM algorithm was used for 3D reconstruction. Furthermore, note that test conditions A and B are distinguished only by whether or not the camera pose is provided.

Known Camera Pose

This experiment corresponds to test condition A, as seen in Table 3.1. The top row of Figure 3.5 shows steps and results from Strategy I as well as results from Strategy II.

Unknown Camera Pose

This experiment corresponds to test condition B, as seen in Table 3.1. The bottom row of Figure 3.5 shows steps and results from Strategy I and results from Strategy II.

3.3.2 577 Dataset

Unlike MS Dataset, 577 Dataset includes video recordings of static surgical scene with moving cameras. Test conditions C,D correspond to 577 Dataset, and are distinguished only by whether or not camera pose is known. Both Strategy I and II were tested in both C and D. Since the cameras were moving, all SFM, SLAM, and VO were tested for condition D.

Known Camera Pose

This experiment corresponds to test condition C, as seen in Table 3.1. The left column of Figure 3.6 shows steps and results from both Strategy I and Strategy II.

Unknown Camera Pose

This experiment corresponds to test condition D, as seen in Table 3.1. The right column of Figure 3.6 shows steps and results from both Strategy I and Strategy II. The output 3D models from the three 3D reconstruction algorithms were generated, and results using SFM were shown for both Strategy I and Strategy II in Figure 3.6.

3.4 Results and Discussion

Table 3.2 shows 3D reconstruction RMSE and R_{Err} values and Sørensen-Dice indices QS for each combination of strategy and test condition A-D. MS Dataset lacks 3D ground truth data. Thus Sørensen-Dice index QS was computed from manually labeled 2D images for

Table 3.2: Performance Analysis

Dataset			MS		577	
Parameter \ Test ID			A	B	C	D
Known Cam Pose			○	⊗	○	⊗
Strategy I	QS	SFM	0.732	0.574	0.895	0.612
		SLAM	–	–	–	0.548
		VO	–	–	–	0.593
	RSME [mm]	SFM	–	–	6.628	8.398
		SLAM	–	–	–	10.78
		VO	–	–	–	11.22
	R_{Err}	SFM	–	–	0.070	0.115
		SLAM	–	–	–	0.186
		VO	–	–	–	0.125
Strategy II	QS	SFM	–	–	0.808	0.713
		SLAM	–	–	–	0.631
		VO	–	–	–	0.691
	RSME [mm]	SFM	–	–	11.02	11.25
		SLAM	–	–	–	13.98
		VO	–	–	–	19.03
	R_{Err}	SFM	–	–	0.086	0.109
		SLAM	–	–	–	0.157
		VO	–	–	–	0.162

QS denotes the average Sørensen-Dice index based on segmentation result and appropriate ground truth. RSME and R_{Err} are measures of 3D reconstruction accuracy.

Strategy I 2D segmentation. Strategy II Sørensen-Dice index, 3D RMSE and R_{Err} do not exist for MS Dataset experiments.

In terms of time complexity, both segmentation and 3D reconstruction were achieved with O(n) runtime. Bundle adjustments accelerated 3D reconstruction [229]. With known camera pose, 3D model alignment time is reduced. Furthermore, segmentation was more efficient because camera pose informed robot kinematics prior, as shown in Table 3.2.

In general, Strategy I reduced 3D reconstruction time by isolating the local region of interest. However, this amount of time saved clearly depends on the relative portion of the image frame that the surgical tool covers. It was also of interest to compare time complexity and density of resulting point clouds generated by the three 3D reconstruction algorithms, SFM, SLAM and VO in Strategy II for test condition D.

3.4.1 Strategy I

In Strategy I, 2D segmentation is executed prior to 3D reconstruction. As a result, only 3D reconstruction of the proximal region surrounding surgical tool is performed.

- Known Camera Pose: Surgical tool pose configuration was overlaid on the image to enhance feature point detection within surgical tool region.
- Unknown Camera Pose: Segmented areas were padded with feature filled pattern to enhance 3D tool reconstruction.
- False positives: Result contained more false positives, i.e. tissue falsely labeled as tools.
- Risk Assessment: While this method generally saves time, it is a high-risk-high-reward approach, particularly with unknown camera pose. Although local 3D reconstruction from the segmented tool region is time efficient, the result is erroneous if segmented tool region is inaccurate, which is likely without camera pose and thus kinematics prior.

3.4.2 Strategy II

In Strategy II, 3D reconstruction is executed prior to segmentation. Segmentation is performed with 3D data.

- Additional Segmentation Cues: Tool segmentation is informed by 3D geometry in addition to color filtering.
- Less Feature Points: Due to the tool's metallic material, thin geometric shape and large depth discontinuities between tool edge and background, less feature points are

available for matching during 3D reconstruction. Segmentation is required prior to tool feature enhancement.

- False Negatives: Results contained more false negatives, i.e. tool pixels falsely labeled as tissue pixels. False negatives can occur from either tissue colored reflections on the tool surface or imprecise reconstruction. It can be difficult to discriminate false negatives from real tissue occluding surgical tool.
- Time Efficiency: 3D reconstruction was performed on the entire image. This is computationally inefficient, particularly with known camera pose and distant surgical tool resulting in small region of interest.
- Risk Assessment: In general, this is a safer strategy; it guarantees 3D reconstruction of the area of interest.

3.4.3 Sources of Error

- Strategy I False Positives: False positives in this context are tissue falsely labeled as surgical tool. This can be attributed to the fact that following segmentation, applied feature enhancement encourages additional features for reprojection into 3D, even neighboring tissue. With unknown camera pose, there is even greater probability that feature enhanced non surgical tool points are segmented as tool points. With known camera pose, segmentation is more accurate. However, tool borders are slightly outstretched, and edge points may actually correspond to tissue.
- Strategy II False Negatives: False negatives in this context are tool regions falsely labeled as tissue. Since in this strategy, 3D reconstruction is done first, areas of the surgical tool shaft appear relatively featureless in 2D images. These areas are difficult to match and reproject to the 3D model. 2D points which are correctly segmented as tool are strong feature points on the surgical tool surface which generate highly accurate depth information by triangulation. On the other hand, tissue (most of the area on the images) is more likely to be falsely matched and reprojected to the tool

region. Since tissue might occlude the tool, it is algorithmically impractical to mark all tissue points within the tool region as either tool points or outliers.

3.4.4 Summary

Vision based force estimation requires an accurate 3D model of the real-time surgical scene with tool segmented from tissue. Two main strategies were proposed: either with tool segmentation followed by 3D reconstruction or vice versa. Given the authors' belief that the preferable strategy may change depending on the problem parameters, a hypothesis matrix was synthesize to study different parameters and conduct 3D reconstruction and segmentation using both strategies [91]. Results indicate that:

- Camera pose information saves time in both strategies under all problem assumptions, and furthermore increases 3D segmentation and reconstruction accuracy.
- 3D point clouds in the tool region were generally denser using Strategy I, since feature enhancement can occur prior to 3D reconstruction.
- Strategy I is more time efficient than Strategy II.
- With known camera pose, Strategy I tends to be comparable to or even better than Strategy II. With unknown camera pose, Strategy I becomes rather risky since local 3D reconstructions may become erroneous.
- The optimal strategy depends on problem parameters. With known camera pose, Strategy I is generally better considering runtime. With unknown camera pose, Strategy II is a slower but safer option to guarantee including the region of interest.
- The optimal strategy depends on tolerance to false positives and false negatives. Strategy I is preferable if false negatives are less tolerable, and Strategy II is suggested if false positives are less tolerable.

Chapter 4

MULTICAMERA DYNAMIC SURGICAL CAVITY 3D RECONSTRUCTION

Dynamic 3D reconstruction of surgical cavities is essential in a wide range of computer-assisted surgical intervention applications, including but not limited to surgical guidance, pre-operative image registration and vision-based force estimation. According to a survey on vision based 3D reconstruction for abdominal minimally invasive surgery (MIS) [123], real-time 3D reconstruction and tissue deformation recovery remain open challenges to researchers. The main challenges include specular reflections from the wet tissue surface and the highly dynamic nature of abdominal surgical scenes. To identify the flaws in existing 3D reconstruction algorithms in robot-assisted MIS, preliminary experiments using state-of-the-art 3D methods were performed to generate 3D surgical scene models from a stereo video sequence. While SFM provides the highest accuracy, it does not meet the real-time requirement, and SLAM is most suitable for online applications, but the 3D model resolution is not ideal. This study is later published in [207] and covered in **section 4.2**.

These results validated the need for a specialized 3D reconstruction framework for robot-assisted MIS. Moreover, [190, 211] emphasize the benefits of using multiple independently moving RGB cameras to generate an accurate measurement of tissue deformation at the volume of interest (VOI). To that end, a novel camera grouping and pair sequencing algorithm that handles multicamera 3D surgical scene reconstruction with known camera poses is proposed [205] and evaluated with the Raven-II [17] surgical robot system for tool navigation, the Medtronic Stealth Station s7 surgical navigation system for camera pose monitoring, and the Space Spider white light scanner to derive the ground truth 3D model.

.

4.1 Background

4.1.1 Motivation

3D reconstruction is an essential premise for numerous robot-assisted laparoscopic surgical applications, including vision-based force estimation, surgical guidance, and medical image registration in which the pre-operative data (computed tomography (CT) or magnetic resonance imaging (MRI) scan image slices) are overlaid on patient anatomy during surgery [123]. Although several research approaches are promising in this field, dense 3D reconstruction in real-time for minimally invasive surgery (MIS) remains an open challenge. Prominently, identifying critical and corresponding feature points from images with smoke, low contrast, specular reflections or homogeneous surfaces is a difficult task. The narrow view involved in laparoscopic imaging combined with the dynamic nature of the cavity, resulting from both surgical instrument and tissue motion, present additional obstacles for 3D reconstruction. Using monocular RGB cameras without active camera motion planning or additional sensory elements limits reconstruction to only planar surfaces or sparse, insufficient 3D models of the surgical scene.

4.1.2 Related Work

Video-based 3D reconstruction of surgical scenes can be classified based on scene type and use of camera motion [123]. Various SLAM methods are able to estimate 3D structures as well as camera motion depending on whether the scene is either rigid or deformable [13, 73, 120, 216]. For an active or frequently moving laparoscopic camera, structure-from-motion algorithms are suitable for generating 3D surface models. Methods which rely only on one camera and its motion must incorporate dynamic view expansion (DVE) to construct and expand dense 3D models due to the limited field-of-view (FOV) of laparoscopic cameras

[143,222]. While promising, this approach highly constrains scene motion to ensure accurate frame matching, which is not amenable to the highly dynamic nature of real world, online surgical scenes. Meanwhile, without the use of camera motion, reconstruction can be achieved using other visual cues, such as stereo vision [175, 195], active projection of spatial patterns [82, 166], and use of shading and shadows [122, 140]. To extract feature points, descriptor-based tracking methods like speeded up robust features (SURF) and scale-invariant feature transform (SIFT) are favorable for their high performance and fast computation, though descriptors and detectors specifically designed for surgical scenes show superior results [124].

Single moving camera surgical cavity reconstruction approaches have utilized either monocular [77] [46] [88] or stereo [121] [144] [131] vision sensors. With monocular cameras, the extracted 3D shape is represented by a linear combination of predefined basis shapes [23]. Spatial and temporal smoothness constraints were imposed in [157]. [231] [84], followed by relaxing of orthographic assumptions of the camera model. With stereo vision, [56] extended the factorization approach from [23], and [130] distinguished rigid and moving points based on a global Euclidean transformation check.

Multiple cameras allow for tracking of dynamic tissue shape changes with minimal camera repositioning. Such an approach for MIS was developed for which the cameras were mounted to a single insertable unit through a trocar to avoid multiple additional incision entries [190]. For this, the relative positions of the cameras were fixed. However, recent technical advances in magnetic cameras [236] [142], which can be inserted into the abdominal cavity and controlled by external magnets, can help overcome this limitation. In fact, high precision wireless control of magnetic cameras was achieved for single-incision laparoscopic surgery (SILS) [212] [59]. This technology can allow multiple independently moving cameras that simultaneously record the surgical cavity from multiple viewpoints.

COSLAM achieves visual reconstruction using multiple independent cameras in dynamic environments [241] and serves as the primary inspiration of the novel multicamera surgical scene 3D reconstruction framework [205] presented in this chapter.

4.2 Preliminary Study

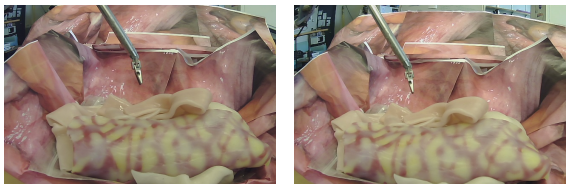


Figure 4.1: Sample stereo image pair of phantom surgical cavity environment. Images were acquired at 30 HZ and streamed various viewpoints of the surgical scene.

In this work, three different methods of 3D reconstruction from stereo laparoscopic image streams including simultaneous localization and mapping (SLAM), visual odometry (VO), and structure from motion (SFM) are compared. As shown in Figure 4.1, a novel dataset was generated using a pre-calibrated stereo camera viewing a realisitc phantom surgical cavity with Raven II [17] surgical end effector, and a ground truth model was acquired via high fidelity 3D scanning.

4.2.1 Preliminary Experiment Design

In this work, a new stereo video dataset of a surgical scene with surgical robot was collected, and a ground truth model was derived using a high precision 3D White Light scanner. Three reconstruction approaches, SLAM, VO and SFM methods, were then compared quantitatively via performance of 3D reconstruction of the synthetic surgical scene against the ground truth model. Reconstruction error and point density of the different methods were of particular interest, with motivations to determine the best technical direction for online pre-operative scan registration and real-time vision-based force estimation.

4.2.2 SLAM

In general, SLAM fuses quantitative data from sensors navigating a previously unknown, primarily static environment. The environment is incrementally mapped and the sensors localized within. Several versions have been successfully implemented in real-time on an ordered sequence of images. Of the actively researched open source SLAM algorithms, oriented features from accelerated segement test (FAST) and rotated binary robust independent ele-

mentary features (BRIEF) SLAM2 (ORB SLAM2) [145] is of particular interest. This is due to ORB SLAM2's notable accuracy, fast operation due to minimal computation, and range of supported imaging setups. Modifications to the ORB SLAM2 method were implemented to better suit the reconstruction task.

MIS Vocabulary Generation: ORB SLAM2 operates with FAST keypoints [178] and BRIEF descriptors. However, ORB's default Bag of Words (BoW) vocabulary for feature detection is ill-suited for surgical scenes found in MIS. With that said, I believe a feature detector and descriptor specific to surgical scenes may exhibit improved robustness for the application of interest in this work. For ORB SLAM2 to benefit from loop closure and trajectory refinement, a visual vocabulary characteristic of the surgical cavity imagery was required. This was achieved using the open-source DBoW2 project [70]. A vocabulary structure identical to ORB SLAM2 in terms of levels and branches was creating using numerous images from the surgical scene to build the vocabulary. In this way, the computational efficiency benefits that comes from the structure of ORB SLAM2 were preserved while features were modified to be specific to MIS tasks within a surgical cavity.

Algorithm 1 SLAM 3D Reconstruction from Trajectory

for each consecutive image pair I^i, I^{i+1} **do**
    extract feature points
        MIS BoW
        FAST keypoints
        extract BRIEF descriptors
    match feature points
        BRIEF descriptors
    load camera pose
        ORB SLAM2 localization
    save attributes:
        camera pose
        image points
        inlier matches between I^i and I^{i+1}
    match feature points
        across aggregate views processed
    reproject matched points
        triangulate from multiple views
    refine reprojection
        bundle adjustment
end for

Reconstruction: While SLAM simultaneously tracks localization and environment reconstruction, the reconstruction procedure in ORB SLAM2 was modified for the presented MIS task. The general method is described in Algorithm 1.

Note that feature matching was performed twice per successive frame pair. The first instance discovers matched points using BRIEF descriptors from a very short time scale – consecutive images only. In contrast, the second is used to track feature points in aggregate over a longer period of time. In this implementation, it was observed that performance increased with downsampling of image frames; this created starker viewpoint changes between successive frames. Additionally, runtime was decreased by this downsampling.

4.2.3 Visual Odometry

VO utilizes sequential images to estimate incremental relative camera poses. After computation, incremental transformation matrices between subsequent time instances can be obtained. This can then be accumulated to recover the trajectory of the camera throughout the video stream. With an estimated camera trajectory, 2D corresponding feature points from subsequent image frames can be reprojected into 3D space through triangulation. In this way a 3D model of the scene can be created.

VO methods can be implemented with either monocular or stereo camera setups. However, a major drawback of using monocular visual odometry for scene reconstruction is that scale is unknown. As previously described, the video dataset in this work was captured using a stereo camera, which provides both depth and scale. VO algorithms can also suffer from drift error; recursively multiplying estimated incremental camera transformation matrices together may lead to accumulation error, as the estimated trajectory drifts further from the true trajectory. An effective solution to this drift error is to provide easy-to-detect feature points with known 3D locations, used to occasionally correct for camera pose error. The VO implementation used in this work is outlined in Algorithm 2.

In line 15 of Algorithm 2, inlier point detection is required. This procedure was built under the assumption of a rigid scene, and thus 3D points should be positioned similarly relative to one another between subsequent frames. In other words, the distance between any two features points in P_t should match the distance between the corresponding points in P_{t+1}. Suppose $d(i, j)$ computes the Euclidean distance between enumerated feature points i, j. This is enforced with consistency matrix $A(t)$ for consecutive frames at indices $t, t+1$, generated such that $A(t)_{ij} = 0$ by default and:

$$A(t)_{ij} = 1$$

if $d(i, j)$ identical between P_t and P_{t+1}.

The goal is then to select a large subset of feature points whose consistency score is higher with one another. This can also be viewed as solving the Maximum Clique Problem with $A(t)$ as the adjacency matrix [18]. From the generated camera trajectory, the surgical scene can be reconstructed similarly to the method described in **section 4.2.2**.

Algorithm 2 Visual Odometry for MIS

1: **denote** L_i, R_i as the i^{th} frame in the left and right cameras respectively
2: **for** consecutive stereo pair L_i, R_i and L_{i+1}, R_{i+1} **do**
3: **preprocess** using camera calibration parameters
4: undistort
5: rectify
6: **compute** disparity maps
7: $L_i, R_i \rightarrow D_i$
8: $L_{i+1}, R_{i+1} \rightarrow D_{i+1}$
9: **load** camera pose
10: ORB SLAM2 localization
11: **extract** and match FAST feature points
12: **calculate** 3D feature point position from D_i, D_{i+1}
13: from D_i, D_{i+1}
14: generates point clouds P_i, P_{i+1}
15: **determine** inlier points between P_i, P_{i+1}
16: **estimate** transformation matrix between inlier points
17: rotation $\mathcal{R}_i$
18: translation $\mathcal{T}_i$
19: **derive** trajectory via recursive multiplication
20: **end for**

4.2.4 Structure from Motion

SFM estimates 3D structure of a scene from a set of 2D images and is capable of working with large, unordered sets of images [73]. Without processing time constraints, SFM often exhibits excellent results. With images captured from a monocular camera, 3D structure and camera motion can be recovered only up to scale.

Computing the actual scale in world units requires other information including size of an object in the scene and information from additional sensor, e.g. an odometer.

This method is suitable for extension to 3D reconstruction with multiple cameras or viewpoints; so long as the FOV captured in the image pairs are not completely disjoint, no additional information about relative camera pose is required for reconstruction. Scale is therefore not an issue with the stereo dataset. There are two primary potential utilizations of SFM in this work:

- reconstruction from entire sequence;
- point cloud generation between subsequent frames, followed by alignment and stitching to form the aggregate structure.

Algorithm 3 SFM Procedure For Consecutive Image Pairs

1: **for** each consecutive image pair I^i, I^{i+1} **do**
2: **extract** and match feature points
3: detect corners in I^i
4: match corners in I^{i+1}
5: **estimate** camera matrix
6: camera essential matrix
7: **find** epipolar inlier points
8: matched points are along a horizontal line
9: **compute** camera motion
10: **extract** and match feature points
11: match dense set of points between I^i, I^{i+1}
12: smaller similarity threshold
13: **generate** pointcloud
14: 3D triangulation of matched points
15: **end for**

In the introduced dataset, overall camera view changes for the entire time span resulted in difficulty extracting corresponding feature points consistently. A shorter time scale was

thus more amenable, and the latter technique was pursued. This begins with the process outlined in Algorithm 3, and results in a point cloud for each pair of consecutive images. Each of these point clouds were then aligned and combined to the current aggregate point cloud. This stitching procedure is visualized in Figure 4.2.

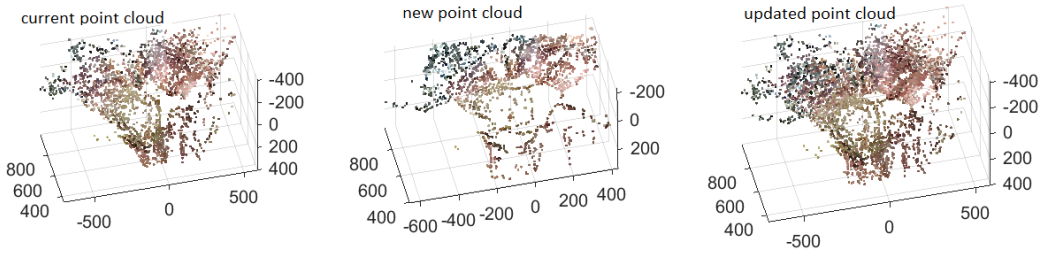


Figure 4.2: Align and stitch the current aggregate point cloud with new point cloud.

Point Cloud Stitching: The SFM generated point clouds are merged both to extend the overall FOV as well as to create a dense aggregate reconstruction. The process is completed using the Iterative Closest Point (ICP) algorithm. The main components of this process are outlined as:

1. *Downsample Point Clouds*: Downsampling of both the generated point cloud and the ground truth point cloud helps to filter noise and improve consistency. The quality of point cloud registration depends heavily on data noise, and downsampling with a box grid filter is an effective way to ameliorate this [83]. The size of the box grid filter can be carefully designed such that the point clouds, after downsampling, have a similar and consistent number of points.

   In this work, the grid size, g was selected as $g = 0.1$. Downsampling was then achieved using the grid average downsampling technique:

$$\begin{aligned} s &= \frac{|P_i|}{|P_{i+1}|} \\ \text{, where } P_i' &= \mathbb{DS}(P_i, g), \text{ and } P_{i+1}' = \mathbb{DS}(P_{i+1}, gs); \end{aligned}$$

Note that $|\cdot|$ denotes number of points in the point cloud, s is a scaling factor, P' is the downsampled version of point cloud P, and $\mathbb{DS}$ is the downsampling function that takes a point cloud as its first argument and grid step size as its second.

2. *Find Transformation and Align*: Determine the transformation that aligns the second point cloud with the first point cloud via ICP, and transform to the reference coordinate system defined by the first point cloud.

3. *Merge the Point Clouds*: Merge point clouds with a specified merge size, as determined by user-defined processing time and resulting resolution requirements.

4. *Repeat Steps Above*: Repeat the steps above taking one new point cloud at a time and merging all the point clouds together.

The resultant aggregate point cloud from merging the individual point clouds is the reconstruction result using SFM.

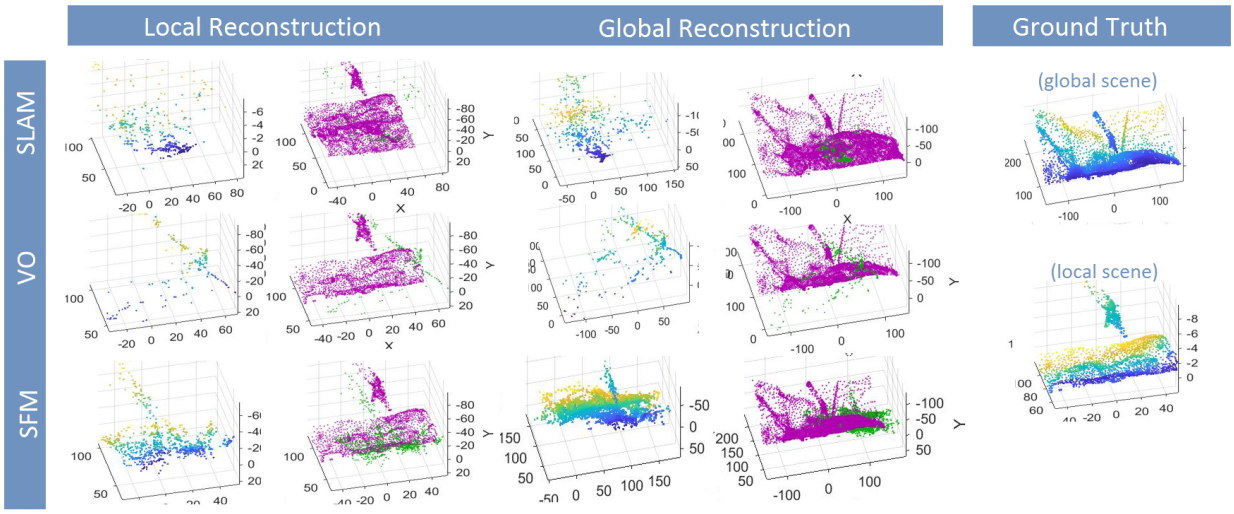


Figure 4.3: Point clouds generated using the three methods of interest and the ground truth.

4.2.5 Preliminary Experiment Results

The three methods of 3D reconstruction described above were performed on the collected surgical scene dataset, and the results are presented in the following section. Recall that reconstruction error against the ground truth model and point density were the quantitative measures of interest. In all three cases, an estimated camera trajectory is used as part of the reconstruction. Ideally, the three methods should produce very similar camera motion estimates from the same dataset. The generated trajectories shown in Figure 4.4 are relatively consistent, indicating that differences in trajectory estimation play only a negligible role in reconstruction. Qualitative visual surface reconstruction results from each of the three methods are shown below in Figure 4.5.

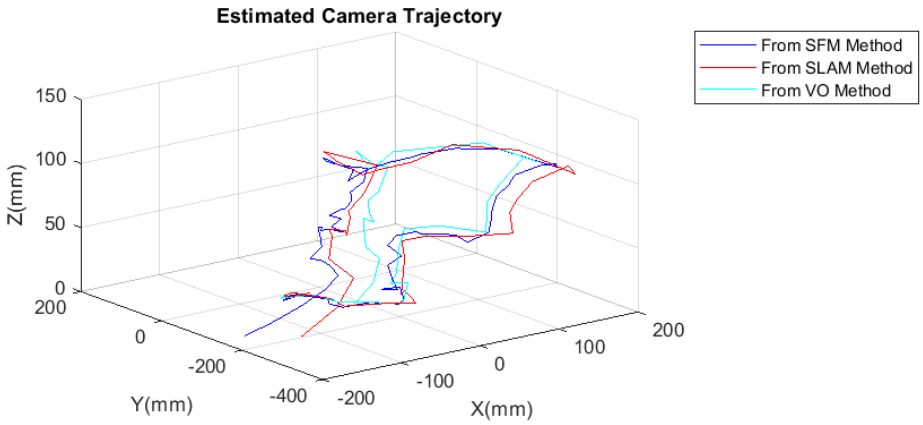


Figure 4.4: The camera trajectory estimated using the three algorithms.

Note that in Figure 4.5 all reconstructed points are merely interpolated; in this visualization, high frequency noise and outliers are emphasized through interpolation. The generated surface does not necessarily reflect either the density or accuracy of generated point clouds. The generated reconstruction results visualized as point clouds are shown in Figure 4.3, and give a better indication of point density and general shape of reconstruction as compared to interpolated results in Figure 4.5. Magenta ground truth point clouds are overlaid to better depict reconstruction error.

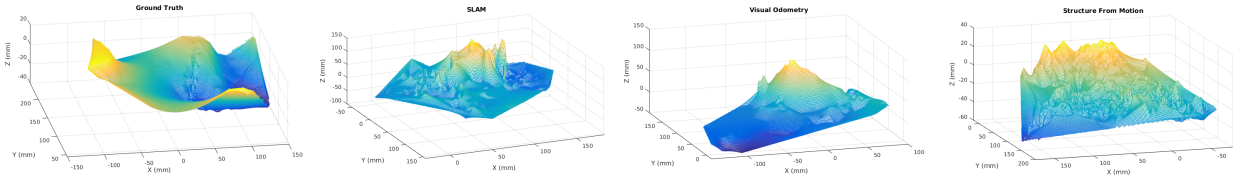


Figure 4.5: The comparison of ground truth point cloud and point cloud with the three tested surface reconstructing algorithms. Surface reconstruction is performed using a subset of the points generated in the point cloud maps and using the interpolation method based on Voronoi tessellation.

To evaluate 3D reconstruction, the root mean squared error (RMSE) of reconstructed points clouds from each algorithm was calculated relative to the ground truth point cloud. Supposing that an algorithm generates reconstructed point cloud P_A, it is compared to the ground truth point cloud P_{GT}. Then the RMSE is based on the Euclidean distance between each point in P_A and the closest point in P_{GT}, specifically

$$\text{RMSE} = \sqrt[2]{\frac{\sum_{i=1}^{N} |P_A(i) - P_{GT}(i)|^2}{N}} \tag{4.1}$$

where $P_A(i)$ indicates the i^{th} point in pointcloud P_A, $P_{GT}(i)$ the closest ground truth point to $P_A(i)$, and $N = |P_A|$, the number of points in P_A. Table 4.1 shows the quantitative surgical scene reconstruction results from the three algorithms.

Table 4.1: Point Cloud Density and Accuracy

	N	**RMSE[mm]**
SLAM	1061	13.9759
VO	444	19.0315
SFM	1312	11.2547

N denotes the number of points in the point cloud.

Based on the results N in Table 4.1, the SFM algorithm was able to generate the most structure from the dataset, whereas the VO point cloud was the sparsest, and thus interpolation results are smoothest, as observed in Figure 4.5. Note that a denser point cloud

is preferred since it both provides more information and through a simple down-sampling process can be transformed into a sparse point cloud if needed. Beyond density of reconstruction, the accuracy of the generated surface is critical to MIS. With regard to the RMSE, SFM resulted in the most accurate reconstructed surgical scene point cloud. These results can be visualized in Figure 4.3. With life-critical telerobotic tasks like RMIS, an emphasis on accuracy of spatial information over quantity is logical and safer.

The algorithms here were implemented on the same machine without hardware optimization. Qualitatively, SFM was the least optimal. With future goals of online implementation, the SFM method's remarkably prolonged computation time make it an unlikely candidate moving forward. Although VO and SLAM both appeared promising in terms of runtime, the SLAM method is preferred with the far denser and more accurate point clouds.

While the SLAM algorithm is encouraging, drawbacks do exist. The SLAM generated point cloud is dense yet diverged. Auxiliary methods such as additional Kalman filtering or particle filtering may remove these outlier points over longer periods of time. In general, when a feature point is repeatably undetectable over a length of time it is removed. This approach is inspired by promising work presented by Zou et al. [241].

4.2.6 Preliminary Study Contribution

To the best of the authors' knowledge, this work is the first to: (1) introduce the stereo-camera video stream of the realisitc phantom surgical cavity dataset with ground truth captured from high fidelity 3D scanning; (2) implement SLAM, VO and SFM 3D dense reconstruction algorithms on the stereo laparoscopic dataset; and (3) comparatively analyze and quantify the performance of said approaches. All in all, this preliminary study provides insight into methods towards real-time surgical scene 3D reconstruction methods for laparoscopic robot-assisted MIS procedures, and lays the groundwork for future work in other areas of medical robotics research, e.g. vision-based force estimation with multiple independently moving cameras with known camera poses (see **section 4.3**).

4.3 Methods

4.3.1 Fundamental Concepts

In MIS, it is undesirable to constantly move the camera around while the surgeon is performing the operation. In fact it can be very distracting if the camera motion is frequent. However, all of the proposed methods require camera motion to get a dense 3D reconstructed scene. To compensate for this, a worthwhile approach may be to reconstruct the surgical scene from multiple cameras from different viewpoints. It is shown that the multiple viewpoint autostereoscopic display (AD) technology prevents the surgeon from losing the 3D perception of the scene when a camera is repositioned [211]. By doing this, the cameras can all stay relatively still but provide multiple view points at every time instance. In fact, this approach is even more powerful in dynamic scenes which is very common considering the respiration and heart beat of the patient. This method also does not necessarily require more incision ports during minimal invasive surgery as the additional cameras can be attached to the inside of the abdomen and provide multiple views once the abdomen is insufflated.

Such a multi-cam device tailored for minimally invasive surgeries (MIS) has already been invented [190]. In that design, the cameras were all mounted on one insertable unit and inserted through a trocar in order to avoid the need to create multiple incision entries. The drawback is that the relative positions of the cameras are fixed.

However, with recent technical advances in magnetic fields, magnetic cameras [236] [142] which can be inserted into the abdominal cavity and controlled by external magnets on the outside of the abdominal wall are developed. Also, the magnetic camera steering can be achieved with high precision with wireless control and thus, single-incision laparoscopic surgery (SILS) can be demonstrated [212]. Moreover, [59] shows a magnetic levitation camera (MLC) design which can reduce the invasiveness to the patients and improve the camera motion dexterity. Potentially, this will allow multiple independently moving cameras that simultaneously record the surgical cavity from different viewpoints.

With this idea in mind, and a conclusion drawn from the preliminary study that SLAM is the 3D reconstruction algorithm most feasible to real-time applications compared with VO and SFM. A novel surgical scene 3D reconstruction framework using multiple independently moving camera with known poses has been developed based on the existing COSLAM algorithm [241]. There are four key points where our implementation differs from COSLAM:

- In COSLAM, the camera poses are assumed to be unknown, however, in our implementation, both the surgical tool is held by a robot arm and the cameras' position and orientation are tracked. So, the pose of the cameras is a known parameter instead of the algorithm output. Theoretically, with additional known information we have, the precision of the 3D reconstructed scene could be improved.
- In the COSLAM algorithm, the image feature points from different cameras are matched. The feature points are then classified into static and dynamic (moving) points depending on the relative locations of the matched feature points across different camera viewpoints. The dynamic points are being referred to a moving object or noise, and that only the static feature points contribute to 3D reconstruction of the scene. In our implementation, not all the objects are rigid, so a third class of feature points - the deformed points - should be created.
- The COSLAM algorithm is tested on a larger scaled environment, such as a room, whereas our implementation will be targeted toward a scene of a surgical cavity whose size is much smaller.
- The camera motion in the original COSLAM as well as other SLAM algorithms tend to be close to a linear trajectory, but in our implementation, the cameras are all moving roughly around an orbit centered at the surgical cavity. So, the camera motion patterns are different, which may or may not affect the performance. So, I intend to look into this problem and do some further investigations.
- The compilation of the COSLAM algorithm is not trivial due to its dependencies on many 3rd party libraries and version restrictions. So my aim is to minimize the dependencies one will need to compile my implementation of the algorithm.

As with in [241], feature matching in our proposed surgical scene 3D reconstruction framework is conducted across both time and space. Camera matching across time, or intracamera matching, matches feature points among images from the same camera at different time instances. Camera matching across space, intercamera matching, matches feature points from concurrent images from different cameras. In this method, cameras are grouped together if fields of view (FOVs) overlap by a predefined threshold. Cameras can exist in multiple groups, and features are matched only between cameras within the same camera group. This is conveyed in Figure 4.6. This example features seven cameras classified into two camera groups. Each cameras undergoes intracamera matching, but only cameras within the same group undergo itercamera matching.

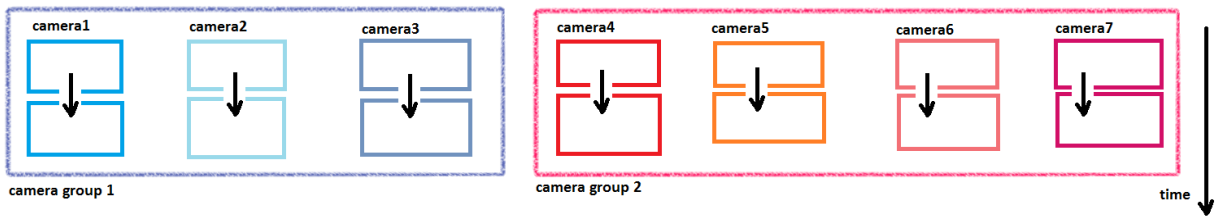


Figure 4.6: Camera grouping and intra/inter camera matching.

While there are many image pairs that can be selected, not all camera pair selections will produce good matching results. This work proposes a carefully designed strategy for image pair selection to increase time efficiency and reduce outliers.

4.3.2 Overall Workflow

Since acquired camera images are assumed 2D, 3D reconstruction requires at least two images for comparison and triangulation. These two images can be selected either from the same camera at subsequent time instances or concurrently from two different cameras. In this work, a basic assumption that some 3D points may be generated from an exhaustive search of all possible image pairs. The static environment case is straightforward. With ideal camera calibrations, computational cost is not a constraint, and with ideal camera triangulation

error in reconstruction arises only from feature matching. Finally, assuming a Gaussian model for outlier noises, an average 3D pointcloud of the scene can be generated. However, practical challenges for MIS exist. Time efficiency is a requirement for online applications, so determining the minimum number of image pairs to derive a 3D model is critical. Moreover, in dynamic environments, image pairs from different time instances may be erroneous. The reliability of the generated 3D information needs to be prioritized in this online, dynamic scenario. Figure 4.7 illustrates the four sub-tasks within the overall proposed approach, detailed in the following subsections.

4.3.3 VOI Coverage Check

Only cameras with views of the VOI should be considered for grouping and subsequent reconstruction. Some basic assumptions about the cameras allow for a geometric approach to the problem.

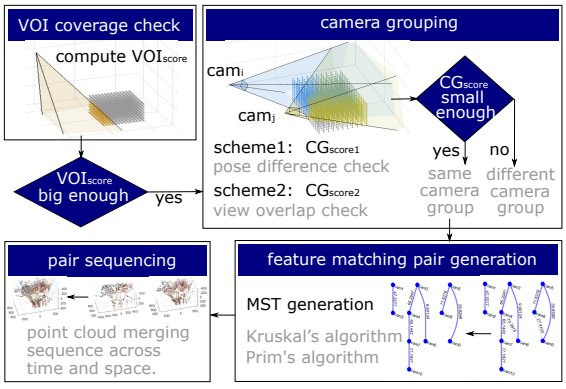


Figure 4.7: The workflow for camera grouping and pair sequencing in multiple camera 3D reconstruction of dynamic surgical cavities.

Suppose each camera has a rectilinear lens with perspective center at the center of its entrance pupil [98]. A pinhole model then applies, as illustrated in Figure 4.8. The camera angle of view (AOV), used interchangeably with FOV [60], describes the angular extent of a given scene that is imaged. The directional components of horizontal, vertical, and diagonal AOV satisfy the relation:

$$\tan\left(\frac{\alpha_i}{2}\right) = \frac{i}{2 \cdot S_2} \tag{4.2}$$

where i can take on the designations of h, v, d, the horizontal, vertical or diagonal specification of the image. S_2 is the distance from camera center to image plane.

The following are necessary and sufficient conditions for a 3D point, p, to appear within the AOV. The 3D point:

- remains within the horizontal AOV ($\omega_h < \alpha_h$).
- remains within the vertical AOV ($\omega_v < \alpha_v$).
- is the closest point to the camera along the ray cast from camera center to the point.

With these assumptions, a VOI coverage check for each camera ensures that all grouped cameras sufficiently view the VOI. Cameras that fail the check will not be taken into consideration until relocated. First the predefined VOI is discretized into a set of 3D points. The 3D points can be distributed uniformly in space, or in a weighted distribution to emphasize particular regions of the VOI such as the tool-tissue contact point.

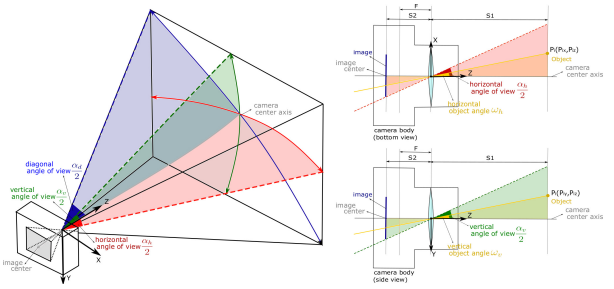


Figure 4.8: The horizontal, vertical and diagonal AOVs shown in red, green and blue respectively. S_1 is the object distance from camera, S_2 the distance from camera lens to image plane and F is the camera focus. $S_2 = F$ is required for sharp projection of P_i.

Each camera is then scored for VOI coverage, denoted $\text{VOI}_{\text{score}}$. First, let $\vec{i}, \vec{j}, \vec{k}$ represent unit vectors in the camera cartesian frame. Let P be the set of all sampled points of the VOI. Iterating through each point $P_q \in P$, compute the normalized projections onto i, j

$$\sin(\omega_v) = \frac{|\vec{P_q} \cdot \vec{j}|}{|\vec{P_q}|} = \frac{|P_{qy}|}{|\vec{P_q}|} \tag{4.3}$$

$$\sin(\omega_h) = \frac{|\vec{P_q} \cdot \vec{i}|}{|\vec{P_q}|} = \frac{|P_{qx}|}{|\vec{P_q}|} \tag{4.4}$$

If the point is within the AOV, coverage is increased. In other words, if $\left(\sin(\omega_h) < \sin\left(\frac{\alpha_h}{2}\right)\right)$ and $\left(\sin(\omega_v) < \sin\left(\frac{\alpha_v}{2}\right)\right)$, then increment

$$\text{VOI}_{\text{score}} = \text{VOI}_{\text{score}} + \frac{1}{q+1}\left(1 - \text{VOI}_{\text{score}}\right)$$

where $P_q = (P_{qx}, P_{qy}, P_{qz})$ is the q^{th} interated point in P, represented in camera frame C with positive depth P_{qz}. Further, ω_h and ω_v are the horizontal and vertical angles between $\vec{k}$ and the ray cast from camera center to P_q. Each camera's $\text{VOI}_{\text{score}}$ represents the VOI coverage and is valued between 0 and 1, 1 representing full coverage. Figure 4.10 depicts several $\text{VOI}_{\text{score}}$ values for different configurations, i.e. various geometries and sample point distributions for the VOI. A simple predetermined $\text{VOI}_{\text{score}}$ threshold informs a binary classification distinguishing eligible cameras with enough visibility of VOI from cameras that do not view critical features of interest. As an example, consider a set of ten cameras within a surgical cavity. These cameras provide various viewpoints within the cavity, and their camera poses are known apriori. This scenario is depicted in Figure 4.9.

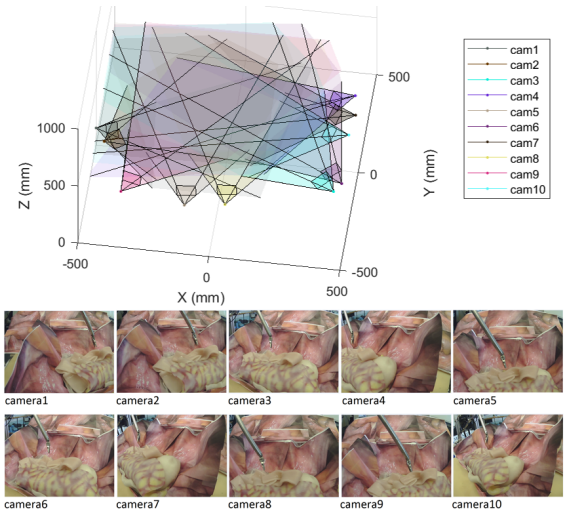


Figure 4.9: This is a set of 10 images from different camera viewpoints and the visualization of the camera poses.

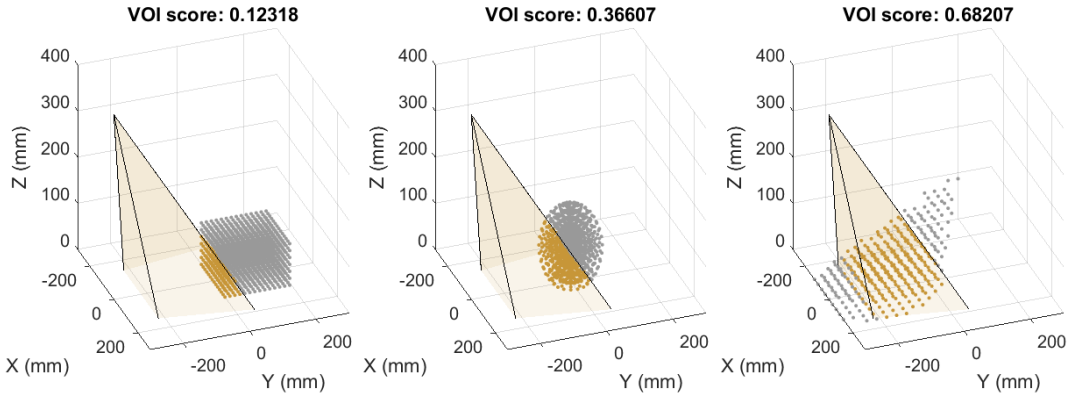


Figure 4.10: The $\text{VOI}_{\text{score}}$ values for three different VOI sample point distributions. Sample points from left to right: (1) uniformly distributed in a cube (2) uniformly distributed in the spherical coordinate system (azimuth, elevation, radii), points appear denser near the center in Cartesian space; (3) distributed along a cone shape similar to the tool range of motion under the Remote Center of Motion (RCM) constraint.

The $\text{VOI}_{\text{score}}$ can then be calculated for each camera given a known workspace geometry and VOI sampling distribution. For this, a cubic workspace and uniform sampling is chosen, and $\text{VOI}_{\text{score}}$ values are determined as shown in Figure 4.11.

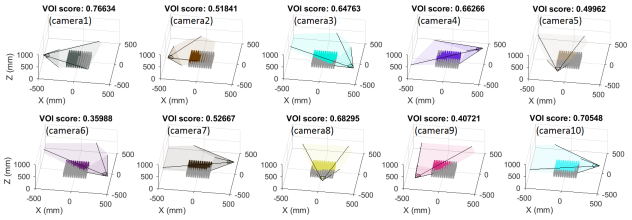


Figure 4.11: $\text{VOI}_{\text{score}}$ values for all 10 cameras from Figure 4.9. VOI defined as a cubic workspace spanning $X, Y = [-150, 150]$ and $Z = [0, 150]$ and with uniform sampling.

4.3.4 Camera Grouping

Of the cameras with sufficient $\text{VOI}_{\text{score}}$, camera groupings for optimal feature matching must be formed. This is achieved through a graph-based approach and another metric, CG_{score}. A fully connected graph is constructed with each vertex a camera and each edge weighted by a CG_{score}. Edges with weights greater than a CG_{score} threshold are broken. Subsequently, initial non-overlapping camera groups are formed as the remaining isolated sub-graphs. These are divided into mutually overlapping camera groups by dividing into the largest complete subgraphs. A critical component of this procedure is the calculation of CG_{score}. Two proposed methods for described, based on (1) pose difference and (2) view overlap.

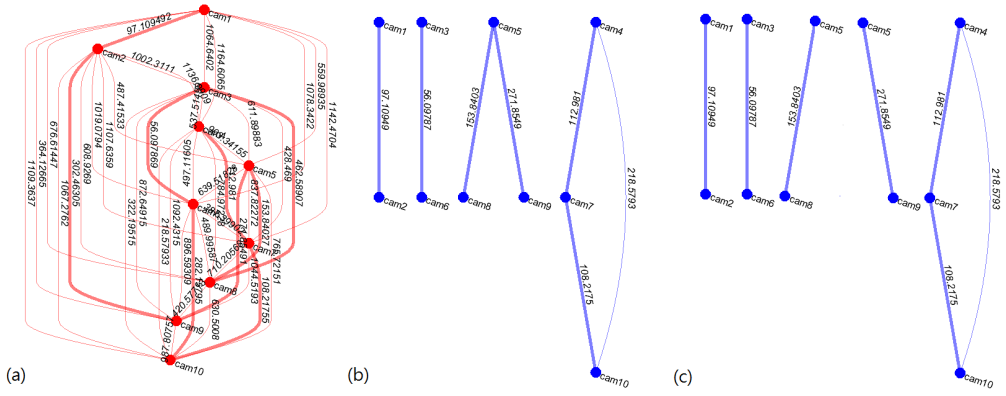


Figure 4.12: (a) $\text{CG}_{\text{score1}}$ of each camera pair (b) and (c) are the non-overlapping and overlapping camera grouping result.

$\mathbf{CG_{score1}}$ (Pose Difference): This score simply compares the relative configuration between two cameras. Let $\vec{t}_{ij}$ denote the translation between camera i coordinate frame and camera j. Also, suppose R_{ij} is the relative rotation between cameras i and j. Then define

$$\mathrm{CG}_{\mathrm{score1}}(i,j) = \left|\left|\vec{t}_{ij}\right|\right|_2 + \left|\cos^{-1}\left(\frac{\mathrm{tr}(R_{ij})-1}{2}\right)\right|$$

which ranges from 0 to ∞; smaller values of $\mathrm{CG}_{\mathrm{score1}}$ indicate more similar camera viewpoints. Camera groups using $\mathrm{CG}_{\mathrm{score1}}$ for the scenario in Figure 4.9 are shown in Figure 4.12.

$\mathbf{CG_{score2}}$ (View Overlap): This grouping scheme inherits the previously computed $\mathrm{VOI}_{\mathrm{score}}$ value utilizes the Sørensen-Dice index [113]. Again, let P denote the set of sampled VOI points. For arbitrary camera i, let Cam_i be the the subset of P viewable by camera i, as determined by (4.3) and (4.4). The quantifiable metric $\mathrm{CG}_{\mathrm{score2}}$ is then defined as:

$$\mathrm{CG}_{\mathrm{score2}}(i,j) = -\frac{2\left|\mathrm{Cam}_i \cap \mathrm{Cam}_j\right|}{\left|\mathrm{Cam}_i\right| + \left|\mathrm{Cam}_j\right|} + 1$$

where $|.|$ denotes cardinality. $\mathrm{CG}_{\mathrm{score2}}$ ranges from 0 to 1; smaller values of $\mathrm{CG}_{\mathrm{score2}}$ indicate more similar camera viewpoints. Camera groups using $\mathrm{CG}_{\mathrm{score2}}$ for the scenario depicted in Figure 4.9 are derived and subsequently depicted in Figure 4.13

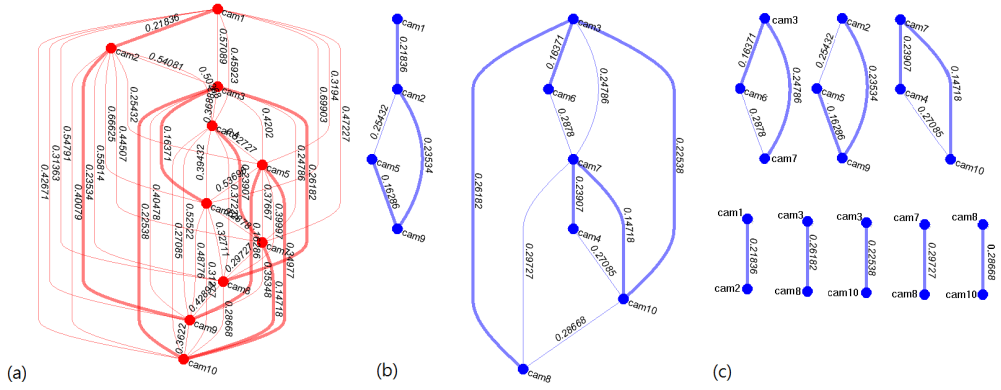


Figure 4.13: (a) $\mathrm{CG}_{\mathrm{score2}}$ of each camera pair (b) and (c) are the non-overlapping and overlapping camera grouping result.

4.3.5 Threshold Derivation

The camera grouping threshold is determined by Algorithm 4 for graph G [94]. r calculated in step 8 is the ratio between the mean intra-cluster and inter-cluster edge weight.

In Figure 4.12-(b), four camera groups are generated with a threshold of 280, separating camera pairs with large CG_{score1}. In this case, each camera belongs to exactly one camera group. Figure 4.12-(c) illustrates overlapping camera groups where a camera can exist in multiple camera groups simultaneously, which is achieved by mandating a complete graph. A threshold of 0.3 for CG_{score2} results in the camera groups depicted in Figure 4.13. The viewpoint overlaps for the scenario depicted in Figure 4.9 using the two proposed CG_{score} algorithms can be visualized graphically, as illustrated in Figure 4.14.

Algorithm 4 $\text{Thresh}_{CG}(G)$

1: **set** s as threhold increment step size
2: **set** $r_{best} = \infty$
3: **set** e_{min} = least edge weight in G
4: **set** th_c = greatest edge weight in G
5: **set** $th_{best} = th_c$
6: **while** $th_c > e_{min}$ **do**
7: remove edges with weight $\geq th_c$
8: calculate weight ratio for remaining graph, r
9: **if** $(r < r_{best})$ **then**
10: $r_{best} = r$
11: $th_{best} = th_c$
12: **end if**
13: $th_c = th_{best} - s$
14: **end while**
15: **return** th_{best}

4.3.6 Feature Matching Pair Generation

After camera groups are formed, optimal camera pairs for triangulation with groups must be determined; using all pairs can be redundant. To facilitate this, a minimum spanning

tree (MST) approach is utilized. Two popular methods exist to find the MST: Kruskal's algorithm and Prim's algorithm [99]. The MST results are distinguished by thicker edges in Figure 4.3.4 and Figure 4.3.4. Table 4.2 shows the average run time for both algorithms using the various graphs and CG_{score} values.

Each MST algorithm will be briefly explained given $N \in \mathbb{N}$ cameras, and two undirected graphs $G_1(V, E_1), G_2(V, E_2)$ where vertices V is the set of N cameras and the edges E_1, E_2 are defined by the two camera group scores, $\text{CG}_{\text{score1}}$ and $\text{CG}_{\text{score2}}$. Specifically, $E_1 = \{\text{CG}_{\text{score1}}(i,j)\}$ and $E_2 = \{\text{CG}_{\text{score2}}(i,j)\}$ for $i, j \in \{1, 2, ..., N\}$. Section 4.3.4 detailed decomposing G_1 and G_2 into sub-graphs via heuristically tuned CG_{score} thresholds. The following subsections describe MST derivation for sub-graph $g(v, e)$ using Kruskal's and Prim's algorithms.

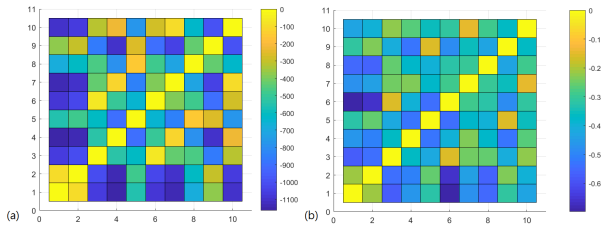


Figure 4.14: View overlap using procedures detailed in 4.3.4 and 4.3.4. On the left is $-\text{CG}_{\text{score1}}(i,j)$ and on the right is the $-\text{CG}_{\text{score2}}(i,j)$ scores. The x and y axes are camera indices. Warmer colors indicate more view overlap.

Graph			Runtime [ms]	
Figure	Overlap	CG_{score}	Kruskal's	Prim's
Figure 4.12-(a)	-	$\text{CG}_{\text{score1}}$	4.7123	4.0939
Figure 4.12-(b)	no	$\text{CG}_{\text{score1}}$	5.0104	5.4402
Figure 4.12-(c)	yes	$\text{CG}_{\text{score1}}$	5.3571	5.9124
Figure 4.13-(a)	-	$\text{CG}_{\text{score2}}$	3.9528	3.2198
Figure 4.13-(b)	no	$\text{CG}_{\text{score2}}$	3.8232	3.4516
Figure 4.13-(c)	yes	$\text{CG}_{\text{score2}}$	4.0016	4.6392

Table 4.2: Mean run time for camera feature matching pair generation using different MST algorithms and weighting functions over ten trials. Blue denotes the better performing MST algorithm for the given test condition. Note: ZNCC and ORB are adopted respectively for feature matching and feature points extraction.

MST via Kruskal's: In Kruskal's algorithm, the MST structure begins by selecting the edge with minimum weight, as determined by CG_{score}. The remaining edges are added

one-by-one to the MST structure based on edge weight, so long as their addition does not create a closed loop in the MST. Once all nodes are included, the MST is complete.

MST via Prim's: In Prim's algorithm, the MST begins by arbitrarily selecting a vertex of the sub-graph, and is removed from the set of remaining vertices. From the remaining vertices, the vertex connected to any element in the MST with least edge weight, as determined by $\mathrm{CG}_{\mathrm{score}}$ is added to the MST and removed from the pool of remaining vertices. The process continues until all vertices are added, at which point the MST is complete.

4.3.7 Pair Sequencing in Time and Space

The 3D model of the surgical cavity is obtained by merging point clouds derived from triangulating matched feature points from every generated image pairs, including both intra- and intercamera pairs. The intercamera pairs are pairs of connected vertices in the MST. However, the accuracy of the final 3D model is affected by the merging sequence. Severla considerations are made in determining both intra- and intercamera pair sequencing. The methods used in this work are described in the following subsections.

Intercamera Pair Matching: (1) *Sub-Graph Point Cloud Generation:* At this point each camera group is represented by the MST of some sub-graph g. Select arbitrarily s, a leaf vertex in g. An initial point cloud is generated from s and an adjacent vertex. Subsequent point clouds are merged by traversing the remainder of the MST via a depth first search approach. Each merge step contributes to a cumulative point cloud, PC_{gi} where i denotes the merge iteration. A final point cloud is generated for sub-graph g, denoted PC_g. To ensure tolerance of map point position uncertainties, a Gaussian feature detection error $N(0, \sigma^2 I)$ is imposed for every merged point. (2) *Reprojection Error Check:* Prior to each merge step i in sub-graph point cloud generation, PC_{gi-1} is reprojected to each image frame in g. Only the points whose maximum reprojection error $\mathrm{REP}_{\mathrm{err}}$, which is simply the Mahalanobis distance between reprojected point and nearest feature point within each camera frame [138].

Intracamera Pair Matching: Intracamera matching classifies dynamic vs static areas of PC_g. For each camera vertex within sub-graph g, REP_{err} of points in PC_g is calculated and compared between the current and previous camera frame (in this work, a framerate of 60 Hz is utilized). If the difference in reprojection error is greater than a predefined threshold, the point is labeled as non-static.

Fused Camera Group Result: The generated point clouds from each camera group are combined to form an aggregate surgical cavity 3D model. Because camera groups were formed based on workspace visibility, limited overlap will occur between camera group point clouds. Where overlap does occur, uncertainty correction as described for sub-graph point cloud generation is employed.

4.4 Experiments

4.4.1 Data Collection

Data were collected from ten tracked, arbitrarily moving cameras viewing a phantom surgical scene for three minutes. The frame rate for all cameras was 60 Hz. To obtain the ground truth 3D model of the surgical cavity, the Space Spider White Light Scanner was used (Figure 4.15). The 3D resolution of the scanner is 0.1 mm with point accuracy at the 0.05 mm scale.

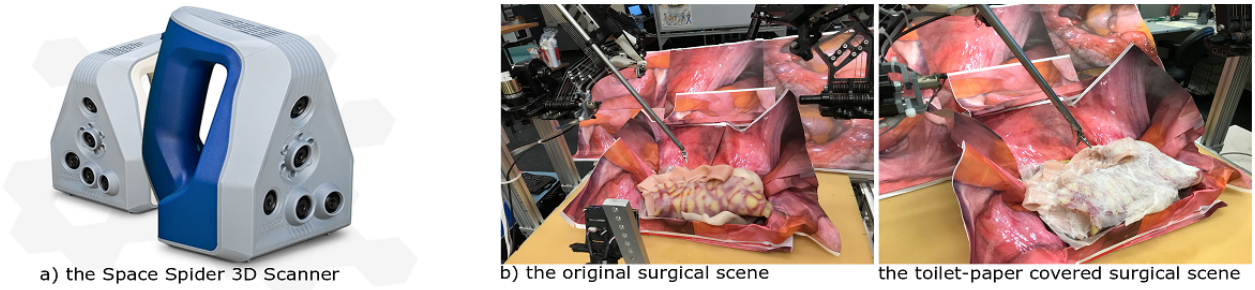


Figure 4.15: a) Space Spider 3D Scanner to generate ground truth 3D model of surgical scene. b) Surgical scene pre and post tissue paper treatment.

The Space Spider is not robust to foam and other elastic plastic materials. Therefore, the original phantom tissue surface was ill-perceived by the scanner. To solve this problem, a thin layer of semi-damped tissue paper covered the tissue surface (Figure 4.15-b). This in effect makes the surface material amenable to 3D scanner detection. Furthermore, damping the thin tissue paper ensured consistent and smooth adhering, resulting in negligible topology changes, and the transparent nature of damp tissue paper helped preserve the color profile of the phantom. A total of 11 scans were required to completely record the geometry of the surgical scene, with each scan containing approximately 700 image frames. Post processing was performed in Artec Studio to generate the 3D ground truth model shown in Figure 4.16.

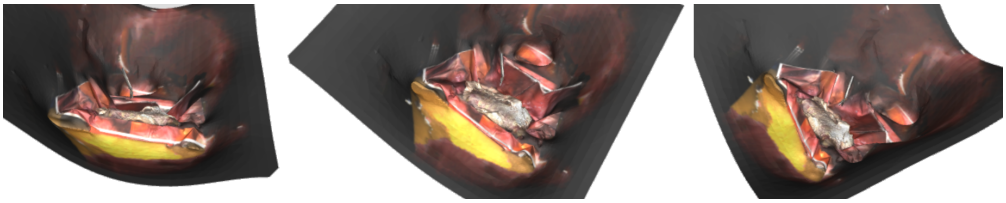


Figure 4.16: The ground truth 3D model of the surgical scene.

After scanning, a few post processing steps are required to construct the 3D model as a whole from multiple scans (Figure 4.17). These include alignment, global registration, noise elimination, fusion, mesh simplification and texture building.

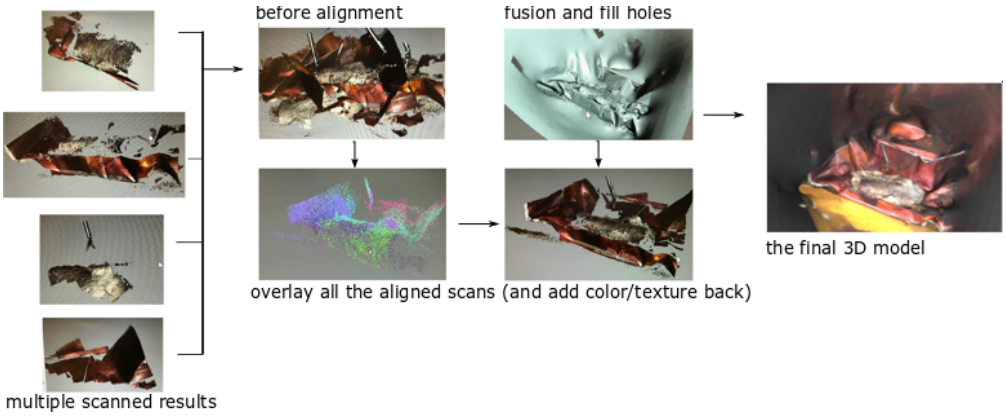


Figure 4.17: Post processing of the ground truth 3D model.

4.4.2 Experimental Setup

A total of six test cases of interest were generated to evaluate different grouping parameters. First, selection of camera grouping method could be classified into four grouping methods: exhaustive pair grouping, nearest pair grouping, CG_{score1}, or CG_{score2}. The latter two methods are divided further into group overlap or no group overlap, as determined via CG_{score} threshold described in Section 4.3.4 – if cameras are allowed to exist in multiple camera groups, overlap is present.

Each experimental condition was evaluated against three metrics of interest, which reflect the density of reconstruction, reconstruction error, and algorithm efficiency respectively:

1. total number of points generated in the 3D surface reconstruction
2. RMSE from ground truth
3. number of camera pairs evaluated

4.5 Results and Discussion

Fig. 4.18 depicts the final surgical cavity 3D reconstruction results from each test condition, and the evaluated metrics are shown in Table 4.3.

Test	Pairing	Overlap	N	RMSE [mm]	Pairs
A	every pair	-	8988	4.2773	45
B	nearest pair	-	1382	2.4424	9
C	CG_{score1}	N	6457	1.7359	6
D		Y	7349	1.6867	6
E	CG_{score2}	N	7056	1.5529	8
F		Y	8678	1.2887	11

Table 4.3: Experimental results showing for each test condition: N - number of points generated, RMSE - error from ground truth point cloud, Pairs - number of camera pairs evaluated. The time efficiency is roughly proportional to the number of triangulated camera pairs. Note: there are N= 16870 points in the ground truth point cloud.

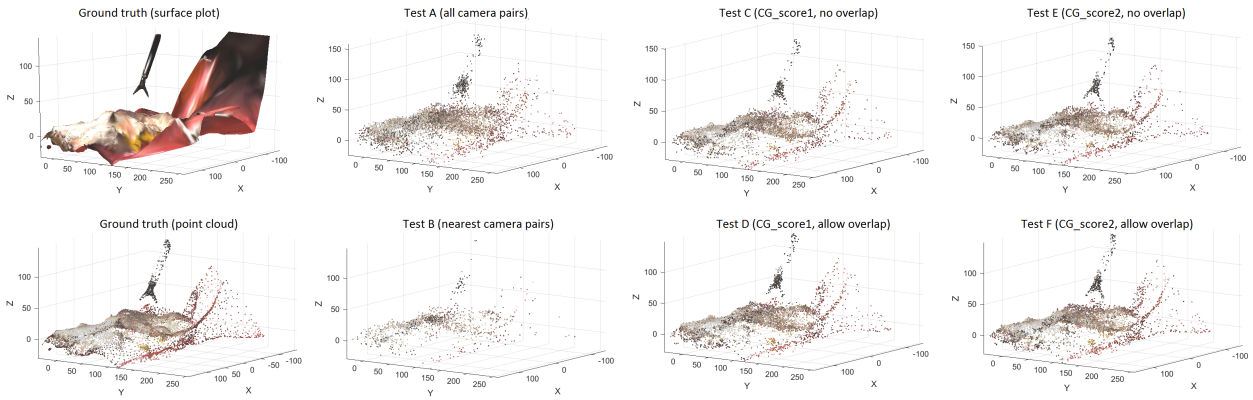


Figure 4.18: 3D reconstruction results using different camera groupings schemes.

Consider the following observations:

- **Test A**: Exhaustive camera pairing results in a dense, noisy point cloud. Since no grouping was performed, reprojection error conditioning as described in Section 4.3.7 is unfeasible, resulting in large N.
- **Test B**: Cameras are paired with nearest neighbor, resulting in one single camera group. The resultant point cloud is of higher precision but sparser.
- **Test C-F**: The proposed camera grouping methods resulted in denser, less noisy and more efficient point clouds than Test A. Tests D and F allow camera group overlap which leads to higher point cloud density and accuracy, yet may cost computational time.
- **Test C,D**: $\text{CG}_{\text{score1}}$ is faster to compute and is robust to workspace size. In contrast, CG_{score2} exhibits runtime proportional to workspace size and sample density (as shown in Fig, 4.10), and is less suitable in larger of time-varying/dynamic workspaces. In this particular experiment, the number of camera pairs does not change between Test C and Test D. Difference in camera groupings result in variation in N and $RMSE$.
- **Test E,F**: View overlap results in more matched feature points, and thus $\text{CG}_{\text{score2}}$ fits the problem objective well. In this condition, camera group overlap has a greater effect on all three metrics as compared with $\text{CG}_{\text{score1}}$.

Chapter 5

TISSUE DEFORMATION ANALYSIS AND FORCE ESTIMATION

Deformable objects and surfaces are ubiquitous in the daily lives of humans – from the garments in fashion to soft tissues within the body. Because of this routine interaction with soft materials, humans are adept and trained in manipulation of deformable objects while avoiding irreversible damage. The dexterity and care involved is largely facilitated through a combination of the human haptic sense of touch and visual observations of object deformation [235]. While this scenario presents itself as a trivially intuitive task, it becomes significantly more difficult and complex with the deprivation of both 3D depth perception and haptic senses. This deprived state is not dissimilar to the scenarios encountered in many robot-assisted minimally invasive surgeries, which can lead to unintentional tissue damage [40]. One approach to remediate these issues combines real-time dynamic 3D reconstruction and vision-based force estimation for haptic feedback. In Chapter 4, a novel approach of camera grouping and pair sequencing [205] framework is introduces. Continuing the endeavors in multicamera 3D reconstruction of dynamic surgical cavities, this work introduces a method for non-rigid, sparse point cloud registration and subsequent point classification into three categories: static, shifting and deforming [203]. Afterward, contact force can be obtained from tissue deformation level by looking up a medical datasheet for a specific tissue type.

Further studies in **section 5.5** seek: (1) an optimal camera pose adjustment algorithm in the multicamera system such that fewer cameras are needed [206]; and (2) a cyber security perspective toward vision derived force feedback [208]. The topics addressed here present open challenges and ongoing research directions for researchers to this day [123], and provide a step towards real-time 3D reconstruction and force feedback in robot-assisted surgery.

5.1 Background

5.1.1 Related Work

Non-rigid Registration: Registering multiple frames of medical imaging is necessary to stitch together volumes or scenes of the region of interest in real-time. Real-time, rigid methods for ultrasound have been proven in clinical validation studies and are adequate only with very slight deformations [184]. However, these methods are often inappropriate for the soft tissues encountered in MIS applications. Non-rigid registration of dynamic point clouds is a field of research on its own.

Oftentimes non-rigid registration is formulated as an energy functional containing both data and regularization terms. Regularization terms help to preserve smoothness, affording the optimization procedure robustness to noise and outliers. Wand et al. and Li et al. created deformation fields to fit data during optimization for non-rigid registration [116, 221]. In another approach, Süßmuth et al. introduced an as-rigid-as-possible energy function to promote smoothness [198]. A high-order graph matching technique with implicit embedding energy was used for registration with high deformation by Zeng et al. [239]. Guo et al. demonstrated that using l_0 regularization in non-rigid registration can improve robustness and accuracy [80]. Methods for sparse non-rigid surface data are elaborated in [86, 234]

The quadratic data terms used in [80, 116, 234] implicitly assume positional errors with Gaussian distributions. Soft tissue deformation resulting from natural breathing or heartbeat are large, piece-wise smooth signals residing on 3D surfaces. On the other hand, indentations due to tool-tissue interactions usually result in larger positional errors close to the incision point, with smaller errors for the remaining surfaces. Therefore, this indicates that the positional errors are sparse, and are thus better suited for and modeled by a heavy-tailed distribution instead of a Gaussian one. Such a model was incorporated in the registration method presented in [117], for which surfaces were assumed piece-wise smooth, and substantial changes in transformations could occur only in relatively local areas.

Deformable Shape Correspondence: Deformation tracking can provide crucial information for force estimation.One approach is to match feature points between two frames of the same deformable object, and estimate a correspondence between those frames; registration, alignment, and matching are special cases of the shape correspondence problem. The complexity of determining correspondence relies on context: partial vs full, dense vs sparse, semantic vs geometry, local vs global etc. [213]. Methods exist for various shape representations. Given implicit surfaces, conformal mappings using diffeomorphisms can be used to produce a space-of-shapes or geodesics, which morph between two shapes [189].

Mesh representations of a surface are amenable to topological approaches for deformable tracking. Given a mesh shape model and a few anchor vertices, a mean-value encoding approach can be used to evaluate shape-preserving and rotation invariant deformations [106]. In another method, a robust mesh correspondence search was achieved via a combinatorial tree traversal that weighted heavily self-distortion energy [240]. Large deformations can be tracked so long as the deformed surface is near-isometric to the original genus zero surface. By first flattening meshes using a mid-edge flattening technique and conformal mapping, Lipman et al. developed a Möbius voting technique that could automatically determine dozens of correspondences between genus zero surfaces under large deformations [126]. A similar Möbius approach was used to ascertain intrinsically symmetric point correspondences in [103]. Other approaches involve first segmenting shapes into semantic parts, followed by registrations between near-isometric shapes within these classes. This was achieved by employing eigenfunctions of the Laplace-Beltrami operator [174].

Schulman et al. developed an algorithm for tracking deformable objects in real-time from sequences of point clouds. The approach utilized a physics engine and a probabilistic generative expectation maximization algorithm to determine point cloud and mesh model correspondences. The solution was robust to occlusion, yet would not recover well from a divergent estimate [185]. Point clouds are also amenable to skeletal approaches. Given even sparse point clouds, curve skeletons can be extracted via an iterative method assuming

shapes are generally cylindrical [199]. By modeling the evolution of competing fronts within an object's volumetric shape, curve skeletons can also be analyzed for both sparse point clouds and meshes [188]. While many of these approaches result in accurate and repeatable shape correspondences, most rely on either a priori assumptions of the object's shape or the presence of numerous distinct features (geometric, color), assets not necessarily available in surgical settings. Furthermore, only a few of these approaches work in real-time.

Dynamic Point Classification: To estimate interaction force based on tissue deformation, the dynamic changes in the surgical cavity surface must be tracked. Locally deforming surfaces should be distinguished from static or merely shifting regions. In [241], re-projection error values of mapped 3D point cloud data between inter and intra camera groups provide indications for distinguishing the dynamic or static nature of the observed geometries. In robot-assisted MIS, it is challenging to isolate and segment the moving surgical tool tip points from nearby deforming tissue points. This is exacerbated by the fact that tissue features and colors are often reflected off the metallic tool surface [123]. To overcome this, this work incorporates robot kinematic information with the constrained optimized non-rigid registration results, thus more selectively distinguishing deforming points from merely shifting ones.

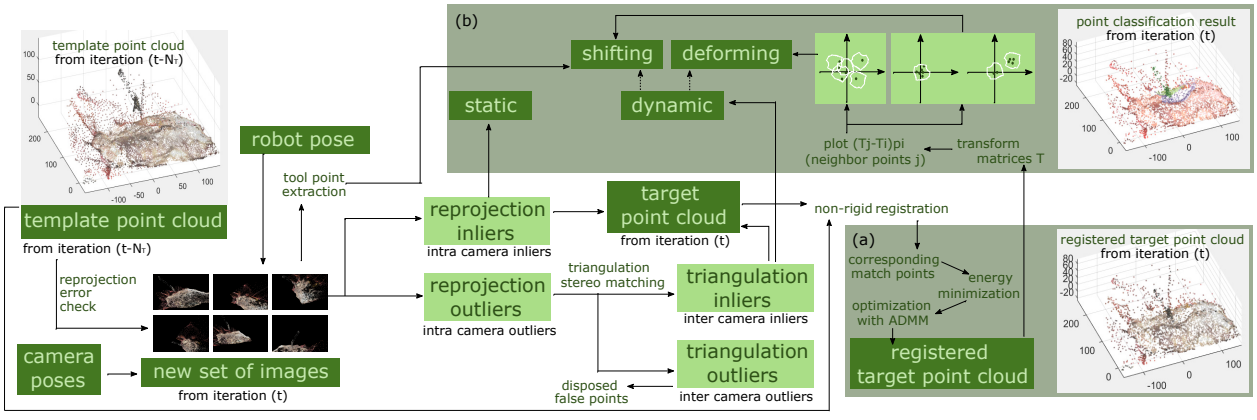


Figure 5.1: The flowchart for (a) non-rigid registration and (b) point classification. Grey boxes demarcate the two main results of the algorithm.

5.1.2 Contribution

In my previous work [205], a graph-based pairwise camera sequencing method for real-time multicamera 3D reconstruction of dynamic surgical cavities is proposed. Towards realization of simultaneous 3D reconstruction and interaction force estimation, the work here extends those previous findings to non-rigid point cloud registration over time and subsequent classification of locally static, shifting, or deforming surfaces [203]. This work utilizes a robot-assisted MIS scenario for which multiple endoscopes are present in the surgical cavity.

This paper presents a constrained optimization framework for 3D information processing from multiple viewpoints in a dynamic surgical environment such that:

- point clouds from successive time frames are optimally registered while simultaneously ensuring shape and smoothness of the resultant 3D model as well as maintaining the dynamic nature of the surgical scenes;
- points are classified into - static or dynamic. Dynamic points are further classified as either deforming or shifting, depending on the relative motion of neighboring points.

5.2 Methods

Since both position and color of surface points are useful indicators for feature registration, surface points in this work are stored in six dimensional color point clouds, i.e. $\vec{p} \in \mathbb{R}^6$ where

$$\vec{p} = \begin{pmatrix} {}^{c}\vec{p} \\ {}^{p}\vec{p} \end{pmatrix}$$

and ${}^{c}\vec{p} \in \mathbb{R}^3$ stores the RGB color values and ${}^{p}\vec{p} \in \mathbb{R}^3$ the Cartesian position vector of point $\vec{p}$. In addition to this, each point position ${}^{p}\vec{p}$ is augmented to form homogenous coordinate, ${}^{h}\vec{p} = ({}^{c}\vec{p} \quad 1)^T$. Given this point cloud representation, the flowchart shown in Figure5.1 conveys the overall workflow described in this work, the details of which are described in the following sections.

5.2.1 Template Point Cloud

The template point cloud, denoted $\mathbb{P}$, contains ordered points collected during the first time instance. In particular, $\mathbb{P}$ is generated accumulatively from multiple view points within the surgical cavity. The process for building the template 3D model from multiple 2D images involves inter-camera matching, pair sequencing, and triangulation – details of which are found in the authors' prior work [205]. A sample of a template point cloud from various viewpoints is shown in Figure 5.2. All points are treated as static at time 0.

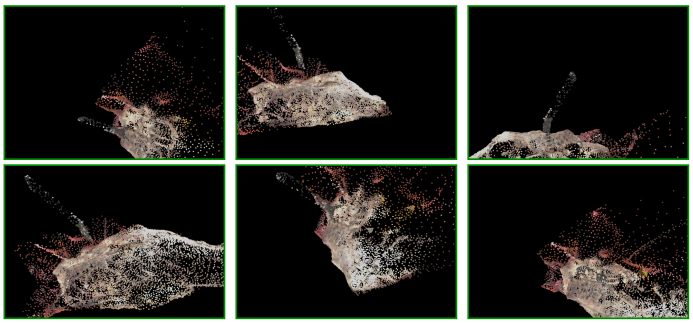


Figure 5.2: Extracted feature points from multiple 2D images captured with 6 cameras from different viewpoints are combined to form $\mathbb{P}$.

5.2.2 Target Point Cloud

For each time step after the initial template point cloud is formed, a new set of images are acquired from all cameras. Visible 3D points in the template point cloud are then reprojected to the new set of images based on current camera poses to determine which regions are viewable at the current time instance. Viewable points which are correctly shown in the separate 2D images will be updated. Then, the union of all non-viewable 3D template points and the updated regions form the target point cloud, $\mathbb{G}$, for that time instance.

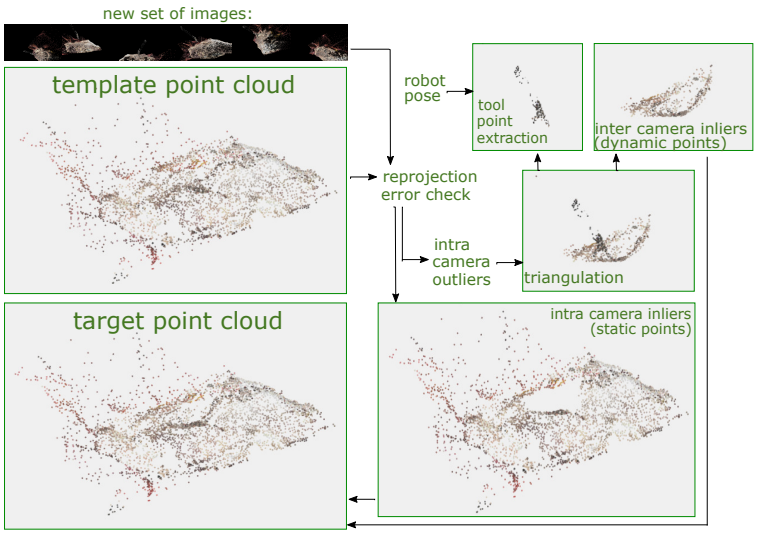


Figure 5.3: The generation of target point cloud $\mathbb{G}$.

Figure 5.3 conveys the process by which the target point cloud is generated. Observe that any feature points within the 2D images which are not yet associated with a 3D point are triangulated, and thereby associated with a new 3D point in the target point cloud, $\mathbb{G}$.

5.2.3 Surgical Tool Segmentation and Removal

Robot kinematic pose information is tracked by the Raven-II system [17]. Given this pose, accurate 3D models of the robot, and camera poses, a 2D mask of the tool shaft can be generated and projected onto each camera image. This in effect eliminates any tool points, both in the 2D images and in the generated 3D point cloud [91, 204]. As a result, both template and target point clouds $\mathbb{P}$, $\mathbb{G}$, consist only of points representing the topology of the soft tissue within the surgical cavity.

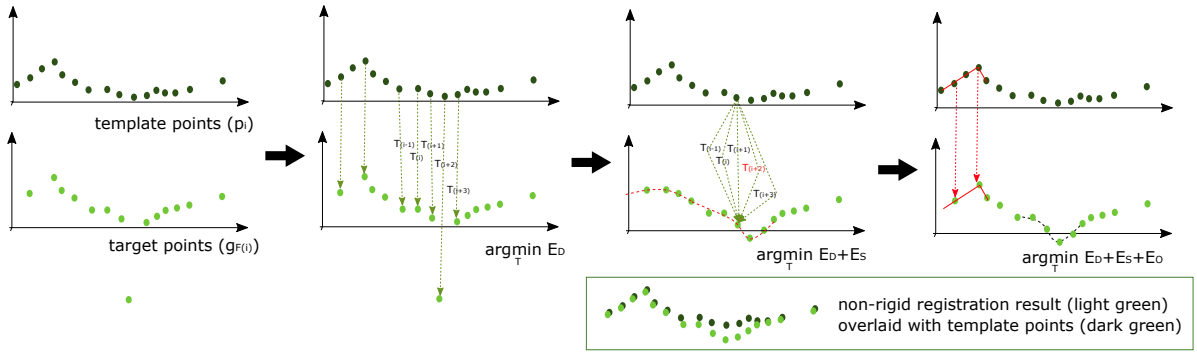


Figure 5.4: Illustration of the three terms in the energy function - $E_{data}, E_{smth}, E_{orth}$.

5.2.4 Energy Function

Suppose that $|\mathbb{P}| = N_{\mathbb{P}}$ and $|\mathbb{G}| = N_{\mathbb{G}}$. Furthermore, denote the set of the first w natural numbers, $\{1, 2, ..., w\}$, as $\mathbb{N}^{[w]}$. With an appropriate energy function defined, the goal is to determine correspondences between each point $\vec{p} \in \mathbb{P}$ to points $\vec{g} \in \mathbb{G}$ in an energy optimal fashion. More explicitly, a correspondence map $\mathcal{F} : \mathbb{N}^{[N_{\mathbb{P}}]} \rightarrow \mathbb{N}^{[N_{\mathbb{G}}]}$ is sought such that $\forall i \in N_{\mathbb{P}}$ the point $\vec{p}_i$ corresponds to $\vec{g}_{\mathcal{F}(i)}$ while minimizing energy. This energy optimization relies on homogenous transformations T_i that transform points $\vec{p}_i \in \mathbb{P}$ to $\vec{g}_i \in \mathbb{G}$.

More succinctly $\mathcal{F}(i) : \mathbb{N}^{[N_{\mathbb{P}}]} \rightarrow \mathbb{N}^{[N_{\mathbb{G}}]}$ can be formulated as

$$\mathcal{F}(i) = \underset{j \in N_{\mathbb{G}}}{\operatorname{argmin}} \left\| \begin{pmatrix} T_i \cdot {}^{h}\vec{p}_i \\ {}^{c}\vec{p}_i \end{pmatrix} - \begin{pmatrix} {}^{p}\vec{g}_j \\ {}^{c}\vec{g}_j \end{pmatrix} \right\|_2 \tag{5.1}$$

and define matrices

$$\mathbf{T} \in \mathbb{R}^{4N_{\mathbb{P}} \times 3}, \quad \mathbf{T} = \begin{pmatrix} T_1 \\ T_2 \\ \vdots \\ T_{N_{\mathbb{P}}} \end{pmatrix}, \text{ where initially } T_i = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \forall i$$

$$\mathbf{P} \in \mathbb{R}^{N_{\mathbb{P}} \times 4N_{\mathbb{P}}}, \ \mathbf{P} = \begin{pmatrix} {}^{h}\vec{p}_1^{T} & & \\ & \ddots & \\ & & {}^{H}\vec{p}_{N_{\mathbb{P}}}^{T} \end{pmatrix}$$

$$\mathbf{G} \in \mathbb{R}^{N_{\mathbb{P}} \times 3}, \quad \mathbf{G} = \begin{pmatrix} {}^{p}\vec{g}_{\mathcal{F}(1)} & {}^{p}\vec{g}_{\mathcal{F}(2)} & \dots & {}^{p}\vec{g}_{\mathcal{F}(N_{\mathbb{P}})} \end{pmatrix}^{T}$$

The point correspondence map $\mathcal{F}$ and transformations $\mathbf{T}$ are iteratively updated to minimize the energy function:

$$E = E_D + \alpha E_S + \beta E_O \tag{5.2}$$

where α, β are real valued weighting scalars and

$$E_D = \|\operatorname{diag}(k_D)\,(\mathbf{PT} - \mathbf{G})\|_1 \tag{5.3}$$

$$E_S = \left\|\operatorname{diag}(\vec{k_S})\mathbf{BT}\right\|_1 \tag{5.4}$$

$$E_O = \|\mathbf{ST} - k_O\|_F^2 \tag{5.5}$$

with the three energy components E_D, E_S, E_O detailed in the following sections and illustrated in Figure 5.4

Data Term E_D:

The data component ensures each transformed template point, ${T_i}^h\vec{p}_i$, is close to the corresponding target point ${}^p\vec{g}_{\mathcal{F}(i)}$ in the 1-norm sense. Expressed point-by-point,

$$E_D \;=\; \sum_{i=1}^{N_{\mathbb{P}}} k_{D_i} \left\| {T_i}^h\vec{p}_i - {}^p\vec{g}_{\mathcal{F}(i)} \right\|_1$$

where $k_D \in \mathbb{R}^{N_{\mathbb{P}}}$ contains real value weights which are determined by

$$k_{D_i} \;=\; \left\{ \left\| \left(\tilde{T}_i^{\,h}\vec{p}_i - {}^p\vec{g}_{\mathcal{F}(i)} \right) \right\|_1 + \epsilon_D \right\}^{-1} \tag{5.6}$$

Note that here $\tilde{T}_i$ indicates the previous iteration value of T_i. $\epsilon_D = 0.001$ is set to a small positive value to prevent division by zero.

Smoothness Term E_S:

The smoothness component achieves piece-wise smoothness of the 3D surface by ensuring neighboring points in $\mathbb{P}$ share similar transformations. In order to do this, the notion of "neighboring" points in $\mathbb{P}$ must be defined. To that end, let the distance between two arbitrary points in $\mathbb{P}$ be the Cartesian Euclidean sense. That is, the distance between two points $\vec{p}_q, \vec{p}_r \in \mathbb{P}$ is simply $||{}^p\vec{p}_q - {}^p\vec{p}_r||_2$. Then neighboring points for point $\vec{p}_i \in \mathbb{P}$ are determined by a neighborhood mapping $\mathcal{B} : \mathbb{N}^{[N_{\mathbb{P}}]} \times \mathbb{N} \to \mathbb{N}^{[N_{\mathbb{P}}]}$, where $\mathcal{B}(i,j)$ returns the index, k, of the j^{th} nearest point $\vec{p}_k \in (\mathbb{P} - \vec{p}_i)$ to point $\vec{p}_i$. For example, $\mathcal{B}(1,2)$ would return the index of the 2^{nd} nearest neighbor to $\vec{p}_1$. In this work, a search of $N_B = 6$ nearest neighbors is used for smoothing. Now define smoothing matrix

$$\mathbf{B} \in \mathbb{R}^{N_B N_{\mathbb{P}} \times 4N_{\mathbb{P}}}, \quad \mathbf{B} = \begin{pmatrix} [b_1] \\ [b_2] \\ \vdots \\ [b_{N_{\mathbb{P}}}] \end{pmatrix}$$

where submatrices $b_q \in \mathbb{R}^{N_B \times 4N_{\mathbb{P}}}$ are defined as

$$b_q = \begin{pmatrix} \cdots & \underbrace{{}^h\vec{p}_q^{\,T}}_{q^{\text{th}}} & \cdots & \underbrace{-{}^h\vec{p}_q^{\,T}}_{\mathcal{B}(q,1)^{\text{th}}} & \cdots \\ & \vdots & & \ddots & \\ \cdots & \underbrace{{}^h\vec{p}_q^{\,T}}_{q^{\text{th}}} & \cdots & \underbrace{-{}^h\vec{p}_q^{\,T}}_{\mathcal{B}(q,N_B)^{\text{th}}} & \cdots \end{pmatrix}$$

Then E_S should regulate transformed neighboring points as

$$E_S = \sum_{i=1}^{N_{\mathbb{P}}} \sum_{j=1}^{N_B} k_S(i,j) \left\| \left(T_i - T_{\mathcal{B}(i,j)}\right) {}^h\vec{p}_i \right\|_1$$

where $k_S \in \mathbb{R}^{N_{\mathbb{P}} \times N_B}$ contains real-valued weights determined by

$$k_S(i,j) = \left\{ \left\| \left(\tilde{T}_i - \tilde{T}_{\mathcal{B}(i,j)}\right) {}^h\vec{p}_i \right\|_1 + \epsilon_S \right\}^{-1} \tag{5.7}$$

Here $\epsilon_S = 0.001$ prevents division by zero. The vector form of k_S is produced by concatenating successive rows of k_S and is denoted $\vec{k_S}$ as in Equation (5.4).

Orthogonal Term E_O:

The orthogonal component ensures preservation of the local surface by enforcing that the orientation portion of each T_i, call it R_i, is close to a proper rotation matrix, i.e. orthogonal with determinant of 1. Since R_i is not necessarily a proper rotation, a proper rotation matrix $\mathcal{R}_i$ that is nearest of R_i is sought via singular value decomposition

$$\begin{aligned} U_i \Sigma_i V_i^T &= R_i \\ \mathcal{R}_i &= \operatorname{sgn}\left(\det\left(U_i V_i^T\right)\right) U_i V_i^T \end{aligned}$$

Now define a row-selection matrix, $\mathbf{S} \in \mathbb{R}^{3N_{\mathbb{P}} \times 4N_{\mathbb{P}}}$, and collect the $\mathcal{R}_i$ into $k_O \in \mathbb{R}^{3N_{\mathbb{P}} \times 3}$ such

that

$$\mathbf{ST} = \begin{pmatrix} R_1 \\ \vdots \\ R_{N_{\mathbb{P}}} \end{pmatrix} \quad , \quad k_O = \begin{pmatrix} \mathcal{R}_1 \\ \vdots \\ \mathcal{R}_{N_{\mathbb{P}}} \end{pmatrix} \tag{5.8}$$

Then E_O should regulate deviation from proper rotation as

$$\begin{aligned} E_O &= \sum_{i=0}^{N_{\mathbb{P}}} \|R_i - \mathcal{R}_i\|_F^2 \\ &= \|\mathbf{ST} - k_O\|_F^2 \end{aligned}$$

as in Equation 5.5, and where $\|\cdot\|_F$ indicates the Frobenius norm. This term takes effect when portions of the target point cloud are significantly distorted from the template.

5.2.5 Optimization

Recall from Equation 5.2 that the goal is to iteratively update transformations $\mathbf{T}$ to minimize energy, in other words to solve for

$$\underset{\mathbf{T}}{\text{argmin}} : E_D + \alpha E_S + \beta E_O$$

where α, β are real values. In this work, $\alpha = 0.5$ and $\beta = 0.3$. The minimization is readily achievable if the formulation is constrained. To that end, a method of Lagrange multipliers is proposed. First define

$$\begin{aligned} Q_1 &= \text{diag}(k_D)\,(\mathbf{PT} - \mathbf{G}) \\ Q_2 &= \text{diag}(\vec{k_S})\mathbf{BT} \end{aligned}$$

and recall relations between Q_1, Q_2 and E_D, E_S, as shown in Equations (5.3,5.4). Now introduce undetermined Lagrangian multipliers Λ_1, Λ_2, and positive real valued parameters

μ_1, μ_2. Then denote the following Lagranian terms

$$\begin{aligned} L_a &= \langle \Lambda_1, Q_1 - \text{diag}(k_D)\left(\mathbf{PT} - \mathbf{G}\right)\rangle \\ L_b &= \langle \Lambda_2, Q_2 - \text{diag}(\vec{k_S})\mathbf{BT}\rangle \\ L_c &= \frac{\mu_1}{2}\left\|Q_1 - \text{diag}(k_D)\left(\mathbf{PT} - \mathbf{G}\right)\right\|_F^2 \\ L_d &= \frac{\mu_2}{2}\|Q_2 - \text{diag}(\vec{k_S})\mathbf{BT}\|_F^2 \end{aligned}$$

where $\langle\cdot\rangle$ is the inner product of argument matrices in vector form. Now define an augmented Lagrangian [75] L_ρ as

$$L_\rho = E + L_a + L_b + L_c + L_d$$

Lagrangian multipliers Λ_1, Λ_2 and parameters μ_1, μ_2 are updated iteratively by

$$\mu_i = \rho_i \tilde{\mu}_i \qquad \text{where } i = 1, 2 \tag{5.9}$$

$$\begin{aligned} \Lambda_1 &= \tilde{\Lambda}_1 + \tilde{\mu}_1\left(Q_1 - \text{diag}(k_D)\left(\mathbf{PT} - \mathbf{G}\right)\right) \\ \Lambda_2 &= \tilde{\Lambda}_2 + \tilde{\mu}_2\left(Q_2 - \text{diag}(\vec{k_S})\mathbf{BT}\right) \end{aligned} \tag{5.10}$$

where $\tilde{\cdot}$ variables are from the previous iteration. The alternate direction method of multipliers (ADMM) [21] is then used to iteratively update $\mathbf{T}, Q_1, Q_2$. First define shrinking operation $S : \mathbb{R}^{m\times n} \times \mathbb{R} \to \mathbb{R}^{m\times n}$, whereby $\hat{\mathbb{X}} = S(\mathbb{X}, \tau)$ is determined element-wise as

$$\hat{\mathbb{X}}(i,j) = \text{sgn}\left(\mathbb{X}(i,j)\right)\max(|\mathbb{X}(i,j)| - \tau, 0)$$

Now consider difference matrices

$$\begin{aligned} \mathbb{D}_1 &= \text{diag}(k_D)(\mathbf{P}\tilde{\mathbf{T}} - \mathbf{G}) - \tilde{Q}_1\tilde{\mu}_1^{-1} \\ \mathbb{D}_2 &= \text{diag}(\vec{k_S})\mathbf{B}\tilde{\mathbf{T}} - \tilde{Q}_2\tilde{\mu}_2^{-1} \end{aligned}$$

where again $\tilde{\cdot}$ variables are from the previous iteration. Q_i is then updated for $i = 1, 2$ as

$$Q_i \quad = \quad \underset{Q_i}{\operatorname{argmin}} L_\rho = S\left(\mathbb{D}_i, \tilde{\mu}_i^{-1}\right) \tag{5.11}$$

Since $\mathbf{T}$ is quadratic, its iterative minimization is solved with a first-order optimization condition:

$$\begin{aligned}\Big(\mu_1 \mathcal{H}(\mathrm{diag}(k_D)\mathbf{P}) + \mu_2 \mathcal{H}(\mathrm{diag}(\vec{k_S})\mathbf{B}) + 2\beta\mathcal{H}(\mathbf{S})\Big)\,\mathbf{T} = \\ \mathbf{P}^T \mathrm{diag}(k_D)^T(\Lambda_1 + \mu_1 Q_1 + \mu_1 \mathrm{diag}(k_D)\mathbf{G}) + \\ \mathbf{B}^T \mathrm{diag}(\vec{k_S}^T(\Lambda_2 + \mu_2 Q_2) + \mathbf{S}^T(2\beta k_O)\end{aligned} \tag{5.12}$$

where $\mathcal{H}(\mathbb{X}) = \mathbb{X}^T\mathbb{X}$. The above detailed operations in concert form the ADMM constrained optimization, summarized below in Algorithm 5.

5.2.6 Point Classification

Only deformed surfaces are indicative of tool-applied force. From section 5.2.2, points in $\mathbb{G}$ inherited directly from $\mathbb{P}$ have minimal reprojection errors, and are thus static. The remaining data must be classified as either deforming or merely shifting. The approach relies on assumptions that shifting points either (1) exhibit near identical transformations as N_B nearest neighbors or (2) exhibit near identical transformations as a subset of neighbors while the remainder shift together. All other points exhibit greater variance in neighbor distribution in $\mathbb{G}$, and are thus deemed deforming.

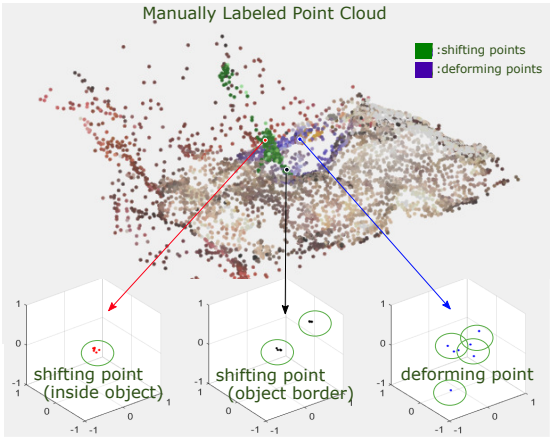


Figure 5.5: $\mathbb{P}$ and three manually labeled sample points. 6 nearest neighbors are shown in $\mathbb{G}$. Shifting points can be concentrated (red) or in two separate groups (black) in $\mathbb{G}$. Neighbors in $\mathbb{G}$ spread widely for deforming (blue).

A score, l_{C_i}, is assigned to each non-static point, $\vec{p}_i$, to determine whether or not it is shifting. l_{C_i} depends on proximity to the robot tool tip as well as the relative motion of the N_B nearest neighbors from the $\mathbb{P}$ to $\mathbb{G}$. First, from tool tip location $R \in \mathbb{R}^3$, determine the distance from $\vec{g}_i$ as

$$d_R = \left\| {T_i}^h \vec{p}_i - R \right\|_2 = \left\| {}^p\vec{g}_i - R \right\|_2$$

and calculate the neighbor index of the furthest neighbor from $\vec{g}_i$ in target point cloud by

$$k = \underset{j \in \mathbb{N}^{[N_B]}}{\arg\max} \left\| {T_i}^h \vec{p}_i - {T_j}^h \vec{p}_i \right\|_2$$

Algorithm 5 Non-rigid Registration ADMM Procedure

input template and target point clouds $\mathbb{P}, \mathbb{G}$
set M_1, M_2: predefined maximum iteration threshold
initialize m_1, m_2: current iteration as 0
while (**T** not converged) and $({}^{++}m_1 < M_1)$ **do**
  calculate G from mapping $\mathcal{F}$ in (5.1)
  update k_D using (5.6)
  update k_S using (5.7)
  while (**T** not converged) and $({}^{++}m_2 < M_2)$ **do**
    update k_O using (5.8)
    update Q_1, Q_2 using (5.11)
    calculate T using (5.12)
    update μ_1, μ_2 using (5.9)
    update Λ_1, Λ_2 using (5.10)
  end while
end while
return T

Then determine whether each neighbor, $\vec{g}_q$, is closer to $\vec{g}_i$ or $\vec{g}_k$ and record the distance in the 1-norm sense

$$a_q = \min\left(\left\| {T_{\mathcal{B}(i,q)}}^h \vec{p}_i - {}^p\vec{g}_i \right\|_1, \left\| (T_{\mathcal{B}(i,q)} - T_{\mathcal{B}(i,k)})^h \vec{p}_i \right\|_1 \right)$$

l_{C_i} is then determined as

$$l_{C_i} = d_R^{-1} \sum_{q=1}^{N_B} \frac{a_q}{\left\| {}^p\vec{p}_i - {}^p\vec{p}_{\mathcal{B}(i,q)} \right\|_2} \tag{5.13}$$

Points are more likely to be classified as deforming if near the tool tip or in dense regions. This is reflected by d_R and the denominator in Equation 5.13. A point p_i is shifting if l_{C_i} is

less than heuristically tuned threshold $\epsilon_C = 0.04$, otherwise it is labeled deforming. Figure 5.5 shows neighbor point distribution in $\mathbb{G}$ from three manually labeled points.

5.3 Experimental Results

5.3.1 Non-Rigid Registration

A robot tool tip deforming soft tissue was recorded from six viewpoints, as depicted in Figure 1.7. In total, 10 seconds were recorded at 30 fps, resulting in 300 sequential scenes. The first 10 frames were used to generate the initial template point cloud. Subsequent target surfaces were paired with the scene 10 frames (1/3 sec) prior as the template. Registration was performed on all scenes, resulting in 16870 points per point cloud. A timeline of the iterative optimization and the registration result of the 199$^{\text{th}}$ frame is shown in Figure 5.7 and Figure 5.6.

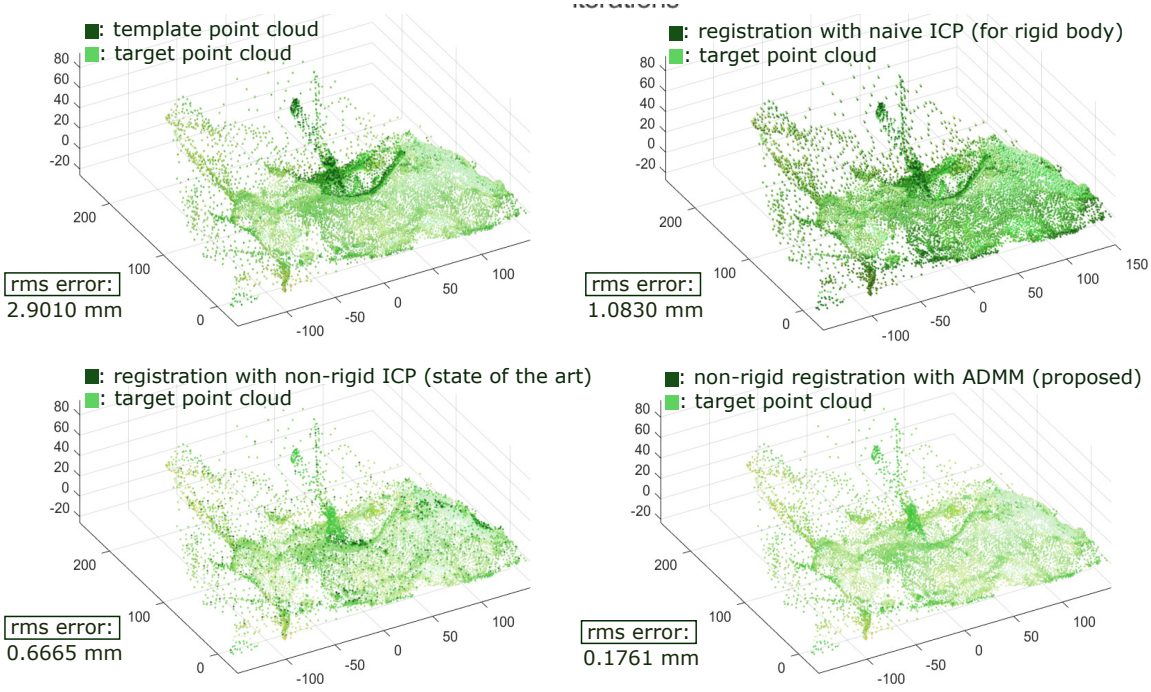


Figure 5.6: Target surfaces overlaid with the template. From top-left clockwise: target and unaltered template; rigid ICP; proposed method; state-of-the-art non-rigid ICP [80].

E_D,E_S significantly improved at the 2$^{\text{nd}}$ epoch, while E_O converged most rapidly. Large errors in rigid ICP indicate deformation, yet state-of-the-art non-rigid ICP fails at edges and tool-tissue borders, where deforming and shifting points meet [80]. The proposed method mitigates these errors.

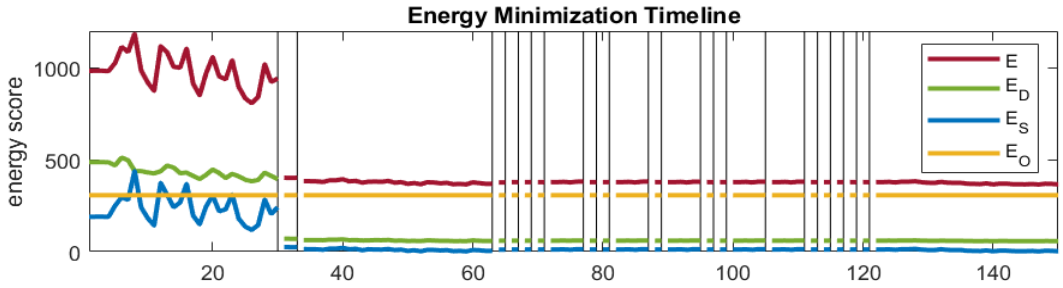


Figure 5.7: Minimization timeline of E, E_D, E_S and E_O. Vertical lines indicate an increment of the outer while loop in Algorithm 5.

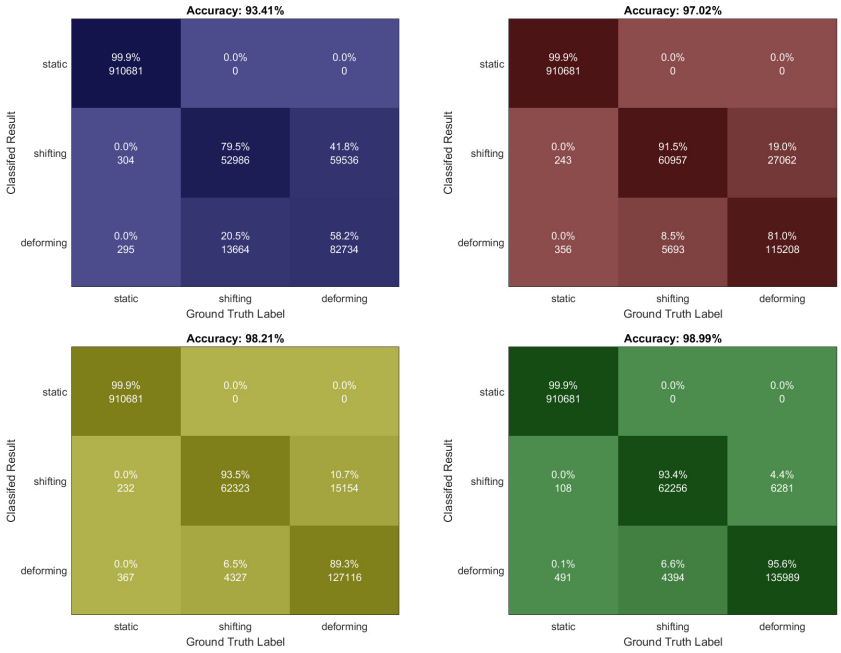


Figure 5.8: Confusion matrices: $\mathbb{A}$ (Blue), $\mathbb{B}$ (Red), $\mathbb{C}$ (Yellow), $\mathbb{D}$ (Green). $\mathbb{D}$ resulted in 98.99% accuracy. Static points propagated directly from the template; thus no false positives. Falsely labeled shifting and deforming points significantly decreased from $\mathbb{A} \rightarrow \mathbb{B} \rightarrow \mathbb{C}$. Including tool-tip proximity, $\mathbb{C} \rightarrow \mathbb{D}$, most significantly improved deforming point classification.

5.3.2 Point Classification

From the entire point cloud sequence, an evaluation set of 60 target point clouds were selected at random and manually labeled. Three additional, increasingly sophisticated classification conditions for $\vec{p}_i$ were used to evaluate each component of the proposed classifier, $\mathbb{D}$:

- $\mathbb{A}$ – a_q is evaluated for neighboring point $\vec{g}_q$ simply as $\left\| T_{\mathcal{B}(i,q)}{}^{h}\vec{p}_i - {}^{p}\vec{g}_i \right\|_1$. This condition ignores shifting object borders and proximity both to tool tip and of neighbors in $\mathbb{P}$.
- $\mathbb{B}$ – a_q is as proposed, but $l_{C_i} = \sum_{q=1}^{N_B} a_q$, ignoring proximity both to tool tip and of neighbors in $\mathbb{P}$.
- $\mathbb{C}$ – $l_{C_i} = \sum_{q=1}^{N_B} \frac{a_q}{\left\| {}^{p}\vec{p}_i - {}^{p}\vec{p}_{\mathcal{B}(i,q)} \right\|_2}$ noe considers neighbor proximity, but ignores tool tip.

Table 5.1: Aggregate Positive Classification Results

Condition	Correct Shifting	Correct Deforming	Accuracy
$\mathbb{A}$	52986	82734	93.41%
$\mathbb{B}$	60957	115208	97.02%
$\mathbb{C}$	**62323**	127116	98.21%
$\mathbb{D}$	62256	**135989**	**98.99%**
Truth	66650	142270	100%

Bold indicates best performer.

All 60 target-template pairs were processed via $\mathbb{A}$–$\mathbb{D}$, and classification results were evaluated via confusion matrices. Figure 5.9 visually compares manually labeled ground truth with the proposed method; the two are near identical.

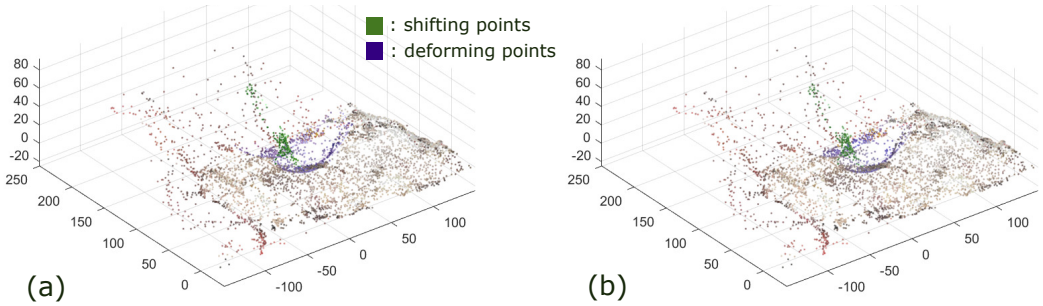


Figure 5.9: (a) manually labeled point cloud (b) classification result from $\mathbb{D}$.

5.4 Conclusion

In surgical scenes, large, piece-wise smooth tissue surfaces coexist with highly deforming tool-tissue interaction regions. These surfaces can be captured and reconstructed from multiple camera views [205]. Tool-tissue interaction forces are characterized by deforming regions, and identifying deformations can be achieved through two steps: registration and classification. The proposed energy-based non-rigid registration method converges rapidly and outperforms state-of-the-art non-rigid ICP methods, as demonstrated in Figure 5.6. Key differences include the use of a heavy-tailed distribution to account for sparse positional errors, and an iterative constrained optimization that preserves both local correspondence and piece-wise smoothness using ADMM.

Following registration, dynamic regions were classified as either deforming or shifting. This work proposed several considerations to discriminate the two, including border regions, relative movement of nearest neighbors, as well as proximity to surgical tool-tip. Parameters were incrementally incorporated into test conditions $\mathbb{A} - \mathbb{D}$, showing overall performance increase with each improvement while achieving accuracy of 98.99% using $\mathbb{D}$. Figure 5.9 depicts that the proposed classification performs close to ground truth. Experiments were conducted on 60 randomly selected deforming scenes and compared against manually labeled data, and numerical results are summarized in Figure 5.8 and Table 5.1. The methods proposed here and accompanying experimental results lay a foundation towards continual research in vision-based force estimation for robot-assisted MIS.

5.5 Further Study

Robot-assisted surgery affords great benefits over traditional surgical practices including the possibility of remote surgical operations in rural areas and higher precision than human dexterity alone. Real-time tissue deformation analysis facilitates intelligent control that improves the surgeons' experience when performing robotic surgery. A few potential future

medical robotics functionalities that leverage tissue deformation information include (1) haptic surgical guidance through virtual fixtures, (2) vision-based force estimation given known tissue property, (3) perception complementarity for telerobotics security, and (4) tissue motion prediction and compensation. My main PhD research revolves around achieving (2), but I got the opportunity to collaboratively work on two other research projects in (3) and (4) respectively and they will be covered in **section 5.5.1** [208] and **section 5.5.2** [233]. Finally, Chapter 6 contains discussion and preliminary experiments with (1).

Furthermore, tissue deformation analysis using a multicamera system naturally led to some interesting questions - what is the minimum number of cameras needed to generate a sufficiently dense 3D tissue model during RMIS? can the cameras learn to move optimally so fewer of them are needed? To answer the questions, an autonomous multicamera viewpoint adjustment algorithm was developed and covered in **section 5.5.3** [206].

5.5.1 Perception Complementarity for Telerobotics Security

Introducing robot systems into surgical tasks provides additional enhancements, including improved precision, remote operation, and an intelligent software layer capable of filtering aberrant motion and scaling surgical maneuvers. However, the software interface in telesurgery also lends itself to potential adversarial cyber attacks. Such attacks can negatively effect both surgeon motion commands and sensory information relayed to the operator.

To combat cyber attacks on the latter, one method to enhance surgeon feedback through multiple sensory pathways is to incorporate reliable, complementary forms of information across different sensory modes. Built-in partial redundancies or inferences between perceptual channels, or perception complementarities, can be used both to detect and recover from compromised operator feedback. In surgery, haptic sensations are extremely useful for surgeons to prevent undue and unwanted tissue damage from excessive tool-tissue force. Direct force sensing is not yet deployable due to sterilization requirements of the operating room. Instead, combinations of other sensing methods may be relied upon, such as non-contact

model-based force estimation. This work presents the design of a surgical simulator software that can be used for vision-based non-contact force sensing to inform the perception complementarity of vision and force feedback for telesurgery. A brief user study is conducted to verify the efficacy of graphical force feedback from vision-based force estimation, and suggests that vision may effectively complement direct force sensing.

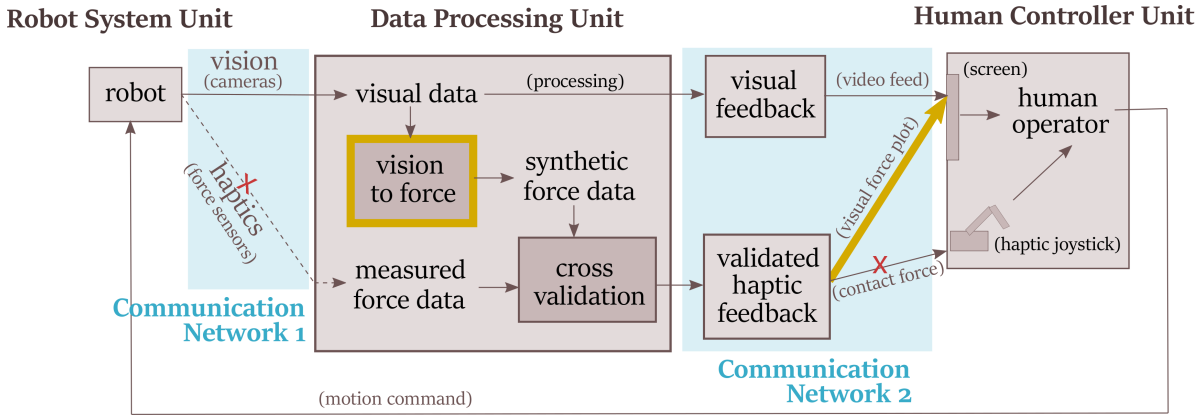


Figure 5.10: Teleoperated RMIS architecture with cybersecurity vulnerabilities identified.

In Figure 5.10, two communication networks in the sensory feedback direction are outlined with a blue background. In Communication Network 1, the dashed arrow indicates a sensory input not currently available in state-of-the-art RMIS; other modes of force estimation, e.g. vision-based, are required. The red 'X' labels are potential cyber attack pathways in the two communication networks addressed. Portioned marked in yellow highlight the two main research areas in this study, which lead to contributions in three directions:

1. introduce the notion of perception complementarities as a partial solution for cybersecurity in robot-assisted MIS;
2. simulate the estimation of haptic information via reconstruction and synthesization through vision, in this case, tissue deformation analysis;
3. evaluate through user studies sensory feedback through a disparate channel, in this case the use of graphical force feedback from vision based-estimation.

The combination of these three contributions provide insight into cross-verification and multi-

pathway transmission of various types of sensory information to enhance RMIS security. Should one type of sensory information be temporarily compromised, distorted or otherwise absent due to adversarial cyber attacks, inconsistency can be detected and thus addressed via perception complementarities.

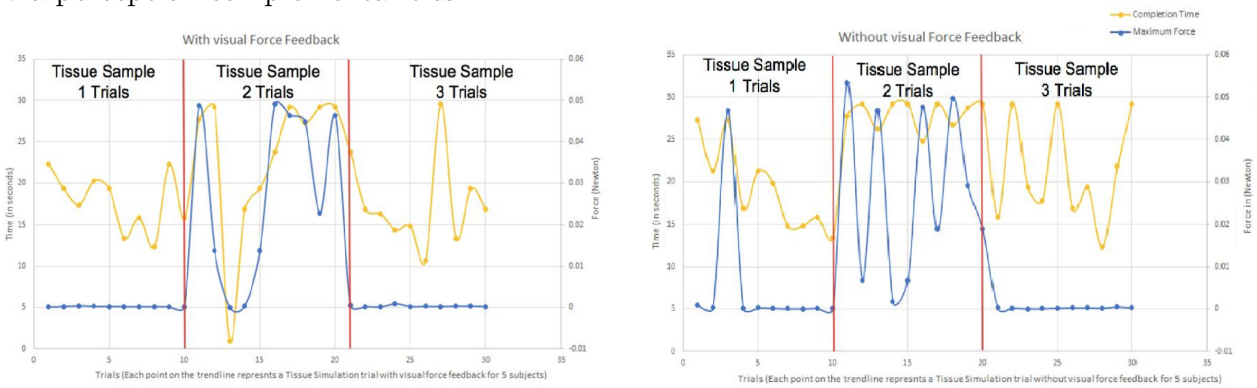


Figure 5.11: Completion time and maximum applied force from each trial conducted by the five subjects using different tissue topologies. On the left is trials with visual force feedback and on the right are trials without. The subjects are tasks to perform telesurgeries on each tissue topology four times (two with visual force feedback and two without). There are 10 sample points in each tissue topology category in the left and right subplot. The primary and the secondary y-axes represent completion time (left [secs]) and maximum force force feedback (right [N]).

The statistical data in Figure 5.11 show positive correlation between the maximum force and the time taken to complete the experiment. However, this is especially true for the trials (sample points on the graph) that are representing the most complex and difficult tissue sample (Tissue Sample 2). For the simpler tissue geometry (Tissue Sample 1), there is little correlation between the completion time and the maximum force feedback. Meanwhile, human operators tend to perform better in both efficiency and applied force with visual force feedback than without visual force feedback. Several users also shared qualitatively that they relied on the visual force plot during situations nearing time limit (30 seconds). As a result, operators needed to act faster while controlling for force or when the tissue geometry is challenging with unavoidable tool-tissue contact. Refer to [208] for more details about the experimental design and quantitative analysis.

5.5.2 Learning-based Tissue Motion Prediction and Compensation

Heartbeat synchronization, a term introduced by Nakamura in 2001 [146], is used to describe the cancellation of the relative motion between the robot-held surgical instruments and the beating heart so that surgeons can operate on the heart as if it was stationary. Similar work can be roughly categorized by research focus on either a) vision-based real-time tissue motion tracking [176, 177]; b) control algorithms for surgical robots to follow the soft tissue motion [14, 220]; or c) implementation of motion compensation on high DOF cable driven surgical manipulators, like the RAVEN-II and the da Vinci Research Kit (dVRK) [125, 180, 181].

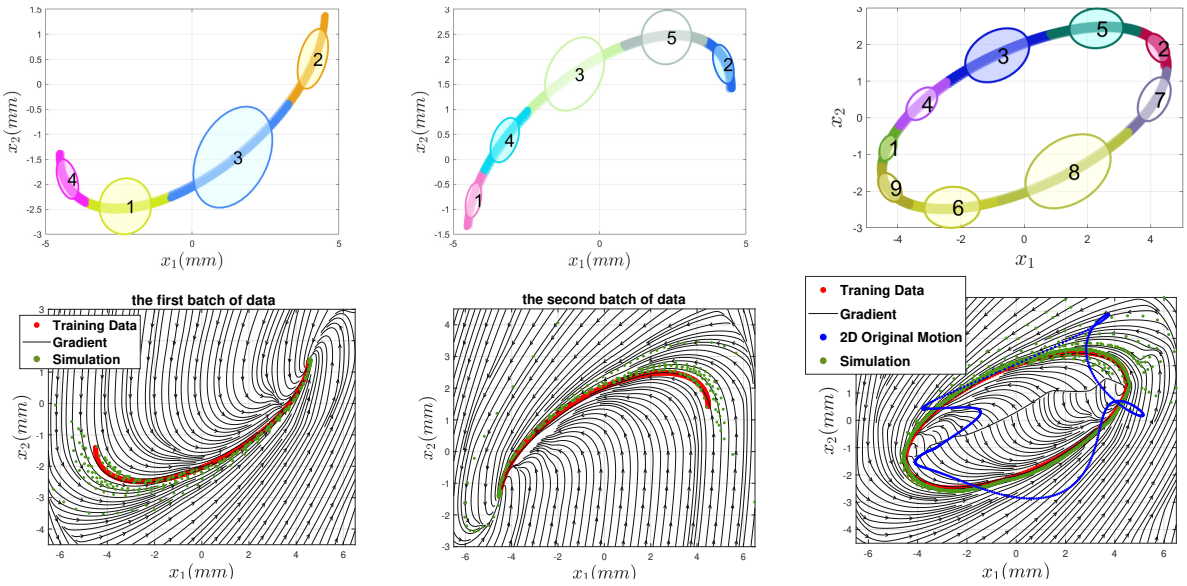


Figure 5.12: The GMM model training process.

In this study, I collaboratively worked with Zhengtong Xu, a visiting student from China on developing a novel soft tissue motion compensation framework for robotic assisted minimally invasive surgery that incorporates (1) primary tissue motion extraction and learning and (2) Model Predictive Control (MPC) [27] for surgical robot end-effector actuation.

First, the primary motion of heartbeat data is extracted through frequency analysis. Then, the Gaussian Mixture Model (GMM) is adopted to fit a spatial gradient map from the primary heart motion data as shown in Figure 5.12. The top row shows the Gaussian

function fit and the bottom row illustrates the learnt dynamic system motion gradient. The three columns from left to right are results from the two data subsets and the combined result. With the GMM based primary motion model, the proposed framework is resilient to tissue motion spanning across different speed, amplitude and on noise level.

Meanwhile, the MPC is formulated to avoid dependencies with robot dynamics, which can be impractical to calculate and error prone for high degree of freedom (DOF) cable-driven surgical robots. An in-depth performance evaluation comparatively with the original tissue motion (simulation) and the motion 10 times slower (RAVEN-II) were analytically conducted. Figure 5.13 illustrates experiments on RAVEN-II [17] using the proposed framework and simple P control. Note that because of the predictive nature of MDP, motion compensation using the proposed framework appears less jittery and follows the reference heartbeat motion more closely. In fact, our approach is able to achieve a RMS error below 1 mm if the reference motion slows down up to 4 times. This work was later submitted to IROS 2020 [208].

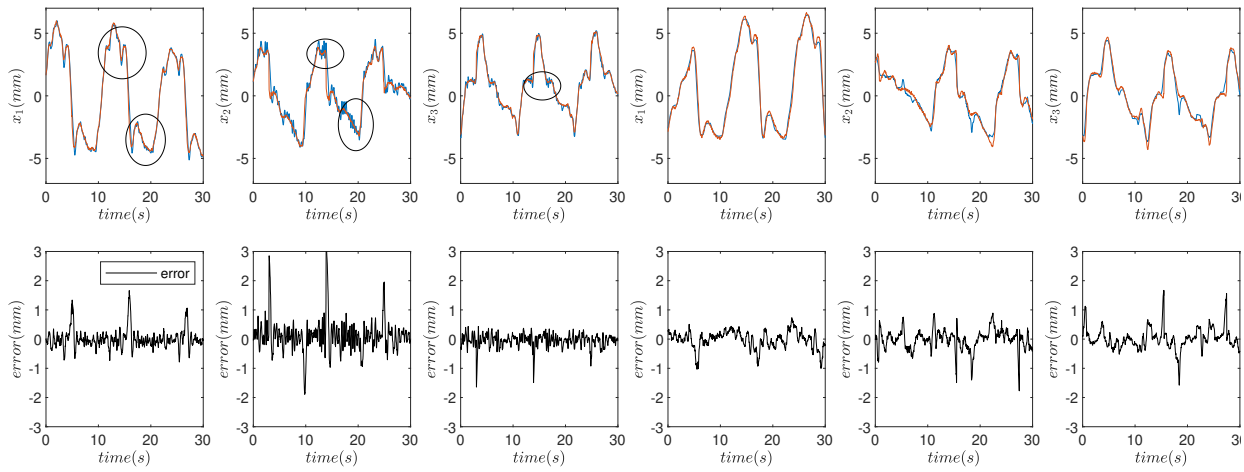


Figure 5.13: The motion compensation result using P control (left three columns) and the proposed algorithm (right three columns) in each of the x_1, x_2, x_3 dimensions across time. The raw tissue motion (orange), the RAVEN-II compensation motion (blue) and the tracking error (black) are illustrated.

5.5.3 Autonomous Multicamera Viewpoint Adjustment

In robot-assisted minimally invasive surgery (RMIS), small keyhole incisions are made in the inflated patient's abdomen. Various robotic surgical tools and laparoscopic optical sensors can then be inserted through these incisions via trocars. Subsequently, real-time vision information from human-positioned laparoscopes informs surgeon teleoperation of the surgical robot system. However, even with experienced human experts in the loop, poor situational awareness due to limited visual and haptic feedback can deteriorate performance. Recent medical robotic research has implications with regard to improving intelligence in RMIS by including augmentations and levels of task autonomy.

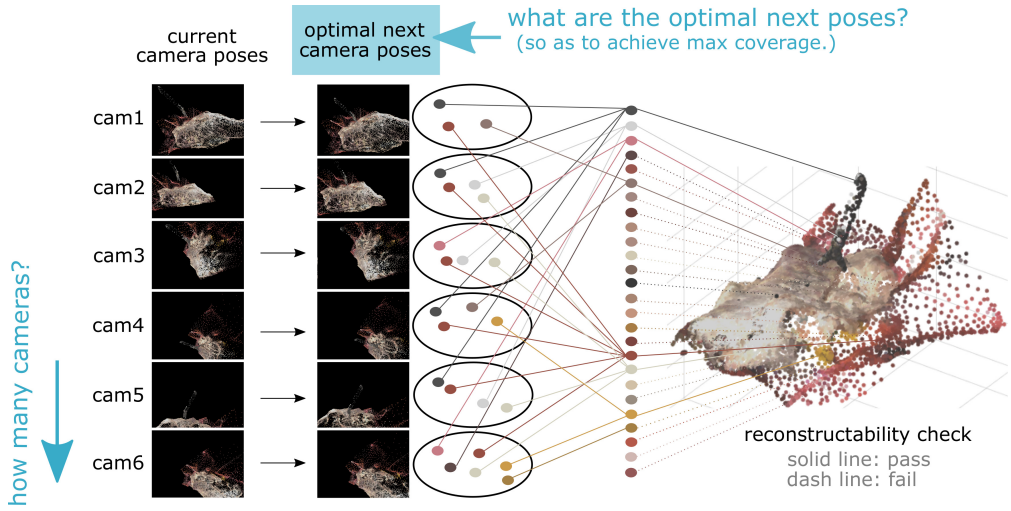


Figure 5.14: Overview of optimal camera viewpoint adjustments formulated as a maximum coverage problem. Each new camera pose is associated with a set of visible points in the 3D surgical cavity model. A point passes the reconstructability check if it is sufficiently visible to at least two camera views. The goal is to find the optimal next camera poses under abdominal wall constraints to achieve maximum number of 3D reconstructable points.

Towards that end, telesurgical visual perception must be addressed. Since manual camera positioning in robotic minimally invasive surgery is suboptimal and error-prone [158], the authors are interested instead in autonomous solutions. Unlike other tool-tracking focused autonomous camera positioning research [192, 223, 238], this work presents a novel

context-aware autonomous multicamera viewpoint adjustment pipeline from the perspective of simultaneously maintaining the surgical tool within view and providing better point coverage for real-time 3D surgical cavity reconstruction.

Motivation

Current robotic laparoscopic systems in RMIS do not function with autonomy. Manual camera positioning is tedious and presents a cost of clinical resources and human attention, and viewpoint control is a key aspect of teleoperation that is conducive for automation [158]. In fact, Urey et al. [211] showed improved surgeon perception of the surgical scene with dense 3D reconstruction from multiple camera viewpoints. Despite these benefits, it is challenging to obtain optimal camera viewpoints through direct human control or verbal communication with a surgical assistant in a dynamic environment while simultaneously reacting effectively to irregular events in surgical scenes – this difficultly is exacerbated with increasing camera number. Thus, instead of encumbering surgeons with the need to both manage cameras and manipulate surgical tools, autonomous robot-controlled camera viewpoint adjustment is a favorable alternative approach and one that is explored in this work.

Related Work

Viewpoint selection is important to several robotic tasks, and may influence navigation, object recognition, environment reconstruction, camera placement and mesh simplification for polygonal models [19]. Salient relevant work include next best view planning and swarm-based mapping as they relate to configuring vision sensors for optimal view coverage.

(1) Next Best View Planners: The next best view (NBV) problem seeks the optimal sensing action to optimize a specified task goal. The NBV strategy or optimal series of view sequences may be task, context, or feature driven, and generally seek the next view to minimize some ambiguity function or incorporate explicit planning algorithms [179]. These active methods perform exploratory sensing and aggregate sensory information to iteratively determine the most suitable sensor view and can be broadly classified into three categories, including surface-based methods, global-based methods, and volumetric methods [1].

Robot kinematic information can be used to collect better views, and may take into account manipulator motion cost as well as view quality [107]. Other objective functions may include reachability, registration constraints related to view overlap, and sensor field of view [215]. These constraints may aid in increasing scan quality while simultaneously reducing required navigation. Dunn et al. developed a hierarchical statistical approach that incoporates camera parameters [61,62], and Potthast et al. used a probabilistic approach for autonomous 3D reconstruction [169]. Scott et al. developed the occluded edges technique, which searches for edges in range imaging data to discover surfaces that have not yet been observed [186, 187]. Multi-phase approaches can reduce the number of viewpoints to be evaluated and also the overall computational complexity of evaluating NBV [172, 214].

(2) Swarm-Based Mapping: Unmanned aerial vehicle (UAV) based swarms allow for distributed processing and have been used for object detection, object tracking, navigation, coordination, environmental monitoring, data collection, and path planning to name a few [43].These approaches require coordination and fusion of multiple vision sensing agents [28, 58, 201]. Optimal path planning for UAV swarms may evaluate computational time against limited sensor and communication capabilities [197], or exhibit adaptive algorithms [50]. Maximizing coverage with swarms of UAVs may also be achieved with simple strategies that leverage individual UAV obstacle avoidance and wall-following with coordinated high-level dispersion from the departure point [137]. Swarm based collaborative simultaneous localization and mapping may achieve decentralized robust, efficient and accurate mapping [242]. UAV swarms are thus amenable to environmental monitoring, mapping and response tasks, including wild fire detection [6] and pollution [153, 183] and water management [31].

While both NBV planning and swarm-based mapping present aspects of optimal vision sensor placement, the application domains of large-scale, openly navigable spaces with rigid objects are not directly extendable to RMIS. Surgical environments, in contrast, are often-times spatially constrained, highly dynamic and deforming, and contain reflective surfaces. These challenges must be addressed for multi-camera optimal view evaluation in RMIS.

Contributions

In this work, the author presents

- a novel autonomous camera viewpoint adjustment framework for RMIS developed as a constrained maximum coverage problem;
- experimental insight regarding the number of cameras in use and their comparative dynamic surgical scene 3D reconstruction performances.

The optimization solution outputs optimally selected next camera poses and the corresponding reconstructability coverage. The two questions in Figure 5.14 succinctly summarize the research aims addressed.

Methods

Autonomous camera viewpoint adjustment was formulated as an optimization problem for achieving maximum coverage [44] of a dynamic 3D surgical cavity model. Due to the highly deforming nature of the surgical scene [123], locations of existing points within the 3D model are frequently updated. Newly observed points update the model, whereas previously observed data are gradually removed from the model if not re-observed - these decay at a heuristically tuned threshold rate. More details of this 3D surgical cavity model update process can be found in [203].

(1) Maximum Coverage Objective Function: Denote the 3D surgical cavity model $\mathbb{P}$ as a set of $N_{\mathbb{P}}$ points, denoted as $\mathbb{P} = \{\vec{p}_1 ... \vec{p}_{N_{\mathbb{P}}}\}$ and $\vec{p}_j \in \mathbb{R}^3$. Due to the fundamental requirements of stereo reconstruction [30], each point is considered visible in the 3D model only when two or more cameras is able to see the point. This requirement is realized by assigning $\forall j = 1...N_{\mathbb{P}}$ a visibility flag $v_j \in [0, 1]$ indicating point $\vec{p}_j$ is viewable if $v_j = 1$, and not viewable otherwise. The maximum coverage problem is then formulated as:

$$\text{Maximize} \sum_{j=1}^{N_{\mathbb{P}}} \mathcal{W}(\vec{p}_j) v_j \tag{5.14}$$

where $\mathcal{W}(\vec{p}_j)$ is a custom weighting function returning a value positively correlated to the

importance of observing point $\vec{p}_j$ during the surgical operation.

Assuming that the surgeon's region of interest is proximal to the surgical tool tissue interaction point, denoted $\vec{q} \in \mathbb{R}^3$ which is obtained through robot kinematics [204], the weighting function $\mathcal{W}$ was defined as

$$\mathcal{W}(\vec{p}_j) = \frac{1}{\max\left(\left\|\vec{p}_j - \vec{q}\right\|_2, \epsilon\right)} \tag{5.15}$$

where $\epsilon = 10^{-3}$, and the range of $\mathcal{W}(\vec{p}_j) \in \mathbb{R}$ is $(0, 1/\epsilon]$.

(2) Constraints on Next Camera Poses: Let the multicamera system contain precisely $N_{\mathbb{C}}$ cameras where $N_{\mathbb{C}} \in \mathbb{N}$ and $N_{\mathbb{C}} \geq 2$. Now denote $\vec{c}_i, \vec{n}_{ik} \in \mathbb{R}^6$ as the current and next candidate camera configurations for camera i respectively, where $i = 1...N_{\mathbb{C}}$, and $k = 1...N_{\mathbb{K}}$ (these are depicted in Figure 5.15). At each time step, each camera i is repositioned and reoriented to one of its next candidate poses under several constraints. If the one-hot representation of the selected next camera pose is $\mathcal{C}_i \in [0, 1]^{N_{\mathbb{K}}}$, then the below constraint holds true.

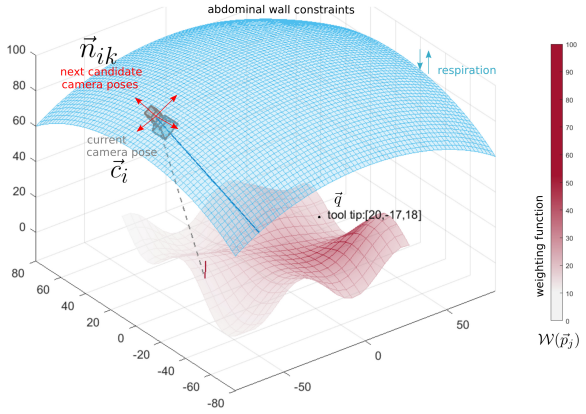


Figure 5.15: Next candidate camera poses $\vec{n}_{ik}$ under abdominal wall constraints. All cameras are mounted on the inside of the abdominal wall via external magnets. Next candidate poses exhibit slight deviations of camera center and orientation from the current pose $\vec{c}_i$. The red colorbar shows the weighting value $\mathcal{W}(\vec{p}_j)$, which is inversely proportional to distance from the tool tip.

$$\left\|\mathcal{C}_i\right\|_1 \quad = \quad \sum_{k=1}^{N_{\mathbb{K}}} \mathcal{C}_i\left(k\right) \leq 1 \ , \ \forall k = 1...N_{\mathbb{K}}, \ \mathcal{C}_i \in \mathcal{C} \tag{5.16}$$

where $\mathcal{C}_i\left(k\right)$ indicates the k^{th} element in $\mathcal{C}_i$ and $\mathcal{C}$ is the set containing all $\{\mathcal{C}_i | i = 1...N_{\mathbb{C}}\}$. Next candidate poses of each camera include the option of remaining at $\vec{c}_i$ or moving to one

of $N_{\mathbb{K}}-1$ other poses. Note that a camera will remain at the current configuration by default if optimal $\mathcal{C}_i^* = [0]^{N_{\mathbb{K}}}$.

To encourage smooth and stable camera pose adjustments, define motion penalty function $\mathcal{M}(i,k) \in \mathbb{R}$, which returns a smaller value when there is less movement or instantaneous velocity change for camera i. This is achieved by setting

$$\begin{aligned} m_1 &= \mathcal{P}(\vec{c}_i) - \mathcal{P}(\vec{n}_{ik}) \\ m_2 &= \mathcal{Q}(\vec{c}_i) \cdot \mathcal{Q}(\vec{n}_{ik}) \\ \mathcal{M}(i,k) &= \|m_1\|_2 + \alpha_1 \ \|m_2 - 1\|_1 \\ &+ \ \alpha_2 \ \left\|m_1 - \hat{m_1^*}\right\|_2 + \alpha_3 \ \left\|m_2 - \hat{m_2^*}\right\|_1 \end{aligned} \tag{5.17}$$

where $\mathcal{P}$, $\mathcal{Q}$ extract from an input vector in $\mathbb{R}^6$ the position and quaternion orientation information respectively. The variables $\hat{m}_1^*$ and $\hat{m}_2^*$ are the m_1, m_2 values obtained from the optimal selected camera pose in the previous time step. Finally, α_1 is a parameter balancing the impact between translational and rotational camera motion, whereas α_2, α_3 are velocity regulation parameters controlling the position and orientation of camera motion respectively.

To regulate jitter in camera motion, a tolerance threshold $T_{\mathcal{M}}$ is introduced. This is used to impose the following constraint

$$\mathcal{L}(\mathcal{C}) = \sum_{i=1}^{N_{\mathbb{C}}} \sum_{k=1}^{N_{\mathbb{K}}} \mathcal{M}(i,k)\, \mathcal{C}_i\,(k) \leq T_{\mathcal{M}} \tag{5.18}$$

(3) Visibility Flag Derivation: The objective function in (5.14) includes a visibility flag, v_j. This flag is determined via visibility function $\mathcal{V}(i,k,\vec{p}_j)$, defined such that

$$\mathcal{V}(i,k,\vec{p}_j) = \begin{cases} 1, & \text{if } \vec{p}_j \text{ is viewable from } \vec{n}_{ik} \\ 0, & \text{if } \vec{p}_j \text{ is not viewable from } \vec{n}_{ik} \end{cases} \tag{5.19}$$

which leads to a subsequent optimization constraint that is imposed as

$$0.5 \sum_{i=1}^{N_{\mathbb{C}}} \sum_{k=1}^{N_{\mathbb{K}}} \mathcal{V}(i,k,\vec{p}_j)\, \mathcal{C}_i\,(k) \geq v_j \quad \forall \vec{p}_j \in \mathbb{P} \tag{5.20}$$

which ensures that v_j is set to 1 only if $\vec{p}_j$ is viewable by at least two cameras. For RMIS, additional considerations are required. In particular, for laparoscopic surgeries a light source is typically operated coaxial with the camera, resulting in severe specular reflection [196] around local patches with angle of reflection $\beta \simeq 0$. Areas with large β appear darker.

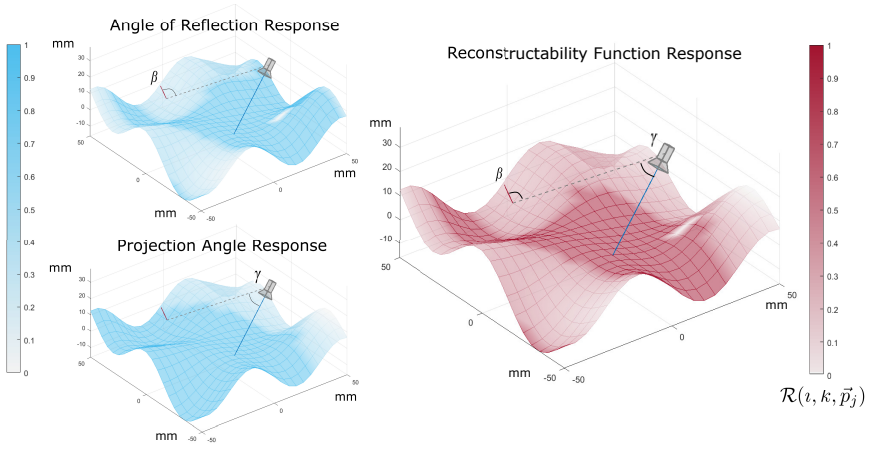


Figure 5.16: Sample reconstructability plot. The red colorbar indicates $\mathcal{R}(i,k,\vec{p}_j)$. Response from angle of reflection β is shown in the top left subplot. The small white dot is a local patch where $\beta < 0.1$, and the reconstructability value is forced to 0. Other smoothly faded patches result from larger β. Bottom left is the projection angle γ response. A point has weaker response when it projects closer to the image border, i.e. γ large.

Furthermore, regions are fuzzier and more more prone to distortion when close to the image border [194]. To account for these considerations, a custom reconstructability function $\mathcal{R}(i,k,\vec{p}_j) \in [0,1]$ is developed to assess a quality score for each point $\vec{p}_j$ from camera viewpoint $\vec{n}_{ik}$, and is calculated as

$$\mathcal{R}(i,k,\vec{p}_j) = \begin{cases} 0.0, & \text{if } \beta < 0.1 \\ \min(\cos(\beta), \cos(\gamma)), & \text{else if } \beta \text{ or } \gamma > 1 \\ 1.0, & \text{else} \end{cases}$$

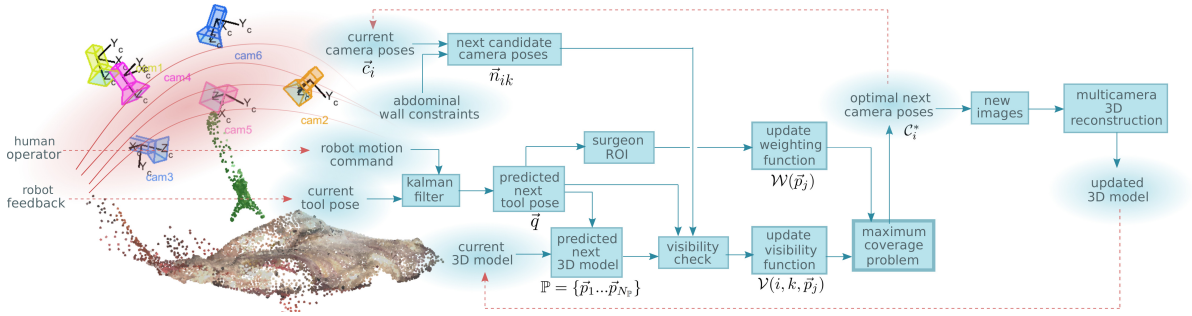


Figure 5.17: One-step look ahead control workflow for autonomous optimal camera viewpoint adjustment. The maximum coverage problem, marked with thicker blue borders, is the main part of this work. The blue arrows in this workflow are updates within the same iteration, and the pink dashed arrows are information passed across subsequent iterations. More details regarding the abdominal wall constraints can be found in Figure 5.15.

where β is the angle of reflection at point $\vec{p}_j$ and γ is the projection angle from camera view $\vec{n}_{ik}$, as depicted in Figure 5.16. The constraint imposed by (5.20) is then reformulated as

$$0.5 \sum_{i=1}^{N_\mathbb{C}} \sum_{k=1}^{N_\mathbb{K}} \mathcal{R}(i,k,\vec{p}_j)\, \mathcal{V}(i,k,\vec{p}_j)\, \mathcal{C}_i\,(k) \geq v_j \quad \forall \vec{p}_j \in \mathbb{P} \tag{5.21}$$

(4) The Optimization Procedure: The previous sections defined objective function and subsequent constraints specific to optimal multicamera configuration for RMIS. With these derivations, the autonomous camera pose adjustment procedure can be formulated as the following budgeted maximum coverage problem:

$$\underset{\mathcal{C}_i, \forall i \in N_\mathbb{C}}{\operatorname{argmax}} \sum_{j=1}^{N_\mathbb{P}} \mathcal{W}(\vec{p}_j) v_j \tag{5.22}$$

subject to constraints (5.16), (5.18), (5.21)

$$\text{where } \mathcal{C}_i \in [0,1]^{N_\mathbb{K}}, \forall i \in N_\mathbb{C} \tag{5.23}$$

$$v_j \in [0,1], \forall j \in N_\mathbb{P} \tag{5.24}$$

Budgeted maximum coverage problems are proven NP-hard [227], even the unit cost version with direct reduction from the set cover problem [2]. Amongst existing approximation approaches [218] [76], the adopted approximate solution achieves an approximation factor of $(1 - 1/e)$ [101], which outperforms alternative options.

The three algorithms below depict the approximation strategy used in this work. The Better Coverage Solver $\mathcal{B}$ compares coverage weight score $\mathcal{S}(h)$ of two input sets of next camera poses h_1 and h_2, then returns the set with greater $\mathcal{S}(h)$.

The Greedy Largest Set $\mathcal{G}$ algorithm returns a set of selected next camera poses derived from a greedy search of increased weight loss ratio. The autonomous camera pose adjustments can be achieved with $\mathcal{G}$, and the approximation factor is $(1 - 1/\sqrt{e})$ [101]. However, for improved approximation factor, algorithm 7 is ultimately adopted.

Algorithm 6 Better Coverage Solver $\mathcal{B}$

input list of camera poses and 3D points $h_1, h_2, \mathbb{P}$
suppose $\mathcal{S}(h)$: returns the $\sum_{j=1}^{N_\mathbb{P}} \mathcal{W}(\vec{p}_j) v_j$ score, with
    - camera poses $\mathcal{C} = h$
    - updated v_j using (5.21)
evaluate $\mathcal{S}(h_1), \mathcal{S}(h_2)$
if $\mathcal{S}(h_1) \geq \mathcal{S}(h_2)$ **then**
    return $h_{\text{Better}} = h_1$
else
    return $h_{\text{Better}} = h_2$
end if

Algorithm 7 Advanced Approximation Strategy $\mathcal{A}$

input list of cameras and 3D points $\mathbb{C}, \mathbb{P}$
set $T_\mathcal{C}$: predefined small set threshold as 3
set $H_1 =$ exhaustively search (5.22) with $\sum_{i=1}^{N_\mathbb{C}} \|\mathcal{C}_i\|_1 < T_\mathcal{C}$
initialize $H_2 = \{\mathcal{C}_i \mid \mathcal{C}_i = [0]^{N_\mathbb{K}}, \forall i \in N_\mathbb{C}\}$
for each $h_2 = \{\mathcal{C}_i \mid \sum_{i=1}^{N_\mathbb{C}} \|\mathcal{C}_i\|_1 = T_\mathcal{C}\}$ **do**
    if (5.16), (5.18) hold true **then**
        $h_2 = \mathcal{G}(h_2, \mathbb{P}, \mathbb{C})$
        $H_2 = \mathcal{B}(h_2, H_2, \mathbb{P})$
    end if
end for
$H_{\text{best}} = \mathcal{B}(H_1, H_2)$
return H_{best}

Algorithm 8 Greedy Largest Set $\mathcal{G}$

input list of camera poses, cameras and 3D points $h, \mathbb{C}, \mathbb{P}$
set $\mathcal{C} = h_{\text{new}} = h$
  E: the explored matrix as $[0]^{N_{\mathbb{C}} \times N_{\mathbb{K}}}$
initialize $E(i,:) = [1]^{1 \times N_{\mathbb{K}}}$ **if** $\|\mathcal{C}_i\|_1 > 0, \forall \mathcal{C}_i \in \mathcal{C}$
while $E \neq [1]^{N_{\mathbb{C}} \times N_{\mathbb{K}}}$ **do**
  find the (i,k) pair in $E(i,k) == 0$ such that $\frac{\mathcal{S}(\mathcal{C}_i(k)=1 \to \mathcal{C}) - \mathcal{S}(h_{\text{new}})}{\mathcal{L}(\mathcal{C}_i(k)=1 \to \mathcal{C}) - \mathcal{L}(h_{\text{new}})}$ is maximized
  if (5.16), (5.18) hold true **then**
    update $E(i,:) = [1]^{1 \times N_{\mathbb{K}}}$
    update $h_{\text{new}} = \mathcal{C}$
  else
    update $E(i,k) = 1$
    reset $\mathcal{C}_i(k) = 0$
  end if
end while
return h_{new}

Finally, the Advanced Approximation Strategy $\mathcal{A}$ entails selecting a small set threshold $T_{\mathcal{C}} \geq 3$. The problem is then divided as

1. exhaustive search of sets with cardinality less than $T_{\mathcal{C}}$
2. comparative greedy algorithm $\mathcal{G}$ with cardinality $T_{\mathcal{C}}$ sets as initial seeds

Optimal results H_1 and H_2 are then compared with the better coverage solver $\mathcal{B}$.

(5) System Update: The workflow for autonomous optimal camera viewpoint adjustment is shown in Figure 5.17. Algorithm 7 is represented as the 'maximum coverage problem' block, with blocks to its left being function and variable settings. Multicamera 3D reconstruction details from the authors' previous work [203, 205] are shown to its right.

Experimental Design

In this work, I am interested in: (1) validating the proposed autonomous multicamera viewpoint adjustment framework using the Advanced Approximation Strategy $\mathcal{A}$ (2) solving for the minimum number of cameras required to maintain a high quality dynamic 3D map. Preliminary experiments show high redundancy with 6 or more cameras and noticeable impact

from occlusion with less than 3 cameras. To that end, three sets of experiments were conducted on the same dynamic surgical scene concerning autonomous multicamera viewpoint adjustment for 3, 4 and 5 cameras respectively. In every set, the proposed approximated viewpoint adjustment method is comparatively evaluated against an exhaustive one-step-look-ahead search in terms of runtime and 3D coverage of the dynamic surgical scene.

(1) Camera Specifications: In these experiments, all cameras were mounted on the inside of the abdominal wall and captured images at a rate of 30 fps with a field of view of 60 degrees. The abdominal wall was a curved 161x161 constantly deforming grid mesh simulating natural motion associated with breathing. Each sample abdomen point was 2mm apart from its adjacent neighbors, and the cameras only existed on the vertices of the abdominal wall mesh. Initially, all cameras were mounted around the center of the surgical scene and oriented straight down. At every time step, a total of 9 possible camera next movements were possible for each camera - moving one sample point in +x,-x,+y or -y direction; tilting 1 degree around +x,-x,+y or -y; and remaining at the current configuration. A motion penalty function $\mathcal{M}$ was defined in (5.17) to impose preference of steady cameras and minimal change in viewpoints.

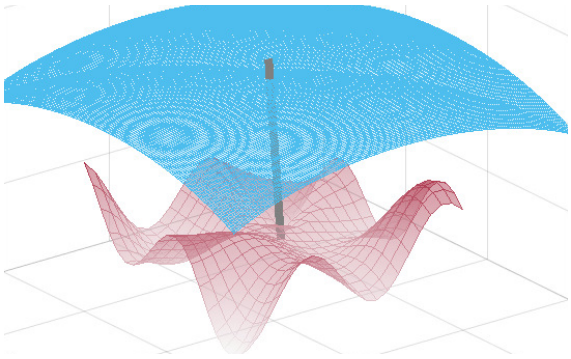


Figure 5.18: Dynamic surgical scene; blue curved surface is the simulated abdominal wall represented by 161x161 grid points slowly warping due to breathing. The gray surgical tool is inserted to palpate the tissue patch at random locations. The pink surface represents the local tissue patch that constantly deforming with a sinusoidal motion pattern and surgical tool tissue contact.

(2) Dynamic Surgical Scene: The tissue surface was represented as a 21x21 sample mesh grid. Each sample tissue point was 10 mm apart from its 4 nearest neighbors and had a timestamped weighting score $\mathcal{W}$ as defined in (5.15) to impose the importance for that particular point to be observed at that time. The dynamic soft tissue patch follows a predetermined sinusoidal motion. Additional tissue deformation occurs when the surgical

tool palpates random locations of the tissue patch. The entire motion sequence is 33.3 seconds, and contains a total of 1000 time iterations (at 30 Hz). The entire simulated surgical environment is illustrated in Figure 5.18. Finally, the objective function described by (5.14) was evaluated in these experiments; namely to find the sequence of optimal poses for each individual camera such that the weighted coverage of the 3D reconstructable tissue points were maximized.

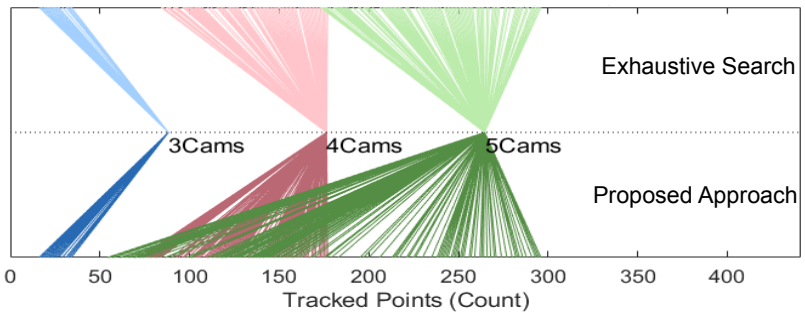


Figure 5.19: Parallel plot depicting the number of tracked points over 1000 iterations using the proposed approximated camera viewpoint adjustment algorithm (lower dark lines) versus with exhaustive search (upper light lines). The horizontal axis is the number of sample tissue points out of 441 that are 3D reconstructable. Each line represents the number of tracked points at one iteration. Results from the three sets of experiments are color coded with blue, red and green respectively for the 3, 4, 5 camera systems.

Observations

(1) Coverage Drop and Recovery: At every time step during the experiments, the number of covered points (3D reconstructable tissue points) were calculated as $\sum_{j=1}^{N_{\mathbb{P}}} v_j$ and are illustrated as color coded lines in Figure 5.19. From this, it is observed that:

1. The instantaneous coverage increases as more cameras are used. Namely, the aggregate number of tracked points increase with increasing number of cameras.
2. The coverage variance across iterations is greater using the proposed approximation method (lower dark lines) compared with exhaustive search (upper light lines).
3. The coverage variance across iterations is greater as more cameras are used. Namely, the tracked points standard deviation increase from 3, to 4, to 5 cameras.

To investigate the effects of the last two observations, the right half of Figure 5.21 shows time variation plots of the tracked point coverage using the proposed method compared with exhaustive search. System robustness is improved by a coverage drop and recovery rule that defines the temporal tracked point count. The temporally tracked point count (in Figure 5.21) differs from the instantaneous number of tracked points (as in Figure 5.19) in that existing points in the 3D map are dropped from the temporal tracked point list only when they are "absent" in the instantaneous tracked point list for 10 successive frames. Similarly, a new 3D point is added to the temporal tracked point list if it repeatedly appears in the instantaneous tracked point list for 10 successive frames.

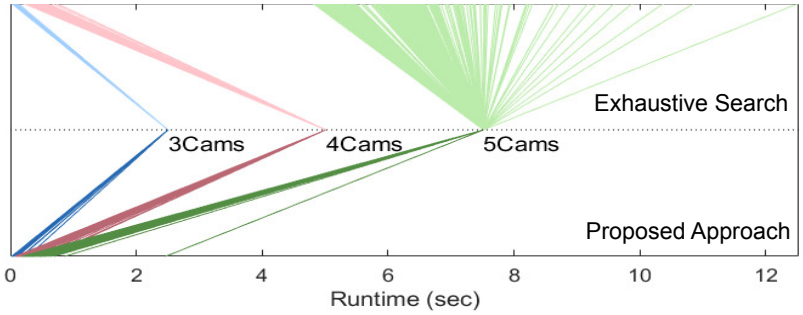


Figure 5.20: Parallel plot showing computational efficiency for each of the 1000 iterations using the proposed camera viewpoint adjustment algorithm (lower dark lines) versus with exhaustive search (upper bright lines). The horizontal axis is runtime for one iteration.

It is observed that tracked points exhibit a vacillating ripple over time. Despite this ripple, coverage remains relatively robust and is able to recover to about 400 tracked points consistently. This suggests that the effects of observations 2 and 3 above may be marginal. Furthermore, coverage oscillation occurs at the same frequency as the periodic sinusoidal tissue motion. Specifically, coverage drops were caused predominantly by the sinusoidal motion near the tissue patch border – critical areas of interest are, in contrast, tracked very robustly, indicating that the weighting function is appropriate. Interestingly, with the proposed method, the effect of increased camera number on coverage drops magnitude is greater as compared to exhaustive search. However, with increasing camera number, the observed coverage drop was distributed across more cameras and with less severity.

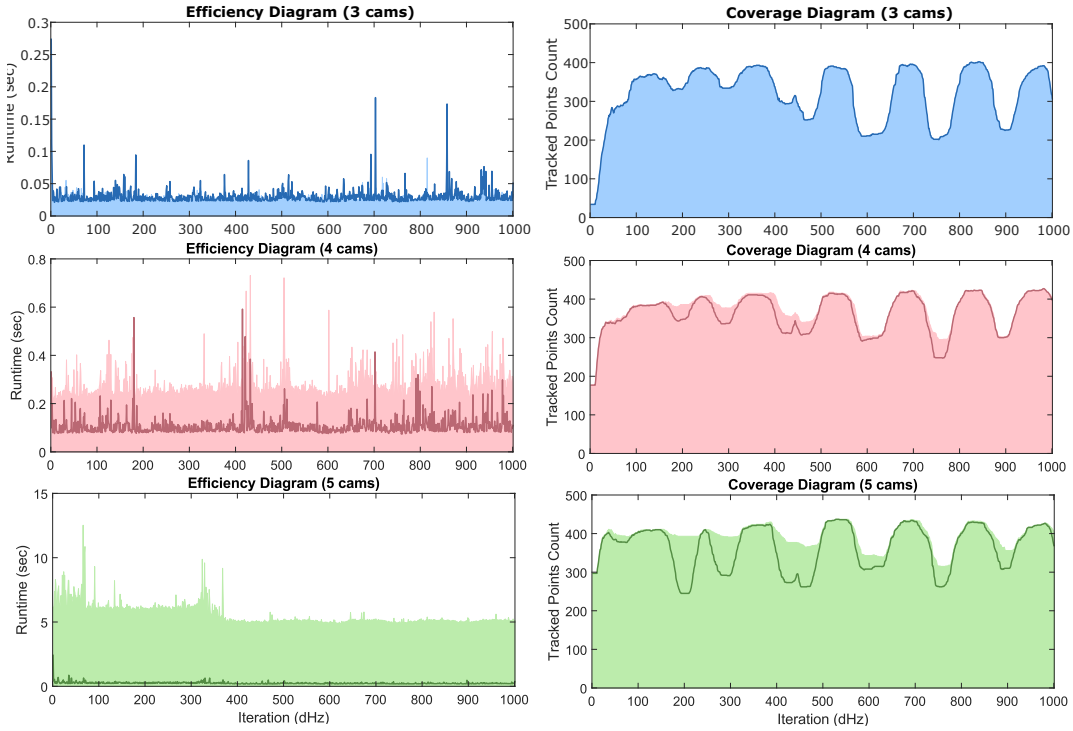


Figure 5.21: Performance plots for the autonomous camera motion adjustment experiments on computational efficiency (left column) and 3D points coverage (right column). The rows correspond to systems with 3, 4 and 5 cameras respectively. Dark lines are for the proposed approximation algorithm $\mathcal{A}$, whereas the shaded area corresponds to exhaustive search.

(2) Computational Efficiency: The computational time efficiency of the proposed method versus exhaustive search is tracked and depicted in Figure 5.20 and the left half of Figure 5.21, illustrated as color coded lines. From these results, one can observe:

1. With increasing number of cameras, runtime improvement from exhaustive search (upper light lines) to the proposed approximation algorithm (lower dark lines) increases.
2. The runtime variance across iterations is greater with the proposed method compared with exhaustive search for fewer cameras (3 or 4).
3. With the proposed method, the best performing trials from each experiment (3, 4 or 5 cameras) performed similarly. This indicates that the computational cost of adding cameras in the proposed method does not increase exponentially with number of cameras. The jump in runtime from 4 to 5 cameras for exhaustive search, in contrast, is stark. It is suspected that greater improvement will be observed with more cameras.

4. Real-time implementation will seek to optimize computational efficiency from the current MATLAB prototype. Yet, runtime results are still indicative of computational efficiency between exhaustive search and the proposed method.

(3) Camera Viewpoint Stability: Qualitatively it was also noticed that with the proposed Advanced Approximation Strategy, $\mathcal{A}$, the average motion penalty value $\mathcal{M}$ for each camera over iterations decreases from 38.88, 36.19, to 31.23 as the camera count increases from 3 to 5, whereas with exhaustive search, the average $\mathcal{M}$ value descends from 38.88, 33.39, to 25.40. This shows that a more stable surgeon view with less individual camera motion can be achieved with more cameras involved.

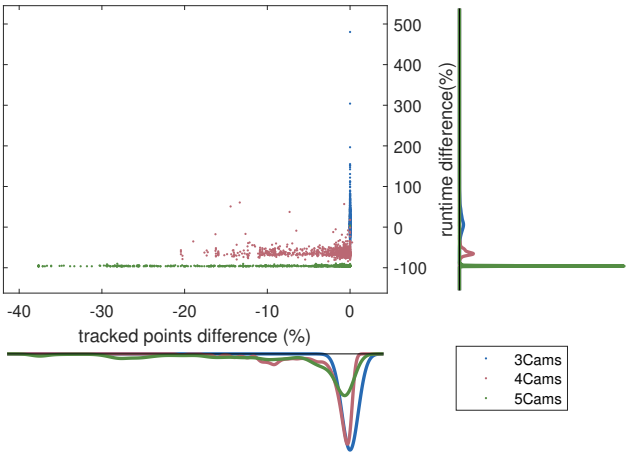


Figure 5.22: Correlation plot of the difference ratio in runtime and coverage between the proposed algorithm and exhaustive search. Every sample point depicts the performance for one iteration. Points are color coded to represent the experiment set, i.e. number of cameras. A difference ratio histogram is calculated for each axis.

Summary

The effectiveness of the novel proposed Advanced Approximation Strategy, $\mathcal{A}$, was assessed in terms of coverage and computational runtime as compared with exhaustive search. Figure 5.22 summarizes the relative improvement from exhaustive search to the proposed method in both metrics. With increasing camera number, runtime is readily decreased with the proposed method at marginal cost in coverage. The proposed approach exhibited stark improvements in computational runtime with increasing number of cameras. The current method relies on a simple one step look ahead approach. Future improvements will seek to reduce the severity of coverage drops due to periodic tissue motion or surgical tool tissue interaction, e.g. palpation. One possible solution is to incorporate Kalman filtering or semantic knowledge of the surgical operation to predict tissue motion.

Chapter 6

HAPTIC FEEDBACK IN ROBOTIC SURGERY

Haptic feedback can render real-time force interactions with computer simulated objects. In several telerobotic applications, it is desired that a haptic simulation reflects a physical task space or interaction accurately. This is particularly true when excessive applied force can result in disastrous consequences, as with the case of robot-assisted minimally invasive surgery (RMIS) and tissue damage. Since force cannot be directly measured in RMIS, non-contact methods are desired. A promising direction of non-contact force estimation involves the primary use of vision sensors to estimate deformation. In Chapter 5, tissue deformation is mapped to contact force by looking up medical datasheets of known tissue properties. However, although the surgical images were captured at 30Hz, due to the computation limitation, the image processing slows it down to roughly 7Hz, whereas haptic feedback update rate is required to be at least 1000Hz for a realistic sensation. Thus, bilateral teleoperation will be feasible only after our code is further optimized for efficiency.

That being said, rather than conducting extensive interpolation with our vision derived force data, and attempting to generate a partially artificial haptic feedback, I put my focus in three aspects of research: (1) a review of existing robot control schemes with both visual and force feedback [119]; (2) the accuracy requirement in haptic feedback for surgeons to have satisfatory performance on a palpation task on a tissue membrane [42]; (3) the possibility of incorporating additional message and warning such as robot joint limits into haptic feedback during RMIS [89]. These endeavors will lay the groundwork for ultimately realizing bilateral teleoperation using the vision-based force estimates in RMIS. This chapter summarizes my findings in (1) and presents (2),(3) in **section 6.5**.

6.1 Visual Guidance

6.1.1 Background

Visual guidance uses feedback information extracted from vision sensors to enhance robot motion [92]. Typically these sensors are cameras. Prior to widespread visual guidance in robotics, robot tasks relied heavily on rigid, repeatable setups and precise control of end-effector position. Moreover, accurate a priori geometries were required to properly acquire objects. These precision positioning systems were expensive, often exceeding the cost of the robot. To reduce overall costs and improve performance, visual guided robots largely replaced precision fixturing [78].

Feddema et al. proposed a method of generating robot trajectories using image feature velocities with known 3D geometrics of the interaction object [67]. Espiau et al. outlined issues in visual guidance, including modeling interaction matrices and visual feature extraction [64]. Corke posed an additional set of critical questions, particularly regarding the dynamics of visual servoing [48]. Problems addressed include latency and stability in feed-forward control paths. In [47, 160], an adaptive visual guidance system was developed to provide a confidence metric and a stochastic controller with Kalman filtering.

6.1.2 Technical Approach

The main objective of vision-based control is to minimize the error $e_s(t) = s(t) - s_D(t)$, where $s(t)$ are visual features and $s_D(t)$ the desired visual feature location [36]. The general approach for reducing $e_s(t)$ is accomplished through a velocity controller, which requires a linear transformation from time variation of s to robot end-effector velocity $\dot{x}$, namely

$$\dot{s}(t) = \dot{e_s}(t) = L_e \, \dot{x}(t) \tag{6.1}$$

where L_e is a $2k \times 6$ matrix dependent on image feature s, and k is the number of feature points. Suppose the robot end-effector velocity $\dot{x}(t)$ is the input to the system, then to reduce

the error $e_s(t)$, consider

$$\dot{x}(t) = -k_v \, \widehat{L_e^{\dagger}} \dot{e_s}(t) = -k_v \, \widehat{L_e^{\dagger}} \, (s(t) - s_D(t)) \tag{6.2}$$

where † represents the Moore-Penrose pseudo inverse operator and k_v is the controller gain. A real-world value of L_e is oftentimes difficult to obtain, and so an approximation denoted as $\widehat{L_e}$ is commonly used instead.

Now define the visual controller as the following

$$U_v(t) = -k_v \, S_v \, \widehat{L_e^{\dagger}} \, e_s(t) \tag{6.3}$$

where S_v is a selection matrix for visual controller activation direction. The motivation for the S_v term is explained in detail in a later Section. This control scheme, $U_v(t)$, allows the system to simultaneously achieve minimal $\left|\left|\dot{e_s}(t) - k_v \, \widehat{L_e} \, \widehat{L_e^{\dagger}} \, e_s(t)\right|\right|_2$ and $||x(t)||_2$.

Visual guidance can be broadly classified by three characteristics, including sensor location, dimensionality and command type. The remainder of this section will elaborate on the distinctions between the different classes.

Sensor Location

Visual guidance systems can be classified into two types by the location of the visual sensing device in the system [90]. The two kinds are

1. eye-in-hand
2. eye-to-hand

In eye-in-hand configurations, the camera is attached to the moving robotic manipulator and the main purpose is to observe the relative position of the target. Usually, the camera is mounted on the end effector or adjacent link. Conversely, in eye-to-hand configurations, the camera is fixed in the world and observing both the target and the robotic manipulator. Figure 6.1 depicts the differences between two configurations.

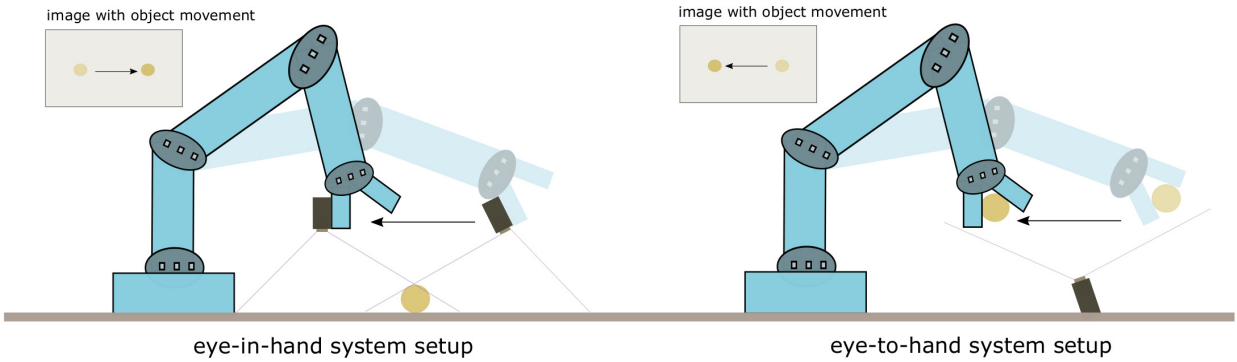


Figure 6.1: Illustration of an eye-in-hand and an eye-to-hand visual guidance system. With the same robot motion, the perceived object motion in the image plane is opposite for the eye-in-hand and eye-to-hand configurations.

Equation 6.1.2 assumes the eye-in-hand configuration. The corresponding equation for eye-to-hand configuration differs only by a negative sign. More derivations can be found in [36]. Specific technical details differentiating the two categories can be found in [35].

Dimensionality

Visual guidance approaches can also be classified by how vision sensor information is interpreted and utilized. At the lowest level of spatial dimensionality, image data directly governs robot motion. Alternatively, higher dimensionality information can be inferred from the combination of image data and a priori knowledge [34]. Broadly speaking, this distinction classifies visual guidance techniques as

1. Image Based Visual Servoing (IBVS)
2. Position Based Visual Servoing (PBVS)
3. Hybrid approaches

Figure 6.2 illustrates the system level differences between IBVS and PBVS classes. IBVS involves using information from the image to directly control robot motion. In PBVS, geometric interpretations of the image are derived, such as estimating the target pose and parameters of the camera. Hybrid approaches combine aspects of the two approaches [34,90].

Image Based Visual Servoing Weiss and Sanderson first proposed a control law based only on the error between current and desired features on the image plane [182] - the pose of the target is irrelevant. The visual features, $s(t)$, are normalized 2D coordinates of feature points, lines or moments of regions on the image plane. Suppose a feature point is represented in the 3D camera coordinates at time t as $(X(t), Y(t), Z(t))$. Then the visual feature is described as $s(t) = [s_1(t)\, s_2(t)]^T = \left[\frac{X(t)}{Z(t)}\ \frac{Y(t)}{Z(t)}\right]^T$. Differentiating with respect to time and expressing in matrix form results in

$$\dot{s} = \begin{bmatrix} \dot{s_1} \\ \dot{s_2} \end{bmatrix} = \begin{bmatrix} \frac{\dot{X}}{Z} - \frac{X \dot{Z}}{Z^2} \\ \frac{\dot{Y}}{Z} - \frac{Y\,\dot{Z}}{Z^2} \end{bmatrix} \tag{6.4}$$

The time derivative of $(X(t), Y(t), Z(t))$ can be expressed as

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = -\dot{x} - \dot{\omega} \times \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{6.5}$$

where $\dot{x}$ and $\dot{\omega}$ are the linear and angular velocities of the robot end effector respectively. This results in

$$\dot{s} = L_e\, \dot{x} \tag{6.6}$$

where

$$L_e = \begin{bmatrix} -\frac{1}{Z} & 0 & \frac{s_1}{Z} & s_1\, s_2 & -(1 + s_1) & s_2 \\ 0 & -\frac{1}{Z} & \frac{s_2}{Z} & 1 + s_2^2 & -s_1\, s_2 & -s_1 \end{bmatrix}$$

The controller input is thus

$$U_v(t) = -k_v\, S_v\, \widehat{L_e^{\dagger}}\, e_s(t) \tag{6.7}$$

where

$$\widehat{L_e^{\dagger}} = \left(L_e^T\, L_e\right)^{-1}\, L_e^T$$

and in general

$$\widehat{L_e^{\dagger}} = \left(L_e^T\, Q\, L_e + W\right)^{-1}\, L_e^T\, Q$$

with weighting matrices Q and W to specify costs of feature error and control input.

IBVF methods tend to struggle with large rotations [33], a symptom known as camera retreat [49]. The visual guidance optimum may converge to only a local minimum, and sometimes the visual system reaches a Jacobian singularity [33]. IBVF is considered visual guidance in 2D, since computations are all performed on the 2D image plane.

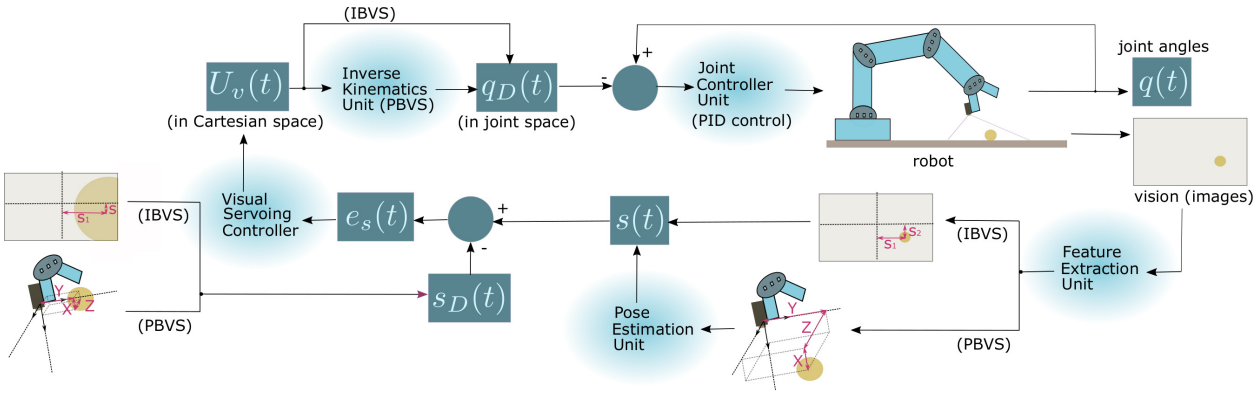


Figure 6.2: Illustration of an Image Based and a Position Based visual servoing system.

Position Based Visual Servoing PBVS is a model-based technique by which pose of the object of interest is estimated with respect to the camera which in turn issues a command to the robot controller. The visual features s are defined as the translation and rotation of the 3D coordinates in the camera frame $[s_1 \;\; s_2 \;\; s_3 \;\; s_{\theta a_1} \;\; s_{\theta a_2} \;\; s_{\theta a_3}]^T$, where θa gives the angle-axis representation of the rotation. Suppose desired feature pose $s_D = [s_{D1} \;\; s_{D2} \;\; s_{D3} \;\; 0 \;\; 0 \;\; 0]^T$. The interaction matrix L_e then becomes

$$Le = \begin{bmatrix} I_{3\times 3} & \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix}_{\times} \\ 0 & L_{\theta a} \end{bmatrix} \tag{6.8}$$

where

$$L_{\theta a} = I_{3\times 3} - \frac{\theta}{2}[a]_\times + \left(1 - \frac{\mathrm{sinc}(\theta)}{\mathrm{sinc}^2\left(\frac{\theta}{2}\right)}\right)[a]_\times^2$$

The pseodo-inverse of the approximated interaction matrix is given as

$$\widehat{Le^\dagger} = \widehat{Le^{-1}} = \begin{bmatrix} -I_{3\times 3} & \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix}_\times L_{\theta a}^{-1} \\ 0 & L_{\theta a}^{-1} \end{bmatrix} \tag{6.9}$$

In this case image features are extracted and used to estimate 3D pose of the target object in Cartesian space. Visual guidance is thus conducted in 3D space.

Hybrid Approaches Hybrid approaches employ some combination of 2D and 3D methods. Some more pertinent hybrid methods include (1) 2.5D Visual Servoing [49, 133], (2) Motion partition-based methods and (3) Partitioned DOF Based Visual Servoing [49]. 2.5D Visual Servoing developed by Malis [133] decouples rotations and translations. Assuming the desired pose is known, rotational information is obtained from partial pose estimates from the homography between camera frame and absolute frame — axis and angle are given by eigenvalues and eigenvectors. Translational control is achieved directly by tracking feature points so long as feature points never leave the field of view and a depth estimate is predetermined off-line. 2.5D servoing is more stable than techniques that preceded it. [45] outlines another hybrid approach whereby visual guidance is split into two parts: (1) keeping the features within the field of view and (2) marking a fixation point as a reference to bring the camera to the desired pose. Without prior knowledge of a depth estimate from an off-line procedure, the depth estimates are obtained from robot odometry and by assuming all features are on a plane.

Command Type

Command type is determined by whether commanded motion is applied at the joint level or as end effector configuration. These two classes are (1) Direct Visual Servoing and (2) Dynamic Look-and-Move.

Distinctly, Direct Visual Servoing algorithms require robot kinematics. A hierarchical scheme was proposed in [225]. This method relied on reliable features of the target object, used as a partial model along with global models of scene and robot.

6.2 Force Control

6.2.1 Background

Force control techniques were developed to assist in automated manufacturing processes [136]. Most manufacturing processes consist of assembly parts and precise contacts between different components of an object. While the parts often meet the dimensional specifications , additional processing are still required to achieve a desired finish. These finishing operations are improved with force control as compared to a dimension driven position-controlled manufacturing process. This can be especially true when assembling delicate parts and physical damage is a real concern. In these cases, joint torques are controlled to match the desired force applied by the end-effector to an external object. For this, a method for external force measurement is required. The most straightforward technique involves direct measurement via attached force sensors at the robot end-effector. Force sensors are often comprised of strain gages, which are able to measure six axis force/torque at the end-effector.

In many real world applications, robots need to follow trajectories while interaction forces may be a constrained in other directions. Force control is oftentimes hybrid force/position control for which the joint torques are computed using two references [173]. For this, the control scheme must be dynamically switched between two operations [100, 237]. Human touch exhibits a form of compliance, a property that position-based machine tools lack.

Compliance in the context of surface finishing is the ability to compensate for mismatch between tool and part surface. This requires maintaining contact rather than position, and thus controlling the amount of applied force. Force control aims to extend robot capabilities in this regard, and allows automated equipment to maintain consistency with contact [226].

6.2.2 Technical Approach

Fundamental to force control is to maintain desired physical contact between the robot device and the environment particular to the manipulation task. Pure motion control proves inadequate due to unavoidable modeling errors and uncertainties, resulting in undesired contact forces or lack thereof, ultimately leading to unstable and unintended behavior during the interaction. This is particularly prevalent in rigid environments [217]. The force controller can be generalized by the following equation

$$U_F(t) = -K_F \, S_F \, (F(t) - F_D(t)) \tag{6.10}$$

where $F(t)$ and $F_D(t)$ are the measured and desired force values, K_F the controller gain and S_F a selection matrix for force controller activation direction.

Force control can be broadly classified by three characteristics, including sensor location, compactness, and control strategy. The remainder of this section will elaborate on the distinctions between the different classes. Mathematical details are found in [217].

Compactness

Compactness of the force controller robotic manipulator system also distinguishes force control techniques. Two commercially accepted methods of force control are: (1) through-the-arm ;(2) around-the-arm. In the former, forces are applied considering all robot axes in unison. The latter focuses on the use of an auxiliary compliant end-of-arm tool to apply forces, while the robtic arm is used for positioning only.

Through-the-arm force control is generally deemed the more elegant approach, as it presents a more natural extension of standard robot control technology. Furthermore, through-the-arm mimics human object manipulation, whereby the manipulation appendage is responsible for both movement in positioning and providing force. Through-the-arm approaches suffer when a stiff, non-compliant media is encountered. The position of this type of media relative to the part surface is critical, since the overall system stiffness increases. In the media mismatch problem, robot manipulators are unable to respond quickly enough to attenuate or eliminate force errors.

Around-the-arm force control decouples force control from the robot controller and uses the robot arm for positioning only. Figure 6.3 shows the high-level difference between through-the-arm and around-the-arm force control systems.

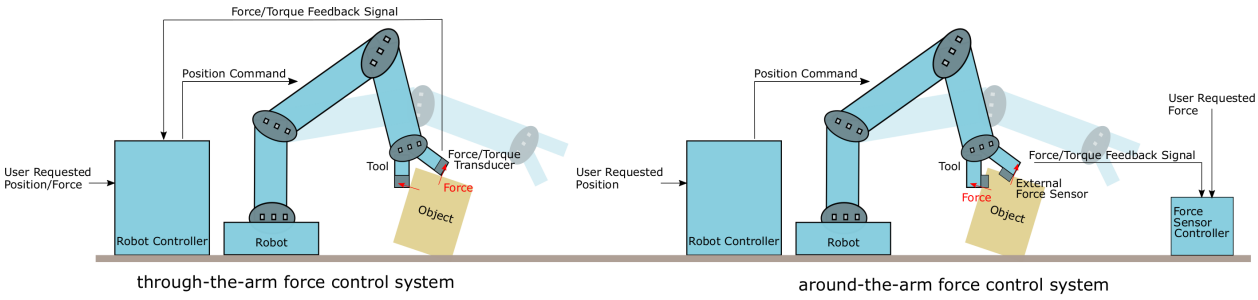


Figure 6.3: Comparison of a through-the-arm and around-the-arm force control system.

Control Strategy

Pneumatic force control methods can be divided into either passive or active methods [217] Passive control involves an open loop control system with no means to adjust for force errors. In contrast, active force control utilizes a controller or regulator to manage a closed-loop system that continuously monitors the applied force and corrects for the errors. This guarantees a much more accurate system than passive force control systems. These two classes can be further described by subcategories, as shown in Figure 6.4.

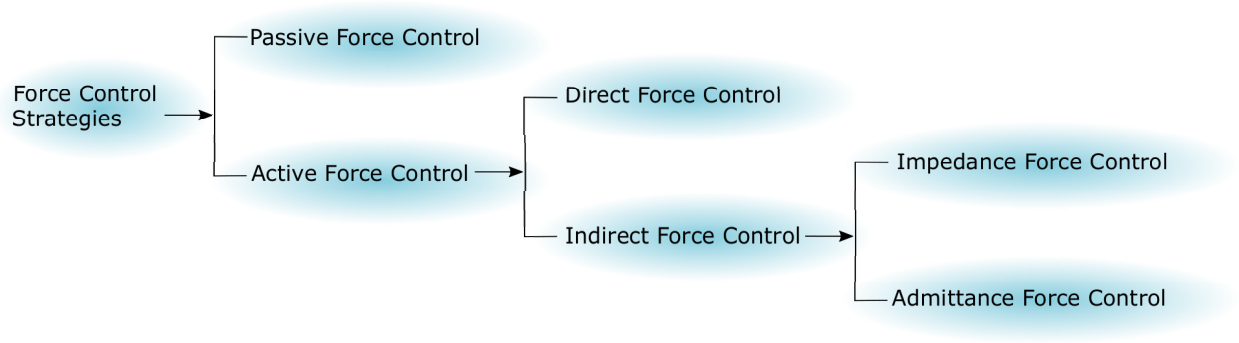


Figure 6.4: Force control strategy classification.

Passive Force Control The use of passive force controllers are typical of robots with special-purpose compliant parts or end-effectors. These parts are compliant with the contact surface during interaction with the environment [66]. Soft robots are often amenable to this type of robot controller; its deformable links are flexible and permit adaptations without breakages. Passive controllers do not require force sensors. Furthermore, with passive controllers there is no guarantee of avoiding high instantaneous contact force. Should the robotic task not tolerate high instantaneous forces, active force controller schemes are suggested.

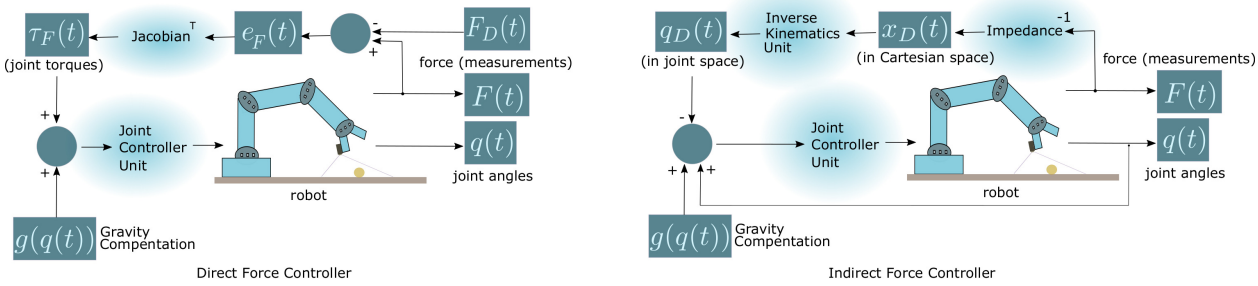


Figure 6.5: Direct vs. indirect active force controllers.

Active Force Control The active force controllers can be divided into two subsets - direct and indirect force controllers. In both cases, a force sensor is required in the system, but the use of the force measurements distinguishes the two. In direct force control, the force is regulated, i.e. there is a desired contact force for the robot to follow. In other words, the interaction force is directly being adjusted via a force control loop based on the error between

measured and desired values. In contrast, for indirect force control, the force measurements are used to regulate robot motion. Here a motion control loop that uses force information guides the motion of the robot manipulator. Figure 6.5 conveys the high level distinction between indirect and direct active control schemes.

Within indirect force control, two main classes emerge: impedance control and admittance control [85]. The overall distinction is depicted graphically in Figure 6.6.

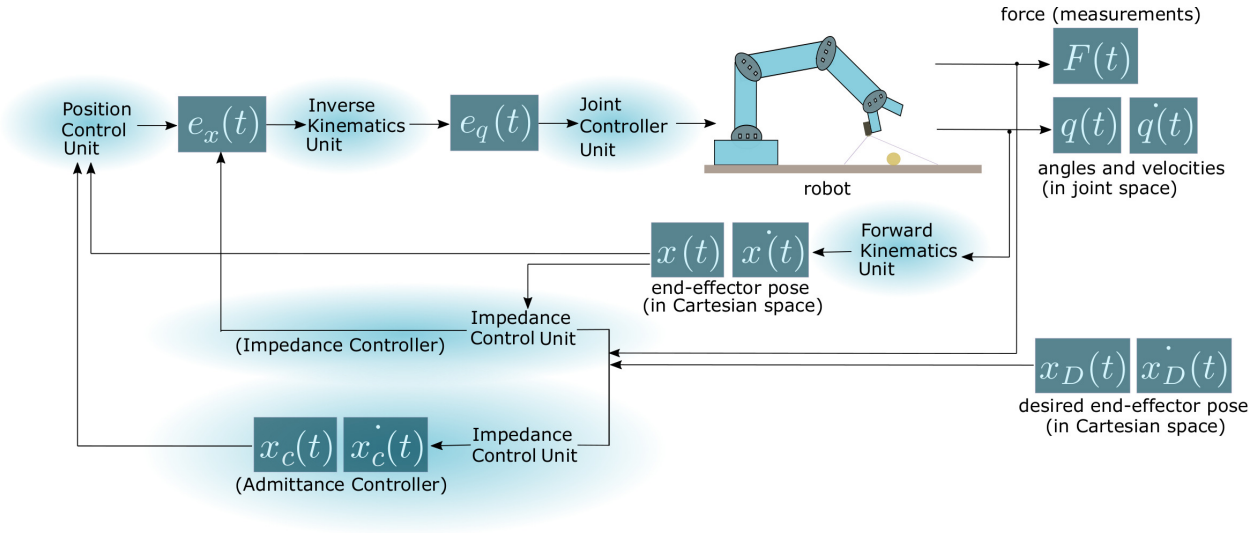


Figure 6.6: A comparison of an impedance and admittance force controller.

Impedance controllers control the dynamic relationship between exerted force and the robot movement error, also known as the mechanical impedance of the robot. This controller generates forces to negate forces involved in motion deviation, and is known to provide better performance with softer movements [114].

On the other hand, admittance controllers generate deviations from desired motion, and predicts the robot will move in the desired motion following the external force. This control method is also called position-based force control, and the desired behavior is implemented using an outer loop surrounding a position or velocity control loop. [69] outlines an admittance controller where the force controller is operated in joint space while force distribution and contact surface orientation are computed online to obtain input force.

6.3 Integrated Vision and Force Control

6.3.1 Resolvability in Sensor Fusion

Sensor fusion within one task dimension require comparison of feedback characteristics from each sensor. Vision and force resolvability [151] evaluates the extraction of useful information from both during manipulation tasks. Information provided from disparate sensors can be assimilated by monitoring resolvability.

Vision resolvability [150] measures the ability to determine object positions and orientations. A typical single camera system can more accurately resolve object locations in a plane parallel to the image plane. Rotations within image plane parallel planes are more accurately resolved than perpendicular ones.

Force resolvability depends both on the force sensor and stiffness of the entire system. This extension of resolvability provides a common measure for both sensors in evaluating when visual servoing or force servoing strategies are appropriate. A nonlinear force/visual servoing algorithm that uses force and vision resolvability to switch between sensing modes demonstrates the advantages of assimilation techniques [10, 168].

6.3.2 Recent Application Spaces

The application space of combined vision and force control spans a wide spectrum. Control strategies are tailored for specific robot-assisted or automation tasks. This section provides an overview of several modern applications.

Haptic Simulation

Basic haptic display involves a purely virtual environment. The subject operates a haptic joystick to send motion commands and receive simulated contact forces [224]. Simultaneously the resultant interactions are reflected visually. As an example, virtual reality micro-robotic

cell injection training uses haptic guidance virtual fixtures [65]. The task exists on the order of micrometers. Without magnified vision-based force feedback the cell can easily break. Virtual training systems combining visual and force feedback can be decomposed into subtasks [128]. In [127], decoupled motion control was implemented whereby simulated force feedback did not directly correct motion deviation. Instead, the trainee regulated motion based on cooperative motion feedback from an expert.

Kimmer et. al showed that mental rotations reduced teleoperation performance despite direct force feedback [104]. Other studies verify that haptic feedback does not reduce user control [79]. In fact, force feedback improves perception in virtual training systems [15]. [93] describes haptic bilateral control with vision-based guidance, where visual information translates to assistive force, while a visual compliance controller relies on force control.

Industrial Robotic Tasks

Robotic manufacturing tasks including painting, material handling, and welding are all non-contact processes. Recent advances incorporate force and vision and permit a new industrial applications [78]. Sensor-based robot controllers lead to better robot performance, lower cost, and new applications [24]. Pomares et. al developed a trajectory tracking system that fused visual and force information with variable weights [168]. Movement flow-based visual guidance and Kalman filters processed detected interaction forces. There is strong interest in combining force and vision for autonomous manipulation in unknown environments [10, 87, 232]. [232] presents a learning control approach. By mapping image features to joints, visual guidance and force servoing are implemented adopting an impedance control law. In [10], both vision and force sensors are mounted on the manipulator. The visual sensor oversees force probe positioning while the force sensor acquires contact forces with a feedforward/fused control. In [87], the robot is visually guided while maintaining a desired contact force using an adaptive control. The task is separated into two subtasks to ensure the end effector force converges to the desired value and image features converge to desired trajectories.

Sometimes the object contact plane is given. Studies like [156] and [171] determine orientation and position of uncalibrated planes. Combined visual and force guidance with unknown planar surfaces is achieved whereby the robot maintains contact forces while marking surface lines [156]. The controller consisted of a force feedback control loop and a vision-based trajectory controller as feed-forward signal. In [139] an electrode wearable bio-signal device controlled a 6DOF manipulator with visual and force sensors attached. Visual and force guidance are also used in autonomous wheel assembly, with wheel loading completed with repeatability less than 2mm [38].

In addition to autonomous manufacturing robots, human-robot collaborative assembly is another form of robot-assisted manufacturing. Methods for humans to work safely alongside robots during assembly were developed [41]. Vision tracks as the operator approaches proximity of the end-effector and the robot stops to allow human operation on the part.

Medical Applications

Robot Assisted Surgery Chatelain et al. optimized ultrasound image quality via visual guidance of robotic ultrasound probes [32]. Both visual and force sensors were attached to the tip of the probe. Patlan-Rosales et. al conducted research studies utilizing ultrasonic and force sensory information to improve robot assisted surgery. In [162], elastography modality and force measurement are inputs to a palpation system controller which achieves robot probe control. The same group developed a robotic control framework for 3D quantitative ultrasound elastography in the context of visual guidance in autonomous palpation [163]. The probe moves in a compression motion with applied force control. Automatic motion compensation by ultrasound visual guidance can estimate strain of moving tissue [164]. Image-based visual guidance, force control and non-rigid motion estimation are achieved given elastic properties of the moving tissue. Robot-assisted laser microsurgery can also benefit from vision and force fusion, whereby fictitious force feedback is created through stereoscopic visualization [154]. Von Sternberg et. al implemented a force-feedback interface for robotic assisted

interventions with real-time MRI guidance [219]. A framework for integrating real-time MRI with robot control in simulated transapical cardiac interventions is presented in [148]. The system provides both visual and force feedback of a simulated transapical aortic valve implantation with virtual robotic manipulator. Result show improved completion time with the addition of force feedback. Blood cell characterization for robot manipulated manipulation is described in [200]. Optical robotic-tweezers manipulate cells, while force calibration and image processing correlates stretch force and deformation.

Rehabilitation A field force controller adapting pelvic motion was designed to provide 3D feedback for stroke rehabilitation [97]. This visually guided robotic system provided guidance force to the user's pelvis. A dissipative haptic display using brake actuated manipulators performed path guidance for rehabilitation [57]. Three controllers were compared – velocity ratio, force cancelling, and force mapping. A haptic feedback, skin stroking glove for the visually impaired was developed to assist navigation [5]. Kinect motion sensing and vibrotactile feedback were combined. Li et al. invented an intelligent wheelchair with path planning and obstacle avoidance using local environment modeling [118]. Visual force field arithmetic mediated target guidance, while a graph method modeled the environment.

6.3.3 Challenges

Combining global information registered by visual sensors with local information registered by force sensors enables thorough task space knowledge and affords better understanding of the object with which the robot interacts. The two information modalities are complementary and essential for complex robot manipulation tasks. The importance is apparent, but general implementation remains challenging.

Data Characteristics

Force measurements are expected on orders of several hundred hertz (up to 1000Hz [4]), while typical vision sensors acquire data at 30-60Hz [63]. Force sensors focus locally near

end-effector and surface contact, whereas visual sensors provide a relatively global view. The bandwidth for data acquisition and measurement space illustrate the innately different characteristics between the two sensor data types.

Sensory Interference

Oftentimes end-effector mass causes inertial coupling effects when visual guidance directs robot motion. This introduces inertial force sensor readings and unstable excitations [152]. To avoid conflicting robot commands from vision and force information, control schemes can be delicately designed. The common goals are to define the dominate sensory data to contextualize and prioritize controllers.

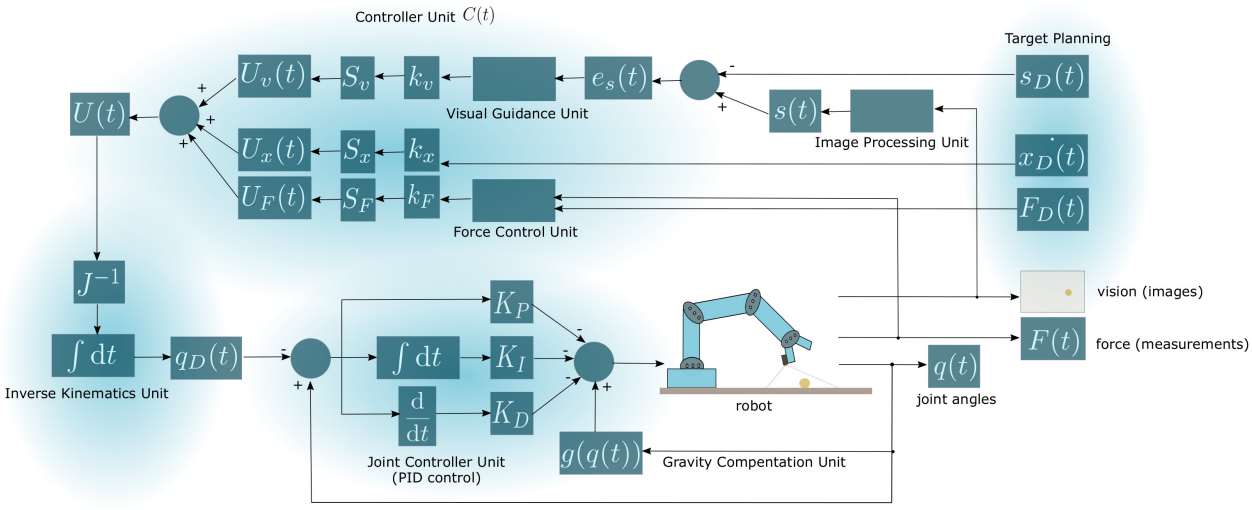


Figure 6.7: Generalized force/vision controller with inner loop PID controllers and gravity compensation.

6.3.4 Technical Approaches

Force and vision sensors provide complementary information; while visual sensors capture the 3D global environment, force sensors collect local information at the interaction site [10]. The two are fundamentally different sensing modalities for which traditional fusion control methods are insufficient – sophisticated control strategies are required [152].

Many approaches up to now rely on hybrid control [173]. Examples include studies like [12] and [11], which extend 'task frame' formalism [26, 55]. Hybrid control strategies are also employed in [87] and [230] for tracking unknown surfaces. External cameras and end effector force sensors are used for real-time grasping in [147]. Several approaches instead rely on impedance control, such as [141], in which an external visual controller loop generates references for the impedance control system. In [151], force and visual sensors act at identical control hierarchy levels. In [210], virtual forces are applied before contact. In [134], different sensor systems were combined employing the task function approach.

6.3.5 Classification of Control Methods

One major benefit of combining vision and force is the ability to approach rigid surfaces at high velocities and initiate stable contact with low impact forces and no bounce. It is proven that force control is crucial when operating visually guided robots in complex environments. However, traditional sensory fusion methods are not amenable [141].

The common goal is prioritizing the dominant data source for a given context. To that end, control strategies can be categorized into three types [152] – traded control, hybrid control and shared control. Each takes a different approach to separate conflicting commands.

The traded controller separates dominant sensor data by time, delineating robot manipulation task into subtasks. Sensor priorities are predefined for each subtask. The hybrid controller classifies dominant sensor by geometry. Robot motion commands generated from vision and force commands are orthogonal. In this way, the two sensors control the robot in different directions. The last control scheme is shared control, which allows both vision and force to induce motion commands at all times and in all directions. Figure 6.7 depicts a general force/visual controller with inner PID controllers and gravity compensation.

Traded Control

In traded control and hybrid control, vision and force information are used separately depending on how the task is divided. For traded control, the task division is temporally based.

The Cartesian euclidean distance $e_x(t)$ between current pose $x(t)$ and target pose $x_D(t)$ is compared with the predefined threshold ϵ. If $\|e_x(t)\|_2 > \epsilon$, only visual guidance is applied while force information is used for monitoring. In contrast, for $\|e_x(t)\|_2 \leq \epsilon$, visual guidance is disabled and only force control is applied. The control law shown in Figure 6.7 as $C(t)$ is:

$$U(t) = \begin{cases} U_v(t) = -k_v \, S_v \, \widehat{L_e^{\dagger}} \, e_s(t), & \text{if } \|e_x(t)\| > \epsilon. \\ U_F(t) = -K_F \, S_F \, (F(t) - F_D(t)), & \text{else.} \end{cases} \tag{6.11}$$

$$\begin{aligned} e_s(t) &= s(t) - s_D(t) \\ e_x(t) &= x(t) - x_D(t) \end{aligned} \tag{6.12}$$

where Sv denotes the selection matrix determining the axes for visual servoing, and is typically the identity matrix. $\widehat{L_e^{\dagger}}$ is the pseudo-inverse of the approximated interaction matrix $\widehat{L_e}$ from Section 6.1.2. The formulation depends on camera location and choice of IBVS or PBVS. For force control, S_F, selects axes for force control, K_F is the controller gain matrix and $F(t)$, F_D represent measured and reference forces respectively. During traded control, manipulator motion is first controlled by visual feedback $U_v(t)$. The controller switches to force control $U_F(t)$ once visual guidance approaches sufficiently near the surface. The desired manipulator state on the image plane $s_D(t)$ represents a state near the surface interaction site.

Traded control affords quick approach to contact surface, and stable contact is possible with 'low gain' force control. However, the main drawback occurs when motion guidance stops once $\|e_x(t)\|_2 \leq \epsilon$. Oftentimes, this can result in high contact forces due to inertial effects, particularly if the approach is optimized for speed.

Hybrid Control

Hybrid control enables visual and force control simultaneously in orthogonal direction. Specifically, visual guidance is enabled in directions orthogonal to all other feedback. As a result, pertinent visual information may go unused, a potential waste of observations. However, one advantage is that due to reduced dimensionality, the visual system does not need to be perfectly calibrated. The hybrid control law is:

$$\begin{aligned} U(t) &= U_v(t) + U_x(t) + U_F(t) \\ &= \left[-k_v\, S_v\, \widehat{L_e^{\dagger}}\, e_s(t)\right] + [-k_x\, S_x\, \dot{x_D}] + [-K_F\, S_F\, (F(t) - F_D(t))] \end{aligned} \tag{6.13}$$

where k_x and represents velocity controller gain, and $\dot{x_D}$ some desired end-effector reference velocity. S_v, S_x and S_F are diagonal selection matrices. To ensure orthogonality, it is required that $S_v + S_x + S_F = I$ and that all the elements in S_v, S_x and S_F are either 1s or 0s.

Many recent works utilize hybrid control, with active research investigating novel combined force and visual robotic control derived from the hybrid control strategy. These are explored below.

Integrated Traded and Hybrid Control In [156], the robotic task involves grasping a pen followed by tracing between whiteboard obstacles, all the while maintaining constant contact force. When the robot end effector is far from the target, only visual guidance is used, similar to traded control. However, when proximal to the surface interaction, force control is added and combined with vision in separate axes. Visual guidance is achieved through 2D IBVS for robustness to camera calibration errors. Constrained motion governs the combined force and vision portion. Two approaches exists for determining these constraints:

1. local estimation from force/torque
2. recursive least square

The latter was used in this case since force measurements were not reliable; low SNR and large friction.

In IBVS, the control error e_s is defined as:

$$e_s(t) \;\; = s(t) - s_D(t) \tag{6.14}$$

where $s(t)$ is the measured position and $s_D(t)$ is the desired position of features. A simple control law to drive $e_s(t)$ to zero is:

$$\begin{aligned} U_v(t) \;\; &= -k_v \, S_v \, \widehat{L_e^{\dagger}} \, e_s(t) \\ &= -k_v \, S_v \;\; \left[J_v^{l,r}(x) \right]^{\dagger} \;\; e_s(t) \end{aligned} \tag{6.15}$$

where x represents Cartesian coordinates and $\dot{x}$ the velocity screw of the end-effector measured in the robot base frame. k_v is a constant gain, and $J_v^{l,r}(x)$ is the image Jacobian, which relates image space velocities $\dot{s}$ of the features to corresponding Cartesian end-effector velocity $\dot{x}$ by

$$\dot{s} \;\; = J_v^{l,r}(x) \, \dot{x} \tag{6.16}$$

$J_v^{l,r}(x)$ is in general a function of Cartesian coordinates r. This is typically provided as the depth of imaged points. When using stereo cameras, the correct combined Jacobian for the stereo system is obtained by stacking Jacobians for the individual cameras

$$J_v^{l,r}(x) = \begin{bmatrix} J_v^l(x) T_b^l \\ J_v^r(x) T_b^r \end{bmatrix} \tag{6.17}$$

where T_b^l and T_b^r are the transformation matrices for screw from the robot base frame to left and right cameras respectively, and $J_v^l(x)$ and $J_v^r(x)$ are the Jacobians for left and right cameras. The exact form for point features can be found in [90]. The screw transformation matrix for the left camera is given by:

$$T_b^l = \begin{bmatrix} R_b^l & t_b^l \times R_b^l \\ 0_{3x3} & R_b^l \end{bmatrix} \tag{6.18}$$

with analogous formulation for the right. Suppose an imposed motion constraint in z that limits end-effector motion along a plane described by normal vector

$$p = [p_1 \;\; p_2 \;\; -1 \;\; p_4]^T$$

Given end-effector position $x = [X\, Y\, Z]^T$, the constrained robot position and velocity need to satisfy the following criteria:

$$p^T \begin{bmatrix} x \\ 1 \end{bmatrix} = [p_1 \;\; p_2 \;\; -1 \;\; p_4] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = 0 \tag{6.19}$$

$$[p_1 \;\; p_2 \;\; -1] \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \end{bmatrix} = 0 \tag{6.20}$$

The motion on the surface of the constrained plane is

$$\dot{s} = J_v^{l,r}(x) \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ p_1 & p_2 \end{bmatrix} \begin{bmatrix} \dot{X} \\ \dot{Y} \end{bmatrix} \tag{6.21}$$

Resulting in

$$\begin{aligned} U_v(t) &= -k_v\, S_v\, \widehat{L_e^\dagger}\, e_s(t) \\ &= -k_v \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ p_1 & p_2 \end{bmatrix} J_v^{l,r}(x)\, e_s(t) \end{aligned} \tag{6.22}$$

where $S_v = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ p_1 & p_2 \end{bmatrix}$ and $\widehat{L_e^\dagger} = J_v^{l,r}(x)$.

Introducing a proportional force controller

$$\begin{aligned} U_F(t) &= -K_F\, S_F\, (F(t) - F_D(t)) \\ &= -\begin{bmatrix} k_{F_x} & 0 & 0 \\ 0 & k_{F_y} & 0 \\ 0 & 0 & k_{F_z} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} (F(t) - F_D(t)) \\ &= \begin{bmatrix} 0_{2x1} \\ k_{F_z}\, (F(t) - F_D(t)) \end{bmatrix} \end{aligned} \tag{6.23}$$

If neglecting the $U_x(t)$ term, and combined with vision-based reference trajectory, the overall control law becomes

$$\begin{aligned} U(t) &= U_v(t) + U_x(t) + U_F(t) \\ &= -k_v \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ p_1 & p_2 \end{bmatrix} J_v^{l,r}(x)\, e_s(t) + 0_{3x1} + \begin{bmatrix} 0_{2x1} \\ k_{F_z}\, (F(t) - F_D(t)) \end{bmatrix} \end{aligned} \tag{6.24}$$

The combined traded and hybrid control offers advantages, including

1. no prior assumptions of negligible friction needed
2. accurate measurements contact force can recover surface normals

However drawbacks exist. Specifically, piecewise linearity of constraints and precise calibration are needed.

Prioritized Hybrid Control Autonomous robots in unknown environments should ideally be both adaptive and calibration free. This is especially true with multiple sensor feedback since interactions between sensory information can be uncertain. Hosoda et. al

presented a hybrid controller with external visual guidance and force controllers [87]. Both exhibited on-line parameter estimation and coordination. This adaptive hybrid controller differs from the typical hybrid controller [152], whereby only one sensory modality governs a given direction; cross sensory interactions are not considered. In this approach, both sensory types in all directions and times are considered. However, a predetermined priority heavily valuing force servoing over visual guidance is required for collision avoidance. For on-line parameter estimation, the visual controller estimates the camera Jacobian every time step, whereas the force controller estimates the unknown constraint surface normal vector.

Visual and force sensory information are not independent, and coordination of the two is crucial. Subtasks are oftentimes dependent on both types, and efforts to carefully design subtasks and separate the two sensory data are not guaranteed. Due to noise and disturbances, extra sensors for backup monitoring are needed. Prioritized hybrid control addresses this with predefined priorities. In completely unknown environments are tasks, criticality of sensor type may not be so straightforward.

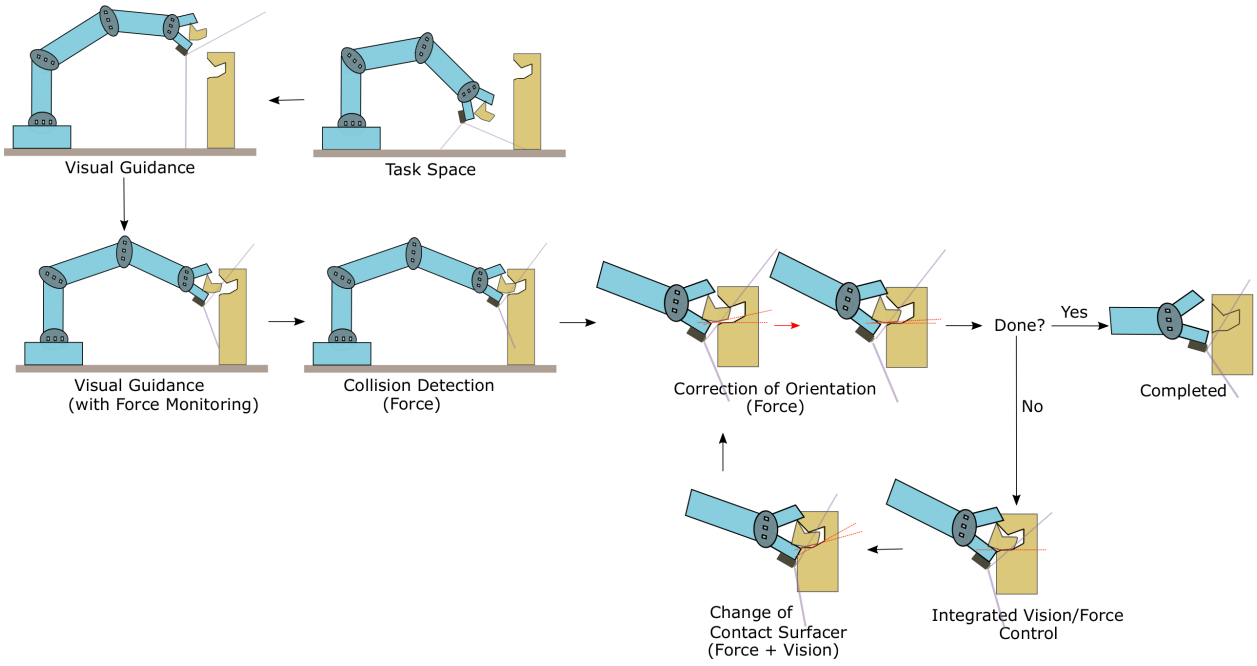


Figure 6.8: An example illustrating vision-force control fusion.

Shared Control

Shared control employs both feedback modalities concurrently and in all directions, without predefined weightings. The key lies in recognizing accelerations induced from visual guidance and subsequently negating from force readings. In practice, measured Cartesian accelerations are derived from joint encoder readings, which requires two differentiations and a transformation from joint to Cartesian space. Thus, calculated Cartesian accelerations are noisy. Measuring end-effector velocity can supplement these readings. Measured Cartesian accelerations and velocities can inform inertial coupling, and should be ignored with regard to force control. The strategy can be represented as:

$$U_v(t) = S_v\, M_v(t) = S_v\, \left(-k_v\, \widehat{L_e^\dagger}\, e_s(t)\right) \tag{6.25}$$

$$U_F(t) = S_F\, M_F(t) = S_F\, \left(-K_F\, (F(t) - F_D(t))\right) \tag{6.26}$$

$$\begin{aligned} c_1 &\equiv \ddot{x}_i(t) > \epsilon_a \\ c_2 &\equiv \dot{x}_i(t)\, \mathrm{sgn}(F_i(t)) < \epsilon_v \\ c_3 &\equiv M_v(t)\, F_i(t) > 0.0 \\ c_4 &\equiv |F_i| < \epsilon_F \end{aligned} \tag{6.27}$$

for each axis i in the task space:

if $(c_1 \wedge c_2) \vee c_3 \vee c_4$:

$S_v[i,i] = 1.0$

$S_F[i,i] = 0.0$

else:

$S_v[i,i] = 0.0$

$S_F[i,i] = 1.0$

Then we have

$$\begin{aligned} U(t) &= U_v(t) + U_x(t) + U_F(t) \\ &= S_v\, M_v(t) + [-k_x\, S_x\, \dot{x_D}] + S_F\, M_F(t) \end{aligned} \tag{6.28}$$

where c_1 - c_4 are four criteria that determine dominant sensory information, ϵ_a, ϵ_v, ϵ_F are sensor noise thresholds, and $\ddot{x_i}$, $\dot{x_i}$ represent measured Cartesian velocities and accelerations along orthogonal task space directions, and F_i denotes measured forces.

Modified Shared Control with Piecewise Thresholds Hybrid control and impedance control aim to resolve sensor data into movement by considering visual and force information at different levels [151]. While the task frame based shared control is promising [26, 55], it relies on high-level descriptors of actions to be carried out in each direction of the workspace at each moment. Thus, the geometric properties of the environment must be known. Modified shared control with piecewise thresholds aims to remove these prerequisites [168]. This work detailed four steps to validate the method:

1. analyze tracked trajectories

2. obtain movement flow from images

3. fuse force and movement flow visual guidance

4. experimental validation

as depicted in Figure 6.8

The sensor fusion approach, step (3), does not require specifying the sensory systems for each direction. A parameter $l(t, t_c)$ indicates whether tracking is correctly developed, where t is current time and t_c is the most recent time of a detected surface change. The following

equations derive $l(t, t_c)$:

$$l(t, t_c) = \frac{f^2(t, t_c)}{h(t, t_c)} \tag{6.29}$$

$$\text{where} \begin{cases} f(t, t_c) = \sum_{j=t_c}^{t} \frac{{}^j\gamma \, g(t,t_c)}{j_v} \\ h(t, t_c) = \sum_{j=t_c}^{t} \frac{g^2(t,t_c)}{j_v} \\ {}^j\gamma : \text{ the innovation value for the Kalman filter at time } j. \\ g(t, t_c) : \text{ the value of } G(t, t_c) \text{ in the surface change direction.} \\ G(t, t_c) : \text{ the effect of surface change on the innovation value.} \end{cases} \tag{6.30}$$

An empirical study centered around $l(t, t_c)$ resulted in the following thresholds:

- L1: Normal functioning of the system; values of $l(t, t_c)$ considered as normal.
- L2: Change in the surface. $l(t, t_c)$ greater than this threshold indicates change in the contact surface and the robot must reorient in relation to the new surface.
- L3. Upper limit of incorrect functioning. This threshold resides between L1 and L2, and characterizes the greatest $l(t, t_c)$ with no surface change.

$l(t, t_c)$ can increase due to irregularities or changes in the contact surface. However, errors in trajectory generated by visual servoing, high velocity established by movement flow, or incorrect tracking prevents the system from maintaining constant force and instead exhibits oscillatory behavior, where $l(t, t_c)$ increases as a result. To correct this, the proportion of information used from the force sensor is augmented with increasing $l(t, t_c)$. Instead of making use of selection matrices $S_v(t)$, $S_F(t)$, the final control action $U(t)$ is a weighted sum obtained from movement flow-based visual servoing $M_v(t)$ and from force sensor $M_F(t)$

$$U_v(t) = p_v \, M_v(t) = p_v \left(-k_v \, \widehat{L_e^{\dagger}} \, e_s(t) \right) \tag{6.31}$$

$$U_F(t) = p_F \, M_F(t) = p_F \, \left(-K_F \left(F(t) - F_D(t) \right) \right) \tag{6.32}$$

where $p_v + p_F = 1.0$, and p_v, p_F range from 0.0 to 1.0.

Control actions are defined depending on $l(t, t_c)$ and the previously defined thresholds as:

1. $l(t, t_c) < L1$

   Normal functioning. Both control actions are weighted the same, $p_v = p_F = 0.5$.

2. $L1 \leq l(t, t_c) < L3$

   A change in the surface begins or the system is non-functioning. The weight of movement flow-based visual guidance system is reduced to correct defects in tracking. The vision controller weight p_v is given as

$$p_v = \frac{p_2 - p_1}{L3 - L1} \ (l(t, t_c) - L1) + p_1 \tag{6.33}$$

   where $p_1 = 0.5$ and $p_2 = 0.5(U_{MIN}/U_{MAX})$, U_{MIN} and U_{MAX} are the minimum and maximum allowed value for $U(t)$. The weight of force control system is $p_F = 1.0 - p_v$.

3. $L3 \leq l(t, t_c) < L2$

   Security margin. The desired behavior is to continue with minimum velocity U_{vMIN} contributed by vision guidance.

4. $l(t, t_c) \geq L2$

   A surface change has occurred. The robot should reorient.

Modified shared control with piecewise thresholds aims to resolve contradicting information. In an unstructured environment, visual guidance cannot result in movements that are considered illegal from the force data. Utilizing the movement flow, different sensory data are combined and 3D trajectories coherent with the spatial restrictions are produced.

Modified Shared Control with Feedforward Loop A feedforward vision controller can be incorporated into shared controller schemes, and a high-level description is found in [10]. Additionally, a hybrid position and force control scheme with an internal velocity controller is used. Unlike other vision and force fusion algorithms, such as [64, 90], a vision Jacobian is not required. Since highlevel task descriptions determine the use of each sensor, probabilistic sensor weightings are not required. Figure 6.9 depicts this approach.

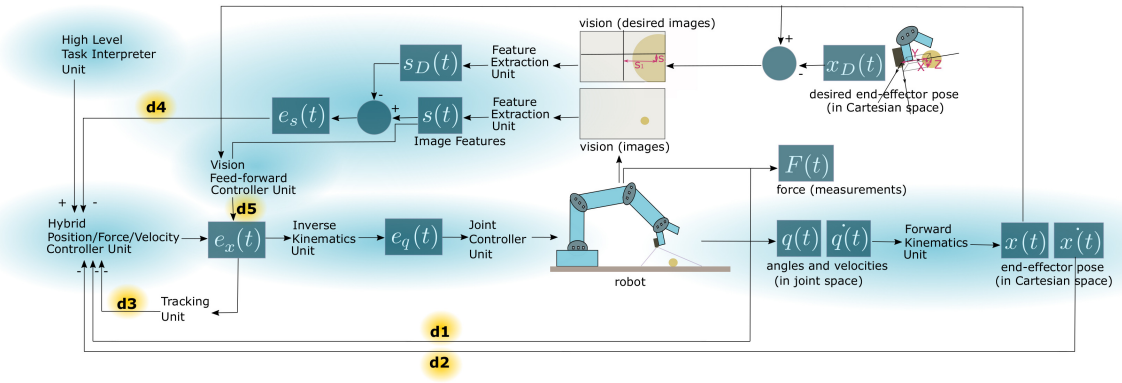


Figure 6.9: Control scheme with a feed-forward loop. The summation applies only to corresponding pairs of desired set points and measured signals.

Mason's compliance (task) frame decomposes the control system into 5 control directions

- d1: force direction: measured contact force between object and robot tool tip used in pos/force velocity controller.
- d2: velocity direction: dominant control direction when no contact.
- d3: tracking direction: automatically generated desired trajectory [55].
- d4: vision direction: camera position relative to image feature of interest.
- d5: feedforward direction: to reduce the error from d1, d3 and d4.

The controller follows four principals for sensor fusion. (1) The task is divided into a sequence of subtasks, (2) subtasks using both sensors must be operating in different directions for different sensory modes, (3) vision feed forward allows force and vision to act in the same direction, (4) the vision algorithm reconstructs 3D position of feature points for position based visual guidance. The framework is promising yet requires experimental validation.

6.4 Conclusions

Force controlled manipulation is a common technique for compliantly contacting and interacting with uncertain environments. Visual guidance is effective for reducing alignment uncertainties between objects using imprecisely calibrated camera-lens manipulator systems. Visual feedback provides information over a relatively large area of the workspace without

requiring contact with the environment, and force feedback provides highly localized and precise information upon contact. These two types of manipulator feedback, force and vision, represent complementary sensing modalities, and have a been individually widely adopted in robotic applications.

Combining the two modes can provide thorough and holistic task space characterization, and can theoretically afford better manipulation performance. Numerous recent research approaches provide insight into promising direction, yet each method has its limitations and disadvantages. The search for a generalized solution is a continued research direction, as most promising and implemented approaches are tailored to specific applications. This document presents high-level introduction of the two fields, and investigates recent advances in integrated visual guidance and force control for robotic manipulation. The methods in visual, force and integrated vision and force are also classified into acknowledged and accepted categories.

In our case, the force and visual information are coupled in a sense that it is originally derived from imaging processing by estimating the level of deformation perceived from the surgical images. The control scheme for autonomy in surgical robotics tasks needs to be very carefully designed in order to combine the visual information and the vision-based force estimation information effectively. Thus, this literature search serves a very useful study.

6.5 Further Study

After the literature review, I was collaboraively involved in two telerobotics research that concerns haptics. An error tolerance user study was conducted for haptic feedback in a teleoperated palpation task and submitted as a paper to EMBC 2020 [42] (see **section 6.5.1**). Later in another research project, we proven the possibility to integrate more information other than contact force, such as robot joint limit warning, to haptic feedback. This work was published to ICARM 2019 [89] (see **section 6.5.2**).

6.5.1 Error Tolerance for Haptic Feedback In Telesurgery

This work investigates the effects of disparity between rendered haptic feedback and visual feedback when interacting with deformable bodies, and is motivated by the use case of haptic feedback in RMIS. Specifically, three variations in rendered force were investigated in a simulated surgical palpation task:

1. (**A**) orientation: tilted towards or away from user;
2. (**B**) orientation: about the visual axis, CCW or CW;
3. (**C**) magnitude.

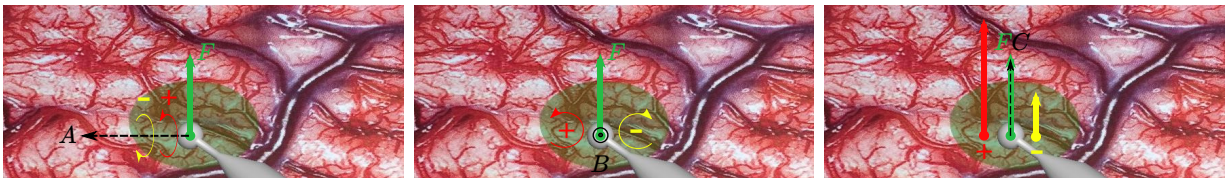


Figure 6.10: Positive and negative directions for test parameters **A**, **B**, and **C**. **Green** indicates the ground truth force vector based on simulated physics, **red** and yellow indicate positive and negative error directions respectively. **Black** indicates axis of error parameter. For **A**, positive error tilts the force vector into the page, and negative error tilts the force out of the page. As viewed from the user, positive error in **B** rotates the force vector CCW, and negative error CW. Parameter **C** scales the magnitude of the perceived force.

The three test variations, each with two test directions, are depicted graphically in Figure 6.10. An adaptive thresholding method is then used to collect the minimum and maximum tolerable errors in force orientation and magnitude of presented haptic feedback to maintain sufficient performance. Positive and negative thresholds for satisfactory performance along these parameters were sought in this study.

Table 6.1 shows the median final threshold values for each parameter. Overall, the results of this study suggest operator sensitivity is not necessarily symmetric to orientation error in vision-based force estimation. In particular, users tended to demonstrate an *increased robustness in force orientation error away and to the left* (CCW). One possible explanation arises when considering typical grasps for right-handed users in stylus haptic interaction (such as during handwriting). A fulcrum is formed with the hand while the load generates

forces at the stylus tip away and to the left. When the ground truth force is rotated towards the direction of the lever link (toward the user and to the right), less force is perceived at the fulcrum, resulting in degraded perception.

Threshold		Median
A	+	+14.4°
	-	-4.4°
B	+	+8.21°
	-	-6.05°
C	+	×1.08
	-	×0.899

Table 6.1: Median Adaptive Thresholds

Furthermore, users maintained acceptable palpation performance even if force magnitude was scaled by about 10%. For the positive threshold of force magnitude, 150% of the pop-through force (2.3 N) saturates the haptic device output at 3.3N. In contrast, no subject successfully completed the palpation task with attenuated force magnitude on the first trial. This suggests that a lack of haptic cues is considerably detrimental to task performance.

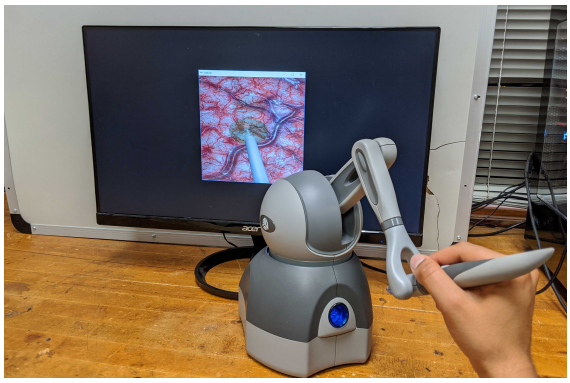


Figure 6.11: The experimental setup in [42]. Motion commands are sent and a deviated force feedback is received.

In general, the results of this study suggest that when considering accuracy of non-contact vision-based force estimation for palpation error tolerance in :

1. orientation is greater away from the user;
2. orientation is greater in the direction away from the user's dominant hand;
3. force magnitude is generally symmetric.

Force estimation confidence should thus consider user handedness and viewing perspective. This work is submitted to EMBC 2020 [42], and we envision the result to inform guidelines for vision-based tool-tissue force estimation.

6.5.2 Haptic Feedback that Reflects Robot Joint Limits

Robotic proxies extend human control to task spaces that would otherwise be unattainable by humans. However, kinematic difficulties associated with operating a remote maneuverable robotic device may arise due to dissimilarities between the reachable workspaces of the robot and the input device. This work reconciled these kinematic complexities using a naive tree structure approach. A 3DOF robot was used to sample joint limits in Cartesian space. The sampled joint limits were then used to compile a synthetic surface point cloud representing the joint limits of the 3DOF end effector. Systematically storing point clouds in a tree data structure, a local point cloud at any given joint limit can be retrieved. Well established haptic rendering techniques can then be used for appropriate 3DOF feedback.

Results show that using this point cloud storage and retrieval method, the joint limits for a 3DOF robot can be better represented, understood and maneuvered in cartesian space. Using joint limit haptic rendering techniques can alert the human operator when a certain limit has been reached, and thus reduce frustration and confusion in teleoperation. There are still issues to be resolved in representing workspace kinematics accurately to an operator using a telerobot, however, techniques used in this study raise the potential for using similar methods for telerobots having extremely complex task environments and numerous degrees of freedom. Direct next steps include algorithmic changes including the replacement of the tree data structure with a more efficient, constant look-up time mapping table. This work is published to ICARM 2019 [89].

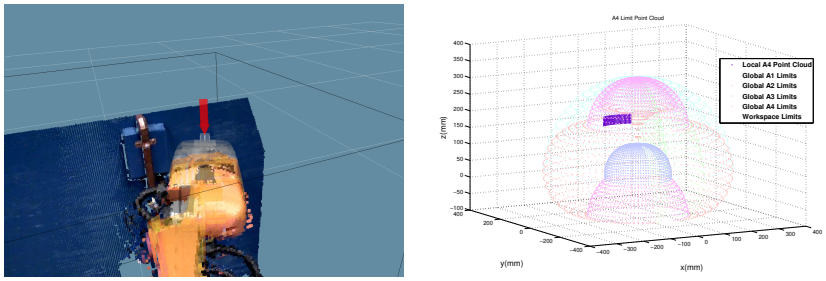


Figure 6.12: (left) Operator-side visual feedback; (right) Local joint limit point cloud when command input violated A4 limits.

Chapter 7

CONCLUSION AND FUTURE WORK

During the course of my PhD, I developed a vision-based force estimation framework for robot-assisted minimally invasive surgery (RMIS) that infers contact force between surgical instruments and soft tissue through real-time tissue deformation analysis. As illustrated in Figure 7.1, this framework is divided into three stages. Stage 1,2 are completed and preliminary experiments are conducted for stage 3. Due to computation limitations, the completion of stage 3 will require software efficiency optimization.

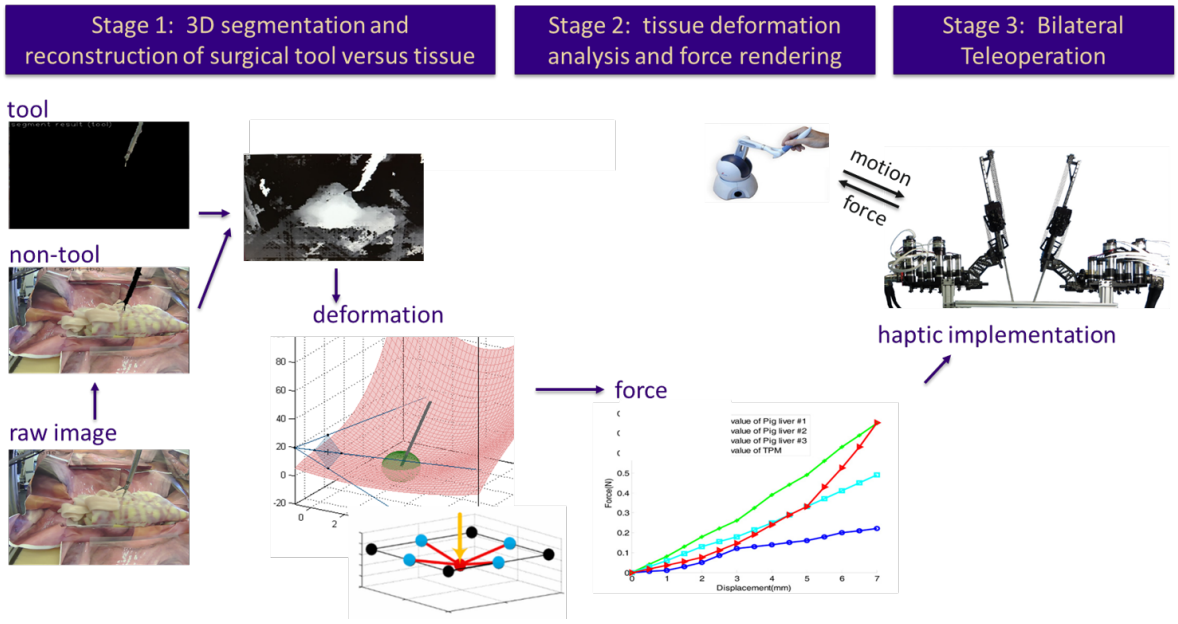


Figure 7.1: My PhD research framework.

Looking at the bigger picture, RMIS has many benefits over traditional open surgery, and combines the skills and decision making of highly-trained surgeons with the dexterity and

precision of machines. This architecture entails the Control Agent (surgeon) sending motion commands to remote Robots (surgical robot). Simultaneously, perceptual information is relayed in the opposite direction, either in the form of direct feedback to human operators or as decision factors for autonomous agents. When aspects of perception are weak or unreliable, a Data Processing Agent can be used to reconstruct or estimate sparse perception to ensure system robustness. This same component is effective for regulating consistency and thus detect and deter cyber attacks. Through this cyber-physical system architecture, however, surgeons lose direct force sensations otherwise present in manual surgery. Realizing accurate force feedback in RMIS remains an open challenge, since direct placement of force sensors at the tool tip is incompatible with required sterilization procedures. Vision-based force estimation from real-time tissue deformation analysis serves as a promising non-contact alternative and is the major theme of my PhD research. Below is a summary of how my endeavors fit in this big picture and a brief description of future research plan.

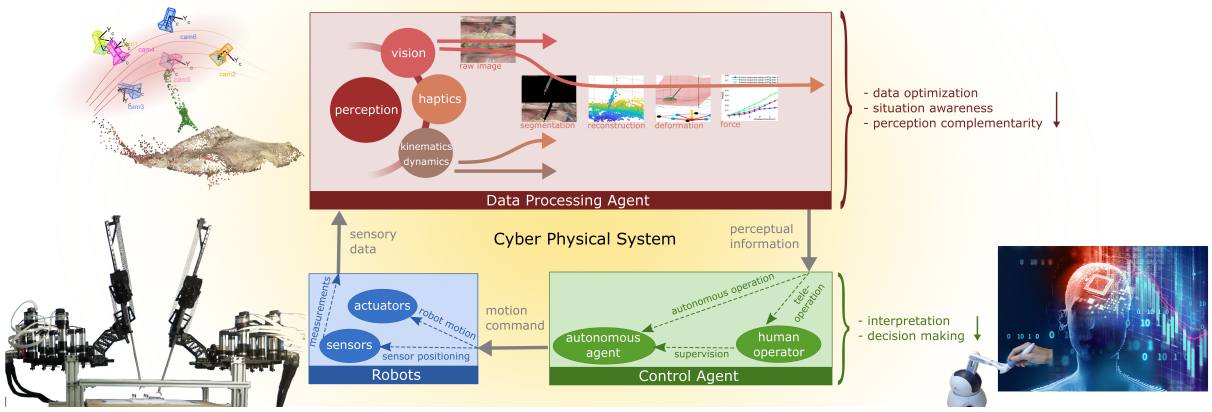


Figure 7.2: The bigger picture.

7.1 Data Processing Agent

The lack of haptic sensations in RMIS can result in undue forces during surgery, and a solution would prevent unintentional tissue damage. To that end, I developed a framework that employs vision to estimate force feedback based on tissue deformation induced by tool-tissue interaction [204].

7.1.1 Situational Awareness

Situational awareness entails understanding environment state through available sensory data. In my research, ascertaining the relevant awareness answers the following question: what do tissue patches in the vicinity of the tool-tissue contact look like? To identify a region of interest (ROI), I implemented and comparatively analyzed multiple approaches for image segmentation of surgical tool pixels from background tissue, including pure computer vision [53], adding robot kinematic prior [204] and finally fusing with a machine learning approach [170]. 3D reconstruction of the dynamic ROI is then implemented [207]. By interchanging the order of segmentation and 3D reconstruction, I also discovered a trade-off between computational efficiency and prone-to-false-positives [91].

7.1.2 Data Optimization

Data optimization improves the quality of sensory input through optimal sensor positioning or pre-processing. Two sensory inputs are utilized in my research - vision and robot kinematics. In vision, external complications like specular reflection, occlusion and deformation pose significant challenges to 3D reconstruction of surgical cavities. A multicamera system is hence considered to yield a more accurate surgical 3D model. I created a graph-based camera grouping and pair sequencing algorithm that automatically generates a high-quality 3D map from multiple, independently moving camera viewpoints [205]. Apart from vision, robot kinematics information is another source of sensory data. However, motor backlash and cable stretch for cable-driven robots like the Raven-II introduce inevitable uncertainties into kinematics calculations. To compensate for this inaccuracy, I collaboratively built a data-driven robot position correction model [81]. In ongoing work, I aim to seek solutions to further improve sensory data quality.

7.1.3 Perception Complementarities

Perception complementarities synthesize available sensory data to reconstruct absent or weak perceptual senses [208]. I developed an autonomous algorithm that distinguishes static, shifting and deforming 3D points. This was used to create tissue deformation maps from temporal

changes in the 3D surgical cavity model using 3D registration [203]. I currently pursue optimizing vision processing software for better time efficiency so that real-time force estimation from the surgical deformation map and tissue dynamics models is possible. Meanwhile, I collaboratively investigated the tolerable vision-derived haptic feedback error such that a surgical palpation task can still be successfully done [42]. Several alternative approaches to force estimation in RMIS rely on static force analysis, i.e. deriving end effector force from joint torques. However, critics are quick to identify that the derived force is extremely noisy, due to friction from within the cable driven joints and the surgical tool shafts when interfacing with the trocar. My future research will combine vision-based and robot-based force estimation approaches into a more vigorous and robust estimation framework. Beyond reflecting tool-tissue contacts, haptic feedback reflecting robot joint limits can also improve operator workspace limit perception issues [89].

7.2 Control Agent

A control agent interprets all forms of perceptual information, including enhanced sensory measurements and artificially generated senses from the perception complementarity. The control agent also commands robot motion.

7.2.1 Interpretation

Interpretation involves incorporating perceptual information to derive an understanding of the surroundings. While this comes intuitively for human operators, integrating various perceptual modes can be challenging for autonomous agents. To address this, I examined control strategies integrating both force and vision [119]. In ongoing research, experiments tailored to analyze perception fusion in autonomous agents are a focus.

7.2.2 Decision Making

Decision making renders action from perceptual information and its interpretation. Depending on whether the target robot role is actuation or sensing, decisions impact either end effector motion or sensor positioning. Addressing the former scenario, I collaboratively im-

plemented an autonomous heartbeat motion compensation agent through imitation learning and model predictive control. This research allows for robotic cardiac surgeries without stopping the heart [233]. For the latter, I developed an autonomous multicamera viewpoint adjustment algorithm formulated as a constrained maximum coverage problem in dynamic surgical cavity 3D reconstruction [206]. This was motivated to require fewer cameras for reconstruction, and relies on adjusting camera angle. My research program aims to develop autonomous solutions using artificial intelligence and transfer learning toward improving robot manipulation or sensor placement.

7.3 Robots

Robots serve as the platforms that perceive and interact with the physical world. With distinct mechanical and software design of robot systems, many perception complementarity and autonomous control frameworks are not readily generalized to other robots. Over the course of my Ph.D, I contributed to the collaborative robotics toolkit (CRTK) [1] [202] and its associated AMBF simulator [2] , an NSF National Robotics Initiatives funded project jointly with Johns Hopkins University and Worchester Polytechnic Institute. The initial motivation was to build a common API for Raven-II and the da Vinci Research Kit (dVRK), telerobotic systems primarily used in surgical applications. However, it also encompasses teleoperation or cooperative control of other robot systems, including prototypical industrial robots. My future research will prioritize making robots compatible with CRTK to expand cross platform support.

Moving forward, my research interest aims to investigate more perception complementarity scenarios; and solidify the vision-based force estimation pipeline. These interdisciplinary endeavors analytically, computationally and experimentally advance all 3 color-coded components (See Figure 7.2) in robot-assisted surgery.

---

[1] collaborative robotics toolkit (CRTK), https://github.com/collaborative-robotics

[2] AMBF simulator, https://github.com/WPI-AIM/ambf

BIBLIOGRAPHY

[1] Randa Almadhoun, Tarek Taha, Lakmal Seneviratne, Jorge Dias, and Guowei Cai. A survey on inspecting structures using robotic systems. *International Journal of Advanced Robotic Systems*, 13(6):1729881416663664, 2016.

[2] Noga Alon, Baruch Awerbuch, and Yossi Azar. The online set cover problem. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 100–105. ACM, 2003.

[3] Pilar Sobrevilla Angelica I. Aviles, Arturo Marban. A recurrent neural network approach for 3d vision-based force estimation. *Image Processing Theory, Tools and Applications (IPTA), 2014 4th International Conference*, 2014.

[4] Csaba Antonya. Force feedback in string based haptic systems. *Procedia Computer Science*, 25:90–97, 2013.

[5] Kashan Aqeel, Urooj Naveed, Faarah Fatima, Farah Haq, M Arshad, Ammar Abbas, M Nabeel, and M Khurram. Skin stroking haptic feedback glove for assisting blinds in navigation. In *Robotics and Biomimetics (ROBIO), 2017 IEEE International Conference on*, pages 177–182. IEEE, 2017.

[6] Brian Argrow, Dale Lawrence, and Erik Rasmussen. Uav systems for sensor dispersal, telemetry, and visualization in hazardous environments. In *43rd AIAA Aerospace Sciences Meeting and Exhibit*, page 1237, 2005.

[7] Peter K. Allen Austin Reiter and Tao Zhao. Feature classification for tracking articulated surgical tools. *Springer-Verlag Berlin Heidelberg*, pages 592–600, 2012.

[8] Tao Zhao Austin Reiter, Peter K. Allen. Marker-less articulated surgical tool detection. 2012.

[9] Hamidreza Azimian, Rajni V Patel, and Michael D Naish. On constrained manipulation in robotics-assisted minimally invasive surgery. In *Biomedical Robotics and Biomechatronics (BioRob), 2010 3rd IEEE RAS and EMBS International Conference on*, pages 650–655. IEEE, 2010.

[10] Johan Baeten, Herman Bruyninckx, and Joris De Schutter. Combining eye-in-hand visual servoing and force control in robotic tasks using the task frame. In *Multisensor Fusion and Integration for Intelligent Systems, 1999. MFI'99. Proceedings. 1999 IEEE/SICE/RSJ International Conference on*, pages 141–146. IEEE, 1999.

[11] Johan Baeten and Joris De Schutter. Hybrid vision/force control at corners in planar robotic-contour following. *IEEE/ASME Transactions on mechatronics*, 7(2):143–151, 2002.

[12] Johan Baeten, Walter Verdonck, Herman Bruyninckx, and Joris De Schutter. Combining force control and visual servoing for planar contour following. *Machine Intelligence and Robotic Control*, 2(2):69–75, 2000.

[13] Adrien Bartoli. Estimating the pose of a 3d sensor in a non-rigid environment. In *Dynamical Vision*, pages 243–256. Springer, 2007.

[14] Ozkan Bebek and M Cenk Cavusoglu. Intelligent control algorithms for robotic-assisted beating heart surgery. *IEEE Transactions on Robotics*, 23(3):468–480, 2007.

[15] Xun Bi, Bo Li, Hongxuan Wang, Letang Xue, and Weiguo Wang. Fuzzy position-force control in training systems with haptic guidance. In *Fuzzy Systems and Knowledge Discovery (FSKD), 2013 10th International Conference on*, pages 207–212. IEEE, 2013.

[16] Zhenshan Bing, Long Cheng, Kai Huang, Zhuangyi Jiang, Guang Chen, Florian Röhrbein, and Alois Knoll. Towards autonomous locomotion: Slithering gait design of a snake-like robot for target observation and tracking. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 2698–2703. IEEE, 2017.

[17] Phillip Roan Daniel Glozman Blake Hannaford, Hawkeye King. Raven-ii: An open platform for surgical robotics research. *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING*, 60(4):954–959, 2013.

[18] Immanuel M Bomze, Marco Budinich, Panos M Pardalos, and Marcello Pelillo. The maximum clique problem. In *Handbook of combinatorial optimization*, pages 1–74. Springer, 1999.

[19] Xavier Bonaventura, Miquel Feixas, Mateu Sbert, Lewis Chuang, and Christian Wallraven. A survey of viewpoint selection methods for polygonal models. *Entropy*, 20(5):370, 2018.

[20] Misagh Mansouri Boroujeni and Ali Meghdari. Haptic device application in persian calligraphy. *IInternational Conference on Computer and Automation Engineering*, 2009.

[21] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[22] G. Bradski. Dr. dobb's journal of software tools. 2000.

[23] Christoph Bregler, Aaron Hertzmann, and Henning Biermann. Recovering non-rigid 3d shape from image streams. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 690–696. IEEE, 2000.

[24] Torgny Brogårdh. Robot control overview: An industrial perspective. *Modeling, Identification and Control*, 30(3):167, 2009.

[25] Robert Grover Brown, Patrick YC Hwang, et al. *Introduction to random signals and applied Kalman filtering*, volume 3. Wiley New York, 1992.

[26] Herman Bruyninckx and Joris De Schutter. Specification of force-controlled actions in the" task frame formalism"-a synthesis. *IEEE transactions on robotics and automation*, 12(4):581–589, 1996.

[27] Eduardo F Camacho and Carlos Bordons Alba. *Model predictive control*. Springer Science & Business Media, 2013.

[28] Mark E Campbell and William W Whitacre. Cooperative tracking using vision measurements on seascan uavs. *IEEE Transactions on Control Systems Technology*, 15(4):613–626, 2007.

[29] Alicia M Cano, Francisco Gayá, Pablo Lamata, Patricia Sánchez-González, and Enrique J Gómez. Laparoscopic tool tracking method for augmented reality surgical applications. In *International Symposium on Biomedical Simulation*, pages 191–196. Springer, 2008.

[30] JF Cardenas-Garcia, HG Yao, and S Zheng. 3d reconstruction of objects using stereo imaging. *Optics and Lasers in Engineering*, 22(3):193–213, 1995.

[31] Haiyang Chao, Marc Baumann, Austin Jensen, YangQuan Chen, Yongcan Cao, Wei Ren, and Mac McKee. Band-reconfigurable multi-uav-based cooperative remote sensing for real-time water management and distributed irrigation control. *IFAC Proceedings Volumes*, 41(2):11744–11749, 2008.

[32] Pierre Chatelain, Alexandre Krupa, and Nassir Navab. Optimization of ultrasound image quality via visual servoing. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 5997–6002. IEEE, 2015.

[33] Francois Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. In *The confluence of vision and control*, pages 66–78. Springer, 1998.

[34] François Chaumette and Seth Hutchinson. Visual servo control. i. basic approaches. *IEEE Robotics & Automation Magazine*, 13(4):82–90, 2006.

[35] François Chaumette and Seth Hutchinson. Visual servo control, part ii: Advanced approaches. *IEEE Robotics and Automation Magazine*, 14(1):109–118, 2007.

[36] François Chaumette, Seth Hutchinson, and Peter Corke. Visual servoing. In *Springer Handbook of Robotics*, pages 841–866. Springer, 2016.

[37] Heping Chen, William Eakins, Jianjun Wang, George Zhang, and Thomas Fuhlbrigge. Robotic wheel loading process in automotive manufacturing automation. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 3814–3819. IEEE, 2009.

[38] Heping Chen, Jianjun Wang, Biao Zhang, George Zhang, and Thomas Fuhlbrigge. Towards performance analysis of wheel loading process in automotive manufacturing. In *Automation Science and Engineering (CASE), 2010 IEEE Conference on*, pages 234–239. IEEE, 2010.

[39] Jiatong Chen and Yong Wang. The guidance and control of small net-recovery uav. In *Computational Intelligence and Security (CIS), 2011 Seventh International Conference on*, pages 1566–1570. IEEE, 2011.

[40] Lei Cheng. Computation and measurement of force and tissue damage for the grasper-tissue interface in robot-assisted minimal invasive surgery. *Doctoral thesis, University of Washington*.

[41] Andrea Cherubini, Robin Passama, André Crosnier, Antoine Lasnier, and Philippe Fraisse. Collaborative manufacturing with physical human–robot interaction. *Robotics and Computer-Integrated Manufacturing*, 40:1–13, 2016.

[42] Digesh Chitrakar, Divas Subedi, **Yun Hsuan Su**, and Kevin Huang. Characterizing limits of vision-based force feedback in simulated surgical tool-tissue interaction. In *2020 International Conference onIn Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2020.

[43] Grzegorz Chmaj and Henry Selvaraj. Distributed processing applications for uav/drones: a survey. In *Progress in Systems Engineering*, pages 449–454. Springer, 2015.

[44] Reuven Cohen and Liran Katzir. The generalized maximum coverage problem. *Information Processing Letters*, 108(1):15–22, 2008.

[45] Christophe Collewet and François Chaumette. Positioning a camera with respect to planar objects of unknown shape by coupling 2-d visual servoing and 3-d estimations. In *transactions on robotics and automation*, pages 322–333. IEEE, 2002.

[46] Toby Collins, Benoît Compte, and Adrien Bartoli. Deformable shape-from-motion in laparoscopy using a rigid sliding window. In *MIUA*, pages 173–178, 2011.

[47] Peter I Corke. Experiments in high-performance robotic visual servoing. In *Experimental Robotics III*, pages 193–205. Springer, 1994.

[48] Peter I Corke. Dynamic issues in robot visual-servo systems. In *Robotics Research*, pages 488–498. Springer, 1996.

[49] Peter I Corke and Seth A Hutchinson. A new partitioned approach to image-based visual servo control. *IEEE Transactions on Robotics and Automation*, 17(4):507–515, 2001.

[50] Christopher T Cunningham and Randy S Roberts. An adaptive path planning algorithm for cooperating unmanned air vehicles. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, volume 4, pages 3981–3986. IEEE, 2001.

[51] Guang Zhong Yang D. Stoyanov. Removing specular reflection components for robotic assisted laparoscopic surgery. *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, (8856983):III – 632–5, 2005.

[52] Maneesh Dewan William Lau Ming Li Henry Lin Panadda Marayong Nicholas Ramey Darius Burschka, Jason J. Corso. Navigating inner space: 3-d assistance for minimally invasive surgery. 2005.

[53] ❖ M. Allan, Alex Shvets, Thomas Kurmann, Zichen Zhang, Rahul Duggal, **Yun Hsuan Su**, Nicola Rieke, Iro Laina, Niveditha Kalavakonda, Sebastian Bodenstedt, et al. 2017 robotic instrument segmentation challenge. *Medical image analysis (MedIA)*, 2019.

[54] Danail Stoyanov Pierre Jannin David Bouget, Max Allan. Vision-based and marker-less surgical tool detection and tracking: a review of the literature. *1361-8415/© 2016 Elsevier B.V.*, pages 633–654, 2016.

[55] Joris De Schutter and Hendrik Van Brussel. Compliant robot motion i. a formalism for specifying compliant motion tasks. *The International Journal of Robotics Research*, 7(4):3–17, 1988.

[56] Alessio Del Bue and Lourdes Agapito. Non-rigid 3d shape recovery using stereo factorization. In *Asian Conference of Computer Vision*, volume 1, pages 25–30, 2004.

[57] Brian Dellon and Yoky Matsuoka. Path guidance control for a safer large scale dissipative haptic display. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 2073–2078. IEEE, 2008.

[58] Ross W Deming and Leonid I Perlovsky. Concurrent multi-target localization, data association, and navigation for a swarm of flying sensors. *Information Fusion*, 8(3):316–330, 2007.

[59] Nicola Di Lorenzo, Livia Cenci, Massimiliano Simi, Claudio Arcudi, Valeria Tognoni, Achille Lucio Gaspari, and Pietro Valdastri. A magnetic levitation robotic camera for minimally invasive surgery: Useful for notes? *Surgical endoscopy*, 31(6):2529–2533, 2017.

[60] Tim Dobbert. *Matchmoving: the invisible art of camera tracking.* John Wiley & Sons, 2006.

[61] Enrique Dunn and Jan-Michael Frahm. Next best view planning for active model improvement. In *BMVC*, pages 1–11, 2009.

[62] Enrique Dunn, Jur Van Den Berg, and Jan-Michael Frahm. Developing visual sensing strategies through next best view planning. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4001–4008. IEEE, 2009.

[63] Steven Eppinger and WARRENP Seering. Understanding bandwidth limitations in robot force control. In *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, volume 4, pages 904–909. IEEE, 1987.

[64] Bernard Espiau, François Chaumette, and Patrick Rives. A new approach to visual servoing in robotics. *ieee Transactions on Robotics and Automation*, 8(3):313–326, 1992.

[65] Syafizwan Faroque, Ben Horan, Michael Mortimer, and Mulyoto Pangestu. Large-scale virtual reality micro-robotic cell injection training. In *World Automation Congress (WAC), 2016*, pages 1–6. IEEE, 2016.

[66] TD Fazio, D Seltzer, and D Whitney. The instrumented remote center of compliance. *Ind. Robot.*, 11(4):238–242, 1984.

[67] John T Feddema and Owen Robert Mitchell. Vision-guided servoing with feature-based trajectory generation (for robots). *IEEE Transactions on Robotics and Automation*, 5(5):691–700, 1989.

[68] Tomonari Furukawa, Frederic Bourgault, Benjamin Lavis, and Hugh F Durrant-Whyte. Recursive bayesian search-and-tracking using coordinated uavs for lost targets. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 2521–2526. IEEE, 2006.

[69] Jose A Galvez, Joaquin Estremera, and Pablo Gonzalez De Santos. A new legged-robot configuration for research in force distribution. *Mechatronics*, 13(8-9):907–932, 2003.

[70] D Gálvez-López and JD Tardós. Dbow2: Enhanced hierarchical bag-of-word library for c++, 2012.

[71] Jian Gao, Alison A Proctor, Yang Shi, and Colin Bradley. Hierarchical model predictive image-based visual servoing of underwater vehicles with adaptive neural network dynamic control. *IEEE transactions on cybernetics*, 46(10):2323–2334, 2016.

[72] Gabriel J Garcia, Juan A Corrales, Jorge Pomares, and Fernando Torres. Survey of visual and force/tactile control of robots for physical interaction in spain. *Sensors*, 9(12):9689–9733, 2009.

[73] Ravi Garg, Anastasios Roussos, and Lourdes Agapito. Dense variational reconstruction of non-rigid surfaces from monocular video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1272–1279, 2013.

[74] Andreas Geiger, Julius Ziegler, and Christoph Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *IEEE Intelligent Vehicles Symposium*, Baden-Baden, Germany, June 2011.

[75] Ronald Glowinski and Patrick Le Tallec. *Augmented Lagrangian and operator-splitting methods in nonlinear mechanics*, volume 9. SIAM, 1989.

[76] Pranava R Goundan and Andreas S Schulz. Revisiting the greedy approach to submodular set function maximization. *Optimization online*, pages 1–25, 2007.

[77] Oscar G Grasa, Javier Civera, and JMM Montiel. Ekf monocular slam with relocalization for laparoscopic sequences. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4816–4821. IEEE, 2011.

[78] Dave Gravel, Frank Maslar, George Zhang, Srini Nidamarthi, Heping Chen, and Tom Fuhlbrigge. Toward robotizing powertrain assembly. In *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on*, pages 541–546. IEEE, 2008.

[79] Chris Gunn, Warren Muller, and Amitava Datta. Performance improvement with haptic assistance: A quantitative assessment. In *EuroHaptics conference, 2009 and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. World Haptics 2009. Third Joint*, pages 511–516. IEEE, 2009.

[80] Kaiwen Guo, Feng Xu, Yangang Wang, Yebin Liu, and Qionghai Dai. Robust non-rigid motion tracking and surface reconstruction using l0 regularization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3083–3091, 2015.

[81] H. Peng, Xingjian Yang, **Y. H. Su**, and Blake Hannaford. Real-time data driven precision estimator for raven-ii surgical robot end effector position. In *2020 International Conference on Robotics and Automation (ICRA)*. IEEE, 2020.

[82] Sven Haase, Sebastian Bauer, Jakob Wasza, Thomas Kilgus, Lena Maier-Hein, Armin Schneider, Michael Kranzfelder, Hubertus Feußner, and Joachim Hornegger. 3-d operation situs reconstruction with time-of-flight satellite cameras using photogeometric data fusion. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 356–363. Springer, 2013.

[83] Yaroslav Halchenko. Iterative closest point (icp) algorithm.

[84] Richard Hartley and René Vidal. Perspective nonrigid shape and motion recovery. In *European Conference on Computer Vision*, pages 276–289. Springer, 2008.

[85] Neville Hogan. Impedance control: An approach to manipulation. In *American Control Conference, 1984*, pages 304–313. IEEE, 1984.

[86] Hidekata Hontani, Takamiti Matsuno, and Yoshihide Sawada. Robust nonrigid icp using outlier-sparsity regularization. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 174–181. IEEE, 2012.

[87] Koh Hosoda, Katsuji Igarashi, and Minoru Asada. Adaptive hybrid control for visual and force servoing in an unknown environment. *IEEE Robotics & Automation Magazine*, 5(4):39–43, 1998.

[88] Mingxing Hu, Graeme P Penney, Daniel Rueckert, Philip J Edwards, Fernando Bello, Roberto Casula, Michael Figl, and David J Hawkes. Non-rigid reconstruction of the beating heart surface for minimally invasive cardiac surgery. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 34–42. Springer, 2009.

[89] Kevin Huang, **Yun Hsuan Su**, Mahmoud Khalil, Daniel Melesse, and Rahul Mitra. Sampling of 3dof robot manipulator joint-limits for haptic feedback. In *2019 International Conference on Advanced Robotics and Mechatronics (ICARM)*. IEEE, 2019.

[90] Seth Hutchinson, Gregory D Hager, and Peter I Corke. A tutorial on visual servo control. *IEEE transactions on robotics and automation*, 12(5):651–670, 1996.

[91] **Yun Hsuan Su** Isaac Huang, Kevin Huang and Blake Hannaford. Comparison of 3d surgical tool segmentation procedures with robot kinematics prior. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2018*, October 2018.

[92] Ishikawa Watanabe Laboratory, University of Tokyo. Basic concept and technical terms. February 2015.

[93] Muhammad Herman Jamaluddin, Tomoyuki Shimono, and Naoki Motoi. Haptic bilateral control system with visual force compliance controller. In *Industrial Electronics (ISIE), 2013 IEEE International Symposium on*, pages 1–6. IEEE, 2013.

[94] Prasanta K Jana and Azad Naik. An efficient minimum spanning tree based clustering algorithm. In *Methods and Models in Computer Science, 2009. ICM2CS 2009. Proceeding of International Conference on*, pages 1–5. IEEE, 2009.

[95] T. G. K. van de Sande and C. G. Snoek. Color descriptors for object category recognition. *In European Conference on Color in Graphics, Imaging and Vision*, 2:378–381, 2008.

[96] Hideki Kadone, Delphine Bernardin, Daniel Bennequin, and Alain Berthoz. Gaze anticipation during human locomotion-top-down organization that may invert the concept of locomotion in humanoid robots. In *RO-MAN, 2010 IEEE*, pages 552–557. IEEE, 2010.

[97] Jiyeon Kang, Vineet Vashista, and Sunil K Agrawal. On the adaptation of pelvic motion by applying 3-dimensional guidance forces using tpad. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(9):1558–1567, 2017.

[98] Douglas A Kerr. The proper pivot point for panoramic photography, 2008.

[99] A Kershenbaum and R Van Slyke. Computing minimum spanning trees efficiently. In *Proceedings of the ACM annual conference-Volume 1*, pages 518–527. ACM, 1972.

[100] Oussama Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987.

[101] Samir Khuller, Anna Moss, and Joseph Seffi Naor. The budgeted maximum coverage problem. *Information processing letters*, 70(1):39–45, 1999.

[102] Myungjoon Kim, Chiwon Lee, Nhayoung Hong, Yoon Jae Kim, and Sungwan Kim. Development of stereo endoscope system with its innovative master interface for continuous surgical operation. *Biomedical engineering online*, 16(1):81, 2017.

[103] Vladimir G Kim, Yaron Lipman, Xiaobai Chen, and Thomas Funkhouser. Möbius transformations for global intrinsic symmetry analysis. In *Computer Graphics Forum*, volume 29, pages 1689–1700. Wiley Online Library, 2010.

[104] Stefan Kimmer, Jan Smisek, and Andre Schiele. Effects of haptic guidance and force feedback on mental rotation abilities in a 6-dof teleoperated task. In *Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on*, pages 3092–3097. IEEE, 2015.

[105] Marc D Kohli, Ronald M Summers, and J Raymond Geis. Medical image data and datasets in the era of machine learning—whitepaper from the 2016 c-mimi meeting dataset session. *Journal of digital imaging*, 30(4):392–399, 2017.

[106] Vladislav Kraevoy and Alla Sheffer. Mean-value geometry encoding. *International Journal of Shape Modeling*, 12(01):29–46, 2006.

[107] Michael Krainin, Brian Curless, and Dieter Fox. Autonomous generation of complete 3d object models using next best view manipulation planning. In *2011 IEEE International Conference on Robotics and Automation*, pages 5031–5037. IEEE, 2011.

[108] Anand Kumar, Nirma Yadav, Shipra Singh, and Neha Chauhan. Minimally invasive (endoscopic-computer assisted) surgery: Technique and review. *Annals of maxillofacial surgery*, 6(2):159, 2016.

[109] Jonckheere EA Hayati S Kwoh YS, Hou J. A robot with improved absolute positioning accuracy for ct guided stereotactic brain surgery. *IEEE transactions on bio-medical engineering*, page 153–160, 1988.

[110] Ka-Wai Kwok, Kuen Hung Tsoi, Valentina Vitiello, James Clark, Gary CT Chow, Wayne Luk, and Guang-Zhong Yang. Dimensionality reduction in controlling articulated snake robot for endoscopy under dynamic active constraints. *IEEE Transactions on Robotics*, 29(1):15–31, 2013.

[111] Mathieu Labbé and François Michaud. Memory management for real-time appearance-based loop closure detection. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 1271–1276. IEEE, 2011.

[112] Iro Laina, Nicola Rieke, Christian Rupprecht, Josué Page Vizcaíno, Abouzar Eslami, Federico Tombari, and Nassir Navab. Concurrent segmentation and localization for tracking of surgical instruments. In *International conference on medical image computing and computer-assisted intervention*, pages 664–672. Springer, 2017.

[113] Iro Laina, Nicola Rieke, Christian Rupprecht, Josué Page Vizcaíno, Abouzar Eslami, Federico Tombari, and Nassir Navab. Concurrent segmentation and localization for tracking of surgical instruments. *CoRR*, abs/1703.10701, 2017.

[114] Dale A Lawrence. Impedance control stability properties in common implementations. In *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*, pages 1185–1190. IEEE, 1988.

[115] Stefan Leutenegger, Simon Lynen, Michael Bosse, Roland Siegwart, and Paul Furgale. Keyframe-based visual–inertial odometry using nonlinear optimization. *The International Journal of Robotics Research*, 34(3):314–334, 2015.

[116] Hao Li, Robert W Sumner, and Mark Pauly. Global correspondence optimization for non-rigid registration of depth scans. In *Computer graphics forum*, volume 27, pages 1421–1430. Wiley Online Library, 2008.

[117] Kun Li, Jingyu Yang, Yu-Kun Lai, and Daoliang Guo. Robust non-rigid registration with reweighted position and transformation sparsity. *IEEE transactions on visualization and computer graphics*, 2018.

[118] Tongying Li and Hongbo Zhu. Research on obstacle avoidance intelligent wheelchair design and simulation. In *Chinese Automation Congress (CAC), 2017*, pages 5475–5479. IEEE, 2017.

[119] **Yun Hsuan Su**, Kevin Huang, and Blake Hannaford. A review on integration of vision and force in robot manipulation. *Advanced Robotics (AD)*, 2020.

[120] Bingxiong Lin, Adrian Johnson, Xiaoning Qian, Jaime Sanchez, and Yu Sun. Simultaneous tracking, 3d reconstruction and deforming point detection for stereoscope guided surgery. In *Augmented Reality Environments for Medical Imaging and Computer-Assisted Interventions*, pages 35–44. Springer, 2013.

[121] Bingxiong Lin, Adrian Johnson, Xiaoning Qian, Jaime Sanchez, and Yu Sun. Simultaneous tracking, 3d reconstruction and deforming point detection for stereoscope guided surgery. In *Augmented Reality Environments for Medical Imaging and Computer-Assisted Interventions*, pages 35–44. Springer, 2013.

[122] Bingxiong Lin, Yu Sun, and Xiaoning Qian. Dense surface reconstruction with shadows in mis. *IEEE Transactions on Biomedical Engineering*, 60(9):2411–2420, 2013.

[123] Bingxiong Lin, Yu Sun, Xiaoning Qian, Dmitry Goldgof, Richard Gitlin, and Yuncheng You. Video-based 3d reconstruction, laparoscope localization and deformation recovery for abdominal minimally invasive surgery: a survey. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 12(2):158–178, 2016.

[124] Bingxiong Lin, Yu Sun, Jaime E Sanchez, and Xiaoning Qian. Efficient vessel feature detection for endoscopic image analysis. *IEEE Transactions on Biomedical Engineering*, 62(4):1141–1150, 2015.

[125] Kyle Lindgren, Kevin Huang, and Blake Hannaford. Towards real-time surface tracking and motion compensation integration for robotic surgery. In *2017 IEEE/SICE International Symposium on System Integration (SII)*, pages 450–456. IEEE, 2017.

[126] Yaron Lipman and Thomas Funkhouser. Möbius voting for surface correspondence. *ACM Transactions on Graphics (TOG)*, 28(3):72, 2009.

[127] Guanyang Liu, Keke Lu, and Yuru Zhang. Haptic-based training for tank gunnery using decoupled motion control. *IEEE computer graphics and applications*, 33(2):73–79, 2013.

[128] Lingzhi Liu, Guanyang Liu, and Yuru Zhang. A novel haptic training method through skill decomposition. In *World Haptics*, pages 621–625, 2013.

[129] Manlu Liu, Qiang Ling, Jing Zhang, Liang Xu, Jiangmei Zhang, and Hua Zhang. Bilateral control of teleoperation manipulator based on virtual force aware guidance.

In *Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM), 2017 IEEE International Conference on*, pages 231–236. IEEE, 2017.

[130] Xavier Lladó, Alessio Del Bue, Arnau Oliver, Joaquim Salvi, and Lourdes Agapito. Reconstruction of non-rigid 3d shapes from stereo-motion. *Pattern Recognition Letters*, 32(7):1020–1028, 2011.

[131] Miguel Lourenço, Danail Stoyanov, and Joao P Barreto. Visual odometry in stereo endoscopy by using pearl to handle partial scene deformation. In *Workshop on Augmented Environments for Computer-Assisted Interventions*, pages 33–40. Springer, 2014.

[132] J. Dankelman M. K. Chmarra, C. A. Grimbergen. Systems for tracking minimally invasive surgical instruments. *Minimally Invasice Therapy Allied Technology*, 16(6):328–340.

[133] Ezio Malis. Hybrid vision-based robot control robust to large calibration errors on both intrinsic and extrinsic camera parameters. In *Control Conference (ECC), 2001 European*, pages 2898–2903. IEEE, 2001.

[134] Ezio Malis, Guillaume Morel, and François Chaumette. Robot control using disparate multiple sensors. *The International Journal of Robotics Research*, 20(5):364–377, 2001.

[135] Jonathan Kleefield Harsha Gopal Edward Reardon Bryan T. Ho Frederick A. Kuhn Marvin P. Fried, FACS. Image-guided endoscopic surgery: Results of accuracy and performance in a multicenter clinical study using an electromagnetic tracking system. *COSM Meeting in Orlando, Florida*, 107:594–601, 1997.

[136] Matthew Thomas Mason. Compliance and force control for computer controlled manipulators. 1979.

[137] KN McGuire, C De Wagter, K Tuyls, HJ Kappen, and GCHE de Croon. Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment. *Science Robotics*, 4(35):eaaw9710, 2019.

[138] Goeffrey J McLachlan. Mahalanobis distance. *Resonance*, 4(6):20–26, 1999.

[139] Ludovico Minati, Natsue Yoshimura, and Yasuharu Koike. Hybrid control of a vision-guided robot arm by eog, emg, eeg biosignals and head movement acquired via a consumer-grade wearable device. *IEEE Access*, 4:9528–9541, 2016.

[140] Rajineesh K Mishra, George B Hanna, Stuart I Brown, and Alfred Cuschieri. Optimum shadow-casting illumination for endoscopic task performance. *Archives of surgery*, 139(8):889–892, 2004.

[141] Guillaume Morel, Ezio Malis, and Sylvie Boudet. Impedance based combination of visual and force control. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 2, pages 1743–1748. IEEE, 1998.

[142] Monica Morgan, Ephrem O Olweny, and Jeffrey A Cadeddu. Less and notes instrumentation: future. *Current opinion in urology*, 24(1):58–65, 2014.

[143] Peter Mountney and Guang-Zhong Yang. Dynamic view expansion for minimally invasive surgery using simultaneous localization and mapping. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 1184–1187. IEEE, 2009.

[144] Peter Mountney and Guang-Zhong Yang. Motion compensated slam for image guided surgery. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 496–504. Springer, 2010.

[145] Raúl Mur-Artal and Juan D. Tardós. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[146] Yoshihiko Nakamura, Kousuke Kishi, and Hiro Kawakami. Heartbeat synchronization for robotic cardiac surgery. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, volume 2, pages 2014–2019. IEEE, 2001.

[147] Akio Namiki, Yoshihiro Nakabo, Idaku Ishii, and Masatoshi Ishikawa. High speed grasping using visual and force feedback. In *ICRA*, pages 3195–3200, 1999.

[148] Nikhil V Navkar, Zhigang Deng, Dipan J Shah, and Nikolaos V Tsekos. A framework for integrating real-time mri with robot control: application to simulated transapical cardiac interventions. *IEEE Transactions on Biomedical Engineering*, 60(4):1023–1033, 2013.

[149] P. L. Negre, F. Bonin-Font, and G. Oliver. Cluster-based loop closing detection for underwater slam in feature-poor regions. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2589–2595, May 2016.

[150] Bradley J Nelson and Pradeep K Khosla. The resolvability ellipsoid for visual servoing. In *computer society conference on computer vision and pattern recognition*, pages 829–829. IEEE, 1994.

[151] Bradley J Nelson and Pradeep K Khosla. Force and vision resolvability for assimilating disparate sensory feedback. *IEEE Transactions on Robotics and Automation*, 12(5):714–731, 1996.

[152] Bradley J Nelson, J Dan Morrow, and Pradeep Khosla. Improved force control through visual servoing. 1995.

[153] Philip Odonkor, Zachary Ball, and Souma Chowdhury. Distributed operation of collaborating unmanned aerial vehicles for time-sensitive oil spill mapping. *Swarm and Evolutionary Computation*, 46:52–68, 2019.

[154] Emidio Olivieri, Giacinto Barresi, Darwin G Caldwell, and Leonardo S Mattos. Haptic feedback for control and active constraints in contactless laser surgery: Concept, implementation, and evaluation. *IEEE transactions on haptics*, 11(2):241–254, 2018.

[155] John M Olson. Tactile display technologies as an enabler for space exploration operations. In *Aerospace Conference, 2007 IEEE*, pages 1–12. IEEE, 2007.

[156] Tomas Olsson, Johan Bengtsson, Rolf Johansson, and Henrik Malm. Force control and visual servoing using planar surface identification. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 4, pages 4211–4216. IEEE, 2002.

[157] Marco Paladini, Adrien Bartoli, and Lourdes Agapito. Sequential non-rigid structure-from-motion with the 3d-implicit low-rank shape model. In *European conference on computer vision*, pages 15–28. Springer, 2010.

[158] Abhilash Pandya, Luke Reisner, Brady King, Nathan Lucas, Anthony Composto, Michael Klein, and Richard Ellis. A review of camera viewpoint automation in robotic and laparoscopic surgery. *Robotics*, 3(3):310–329, 2014.

[159] Antonio Paolillo, Anastasia Bolotnikova, Kévin Chappellet, and Abderrahmane Kheddar. Visual estimation of articulated objects configuration during manipulation with a humanoid. In *System Integration (SII), 2017 IEEE/SICE International Symposium on*, pages 330–335. IEEE, 2017.

[160] Nikolaos P Papanikolopoulos and Pradeep K Khosla. Adaptive robotic visual tracking: Theory and experiments. *IEEE Transactions on Automatic Control*, 38(3):429–445, 1993.

[161] Marsic I. Burd R.S. Parlak, S. Activity recognition for emergency care using rfid. *6th International ICST Conference on Body Area Networks*, (1-936968-29-0), 2011.

[162] Pedro A Patlan-Rosales and Alexandre Krupa. Automatic palpation for quantitative ultrasound elastography by visual servoing and force control. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 2357–2362. IEEE, 2016.

[163] Pedro A Patlan-Rosales and Alexandre Krupa. A robotic control framework for 3-d quantitative ultrasound elastography. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 3805–3810. IEEE, 2017.

[164] Pedro A Patlan-Rosales and Alexandre Krupa. Strain estimation of moving tissue based on automatic motion compensation by ultrasound visual servoing. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 2941–2946. IEEE, 2017.

[165] Zhouhua Peng and Jun Wang. Output-feedback path-following control of autonomous underwater vehicles based on an extended state observer and projection neural networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(4):535–544, 2018.

[166] Jochen Penne, Kurt Höller, Michael Stürmer, Thomas Schrauder, Armin Schneider, Rainer Engelbrecht, Hubertus Feußner, Bernhard Schmauss, and Joachim Hornegger. Time-of-flight 3-d endoscopy. *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2009*, pages 467–474, 2009.

[167] Voros S. Hager G.D. Pezzementi, Z. Articulated object tracking by rendering concsistent appearance parts. *Robotics and Automation ICRA*, pages 3940–3947, 2009.

[168] Jorge Pomares and Fernando Torres. Movement-flow-based visual servoing and force control fusion for manipulation tasks in unstructured environments. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 35(1):4–15, 2005.

[169] Christian Potthast and Gaurav S Sukhatme. A probabilistic framework for next best view estimation in a cluttered environment. *Journal of Visual Communication and Image Representation*, 25(1):148–164, 2014.

[170] Fangbo Qin, Yangming Li, **Yun Hsuan Su**, De Xu, and Blake Hannaford. Surgical instrument segmentation for endoscopic vision with data fusion of rediction and kinematic pose. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9821–9827. IEEE, 2019.

[171] Liankui Qiu, Haitao Zhang, Wei Gao, and Yunjian Ge. Vision/force based robot manipulator servo control for uncertain plane surface tracking. In *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on*, pages 977–982. IEEE, 2008.

[172] Phillip Quin, Gavin Paul, Alen Alempijevic, Dikai Liu, and Gamini Dissanayake. Efficient neighbourhood-based information gain approach for exploration of complex 3d environments. In *2013 IEEE International Conference on Robotics and Automation*, pages 1343–1348. IEEE, 2013.

[173] Marc H Raibert and John J Craig. Hybrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 103(2):126–133, 1981.

[174] Martin Reuter. Hierarchical shape segmentation and registration via topological features of laplace-beltrami eigenfunctions. *International Journal of Computer Vision*, 89(2-3):287–308, 2010.

[175] Rogério Richa, Antônio PL Bó, and Philippe Poignet. Robust 3d visual tracking for robotic-assisted cardiac interventions. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 267–274. Springer, 2010.

[176] Rogério Richa, Antônio PL Bó, and Philippe Poignet. Towards robust 3d visual tracking for motion compensation in beating heart surgery. *Medical Image Analysis*, 15(3):302–315, 2011.

[177] Rogério Richa, Philippe Poignet, and Chao Liu. Three-dimensional motion tracking for beating heart surgery using a thin-plate spline deformable model. *The International Journal of Robotics Research*, 29(2-3):218–230, 2010.

[178] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision*, volume 1, pages 430–443, May 2006.

[179] Sumantra Dutta Roy, Santanu Chaudhury, and Subhashis Banerjee. Active recognition through next view planning: a survey. *Pattern Recognition*, 37(3):429–446, 2004.

[180] Angelica Ruszkowski, Omid Mohareri, Sam Lichtenstein, Richard Cook, and Septimiu Salcudean. On the feasibility of heart motion compensation on the davinci® surgical robot for coronary artery bypass surgery: Implementation and user studies. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4432–4439. IEEE, 2015.

[181] Angelica Ruszkowski, Caitlin Schneider, Omid Mohareri, and Septimiu Salcudean. Bimanual teleoperation with heart motion compensation on the da vinci® research kit: Implementation and preliminary experiments. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4101–4108. IEEE, 2016.

[182] Arthur C Sanderson and Lee E Weiss. Adaptive visual servo control of robots. In *Robot vision*, pages 107–116. Springer, 1983.

[183] Matthias Scheutz, Paul Schermerhorn, and Peter Bauer. The utility of heterogeneous swarms of simple uavs with limited sensory capacity in detection and tracking tasks. In *Proceedings 2005 IEEE Swarm Intelligence Symposium, 2005. SIS 2005.*, pages 257–264. IEEE, 2005.

[184] Robert J Schneider, Douglas P Perrin, Nikolay V Vasilyev, Gerald R Marx, J Pedro, and Robert D Howe. Real-time image-based rigid registration of three-dimensional ultrasound. *Medical image analysis*, 16(2):402–414, 2012.

[185] John Schulman, Alex Lee, Jonathan Ho, and Pieter Abbeel. Tracking deformable objects with point clouds. In *2013 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1130–1137. IEEE, 2013.

[186] William R Scott. Model-based view planning. *Machine Vision and Applications*, 20(1):47–69, 2009.

[187] William R Scott, Gerhard Roth, and Jean-François Rivest. View planning for automated three-dimensional object reconstruction and inspection. *ACM Computing Surveys (CSUR)*, 35(1):64–96, 2003.

[188] Andrei Sharf, Thomas Lewiner, Ariel Shamir, and Leif Kobbelt. On-the-fly curve-skeleton computation for 3d shapes. In *Computer Graphics Forum*, volume 26, pages 323–328. Wiley Online Library, 2007.

[189] Eitan Sharon and David Mumford. 2d-shape analysis using conformal mapping. *International Journal of Computer Vision*, 70(1):55–75, 2006.

[190] M Silvestri, T Ranzani, A Argiolas, M Vatteroni, and A Menciassi. A multi-point of view 3d camera system for minimally invasive surgery. *Sensors and Actuators A: Physical*, 202:204–210, 2013.

[191] Richard J. Radkea Siqi Chena, Daniel Cremersb. Image segmentation with one shape prior – a template-based formulation. *Image and Vision Computing*, 2012.

[192] Kai-Tai Song and Chun-Ju Chen. Autonomous and stable tracking of endoscope instrument tools with monocular camera. In *2012 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 39–44. IEEE, 2012.

[193] Nathaniel J Soper, Lee L Swanström, and Steve Eubanks. *Mastery of endoscopic and laparoscopic surgery*. Lippincott Williams & Wilkins, 2008.

[194] James D Stefansic, Alan J Herline, Yu Shyr, William C Chapman, J Michael Fitzpatrick, Benoit M Dawant, and Robert L Galloway. Registration of physical space to laparoscopic image space for use in minimally invasive hepatic surgery. In *5th IEEE EMBS International Summer School on Biomedical Imaging, 2002.*, pages 12–pp. IEEE, 2002.

[195] Danail Stoyanov, Marco Visentini Scarzanella, Philip Pratt, and Guang-Zhong Yang. Real-time stereo reconstruction in robotically assisted minimally invasive surgery. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 275–282. Springer, 2010.

[196] Danail Stoyanov and Guang Zhong Yang. Removing specular reflection components for robotic assisted laparoscopic surgery. In *IEEE International Conference on Image Processing 2005*, volume 3, pages III–632. IEEE, 2005.

[197] PB Sujit and Randy Beard. Multiple uav path planning using anytime algorithms. In *2009 American Control Conference*, pages 2978–2983. IEEE, 2009.

[198] Jochen Süßmuth, Marco Winter, and Günther Greiner. Reconstructing animated meshes from time-varying point clouds. In *Computer Graphics Forum*, volume 27, pages 1469–1476. Wiley Online Library, 2008.

[199] Andrea Tagliasacchi, Hao Zhang, and Daniel Cohen-Or. Curve skeleton extraction from incomplete point cloud. In *ACM Transactions on Graphics (TOG)*, volume 28, page 71. ACM, 2009.

[200] Youhua Tan, Dong Sun, Wenhao Huang, Jinping Cheng, and Shuk Han Cheng. Robotic manipulation of human red blood cells with optical tweezers for cell property characterization. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4269–4274. IEEE, 2010.

[201] WT Luke Teacy, Jing Nie, Sally McClean, Gerard Parr, Stephen Hailes, Simon Julier, Niki Trigoni, and Stephen Cameron. Collaborative sensing by unmanned aerial vehicles. 2009.

[202] **Y. H. Su**, Adnan Munawar, Anton Deguet, Andrew Lewis, Kyle Lindgren, Yangming Li, Russell H. Taylor, Gregory S. Fischer, Blake Hannaford, and Peter Kazanzides. Collaborative robotics toolkit (crtk): Open software framework for surgical robotics research. In *2020 International Conference on Robotics Computing (IRC)*. IEEE, 2020.

[203] **Yun Hsuan Su**, Issac Huang, Kevin Huang, and Blake Hannaford. Multicamera 3d reconstruction of dynamic surgical cavities: Non-rigid registration and point classification. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019.

[204] **Yun Hsuan Su**, Kevin Huang, and Blake Hannaford. Real-time vision-based surgical tool segmentation with robot kinematics prior. In *Medical Robotics (ISMR), 2018 International Symposium on*, pages 1–6. IEEE, 2018.

[205] **Yun Hsuan Su**, Kevin Huang, and Blake Hannaford. Multicamera 3d reconstruction of dynamic surgical cavities: Camera grouping and pair sequencing. In *2019 International Symposium on Medical Robotics (ISMR)*, pages 1–7. IEEE, 2019.

[206] **Yun Hsuan Su**, Kevin Huang, and Blake Hannaford. Multicamera 3d reconstruction of dynamic surgical cavities: Autonomous optimal camera viewpoint adjustments. In *2020 International Symposium on Medical Robotics (ISMR)*. IEEE, 2020.

[207] **Yun Hsuan Su**, Kyle Lindgren, Kevin Huang, and Blake Hannaford. A comparison of surgical cavity 3d reconstruction methods. In *2020 International Symposium on In System Integration (SII)*. IEEE, 2020.

[208] **Yun Hsuan Su**, Yana Sosnovskaya, Kevin Huang, and Blake Hannaford. Securing robot-assisted minimally invasive surgery through perception complementarities. In *2020 International Conference on Robotics Computing (IRC)*. IEEE, 2020.

[209] John Tisdale, Allison Ryan, Zu Kim, David Tornqvist, and J Karl Hedrick. A multiple uav system for vision-based search and localization. In *2008 American Control Conference*, pages 1985–1990. IEEE, 2008.

[210] Toshio Tsuji, Hiromasa Akamatsu, and Makoto Kaneko. Non-contact impedance control for redundant manipulators using visual information. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volume 3, pages 2571–2576. IEEE, 1997.

[211] Hakan Urey, Kishore V Chellappan, Erdem Erden, and Phil Surman. State of the art in stereoscopic and autostereoscopic displays. *Proceedings of the IEEE*, 99(4):540–555, 2011.

[212] P Valdastri, C Quaglia, E Buselli, A Arezzo, N Di Lorenzo, M Morino, A Menciassi, P Dario, et al. A magnetic internal mechanism for precise orientation of the camera in wireless endoluminal applications. *Endoscopy*, 42(6):481, 2010.

[213] Oliver Van Kaick, Hao Zhang, Ghassan Hamarneh, and Daniel Cohen-Or. A survey on shape correspondence. In *Computer Graphics Forum*, volume 30, pages 1681–1707. Wiley Online Library, 2011.

[214] Juan Irving Vásquez and L Enrique Sucar. Next-best-view planning for 3d object reconstruction under positioning error. In *Mexican International Conference on Artificial Intelligence*, pages 429–442. Springer, 2011.

[215] J Irving Vasquez-Gomez, L Enrique Sucar, Rafael Murrieta-Cid, and Efrain Lopez-Damian. Volumetric next-best-view planning for 3d object reconstruction with positioning error. *International Journal of Advanced Robotic Systems*, 11(10):159, 2014.

[216] René Vidal and Daniel Abretske. Nonrigid shape and motion from multiple perspective views. *Computer Vision–ECCV 2006*, pages 205–218, 2006.

[217] Luigi Villani and Joris De Schutter. Force control. In *Springer handbook of robotics*, pages 161–185. Springer, 2008.

[218] Rakesh V Vohra and Nicholas G Hall. A probabilistic analysis of the maximal covering location problem. *Discrete Applied Mathematics*, 43(2):175–183, 1993.

[219] Nicholas C Von Sternberg, Atilla Kilicarslan, Nikhil V Navkar, Zhigang Deng, Karolos Grigoriadis, and Nikolaos V Tsekos. Implementation of a force-feedback interface for robotic assisted interventions with real-time mri guidance. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 4869–4874. IEEE, 2013.

[220] Gustaaf J Vrooijink, Alper Denasi, Jan G Grandjean, and Sarthak Misra. Model predictive control of a robotically actuated delivery sheath for beating heart compensation. *The International journal of robotics research*, 36(2):193–209, 2017.

[221] Michael Wand, Bart Adams, Maksim Ovsjanikov, Alexander Berner, Martin Bokeloh, Philipp Jenke, Leonidas Guibas, Hans-Peter Seidel, and Andreas Schilling. Efficient reconstruction of nonrigid shape and motion from real-time 3d scanner data. *ACM Transactions on Graphics (TOG)*, 28(2):15, 2009.

[222] Alexander Warren, Peter Mountney, David Noonan, and Guang-Zhong Yang. Horizon stabilized—dynamic view expansion for robotic assisted surgery (hs-dve). *International journal of computer assisted radiology and surgery*, 7(2):281–288, 2012.

[223] Oliver Weede, Holger Mönnich, B Müller, and Heinz Wörn. An intelligent and autonomous endoscopic guidance system for minimally invasive surgery. In *2011 IEEE International Conference on Robotics and Automation*, pages 5762–5768. IEEE, 2011.

[224] Chao-Huang Wei and Shang-Ping Chen. Vr-based teleautonomous system for agv path guidance. In *Control, Automation, Robotics and Vision, 2002. ICARCV 2002. 7th International Conference on*, volume 3, pages 1262–1267. IEEE, 2002.

[225] LEEE Weiss, ARTHURC Sanderson, and CHARLESP Neuman. Dynamic sensor-based control of robots with visual feedback. *IEEE Journal on Robotics and Automation*, 3(5):404–417, 1987.

[226] Daniel E Whitney. Historical perspective and state of the art in robot force control. *The International Journal of Robotics Research*, 6(1):3–14, 1987.

[227] Gerhard J Woeginger. Exact algorithms for np-hard problems: A survey. In *Combinatorial optimization—eureka, you shrink!*, pages 185–207. Springer, 2003.

[228] PA Woerdeman, PWA Willems, HJ Noordmans, and JW van der Sprenkel. The analysis of intraoperative neurosurgical instrument movement using a navigation log-file. *International Journal of Medical Robotics and Computer Assisted Surgery*, 2(2):139–145, 2006.

[229] Changchang Wu. Towards linear-time incremental structure from motion. In *3D Vision-3DV 2013, 2013 International conference on*, pages 127–134. IEEE, 2013.

[230] Di Xiao, Bijoy K Ghosh, Ning Xi, and Tzyh Jong Tarn. Intelligent robotic manipulation with hybrid position/force control in an uncalibrated workspace. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 2, pages 1671–1676. IEEE, 1998.

[231] Jing Xiao and Takeo Kanade. Uncalibrated perspective reconstruction of deformable structures. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1075–1082. IEEE, 2005.

[232] Nan Feng Xiao and S Nahavandi. Learning-based visual and force servoing control of a robot in an unknown environment. In *Control and Automation, 2002. ICCA. Final Program and Book of Abstracts. The 2002 International Conference on*, pages 99–100. IEEE, 2002.

[233] Zhengtong Xu, **Yun-Hsuan Su**, Haonan Peng, and Blake Hannaford. learning-based soft tissue Motion prediction and motion compensation for robotic assisted minimally invasive surgery. In *Intelligent Robots and Systems (IROS), 2020 IEEE/RSJ International Conference on*. IEEE, 2020.

[234] Jingyu Yang, Ke Li, Kun Li, and Yu-Kun Lai. Sparse non-rigid registration of 3d shapes. In *Computer Graphics Forum*, volume 34, pages 89–99. Wiley Online Library, 2015.

[235] Shan Yang. *Non-Rigid Body Mechanical Property Recovery From Images and Videos*. PhD thesis, The University of North Carolina at Chapel Hill, 2018.

[236] Gang Yin, Woong Kyu Han, Stephen Faddegon, Yung Khan Tan, Zhuo-Wei Liu, Ephrem O Olweny, Daniel J Scott, and Jeffrey A Cadeddu. Laparoendoscopic single site (less) in vivo suturing using a magnetic anchoring and guidance system (mags) camera in a porcine model: impact on ergonomics and workload. *Urology*, 81(1):80–84, 2013.

[237] Tsuneo Yoshikawa. Dynamic hybrid position/force control of robot manipulators–description of hand constraints and calculation of joint driving force. *IEEE Journal on Robotics and Automation*, 3(5):386–392, 1987.

[238] Lingtao Yu, Zhengyu Wang, Liqiang Sun, Wenjie Wang, and Tao Wang. A kinematics method of automatic visual window for laparoscopic minimally invasive surgical robotic system. In *2013 IEEE International Conference on Mechatronics and Automation*, pages 997–1002. IEEE, 2013.

[239] Yun Zeng, Chaohui Wang, Yang Wang, Xianfeng Gu, Dimitris Samaras, and Nikos Paragios. Dense non-rigid surface registration using high-order graph matching. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 382–389. IEEE, 2010.

[240] Hao Zhang, Alla Sheffer, Daniel Cohen-Or, Quan Zhou, Oliver Van Kaick, and Andrea Tagliasacchi. Deformation-driven shape correspondence. In *Computer Graphics Forum*, volume 27, pages 1431–1439. Wiley Online Library, 2008.

[241] Danping Zou and Ping Tan. Coslam: Collaborative visual slam in dynamic environments. *IEEE transactions on pattern analysis and machine intelligence*, 2012.

[242] Danping Zou, Ping Tan, and Wenxian Yu. Collaborative visual slam for multiple agents: A brief survey. *Virtual Reality & Intelligent Hardware*, 1(5):461–482, 2019.

[243] Tamás Haidegger Árpád Takács, Imre J. Rudas. Reaction force and surface deformation estimation based on heuristic tissue models. *Takács, Á., Rudas, I.J., Haidegger, T. Med Biol Eng Comput*, 2016.

Note: Publications under review are marked with 🏷; those in preparation are marked with 🏷.

Appendix A

LIGHTING REFACTORIZATION

The camera used in this work has an automatic brightness adjustment feature, which can be disturbing because the lighting is inconsistent throughout the video sequence. Some pre-processing of brightness equalization is needed before feeding the images into the three reconstruction algorithms. This is essential because without brightness compensation, it will be difficult for the feature matching algorithm to find correspondence. Using the brightness of the first image I^1 as our standard brightness, Eq.A.1 is applied to all successive images.

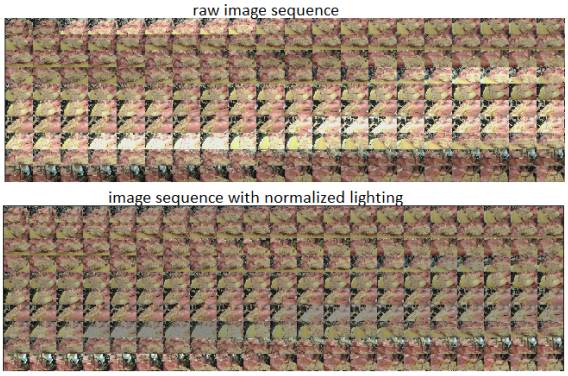


Figure A.1: The image frame samples before and after brightness re-factorization.

$$\begin{aligned} brightness\_scale &= \frac{\sum_{x=1}^{Nx}\sum_{y=1}^{Ny}\sum_{c=1}^{3} I^1_{xyc}}{\sum_{x=1}^{Nx}\sum_{y=1}^{Ny}\sum_{c=1}^{3} I^i_{xyc}} \\ I^i &= I^i .* brightness\_scale \end{aligned} \tag{A.1}$$

Where Nx and Ny are respectively the number of rows and columns in the image and c is the index of the three channels in the RGB color space.

The ith image in the video sequence is I^i and operator $.*$ denotes pixel-wise multiplication of the image and the scaling factor. Rhe raw image samples are shown on the top half of Figure A.1, while the image samples after brightness modification are on the bottom half.

Appendix B

INTEGRATED VISION AND FORCE PAPER LIST

This is a table of all the papers I reviewed in Chapter 6.

Ref	Conf./Year	Role of Force	Role of Vision	Remark
[162]	IROS, 2016	Force sensor: acquire the contact force	Image modality: ultrasound sensors	Robotic palpation system/elastography modality and force measurement as input/probe control
[163]	ICRA, 2017	Force sensor: acquire the contact force	Image modality: ultrasound sensors	compression motion with applied force control/ visual servoing control for autonomous palpation
[164]	IROS, 2017	Force sensor: acquire the contact force	Image modality: ultrasound sensors	mage-based visual servoing, force control and non-rigid motion estimation/find elastic property of a moving tissue via a robot-assisted system
[32]	ICRA, 2015	Force sensor: acquire the contact force	Image modality: ultrasound sensors	ultrasound-based visual servoing framework/ positioning optimization of ultrasound probe/ also utilizes for measurements

[71]	Transactions on Cybernetics, 2016	Force sensor: acquire the contact force	Visual sensor: environment understanding	dynamic positioning of a fully actuated underwater vehicle/NN to reduce velocity tracking error under uncertainty of thrust force/velocity control, with external force measurements
[165]	Transactions on Cybernetics, 2017	Force sensor: acquire the contact force	Visual sensor:reconstruct 3D scene/ environment	output-feedback path-following control of underactuated autonomous underwater vehicles/path-following control, with external force measurements
[129]	CIS RAM, 2017	Force feedback: target guidance	Visual sensor: kinect 3D sensing	real-time virtual force guidance/PHANTOM force feedback device/Kinect image equipment
[154]	Transactions on Haptics, 2017	Force feedback: fictitious force	Visual sensor: stereoscopic visualization	robot-assisted laser microsurgery/fictitious force feedback is created through stereoscopic visualization
[97]	Transactions on Neural Systems and Rehabilitation, 2017	Force feedback:Target guidance	Visual sensor: reconstruct object pose and robot joints configuration	force-field controller provides haptic feedback/guidance force on the user's pelvis/robot assisted stroke patent recovery via visual servoing/

[139]	IEEE Access, 2016	Force sensor:acquire the contact force	Visual sensor: environment understanding	Electrode wearable biosignal controller device/6DOF robot arm with visual and force sensors attached/participants control the robot arm using the electrode wearable device
[65]	World Automation Congress (WAC), 2016	Force feedback: haptic guidance	Visual feedback: virtual environment understanding	Virtual Reality (VR) cell injection training/haptic guidance is provided in the form of virtual fixtures (VFs)/
[104]	SMC, 2015	Force feedback: haptic guidance	Visual feedback: virtual environment understanding	the influence of mental rotations on task performance is studied/ mental rotations decrease teleoperation performance despite the addition of direct force feedback
[15]	FSKD, 2013	Force feedback: haptic guidance	Visual feedback: virtual environment understanding	force feedback to improve the perception of the operators in virtual training systems/ fuzzy-PID controller
[219]	ICRA, 2013	Force feedback: haptic guidance	Image modality: magnetic resonance imaging	forbidden region guided fixtures (FRVF)/human-in-the-loop control of image-guided procedures /with haptic force-feedback devices (FFD)/force guidance from real-time magnetic resonance imaging

[128]	World Haptics Conference (WHC), 2013	Force feedback: haptic guidance	Visual feedback: virtual environment understandin	Skill training with haptic guidance/2-DOF control training experiment/novel training method through skill decomposition
[93]	ISIE, 2013	Force feedback: haptic guidance	Visual feedback: virtual environment understandin	haptic bilateral control method with vision-based guidance/translate the visual information to assistive force/ visual compliance controller based on force control/great block diagram
[127]	Computer Graphics and Applications, 2013	Force feedback: haptic guidance sent via joystick	Visual feedback: virtual environment understanding	decoupled motion control/the force exerted by the trainee's device doesn't directly correct the trainee's hand action. Instead, the trainee regulates his manipulation to cooperate with the expert by feeling motion feedback from the expert
[110]	Transactions on Robotics, 2012	Modeled force: does not have a force sensor	Visual sensor: environment understanding	proximity query (PQ) formulation: derivation of robot motion/ haptic guidance is provided to prevent excessive force applied to the tissue by the robot/

[148]	Transactions on Biomedical Engineering, 2012	Force feedback: Generated from rtMRI derived virtual fixture	Visual sensor: (rtMRI imaging)	image-guided robot assistance/ 20 Hz for visualization and 1000 Hz for force feedback/ simulated Transapical aortic valve implantation with a virtual robotic manipulator/completion time improves with force feedback included
[39]	CIS, 2011	Modeled force: does not have a force sensor	Visual sensor: environment understandin	UAV: Optical visual guidance system/ Direct force control
[200]	IROS, 2010	Modeled force: no force sensor, estimated by cell deformation	Visual sensor: controlling relative position between cell and tweezer	optical robotic-tweezers system for manipulation of human red blood cells (RBCs)/ visual guidance and position feedback control/ force calibration and image processing to find stretching force and the induced deformation correlation/ characterize the biomechanical properties based on the cell mechanical model/related to research
[20]	ICCAE, 2009	Force feedback: sent through joystick	Visual feedback: virtual environment understanding	virtual Persian handwriting learning system/ full guidance mode by force guidance/ partial guidance, combination of haptics and visual feedback guides the operator

[96]	RO-MAN, 2010	Force sensor: acquire the contact force	Visual sensor: environment understanding	top-down organization: gaze turns first, then the head turns, and the walking direction and the feet follow/ humanoid robots based mainly on ZMP which concerns the force acting between the foot and the ground indicating bottom-up Control and organization
[79]	EuroHaptics conference, 2009	Force feedback: sent through joystick	Visual feedback: virtual environment understanding	performance improvement that force feedback can provide in a virtual environment/ haptic guidance did not take control away from the user/compare how much force feedback improve the performance of tasks in virtual world
[57]	ICRA, 2008	Force feedback: Target guidance	Visual sensor: environment understanding	dissipative haptic displays/ Brake Actuated Manipulator (BAM)/path guidance controller comparison/three controllers, velocity ratio, force cancelling, and force mapping/with and without visual feedback

[155]	Aerospace Conference, 2007	Force feedback:Tactile display	Visual sensor: environment understanding	Tactile situation awareness system (TSAS) /tactile displays decreased overall workload/improve performance for flight-test for simulated Space Shuttle
[224]	ICARCV, 2002	Force feedback: sent through joystick	Visual feedback: virtual wall in the 3D virtual scene	VR-based AGV path guidance/ virtual force as force feedback/virtual 3D scene as visual feedback/ tele-autonomous control loop
[5]	ROBIO, 2017	Force feedback: Vibrotactile	Visual sensor: Kinect motion sensing	Skin stroking: painless and comfortable form of a tactile display/ haptic feedback assistive guidance for the blind/ Vibro-tactile feedback mechanism
[118]	CAC, 2017	Force feedback:Target guidance	Image modality: ultrasound sensors	intelligent wheelchair/ path planning and random avoidance/ local environment modeling/ Visual Force Field arithmetic is used to achieved target guidance/ Graph method is used for modeling whole environment

[16]	IROS, 2017	Modeled force: does not have a force sensor	Visual sensor:stable visual info during locomotion	fusing visual and force information/ tracking trajectories/ employs variable weights for each sensor system/movement flow-based visual servoing/ detection of interaction forces processed by a Kalman filter
[168]	Transactions on Systems, Man, and Cybernetics, 2005	Force sensor: acquire the contact force	Visual sensor: reconstruct object pose and robot joints config	fusing visual and force information/ tracking trajectories/ employs variable weights for each sensor system/movement flow-based visual servoing/ detection of interaction forces processed by a Kalman filter/
[232]	ICCA, 2002	Force sensor:acquire the contact force	Visual sensor:reconstruct object pose and robot joints config	learning control approach is presented for visual and force servoing of a robot in an unknown environment/mapping from image features to joints/ force servoing: an impedance control law is obtained
[156]	ICRA, 2002	Force sensor:acquire the contact force	Visual sensor:surface following	combining direct force control and visual servoing in the presence of unknown planar surfaces/ force feedback control loop/ vision based trajectory as a feed-forward signal/robot arm draw lines while maintaining contact force

[152]	American Control Conference, 1995	Force sensor: acquire the contact force	Visual sensor: reducing alignment uncertainties between objects	three different strategies which combine force and vision within the feedback loop of a manipulator/ traded control, hybrid control, and shared control
[10]	MFI, 1999	Force sensor: achieve force controlled contact	Visual sensor: controlling relative position of force probe w.r.t. object	combine force control and visual servoing in an uncalibrated workspace/vision and force sensors both on manipulator/feedforward/fused control method
[87]	IEEE Robotics and Automation Magazine, 1998	Force sensor: acquire the contact force	Visual sensor: Unknown constraint surface estimation	adaptive robot controller/ robot is visually guided/contacting task in unknown environment/force servoing task: force at the tip of the manipulator converge to the desired value/ visual servoing task: to make the image features converge to given desired trajectories
[171]	WCICA, 2008	Force sensor: acquire the contact force	Stereo vision: normal direction of plane surface	uncertain plane surface tracing based on vision and force sensing/validated by simulation with MatLab robotics toolkit

[159]	SII/2017	Force sensor: acquire the contact force	Visual sensor: reconstruct object pose and robot joints config.	In simulation/virtual visual servoing/open paper drawer for printer/ estimating the configuration of an articulated object to be manipulated by a humanoid robot
[37] [38]	IROS, 2009 CASE, 2010	Force sensor: acquire the contact force	Visual sensor: environment understanding	trim-and-final assembly/ Robot assisted wheel loading process/ force control and visual servoing loop/ estimate assembly time and tracking error

: Denotes the underwater vehicle applications.
: Denotes the unmanned aerial vehicle applications.
: Denotes the space exploration applications.
: Denotes the haptic virtual simulation applications.
: Denotes the industrial robotic applications.
: Denotes the medical applications.
: Denotes the other applications.