Towards Autonomous Robot-Assisted Surgery

BY

Ki-Hwan Oh
B.S., Sung Kyun Kwan University, 2018

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois Chicago, 2026

Chicago, Illinois

Defense Committee:
Miloš Žefran, Chair and Advisor
Ahmet Enis Cetin
Shuo Han
Pier Cristoforo Giulianotti, Department of Surgery
Liaohai Chen, Worcester Polytechnic Institute

Accessibility Statement

An accessible EPUB version of this document is available at [URL] or can be obtained by contacting [contact details].

Dedication

To my parents,
whose love and quiet sacrifice
carried me through every step of this journey;

and to the friends and collaborators
who walked beside me along the way —
sharing their time, their insight,
and their belief in this work
when mine sometimes wavered.

This dissertation is as much yours as it is mine.

Acknowledgments

I would like to express my sincere gratitude to my advisor, Miloš Žefran, for his guidance, encouragement, and steady support throughout my doctoral studies. I am also deeply grateful to my former advisor, Liaohai Chen, whose mentorship and early support helped shape the direction of this research.

I am thankful to my colleague, Leonardo Borgioli, for his collaboration, technical support, and many valuable discussions throughout this work. I also thank the members of the Surgical Innovation and Training Lab for their feedback, collaboration, and for creating an environment in which this research could grow.

I would like to thank my committee members, Ahmet Enis Cetin, Shuo Han, and Pier Cristoforo Giulianotti, for their time, expertise, and thoughtful feedback on this dissertation.

Finally, I would like to thank my family and friends for their constant encouragement, patience, and support throughout this journey.

KO

Contribution of Authors

This dissertation was prepared by Ki-Hwan Oh under the guidance of Miloš Žefran.

Ki-Hwan Oh led the formulation of the research problems, the design and implementation of the experimental systems, the collection and annotation of data, the development and training of the models, the analysis of results, and the writing of this dissertation. Miloš Žefran provided research direction, technical feedback, and manuscript review throughout the project.

Leonardo Borgioli contributed to collaborative discussions, data collection, experimental development, and model training for the works incorporated into this dissertation. Members of the Surgical Innovation and Training Lab provided surgical insight, procedural feedback, and support during data collection and evaluation.

Portions of this dissertation are based on collaborative publications cited throughout the text. The integration of those works into a unified dissertation, along with the presentation and interpretation of the material, is the responsibility of Ki-Hwan Oh.

Contents

List of Figures

List of Tables

List of Abbreviations

AdamW	Adam with Decoupled Weight Decay
AP	Average Precision
CDF	Cumulative Distribution Function
COCO	Common Objects in Context
CRCD	Comprehensive Robotic Cholecystectomy Dataset
BCE	Binary Cross-Entropy
DoF	Degrees of Freedom
DT2-kpt	Detectron2 model for object keypoint detection
DT2-seg	Detectron2 model for instance segmentation
dVRK	da Vinci Research Kit
ECM	Endoscope Camera Manipulator
ESU	Electrosurgical Unit
FBF	Fenestrated Bipolar Forceps
FNN	Feedforward Neural Network
FP16	Half-Precision Floating Point
FPN	Feature Pyramid Network
GELU	Gaussian Error Linear Unit
GPU	Graphics Processing Unit
HRSV	High Resolution Stereo Viewer
IBVS	Image-Based Visual Servoing
ICG	Indocyanine Green
IoU	Intersection over Union
L-GBM	Light Gradient Boosting Machine
mAP	mean Average Precision
LND	Large Needle Driver
MaskDINO	Mask Detection with Transformers
MLP	Multilayer Perceptron
MOSFET	Metal-Oxide-Semiconductor Field-Effect Transistor
MSE	Mean Squared Error
MTM	Master Tool Manipulator
OCR	Optical Character Recognition
PBVS	Position-Based Visual Servoing
PCA	Principal Component Analysis
PCH	Permanent Cautery Hook
PoE	Product of Exponentials
PSM	Patient Side Manipulator
RAS	Robotic-Assisted Surgery

ResNet	Residual Network
RF	Random Forest
RGB	Red, Green, Blue
RMSE	Root Mean Square Error
ROS2	Robot Operating System 2
SAM	Segment Anything Model
SAM2	Segment Anything Model 2
SGM	Semi-Global Matching
SLSQP	Sequential Least Squares Programming
SUJ	Setup Joints
SVM	Support Vector Machine
TST	Time Series Transformer
ViT	Vision Transformer
YOLO11-pose	YOLO11 model for object keypoint detection
YOLO11-seg	YOLO11 model for instance segmentation

Notation

Bold lowercase letters denote vectors and bold uppercase letters denote matrices. The following mathematical notations are used throughout this dissertation:

g_{ab}	Homogeneous transformation matrix from frame A to frame B.
ζ_i or ξ_i	Twist at i-th joint, composed of translational velocity v and rotational velocity ω, $[v \quad \omega]^T$.
$\hat{\zeta}$	Twist matrix of joint twist ζ, $\begin{bmatrix} \hat{\omega} & v \\ 0 & 0 \end{bmatrix}$, where $\hat{\omega}$ is the skew-symmetric matrix of the rotational velocity component ω in the joint twist ζ.
θ_i or φ_i	The joint values, angles in radians for revolute joints, and displacement along the axis in meters for prismatic joints, for i-th joint.

Summary

Robotic surgery promises enhanced precision and adaptability over traditional surgical methods, along with the possibility of automating surgical interventions—resulting in reduced stress on the surgeon, better surgical outcomes, and lower costs. Cholecystectomy, the surgical removal of the gallbladder, serves as an ideal model procedure for automation due to its distinct anatomical features, standardized maneuvers, and high clinical volume.

This dissertation makes three contributions toward automating robotic cholecystectomy using the da Vinci surgical system with the da Vinci Research Kit (dVRK).

First, we develop a vision-based autonomous dissection framework that enables the da Vinci robot to dissect the gallbladder from the liver using only stereo endoscopic images. The framework evolves through two versions: an initial single-arm system with offline trajectory planning, and an upgraded bimanual system with automatic grasping, tissue stretching, and online boundary-guided dissection. The perception pipeline progresses from Detectron2 through MaskDINO to YOLO11, with the introduction of a liver bed segmentation class that supports boundary tracking after repeated rounds of energy delivery and lays the groundwork for future multi-round dissection. In single-cycle trials, the upgraded system achieves a 3.3× speed improvement and submillimeter boundary tracking precision.

Second, we create and release the Comprehensive Robotic Cholecystectomy Dataset (CRCD), a large-scale multimodal dataset recorded during *ex vivo* pseudo-cholecystectomy procedures on porcine livers. The CRCD combines stereo endoscopic videos, full kinematic data for all robot arms and console manipulators, pedal signals, and tissue segmentation and instrument keypoint annotations. With over 755,000 stereo frames from seven surgeons with documented experience levels, the CRCD provides a multimodal data foundation for perception, control, and learning on a real surgical procedure.

Third, we introduce a baseline framework for predicting instrument kinematics from stereo endoscopic images. Per-instrument Vision Transformer backbones are pre-trained on instrument segmentation masks and paired with lightweight stereo pose estimation heads that regress end-effector translation. Evaluated on held-out CRCD videos, the models achieve mean translation errors near one centimeter (0.94 cm for PCH and 1.13 cm for FBF), establishing a foundation for full six-degree-of-freedom instrument pose estimation from surgical video alone.

Together, these contributions show that vision-based dissection is feasible, provide the data infrastructure needed to train and evaluate perception and control models, and indicate that instrument state can be recovered from images without relying solely on robot kinematics.

CHAPTER 1

Introduction

This chapter provides the historical context for robotic surgery, introduces the da Vinci surgical system and the da Vinci Research Kit platform, describes the robotic cholecystectomy procedure that serves as the clinical testbed for this dissertation, and reviews related work in surgical automation, perception, datasets, and instrument pose estimation. The chapter concludes with a statement of thesis contributions and an overview of the dissertation organization.

1. History of Surgical Robots

The roots of robotic surgery can be traced back to 1985, when the PUMA 200 robot was used by Kwoh et al. [**5**] to perform CT-guided stereotactic neurosurgical biopsies; the same arm was later adapted for urological and prostate procedures [**6**, **7**]. Shortly thereafter, the Robodoc system [**6**, **8**]—developed at IBM Research in collaboration with the University of California at Davis—performed the first robot-assisted total hip replacement on a human patient in November 1992 at Sutter General Hospital, marking the first time that a robot carried out tissue-altering surgery on a human in the United States, although full FDA 510(k) clearance for Robodoc would not arrive until 2008. The 1990s also saw a shift to master–slave architectures that allowed a surgeon to control robotic movements from a separate console. This paradigm enabled the creation of AESOP [**9**] (1993), a voice-controlled robotic arm used to position the endoscope during an operation. Shortly thereafter, the ZEUS system [**10**] (1994) was introduced, allowing a surgeon to control robotic arms equipped with surgical instruments, demonstrating the feasibility of telepresence surgery.

Intuitive Surgical was established in 1995 and developed successive prototypes, first *Lenny* and then *Mona* [**11**, **12**]. Intuitive came to dominate the surgical robotics landscape after

(A) PUMA (B) ZEUS (C) AESOP

FIGURE 1.1. Historical roots of robotic surgery devices: PUMA, ZEUS, and AESOP.

launching the da Vinci system in 1998. The da Vinci was approved by the FDA for general laparoscopic procedures in 2000, becoming the first operative surgical robot to receive full FDA clearance for general soft-tissue surgery in the United States—predating Robodoc's own 510(k) clearance, which only arrived in 2008 despite its earlier clinical use under trial protocols. Robotic surgery systems such as AESOP, ZEUS, and da Vinci enhance the surgeon's capabilities—including image stability, tremor filtering, and precision instrument control—reduce the need for physical assistance by surgical staff, and enable remote minimally invasive procedures through telepresence. They signified a transformative shift in surgery, introducing robotic-assisted procedures that continue to improve and expand in scope.

2. da Vinci Surgical System

2.1. da Vinci System. Intuitive's da Vinci robotic surgery platform represents a major innovation in robot-assisted surgery (RAS), evolving from foundational telepresence concepts. The system addresses limitations in traditional laparoscopic techniques and features three integrated components: a patient cart, a surgeon console, and a vision cart for advanced imaging (Fig. 1.2). This setup offers seamless control of robotic arms with wrist-like dexterity and incorporates a stereo endoscope that provides 3D depth information to the surgeon.

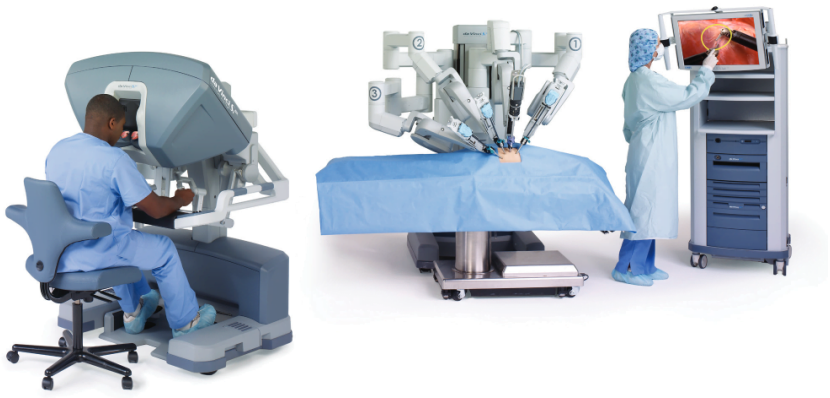


FIGURE 1.2. Major components of the Da Vinci system: surgeon console (left), patient-side manipulators (center), and vision cart (right).

Since its introduction, Intuitive has continually refined the da Vinci system. The S model [**13**] (2006) introduced high-definition imaging, while the Si model [**14**] (2009) added dual-console functionality for enhanced collaboration and training. The Xi model [**15**] (2014) redesigned the patient-side cart and robot arm architecture, improving cart mobility and adding more flexibility to port placement. It also introduced FireFly fluorescence imaging for real-time assessment of tissue perfusion and advanced instruments, such as the Vessel Sealer Extend, for efficient vascular sealing and cutting. These enhancements have expanded the platform's use in abdominal, gynecological, and urological surgeries, promoting minimally invasive approaches that improve procedural precision and reduce recovery times.

2.2. da Vinci Research Kit (dVRK). The da Vinci Research Kit (dVRK) [**16**] is a collaborative effort by Intuitive Surgical and Johns Hopkins University to advance telerobotic surgery research. Retired da Vinci surgical systems are repurposed for academic use, enabling universities to develop innovative surgical technologies and methodologies. The dVRK is currently deployed at over 43 institutes across 10 countries for a wide range of medical robotics research applications.

The dVRK provides programmatic control of the following da Vinci components: the Master Tool Manipulators (MTMs), which are two 6-DoF arms with a gripper that the operator uses to control the patient-side robot arms; the Endoscope Camera Manipulator (ECM), a 5-DoF arm holding the stereo endoscope; and the Patient-Side Manipulators

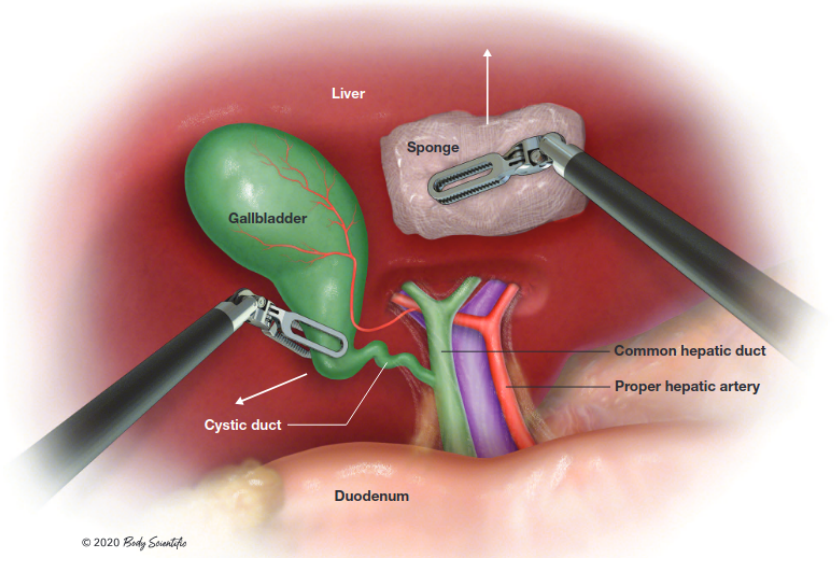


FIGURE 1.3. Anatomy of robotic cholecystectomy [**1**].

(PSMs), which are 7-DoF arms with laparoscopic instruments installed. The dVRK also enables reading of pedal inputs, which the surgeon presses to activate ECM control, the clutch, or instruments connected to electrosurgical units. This comprehensive access to the system's kinematic state and control interfaces makes the dVRK the enabling platform for all the work presented in this dissertation.

3. Robotic Cholecystectomy

Cholecystectomy is a surgical procedure to remove the gallbladder, an organ that stores bile produced by the liver. It is commonly performed to treat gallstones, inflammation, or other gallbladder diseases that cause pain or complications. The procedure can be done through traditional open surgery or minimally invasive laparoscopic methods, including robotic-assisted surgery.

The first laparoscopic cholecystectomy was performed in 1985 by Erich Mühe in Germany [**17**]. The procedure gained wider acceptance after Philippe Mouret's case series of laparoscopic cholecystectomies in 1987 [**18**], which highlighted benefits such as reduced post-operative pain and improved cosmesis (i.e., the cosmetic outcome of surgery, including reduced scarring). On March 3, 1997, Dr. Jacques Himpens performed the world's first robotic

cholecystectomy using an early prototype named *Mona* [**19**]. Unlike previous laparoscopic approaches, the robotic method faced fewer feasibility concerns but was limited by cost. Advances such as Indocyanine Green (ICG) fluorescence imaging, single-port access, and cost-saving strategies have since made robotic cholecystectomy increasingly popular. It is also a valuable procedure for surgeons and support staff to gain robotic surgery experience through a familiar and standardized operation [**17**].

Cholecystectomy stands out as an ideal candidate for automation: its relatively straightforward surgical anatomy, combined with standardized maneuvers and well-defined anatomical features, makes it well-suited for studies aiming to automate surgical procedures using robots. This dissertation uses cholecystectomy as the clinical testbed for all three of its contributions.

3.1. Procedure. The current robotic cholecystectomy procedure is organized into the following eleven steps [**1**, **20**]:

(1) *Exposure of the Working Area:* Retract part of the liver and reposition the gallbladder to uncover the hilum, where structures such as ducts and vessels connect. Key landmarks—the gallbladder neck, cystic artery, and cystic duct—are identified to guide the procedure and avoid damage.

(2) *Retraction of the Gallbladder Neck:* Pull the gallbladder neck to improve visibility of the triangle of Calot, an anatomical area used as a map for safe dissection. Structures connecting the gallbladder to the liver are isolated to prevent accidental damage.

(3) *Opening the Anterior Peritoneal Layer:* Carefully incise the outer covering of the gallbladder to access deeper structures. This step requires precision to avoid harming important tissues, especially if scarring is present.

(4) *Opening the Posterior Peritoneal Layer:* Remove the posterior layer of tissue connecting the gallbladder to the liver. This helps free important structures while avoiding unintended injuries.

(5) *Isolation of the Cystic Duct:* Pull the gallbladder neck to expose the duct that carries bile from the gallbladder. Fluorescent dye or X-rays may be used if the anatomy is unclear.

(6) *Isolation of the Cystic Artery:* Locate the small artery that supplies blood to the gallbladder. Ensuring only two structures connect to the gallbladder helps avoid errors and protects nearby arteries essential for liver function.

(7) *Clipping of the Cystic Duct:* Clear the area around the duct and securely close it with special clips to prevent bile leakage. Stones in the duct, if present, can be removed at this stage.

(8) *Clipping the Cystic Artery:* Securely close the blood vessel with clips to prevent bleeding.

(9) *Division of the Cystic Duct and Artery:* Cut the duct and artery using scissors to avoid heat-related tissue damage. This step disconnects the gallbladder from its attachments.

(10) *Dissection of the Gallbladder from the Liver:* Separate the gallbladder from the liver along a natural tissue layer. Different techniques are used depending on whether inflammation is present, with care taken to avoid bleeding or bile leakage.

(11) *Specimen Retrieval:* After removing the gallbladder, inspect the area to ensure there is no bleeding or leakage. The gallbladder is removed through a small incision using a bag to prevent spillage.

The primary actions in robotic cholecystectomy are grasping and dissecting. Grasping involves securely holding structures such as the gallbladder neck to maintain visibility and tissue tension. Dissecting refers to carefully separating the gallbladder from surrounding tissues using energy delivery (monopolar, bipolar, or ultrasonic instruments). Both actions demand high levels of dexterity and control. Step 10—dissection of the gallbladder from the liver bed—is the specific subtask that this dissertation automates (Chapters 5–6), while the full procedure is the basis for the dataset contribution (Chapter 3).

4. Related Work

This section surveys the literature relevant to the three pillars of this dissertation: autonomous surgical task execution, visual perception in surgical robotics, and vision-based instrument state estimation. We identify the research gaps that motivate each contribution.

4.1. Surgical Task Automation. RAS has spurred growing research into automating specific surgical subtasks. Substantial effort has been devoted to suturing, with approaches ranging from visual tracking of suture threads [**21**] and image-guided knot tying [**22**] to single-arm [**23**] and multi-throw [**24**] automated suturing systems. Thread tracking [**25**] remains a central challenge in this line of work. Beyond suturing, Ayvali et al. [**26**] explored Bayesian optimization-guided probing for tissue ablation, while Shademan et al. [**27**] demonstrated supervised autonomous soft tissue surgery using point clouds generated from plenoptic cameras. However, the reliance on non-standard cameras (plenoptic or external stereo rigs) makes these methods incompatible with the standard endoscopic setup used in clinical RAS. More recently, Kim et al. [**28**, **29**] introduced the Surgical Robot Transformer (SRT) and its hierarchical extension SRT-H for autonomous cholecystectomy clipping and cutting on *ex vivo* tissue, but—following the Action Chunking Transformer [**30**] framework—they augment the endoscope with non-standard wrist cameras mounted at the instrument tips and target the post-dissection phase rather than the energy-based dissection itself.

Vision-based cutting control has also been investigated: Han et al. [**31**] introduced a visual servoing algorithm that automatically cuts deformable objects along a predetermined path with a scalpel. While effective in a bench-top setting, this approach uses a scalpel—an instrument never used in RAS due to bleeding risk—and relies on feature-point tracking rather than tissue-level semantic understanding. Other studies have employed segmentation models to detect loose connective tissues and visualize safe dissection planes during robot-assisted gastrectomy [**32**], but these operate at the pixel level without establishing logical connections between individual tissue types.

Compared to auto-suturing, there is notably less work on auto-dissection—a gap highlighted in a comprehensive review of autonomy in surgical robotics [**33**]. Energy-based

dissection along tissue boundaries, which is a core task in cholecystectomy, remains largely unaddressed. This dissertation fills that gap with a vision-based autonomous dissection framework that uses only standard endoscopic images (Chapters 5–6).

4.2. Image Segmentation and Detection in Surgical Robotics. Accurate scene understanding is a prerequisite for any vision-based surgical automation system. In minimally invasive surgery, this requires both tissue segmentation (to delineate anatomical structures) and instrument detection (to localize the robot's tools).

4.2.1. *Tissue Segmentation.* Early approaches relied on handcrafted features and shallow classifiers, but the field has since converged on deep learning architectures. Datasets such as CholecSeg8k [**34**] and CholecTriplet [**35**] provide semantic annotations for cholecystectomy scenes, while Endoscapes [**36**] provides annotations for assessing the Critical View of Safety (CVS)—a standard intraoperative checkpoint in laparoscopic cholecystectomy in which the surgeon must clearly expose the cystic duct and cystic artery before clipping or cutting them, in order to avoid bile duct injury. However, most public datasets annotate only a few semantic classes (e.g., "organ" vs. "instrument") and do not provide the fine-grained instance-level masks needed for boundary-guided dissection.

4.2.2. *Instance Segmentation Architectures.* The Mask R-CNN framework [**37**], implemented in Detectron2 [**38**] with a ResNet-50-FPN [**39**, **40**] backbone, was the workhorse of early surgical segmentation research. More recently, Transformer-based architectures have advanced the state of the art: MaskDINO [**41**] combines the DINO detector with a masked attention mechanism for joint detection and segmentation, while Mask2Former [**42**] offers a unified framework for semantic, instance, and panoptic segmentation. The YOLO family has also evolved rapidly: YOLO11 [**43**] provides real-time instance segmentation (YOLO11l-seg) and keypoint detection (YOLO11l-pose) in a single architecture, making it particularly attractive for latency-sensitive surgical applications. Chapter 4 evaluates Detectron2, MaskDINO, and YOLO11 on our custom cholecystectomy dataset.

4.2.3. *Annotation Tools.* Generating high-quality training data for surgical scenes is labor-intensive. The Segment Anything Model (SAM) [**44**] and its successor SAM2 [**45**]

have dramatically reduced annotation effort by enabling point-prompted, propagation-based segmentation across video frames. This dissertation uses SAM for the initial perception dataset and a SAM2-based workflow for the expanded CRCD annotations that support the later perception and kinematics contributions. The full CRCD annotation release is described in Chapter 3, while Chapter 4 focuses on how those annotations are used to train and evaluate the models.

4.3. Surgical Datasets. The development of data-driven surgical automation systems is fundamentally constrained by the availability of large-scale, multimodal datasets. Most existing surgical datasets focus on a single modality—typically endoscopic video with instrument or phase annotations [**34**, **46**–**50**]. While valuable for training perception models, these datasets lack the kinematic data needed to train or evaluate control algorithms.

The few datasets that do include kinematics are limited in scale or task complexity. JIGSAWS [**51**, **52**] pairs video with kinematics for 120 demonstrations of three elementary bench-top drills (suturing, needle passing, knot tying)—toy tasks far removed from real surgical procedures. Rivas-Blanco et al. [**53**] recorded kinematics during simple peg-transfer exercises with external (non-endoscopic) cameras. Colleoni et al. [**54**] used kinematics to improve instrument segmentation data augmentation, but the recorded motions were unrelated to surgical procedures.

More recently, ImitateCholec [**55**] contributed over 18,000 *ex vivo* cholecystectomy demonstrations with dVRK kinematics, but targets the clipping and cutting phase for imitation learning [**28**, **29**] rather than dissection. General-purpose robot learning corpora such as Open X-Embodiment [**56**] and DROID [**57**] are orders of magnitude larger but do not include surgical tasks.

Among the public datasets discussed above, we did not identify another resource that combines stereo endoscopic video, full robot kinematics (for all arms and manipulators), pedal signals, and tissue segmentation and instrument keypoint annotations from actual cholecystectomy procedures. Concurrent community efforts such as Open-H-Embodiment [**58**]

are beginning to aggregate large-scale, multimodal data across many medical robotics platforms and tasks, but no single procedure-specific cholecystectomy resource yet provides the full combination of modalities targeted in this work. The CRCD (Chapter 3) was designed to address this combination of needs. A detailed comparison with existing datasets is provided in Chapter 3, Table 3.1.

4.4. Vision-Based Instrument Pose Estimation. Accurate knowledge of instrument pose is central to surgical automation. While the dVRK provides kinematic readings through its joint encoders, these signals suffer from cable-driven transmission errors and calibration drift that can reach up to 5 cm [**28**, **29**]. Vision-based approaches that estimate instrument pose directly from endoscopic images offer a potentially more reliable alternative.

Existing benchmarks for visual instrument pose estimation are small: the SurgRIPE challenge [**59**] provides only 2,841 labeled frames, and SuPer [**60**] offers roughly 2,000 frames. SurgPose [**61**] contributes approximately 120,000 keypoint annotations but targets joint localization rather than full end-effector Cartesian state, while SurgeoNet [**62**] relies entirely on synthetic data. In all cases, the datasets are too small or too constrained to train robust pose estimation models for real surgical procedures.

The Vision Transformer (ViT) [**63**] has emerged as a powerful architecture for visual recognition. Self-supervised pre-training strategies such as DINO [**64**] have shown that ViT encoders develop attention maps that closely correspond to object boundaries, suggesting inherent suitability for spatial localization. In surgical robotics, Transformer-based models have been adopted for imitation learning—the Surgical Robot Transformer (SRT) [**28**] and its hierarchical extension SRT-H [**29**] map visual observations to robot actions. Both build on the Action Chunking Transformer (ACT) framework [**30**] and rely on additional wrist cameras mounted at the instrument tips—a sensing modality that is not part of the standard clinical RAS setup—and they train a single model for both instruments jointly, coupling the representations of tools whose motions may be only loosely correlated. Outside the surgical domain, the *Decoupled Interaction Framework* (DIF) of Jiang et al. [**65**]—evaluated

on general bimanual manipulation tasks rather than RAS—demonstrates that decoupled, per-arm models can outperform integrated dual-arm control by over 20%.

Chapter 7 presents a ViT-based pipeline for estimating instrument translation from stereo endoscopic images, leveraging the CRCD's synchronized video-kinematics pairs to train and evaluate the model on real cholecystectomy data.

5. Thesis Overview and Contributions

5.1. Contributions. The literature review reveals three interconnected gaps that this dissertation addresses. Each contribution is motivated by and directly targets one of these gaps:

(a) **A vision-based autonomous dissection framework for robotic cholecystectomy with bimanual manipulation.** Despite substantial progress in automating suturing and other surgical subtasks, energy-based dissection along tissue boundaries—the core action in cholecystectomy—remains largely unaddressed. Existing vision-based cutting methods use non-surgical instruments (scalpels), non-standard cameras (plenoptic), or operate without tissue-level semantic understanding. To fill this gap, we develop a complete pipeline that enables the da Vinci robot to autonomously dissect the gallbladder from the liver using only endoscopic images. The framework evolves through two versions [**2**, **3**]: an initial single-arm system with offline trajectory planning, and an upgraded bimanual system with automatic grasping, tissue stretching, and online boundary-guided dissection. The perception pipeline progresses from Detectron2 to MaskDINO and YOLO11, with the introduction of a liver bed segmentation class that supports boundary tracking after repeated rounds of energy delivery and lays the groundwork for future multi-round dissection. In the single-cycle trials reported in Chapter 6, the upgraded system achieves a 3.3× speed improvement over the original while maintaining sub-millimeter boundary tracking precision.

(b) **The Comprehensive Robotic Cholecystectomy Dataset (CRCD).** Existing publicly available surgical datasets provide only partial coverage of the modalities required to study perception, control, and surgeon intent jointly. No prior dataset, to our knowledge, simultaneously provides stereo endoscopic video, full robot kinematics for all arms and console manipulators, pedal activation signals, and per-frame tissue segmentation and instrument keypoint annotations recorded from multiple surgeons performing the same cholecystectomy procedure. To address this gap, we create and release a large-scale, multimodal dataset recorded during *ex vivo* pseudo-cholecystectomy procedures on porcine livers using the dVRK [**66**, **67**]. The CRCD combines stereo endoscopic videos (60 fps, 1280×720), full kinematic data for all robot arms and console manipulators (100 Hz), pedal signals (230 Hz), and tissue segmentation and instrument keypoint annotations in COCO format. With over 755,000 stereo frames, 34,678 annotated segmentation frames, and 15,999 instrument keypoint instances from seven surgeons with documented experience levels, the CRCD provides a multimodal foundation for the perception, pedal, and kinematics studies in this dissertation.

(c) **A baseline framework for predicting instrument kinematics from stereo endoscopic images.** Existing pose estimation benchmarks are too small for robust model training, and most approaches either rely on synthetic data or do not target the end-effector Cartesian state; the few datasets that pair video with kinematics are limited to toy tasks, leaving a gap in evaluating vision-to-pose models on real surgical data. To establish such a baseline, we introduce a two-stage pipeline in which per-instrument Vision Transformer backbones are pre-trained on instrument segmentation masks and then paired with lightweight stereo pose estimation heads to regress end-effector translation. Evaluated on held-out CRCD videos, the models achieve mean translation errors of approximately 1 centimeter, establishing a foundation for full $SE(3)$ instrument pose estimation from surgical video alone.

These three contributions are interconnected: the autonomous dissection framework (a) motivates the need for robust perception and instrument localization; the CRCD (b) provides the data that enables training and evaluation of both the perception models used by the dissection framework and the kinematics prediction models; and the kinematics prediction work (c) demonstrates an alternative approach to instrument localization that could ultimately replace the keypoint-based method used in the dissection pipeline.

5.2. Dissertation Organization. The remainder of this dissertation is organized as follows:

Chapter 2: System Setup: describes the hardware and software infrastructure: the dVRK platform with Si endoscope, custom fiducial-marker-based arm calibration, stereo 3D scene reconstruction, electrosurgical unit control via Arduino, and the ROS2-based system architecture.

Chapter 3: Comprehensive Robotic Cholecystectomy Dataset: describes the CRCD in detail: its motivation, components (video, kinematics, pedals, annotations), recording protocol, surgeon profiles, and preliminary applications including pedal intent recognition and instance segmentation.

Chapter 4: Perception: builds on the CRCD annotations to present the evolution of the perception pipeline, the training of three perception model families (Detectron2, MaskDINO, and YOLO11), and a comparative evaluation on the expanded cholecystectomy dataset.

Chapter 5: Autonomous Dissection: details the methodology for autonomous dissection, from the initial single-arm offline approach to the upgraded bimanual online system with grasping, tissue stretching, and real-time boundary tracking [**2**, **3**].

Chapter 6: Experimental Results: presents the downstream motion evaluation of both system versions, including grasping trials and dissection results across multiple model configurations. The evaluation of the perception model is presented in Chapter 4.

Chapter 7: Kinematics Prediction from Endoscopic Images: presents the ViT-based pipeline for predicting instrument translation from stereo video, including the dataset preparation from the CRCD, the two-stage training procedure, and a comprehensive experimental evaluation with ablation studies.

Chapter 8: Discussion and Future Work: synthesizes the findings across all contributions, discusses limitations, and outlines future research directions toward fully autonomous robotic surgery.

Chapter 9: Conclusion: summarizes the contributions and their significance.

Appendix A: details the fiducial-marker-based arm calibration procedure.

Appendix B: presents the inverse kinematics formulation.

CHAPTER 2

System Setup

This chapter describes the hardware and software infrastructure that underpins all experimental work in this dissertation. The da Vinci surgical robot integrated with the da Vinci Research Kit is introduced in Section 1. Section 2 presents the custom fiducial-marker-based arm calibration that reduces the positional error of the dVRK's forward kinematics from centimeters to millimeters—a prerequisite for the automated control algorithms in Chapter 5. Section 3 describes the stereo 3D scene reconstruction pipeline that converts endoscopic images into point clouds. Section 4 details the electrosurgical unit interface that enables programmatic control of energy delivery. Finally, Section 5 presents the numerical inverse kinematics used to command the robot to desired configurations.

1. Hardware Overview

Our system uses a first-generation da Vinci surgical system integrated with the da Vinci Research Kit (dVRK) [**16**]. The da Vinci robot comprises three Patient Side Manipulators (PSMs), an Endoscope Camera Manipulator (ECM), and two Master Tool Manipulators (MTMs) at the surgeon console. In our initial work [**2**], a single PSM (PSM1) equipped with a Permanent Cautery Hook (PCH) was used for dissection while the endoscope remained stationary. The upgraded system [**3**] introduced bimanual manipulation by adding PSM2 with a Fenestrated Bipolar Forceps (FBF) for grasping and tissue stretching, as described in Chapter 5.

In a departure from conventional configurations, we replaced the original endoscope with the Si model [**14**], which provides superior image quality and notably reduced noise characteristics. These improvements are critical for the perception pipeline (Chapter 4), where the segmentation and keypoint detection models operate directly on the endoscopic images.

The dVRK lacks direct control over the external power supply that governs the voltage output of monopolar and bipolar instruments. To enable programmatic energy delivery during automated dissection, we interfaced the electrosurgical unit cables and the console pedals with an Arduino, as described in Section 4.

For the upgraded system, we migrated the implementation platform from ROS to ROS2 [**68**]. ROS2 provides improved real-time performance, more robust inter-process communication, and a modern middleware layer, all of which better support the time-critical perception-to-control loop required for online boundary tracking.

2. da Vinci Arm Calibration

2.1. Motivation. In the full da Vinci system, the forward kinematics are derived from the Setup Joints (SUJs), which serve as the base of the robot. Each SUJ is equipped with potentiometers that measure voltages, which are then converted into joint positions to estimate the base frame of the PSM relative to the ECM. However, due to the inherent inaccuracies of the SUJs, the positional discrepancy between the PSM tip and the ECM tip remains within approximately $\pm 5\,\mathrm{cm}$ for translation and between 5–10 degrees for orientation [**16**]. While such an error margin is tolerable for manual teleoperation using the MTMs—where the surgeon continuously corrects for positional offsets through visual feedback—it is unacceptable for automated control, where the robot must reach precise target positions without human intervention. To address this limitation, we developed a custom calibration method that uses an external camera and fiducial markers to refine the forward kinematics.

2.2. Product of Exponentials Formulation. We formulate the forward kinematics using the Product of Exponentials (PoE) [**69**] representation. For a PSM with six joints (configuration string RRPRRR, where R denotes a revolute joint and P a prismatic joint), the forward kinematics from the base frame R to the instrument tip frame T are:

$$g_{rt}(\vartheta) = e^{\hat{\zeta}_1 \vartheta_1}\, e^{\hat{\zeta}_2 \vartheta_2}\, e^{\hat{\zeta}_3 \vartheta_3}\, e^{\hat{\zeta}_4 \vartheta_4}\, e^{\hat{\zeta}_5 \vartheta_5}\, e^{\hat{\zeta}_6 \vartheta_6}\, g_{rt}(0) \tag{2.1}$$

For the ECM with four joints (configuration string RRPR), the forward kinematics from the base frame S to the endoscope tip frame E are:

$$g_{se}(\varphi) = e^{\hat{\xi}_1\varphi_1}\, e^{\hat{\xi}_2\varphi_2}\, e^{\hat{\xi}_3\varphi_3}\, e^{\hat{\xi}_4\varphi_4}\, g_{se}(0) \tag{2.2}$$

where g_{ab} represents the homogeneous transformation matrix between frames A and B. Vectors $\zeta_i \in \mathbb{R}^6$ and $\xi_i \in \mathbb{R}^6$ are the unknown joint twist parameters for the PSM and ECM, respectively. The joint angles ϑ and φ are derived from the dVRK encoder measurements.

Each twist parameter vector has the form:

$$\zeta^R = \begin{bmatrix} v_x & v_y & v_z & \omega_x & \omega_y & \omega_z \end{bmatrix}^T \tag{2.3}$$

for a revolute joint, where $\omega \in \mathbb{R}^3$ is the unit vector along the axis of rotation and $v = -\omega \times q \in \mathbb{R}^3$ encodes a point q lying on that axis. For a prismatic joint:

$$\zeta^P = \begin{bmatrix} v_x & v_y & v_z & 0 & 0 & 0 \end{bmatrix}^T \tag{2.4}$$

where $v \in \mathbb{R}^3$ is the unit vector along the direction of translation. The same definitions apply to the ECM twist parameters ξ_i, using the appropriate joint type for each of the four ECM joints.

2.3. Joint Angle Calibration. The dVRK computes its reported joint angles θ_d from raw encoder readings ϵ_d via a linear mapping:

$$\theta_d = \alpha_d \epsilon_d + \beta_d \tag{2.5}$$

where α_d and β_d are the factory-set linear parameters. We seek to compute calibrated joint angles θ_c from the same raw encoder readings:

$$\theta_c = \alpha_c \epsilon_d + \beta_c = \alpha' \theta_d + \beta' \tag{2.6}$$

TABLE 2.1. Joint limit settings used during calibration dataset recording.

Joint (metric)		1 (deg)	2 (deg)	3 (m)	4 (deg)	5 (deg)	6 (deg)
ECM	Min	−70	−45	0.03	−85	—	—
	Max	70	40	0.23	85	—	—
PSMs	Min	−90	−45	0	−60	−30	−80
	Max	90	45	0.24	60	50	80

TABLE 2.2. Number of recorded instances for calibration of each arm.

Arm	**PSM1**	**PSM2**	**ECM**
Number of instances	1950	1950	1700

where $\alpha' = \alpha_c/\alpha_d$ and $\beta' = \beta_c - \alpha'\beta_d$. The dVRK's reported joint angles are thus linearly related to the calibrated joint angles. The parameters α' and β' are determined jointly with the twist parameters through the optimization procedure described in Section 2.5.

2.4. Calibration Data Collection. Our calibration approach is influenced by [**70**], which uses an optical tracking system with custom adapters for instrument tips. Given the limited availability of such systems in most research laboratories, we opted for the more accessible ArUco fiducial markers [**71**] attached to the instrument tips, tracked by a ZED-mini external stereo camera. We selected base frames for each arm (Fig. 2.2) that remain visible even when a tray is placed in the da Vinci workspace, ensuring consistent performance regardless of the physical setup.

The calibration dataset was collected by sweeping each joint from its minimum to maximum position while holding all other joints at zero. Table 2.1 lists the joint limit settings used during recording. At each configuration, the robot was momentarily stopped and the ArUco marker position was averaged over ten frames to reduce noise. The resulting dataset sizes are summarized in Table 2.2.

2.5. Optimization. The joint twist parameters (ζ_i for the PSMs, ξ_i for the ECM) together with the linear calibration parameters (α', β') from Equation (2.6) are determined

by minimizing the discrepancy between the predicted instrument tip pose and the ArUco-measured ground truth. The total number of unknowns is 128: 48 per PSM (six twist vectors of six parameters each, plus six α' and six β' values) and 32 for the ECM (four twist vectors, plus four each of α' and β').

The distance between two homogeneous transformation matrices is computed as a weighted sum of translational and rotational components:

$$d(g_1, g_2) = 0.7 \cdot d_{\text{trans}}(g_1, g_2) + 0.3 \cdot d_{\text{rot}}(g_1, g_2) \tag{2.7}$$

The translational distance is the Euclidean distance between the position components. The rotational distance is based on the quaternion representation [**72**]:

$$d_{\text{rot}} = 2\cos^{-1}\big(\text{Re}(z)\big), \qquad z = q_1 \cdot \overline{q}_2 \tag{2.8}$$

where q_1 and q_2 are the rotation quaternions extracted from the two transformations, $\overline{q}_2$ denotes the conjugate of q_2, and $\text{Re}(z)$ is the real part of the quaternion product. The optimization objective is to minimize the mean squared error of these distances across all recorded configurations.

Nonlinear constraints enforce physical consistency: the norm of the rotation-axis component ω must equal unity for revolute joints, and the last three elements of the twist vector must be zero for prismatic joints (as defined in Equations (2.3) and (2.4)).

We used MATLAB's Global Optimization Toolbox [**73**] to solve this constrained optimization problem. The optimized twist parameters and linear calibration coefficients are reported in Table A.1 in Appendix A.

2.6. Calibration Results. To evaluate the calibrated forward kinematics, we recorded the predicted positions from both the factory dVRK kinematics and our custom calibration while moving the arms through random configurations, with ArUco marker positions serving as ground truth. Figure 2.1 shows the tracked positions over time, and Table 2.3 summarizes the mean error and standard deviation for each axis.

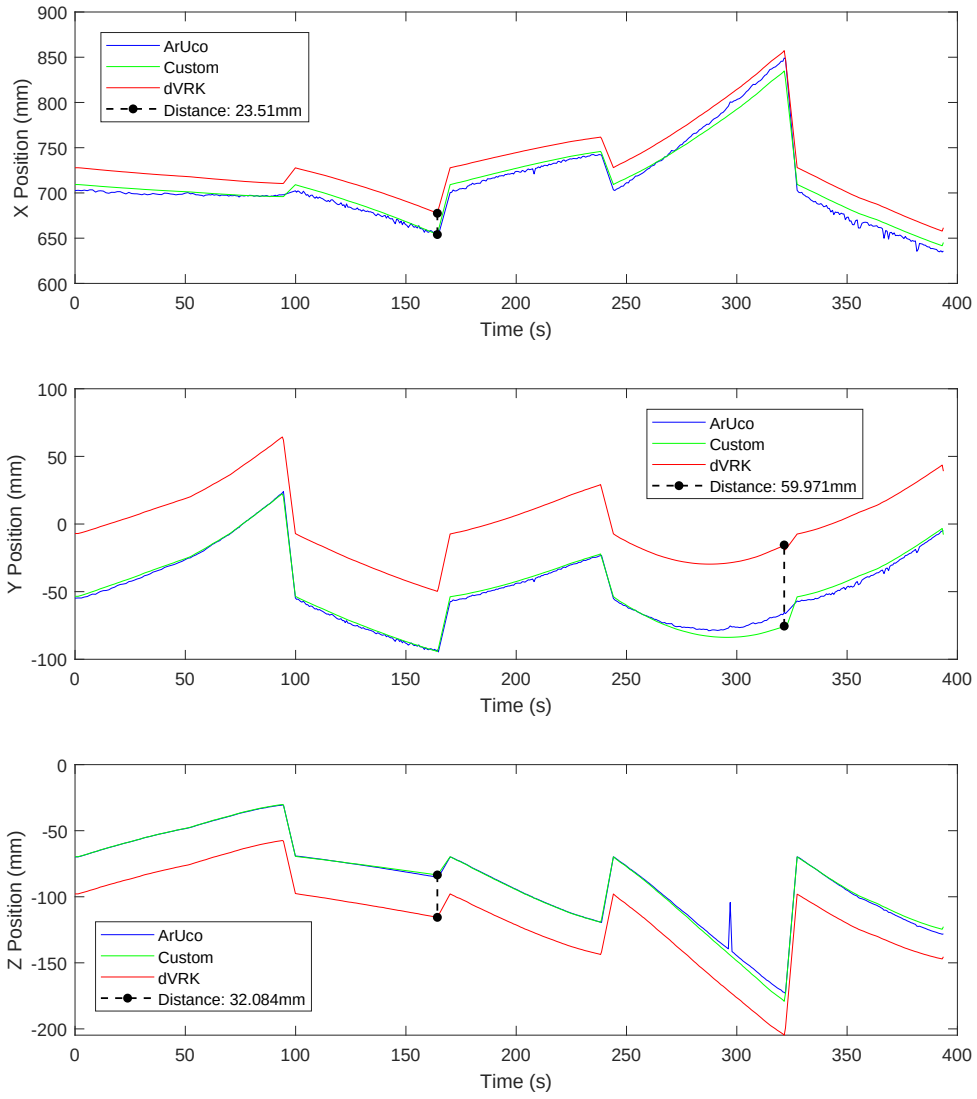


FIGURE 2.1. Position predicted by the dVRK's factory forward kinematics (blue) compared to the ArUco marker ground truth (red) and the calibrated custom kinematics (green) during a random arm motion. The calibrated kinematics closely track the ground truth, while the dVRK kinematics exhibit large offsets, particularly in the Y and Z axes.

The custom calibration reduces the mean error by a factor of 1.96× for X, 20.04× for Y, and 19× for Z. The improvement is most dramatic along the Y and Z axes, where the factory SUJ-based kinematics exhibit errors exceeding 2–5 cm. The calibrated kinematics bring these errors to the low-millimeter range, making automated instrument positioning feasible for the control algorithms in Chapter 5.

TABLE 2.3. Mean error and standard deviation between different kinematics and the ArUco marker ground truth during random arm motions (Fig. 2.1).

Measure (distance)	**dVRK ⇔ ArUco**	**Custom ⇔ ArUco**
Mean Error	X: 11.0 mm Y: 50.1 mm Z: 20.9 mm	X: 5.6 mm Y: 2.5 mm Z: 1.1 mm
Standard Deviation	X: 4.9 mm Y: 2.0 mm Z: 3.9 mm	X: 3.6 mm Y: 2.1 mm Z: 2.2 mm

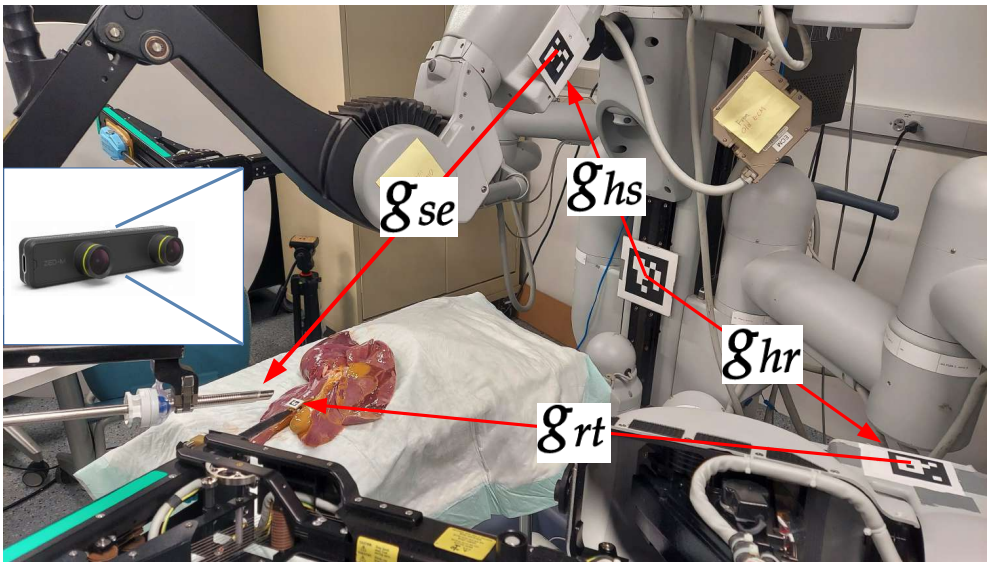


FIGURE 2.2. The calibration setup showing all coordinate frames and transformations. The ArUco fiducial markers on the instrument tips and arm bases are tracked by an external ZED-mini camera. The Helper (H) frame bridges the two sides of the workspace, enabling the second PSM to be referenced to the ECM even when the ECM and first PSM base frames are occluded. The arrows indicate the direction of each transformation, ultimately establishing the transformation g_{et} between the ECM tip and the PSM tip.

2.7. Multi-Arm Configuration. After determining g_{rt} and g_{se} from the calibration, the relative configuration of a PSM instrument tip with respect to the ECM tip must be established. This requires the Helper (H) frame shown in Fig. 2.2. The Helper frame may seem unnecessary when only a single PSM is used; however, it becomes essential when adding a second PSM. Both the ECM base frame (S) and the first PSM base frame (R) are physically located on the same side of the workspace, making them occluded from the opposite side

where the second PSM is mounted. The Helper frame, placed in a position visible to both sides, facilitates the chain of transformations:

$$g_{et} = g_{se}^{-1} \cdot g_{hs}^{-1} \cdot g_{hr} \cdot g_{rt} = g_{es} \cdot g_{sh} \cdot g_{hr} \cdot g_{rt} \tag{2.9}$$

where g_{sh} and g_{hr} are obtained from the fiducial markers via the external ZED-mini camera. If the SUJ locations are altered between experiments, only g_{sh} and g_{hr} need to be re-measured; the calibrated twist parameters remain valid. The kinematic data in the CRCD dataset (Chapter 3) uses this transformation chain to provide PSM tip poses relative to the ECM tip.

3. 3D Scene Reconstruction

The autonomous dissection framework requires 3D coordinates of tissue boundaries and instrument positions to plan and execute trajectories. This section describes the stereo reconstruction pipeline that converts 2D endoscopic images into 3D point clouds.

3.1. Stereo Camera Calibration. The Si endoscope provides a stereo pair of images. We calibrated the stereo camera system by capturing multiple images of a 9×6 chessboard in different positions and orientations. Following the approach of Zhang [**74**], we used both the MATLAB Stereo Camera Calibration Toolbox [**75**] and OpenCV [**76**] to compute the camera parameters. Images with low reprojection error were selected for the final calibration. The resulting parameter set includes:

- Distortion coefficients for each endoscope
- Intrinsic camera matrices (focal lengths and principal points)
- Rectification matrices
- Projection matrices for both left and right endoscopes

These parameters facilitate the recovery of 3D point clouds from the recorded stereo videos and are included in the CRCD dataset (Chapter 3) to enable other researchers to reproduce the 3D reconstruction.

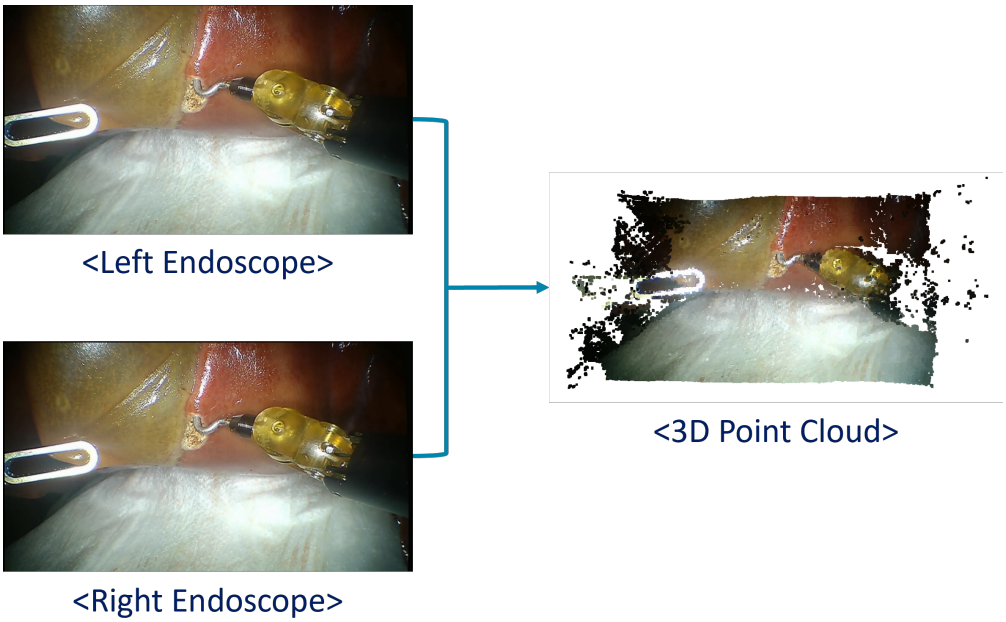


FIGURE 2.3. A 3D point cloud generated from stereo endoscopic images using Semi-Global Matching. The gallbladder (green), liver (brown), and instrument surfaces are visible. The point cloud format $\{(u, v) \mid (x, y, z)\}$ maps each 2D pixel to its estimated 3D position.

3.2. Stereo Disparity and Point Cloud Generation. Using the intrinsic and extrinsic camera parameters, we applied the modified Semi-Global Matching (SGM) algorithm [**77**] to produce stereo disparity maps from the rectified stereo endoscopic images. Before applying the SGM, the images are passed through a bilateral filter to reduce noise while preserving edges—particularly the tissue boundaries that the dissection algorithm must track.

The disparity map is converted to a 3D point cloud using the baseline distance and focal length of the stereo cameras. The resulting point clouds are dictionaries of the form $\{(u, v) \mid (x, y, z)\}$, where the estimated 3D point (x, y, z) is mapped to each pixel (u, v) in the 2D image (Fig. 2.3). To filter outliers—particularly points at large distances that arise from regions of low texture or specular reflections—we average each 3D point within a fixed-size 10×10 pixel window and discard points representing infinite distances from the disparity map.

While the SGM algorithm does not match the accuracy of recent learning-based stereo methods [**78**], it offers two practical advantages for our setting: it does not require a GPU for

real-time operation, and it generalizes to new surgical scenes without retraining. Its performance was adequate for the submillimeter dissection accuracy demonstrated in Chapter 6.

4. Electrosurgical Unit Control

Energy delivery is fundamental to surgical dissection: the monopolar instrument (PCH) uses electrical current to cauterize and separate tissue. This section describes the interface that enables both manual and programmatic control of the electrosurgical unit.

4.1. Console Pedals. The da Vinci console provides four pedals: *camera* (repositions the endoscope), *clutch* (disengages the MTMs from the PSMs for repositioning), *monopolar* (activates monopolar energy delivery), and *bipolar* (activates bipolar energy delivery). The dVRK provides pedal signals only at the moments of press and release events. To produce continuous signals suitable for synchronization with image and kinematic data, we interpolate the pedal state: each signal is held at 0 by default and set to 1 for the duration that the pedal is pressed. When the camera pedal is engaged, both MTMs are linked and jointly control the ECM rather than the PSMs. The pedal signals are sampled at 230 Hz.

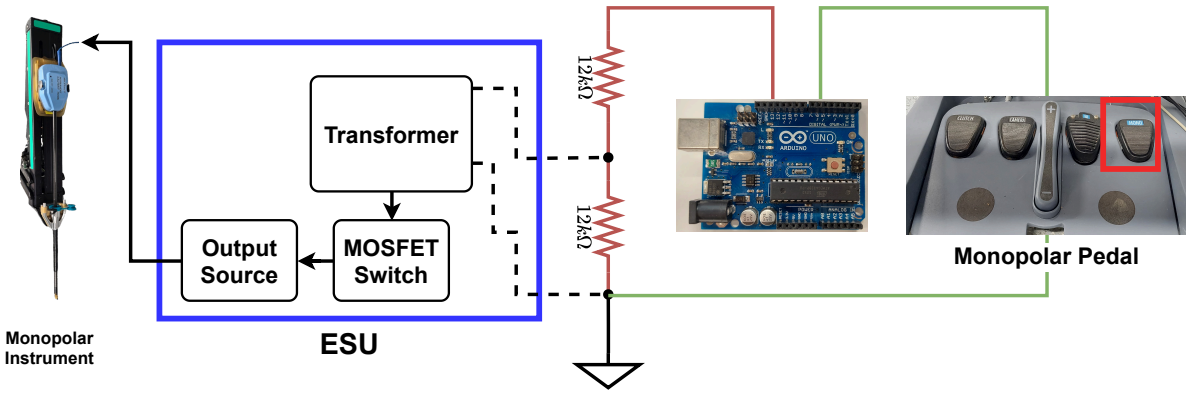
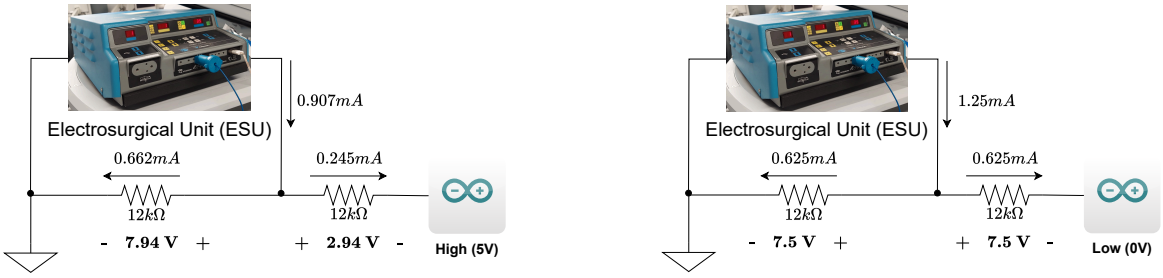


FIGURE 2.4. Schematic of the interface between the Arduino, the da Vinci console pedals, and the Pfizer Valleylab Force 2 electrosurgical generator. The Arduino intercepts the pedal signals and can independently trigger the monopolar output for automated energy delivery.

4.2. ESU Interface. The dVRK does not provide direct programmatic control over the electrosurgical generator (ESU) that regulates the voltage output to monopolar and bipolar

instruments. Our setup uses the Pfizer Valleylab Force 2 ESU. Based on the design principles described in [**79**], the internal structure of the generator includes a transformer between the input circuit and a MOSFET switch that determines whether to activate the monopolar output (Fig. 2.4). A minimum current of 1 mA must flow through the input cable (originally connected to the foot pedals) to activate the output.

We interfaced the generator's input cable with the da Vinci console pedals using an Arduino. In the default state (Fig. 2.5a), the Arduino's write pin is set to high (5 V). The resulting voltage distribution keeps the current below the activation threshold, and the monopolar output remains inactive. When the Arduino's write pin is set to low—either programmatically during automated dissection or when the surgeon presses the monopolar pedal—the voltage drops to 0 V, the current exceeds the threshold, the MOSFET switch activates, and the monopolar output is delivered to the instrument tip (Fig. 2.5b).



(A) Deactivated: write pin HIGH.

(B) Activated: write pin LOW.

FIGURE 2.5. Circuit states of the ESU interface from Fig. 2.4. (a) Default state: the Arduino write pin is HIGH (5 V), current remains below threshold, and the monopolar output is inactive. (b) Activated state: the write pin is LOW or the pedal is pressed, current exceeds the threshold, and the MOSFET activates the monopolar output.

During the recording of the CRCD dataset (Chapter 3), occasional Arduino shutdowns were observed when high current was applied to the instrument tip during prolonged energy delivery, which corrupted the pedal recordings for some trials.

4.3. Monopolar and Bipolar Operation. In cholecystectomy, monopolar energy is the primary modality for dissection: the PCH delivers current through the tissue at the point of contact, cauterizing and separating the tissue layers. Bipolar energy, delivered through

the FBF, passes current between the two jaws of the instrument and is used for coagulation of blood vessels to prevent bleeding. Our autonomous dissection framework currently controls the monopolar output for the PCH; bipolar control follows the same Arduino interface principle and can be activated when the FBF requires energy delivery.

5. Inverse Kinematics

During automated procedures, the robot must be commanded to desired Cartesian configurations computed from the perception and planning pipeline (Chapter 5). We implemented numerical inverse kinematics using the Sequential Least Squares Programming (SLSQP) algorithm [**80**], which is well suited for minimizing scalar functions of several variables subject to bounds and constraints. The objective function uses the same weighted distance metric defined in Equation (2.7), measuring the discrepancy between the current and desired end-effector poses. The constraints ensure that the computed joint angles remain within the dVRK's physical joint limits [**16**]. These limits are adjustable and are set to the ranges reported in Table 2.1.

CHAPTER 3

Comprehensive Robotic Cholecystectomy Dataset (CRCD)

This chapter presents the Comprehensive Robotic Cholecystectomy Dataset (CRCD) [**66**, **67**], a multimodal dataset recorded during *ex vivo* pseudo-cholecystectomy procedures on porcine livers using the da Vinci Research Kit (dVRK). Within the set of publicly available datasets summarized in Table 3.1, the CRCD brings together stereo endoscopic videos, full kinematic data for all robot arms and console manipulators, pedal signals, and tissue segmentation and instrument keypoint annotations. This chapter describes the motivation, composition, recording protocol, and preliminary applications of the dataset. The perception models trained on the CRCD annotation data are evaluated in Chapters 4 and 6.

1. Motivation and Related Datasets

Training state-of-the-art machine learning models for robot-assisted surgery (RAS) requires extensive datasets that comprehensively characterize the surgeon's actions alongside the corresponding robot motion and endoscopic video. In recent years, considerable effort has been devoted to creating public surgical datasets with expert annotations [**81**,**82**]. However, most existing datasets focus narrowly on instrument segmentation [**47**, **48**] and/or organ segmentation [**34**, **49**, **50**] from endoscopic video. For example, Twinanda et al. [**46**] created a video dataset with instrument segmentation labels and surgical phase annotations, but without any kinematic data. This makes it difficult to estimate the 3D position of detected instruments or to calculate their distance to tissues—capabilities that kinematic data has been shown to improve [**83**, **84**].

Few datasets incorporate kinematic data. Rivas-Blanco et al. [**53**] recorded arm and controller kinematics with external stereo cameras, but the cameras were in fixed locations, and the tasks were limited to basic exercises (moving pegs, following wires). The JIGSAWS

TABLE 3.1. Comparison of publicly available surgical robotics datasets. Relative to the other datasets in this table, the CRCD combines kinematics for all da Vinci arms and console manipulators with pedal signals and dense annotations. Annotation volume reports labeled frames when available; for the expanded CRCD, both segmentation-frame and keypoint-instance counts are shown.

Name	Year	Data	Procedure	Annotations	Annotation Volume
JIGSAWS [52]	2014	103 videos + kinematics	In-vitro experiments	Gestures, scoring	–
EndoVis 2017 [36]	2017	8 videos	Porcine procedures	Tool segmentation	3,000
FlapNet [86]	2020	1 video	Lobectomy	Tissue flap and tools	2,160
UCL dVRK [54]	2020	14 videos + kinematics	Synthetic background	Tool segmentation	4,200
RoboTool [87]	2021	20 videos	Various	Tool segmentation	514
AutoLaparo [88]	2022	21 videos	Hysterectomy	Task perception	5,936
HemoSet [89]	2024	11 videos	Thyroidectomy	Blood segmentation	857
CRCD [66,67]	2024	16 videos + kinematics + pedals	Cholecystectomy	–	–
Expanded CRCD [66,67]	2024	16 videos + kinematics + pedals	Cholecystectomy	Tool keypoints, Tissue segmentation	34,678 seg. frames 15,999 kpt instances

dataset [52] included more advanced tasks such as suturing and knot tying, but was limited to in vitro toy experiments. Colleoni et al. [54] recorded kinematics to improve instrument segmentation robustness, but the arm movements were unrelated to actual surgical procedures.

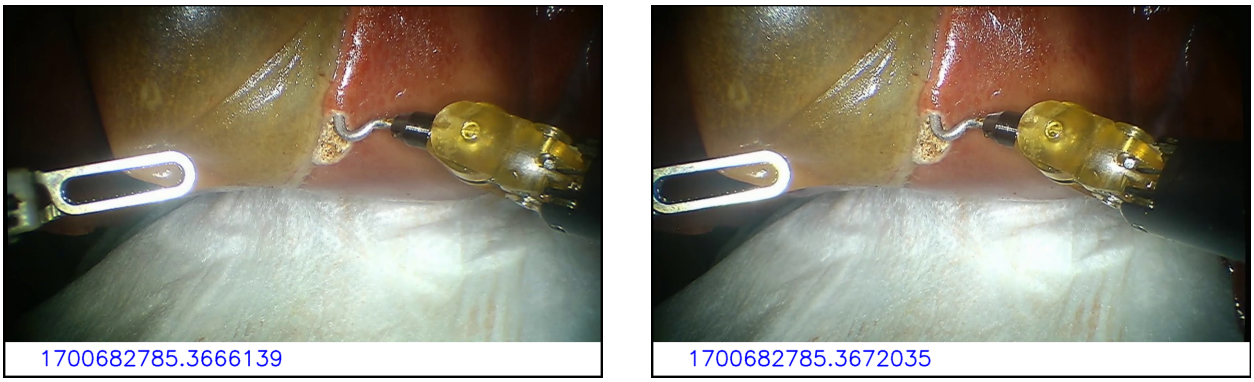
A significant yet often overlooked set of interaction signals is the pedals of the robotic surgery console. Surgeons use pedals to clutch the controllers, move the endoscope, and deliver monopolar/bipolar energy. Analyzing these interactions and automating such secondary tasks is vital for alleviating cognitive workload during prolonged interventions [85]. The datasets compared in Table 3.1 do not report publicly released pedal traces, which motivated their inclusion in the CRCD.

Table 3.1 compares the CRCD with publicly available surgical robotics datasets. Apart from CRCD, only JIGSAWS [52] and UCL dVRK [54] include kinematic data, but JIGSAWS lacks ECM information and involves only in-vitro tasks, while UCL dVRK omits MTM data. Neither dataset reports released pedal traces. Most datasets also annotate only a small subset of their video frames, whereas the expanded CRCD provides 34,678 annotated segmentation frames together with 15,999 instrument keypoint instances.

2. Dataset Components

2.1. Stereo Endoscopic Images. The robotic platform consists of the first-generation da Vinci surgical system controlled through the dVRK [**16**]. The original endoscope was replaced with the Si model [**14**] for its superior image quality and reduced noise (the full hardware setup is described in Chapter 2). The stereo endoscope cameras were calibrated using OpenCV [**76**] based on the approach of Zhang [**74**] and the ROS camera calibration toolbox [**75**], determining the intrinsic and extrinsic parameters for each camera. The dataset includes distortion parameters, intrinsic camera matrix, rectification matrix, and projection matrix for both cameras of the stereo endoscope, enabling 3D point cloud recovery from the videos.

Individual images from each camera are recorded separately, with a Robot Operating System (ROS) [**90**] timestamp at the bottom of the image (Fig. 3.1). These timestamps can be extracted using OCR engines such as Tesseract [**91**] and link each frame to the corresponding kinematic data and pedal signals. Videos are recorded at 60 frames per second with a resolution of 1280×720 pixels, encoded with the AVC1 FourCC identifier for the H.264/AVC codec [**92**], and compressed to MP4.



(A) Left endoscope (B) Right endoscope

FIGURE 3.1. Sample stereo endoscopic images from the CRCD. Each frame includes a ROS timestamp at the bottom for synchronization with kinematic and pedal data.

2.2. Pedal Signals. The da Vinci console includes four pedals: *camera*, *clutch*, *monopolar*, and *bipolar*. The dVRK provides pedal signals only at the moment pedals are pressed

or released. To achieve synchronization with the image and kinematic data, the signals were interpolated into binary state traces that remain at 0 by default and switch to 1 while the corresponding pedal is pressed. Pedal inputs are recorded at 230 Hz, a rate dictated by the Arduino's serial communication loop (the 230,400 baud link to the host yields one packet roughly every 1/230 s). Although this is well above the 60 Hz video rate, oversampling the pedal channels is inexpensive (each signal is a single bit) and ensures that brief press/release transitions are not missed during alignment with the image and kinematic streams.

The dVRK lacks direct control over the electrosurgical unit (ESU) that regulates the energy delivered to the instruments. We extended the capability of the dVRK by interfacing a Pfizer Valleylab Force 2 ESU with the da Vinci console pedals through an Arduino, as illustrated in Fig. 2.4. This added bridge allows the monopolar and bipolar pedals to be read programmatically and the corresponding energy delivery to be triggered either by the surgeon or by the autonomous dissection framework. The full circuit design and operating principle are described in Chapter 2, Section 4.

2.3. Kinematic Data. The kinematic data in the CRCD is based on the forward kinematics of the da Vinci robot derived from our custom calibration using fiducial markers [**2**]. This calibration determines the position of the PSM instrument tip relative to the ECM tip. The full calibration procedure, including the Product of Exponentials formalism and the optimization of joint twist parameters, is described in Chapter 2.

The coordinate frame assignments and transformation chain used to compute g_{et}—the relative configuration of the PSM instrument tip with respect to the ECM tip—are illustrated in Fig. 2.2 and defined in Equation (2.9). If the Setup Joints (SUJs) are repositioned, only the fiducial-marker-based transformations g_{sh} and g_{hr} need to be updated; the calibrated twist parameters remain valid.

Table 3.2 lists the complete set of kinematic variables recorded in the dataset. For each PSM, the dataset includes g_{rt} (base-to-tip), g_{et} (ECM-tip-to-PSM-tip), the joint states (position, velocity, effort), and the jaw joint states. For the ECM, it includes g_{se} (base-to-tip), g_{he} (helper-to-tip), and the joint states. For each MTM, it includes the tip pose relative

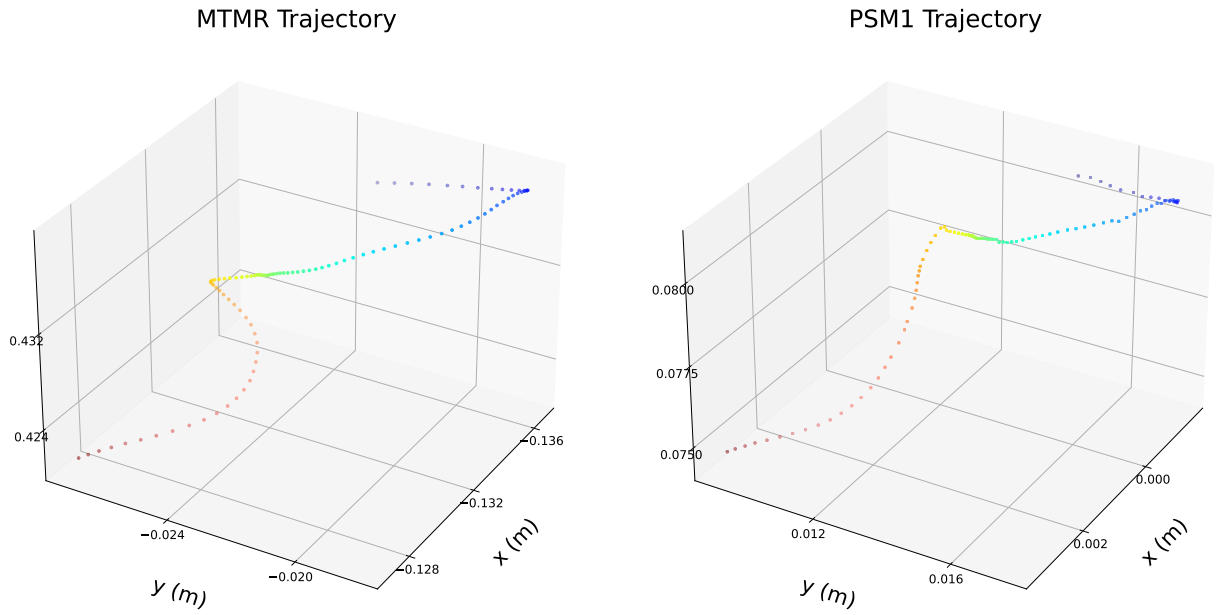
TABLE 3.2. Complete list of kinematic variables and pedal signals in the CRCD. "Local" tip Cartesian pose relates the arm tip to its base frame (e.g., g_{rt} for PSM, g_{se} for ECM). The non-local pose relates the tip to its reference frame (Helper frame for ECM, ECM tip for PSMs, HRSV frame for MTMs).

Type	Features	Dim	Description (Units)
ECM	Endoscope Tip Cartesian Pose (g_{he})	7	Translation $\{x, y, z\}$ (m), Quaternion $\{x, y, z, w\}$
	Local Endoscope Tip Cartesian Pose (g_{se})	7	Translation $\{x, y, z\}$ (m), Quaternion $\{x, y, z, w\}$
	Arm Joint State	12	Joint Position $\{\theta(4)\}$ (rad), Velocity $\{\dot{\theta}(4)\}$ (rad/s), Effort $\{\tau(4)\}$ (N)
MTML	Manipulator Tip Cartesian Pose	7	Translation $\{x, y, z\}$ (m), Quaternion $\{x, y, z, w\}$
	HRSV-relative Tip Cartesian Pose	7	Translation $\{x, y, z\}$ (m), Quaternion $\{x, y, z, w\}$
	Manipulator Joint State	18	Joint Position $\{\theta(6)\}$ (rad), Velocity $\{\dot{\theta}(6)\}$ (rad/s), Effort $\{\tau(6)\}$ (N)
	Gripper Joint State	1	Joint Position $\{\theta(1)\}$ (rad)
MTMR	Manipulator Tip Cartesian Pose	7	Translation $\{x, y, z\}$ (m), Quaternion $\{x, y, z, w\}$
	HRSV-relative Tip Cartesian Pose	7	Translation $\{x, y, z\}$ (m), Quaternion $\{x, y, z, w\}$
	Manipulator Joint State	18	Joint Position $\{\theta(6)\}$ (rad), Velocity $\{\dot{\theta}(6)\}$ (rad/s), Effort $\{\tau(6)\}$ (N)
	Gripper Joint State	1	Joint Position $\{\theta(1)\}$ (rad)
PSM1	Instrument Tip Cartesian Pose (g_{et})	7	Translation $\{x, y, z\}$ (m), Quaternion $\{x, y, z, w\}$
	Local Instrument Tip Cartesian Pose (g_{rt})	7	Translation $\{x, y, z\}$ (m), Quaternion $\{x, y, z, w\}$
	Arm Joint State	18	Joint Position $\{\theta(6)\}$ (rad), Velocity $\{\dot{\theta}(6)\}$ (rad/s), Effort $\{\tau(6)\}$ (N)
	Jaw Joint State	3	Joint Position $\{\theta(1)\}$ (rad), Velocity $\{\dot{\theta}(1)\}$ (rad/s), Effort $\{\tau(1)\}$ (N)
PSM2	Instrument Tip Cartesian Pose (g_{et})	7	Translation $\{x, y, z\}$ (m), Quaternion $\{x, y, z, w\}$
	Local Instrument Tip Cartesian Pose (g_{rt})	7	Translation $\{x, y, z\}$ (m), Quaternion $\{x, y, z, w\}$
	Arm Joint State	18	Joint Position $\{\theta(6)\}$ (rad), Velocity $\{\dot{\theta}(6)\}$ (rad/s), Effort $\{\tau(6)\}$ (N)
	Jaw Joint State	3	Joint Position $\{\theta(1)\}$ (rad), Velocity $\{\dot{\theta}(1)\}$ (rad/s), Effort $\{\tau(1)\}$ (N)
Pedals	Clutch Pedal State	1	Binary state interpolated from pedal events (0 idle, 1 pressed)
	Camera Pedal State	1	Binary state interpolated from pedal events (0 idle, 1 pressed)
	Monopolar Pedal State	1	Binary state interpolated from pedal events (0 idle, 1 pressed)
	Bipolar Pedal State	1	Binary state interpolated from pedal events (0 idle, 1 pressed)

to both the arm base and the High-Resolution Stereo Video (HRSV) frame, the joint states, and the gripper angle. The PSM1 is associated with MTMR (MTM Right) and PSM2 with MTML (MTM Left); when the camera pedal is engaged, both MTMs control the ECM. All kinematic data is sampled at 100 Hz.

Figure 3.2 shows a sample of the recorded 3D trajectories for the MTMR and PSM1 during a brief segment of the procedure, illustrating the correspondence between the surgeon's console manipulations and the resulting instrument motion. The two trajectories are not identical because the dVRK does not enforce a rigid one-to-one mapping between the MTM and PSM tips: the master-side motion is expressed in the console (HRSV) frame and the slave-side motion in the ECM-tip frame, the two are related through a motion-scaling

factor (typically less than one to provide fine-motion control), and each press of the clutch pedal disengages the coupling so that the surgeon can re-center the MTM workspace without moving the PSM. The result is that the PSM1 trace follows the overall shape of the MTMR motion but is compressed in space and contains discontinuities relative to it wherever the clutch was engaged.



(A) MTMR (B) PSM1

FIGURE 3.2. Sample 3D trajectories of the (a) MTMR console manipulator and (b) corresponding PSM1 instrument arm during a brief segment of the procedure.

2.4. Segmentation and Keypoint Annotations. The expanded CRCD [**66**, **67**] augments the endoscopic images with tissue segmentation and instrument keypoint annotations. This subsection summarizes the final annotation release and resulting dataset statistics; Chapter 4 gives the full segmentation-dataset discussion in Sections 1.1 and 1.2, and evaluates the trained perception models in Section 3.

2.4.1. *Segmentation Annotations.* In our initial work [**2**], a custom segmentation dataset was created by manually annotating individual frames using SAM [**44**]. The porcine portion of that dataset contained 1,430 training frames and 356 test frames with only two tissue

classes, liver and gallbladder. This limited scale made it difficult for the model to generalize to new specimens or to tissue whose appearance changed after energy delivery.

To address these limitations, we annotated CRCD surgical videos with the Segment Anything Model 2 (SAM2) [**45**], which propagates masks across video frames from point-based prompts. The selected clips span multiple surgeons and multiple stages of dissection, capturing the tissue deformation, color variation, and cauterization patterns that were absent from the initial dataset. After mask propagation, we manually assigned semantic labels to each region to create a structured dataset suitable for model training.

The final segmentation release contains three tissue classes: *liver*, *gallbladder*, and *liver bed*. The liver bed denotes the exposed liver surface that was previously attached to the gallbladder. This class becomes visible only after partial dissection and is essential for distinguishing intact liver from the cauterized dissection frontier after repeated rounds of energy delivery.

2.4.2. *Keypoint Annotations.* Keypoint annotations for surgical instruments—Fenestrated Bipolar Forceps (FBF) and Permanent Cautery Hook (PCH)—were performed manually using the COCO annotator [**93**]. The annotation schema places landmarks at joints, tips, and other visually distinctive features to maximize localization robustness. The full instrument-specific layouts and qualitative prediction examples are presented in Chapter 4.

Both datasets adhere to Microsoft's COCO format [**4**], ensuring compatibility with standard computer vision frameworks. Table 3.3 summarizes the final annotation counts. In total, the segmentation release contains 34,678 annotated frames, and the keypoint release adds 15,999 instrument instances.

3. Surgical Task

3.1. Setup. The recordings took place in the setup shown in Fig. 3.3, where the surgeon controls the robot from the da Vinci console and performs the assigned task on a porcine liver placed on a table with the gallbladder covered by the liver, closely matching the actual *in vivo* positioning. The dataset comprises seven surgeons, denoted alphabetically from "A"

TABLE 3.3. Annotation statistics for the expanded CRCD used throughout this dissertation. The segmentation split contains 25,988 training frames and 8,690 test frames; liver and gallbladder are labeled in every frame, while the liver bed appears only after partial dissection. Keypoint annotations were performed manually using the COCO annotator.

Data Type	Categories	Train	Test
Segmentation	Liver	25,988	8,690
	Gallbladder	25,988	8,690
	Liver Bed	21,660	7,261
Keypoints	FBF	5,476	1,372
	PCH	7,320	1,831

TABLE 3.4. Contribution of each surgeon to the CRCD. The experience column reports the total number of laparoscopic procedures performed. Red entries indicate incomplete data due to video compression damage (Surgeon A, D) or Arduino shutdown (Surgeon F).

Surgeon	Video	Kinematics	Pedals	Experience (# Procedures)
A	1	3	3	150
B	3	3	3	1,500
C	3	3	3	225
D	0	3	3	65
E	3	3	3	1,000
F	3	3	0	225
G	3	3	3	1,000
Total	16	21	18	–

through "G," all with experience in surgical robotic cholecystectomy. Each surgeon performed the task three times, using a new liver for each attempt.

The duration of each trial varied with the difficulty of the task, influenced primarily by the degree of liver decay. Challenges arose when the liver and gallbladder were similar in color, making it difficult to distinguish between the two organs. Table 3.4 provides details on the data recorded for each surgeon. Some videos were damaged during compression and were excluded from the dataset. Occasional shutdowns of the Arduino occurred when high current was applied to the instrument tip, corrupting the pedal recordings for the affected trials.

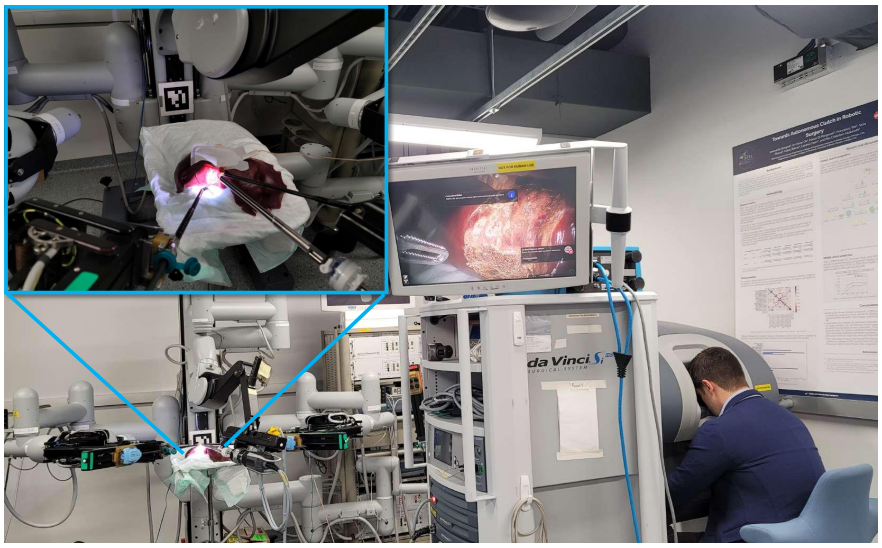


FIGURE 3.3. Environment setup for the *ex vivo* cholecystectomy. The surgeon operates the da Vinci console while the porcine liver with attached gallbladder is positioned on the surgical table.

3.2. Surgeon Profiles. Understanding the backgrounds and experience levels of surgeons is critical for analyzing RAS data. The skills and expertise of surgeons significantly impact the successful execution of robotic procedures, given the nuanced control and precise manipulation required. The expanded CRCD includes the surgical background of each participant: the total number of procedures, the breakdown of laparoscopic versus robotic cases, and the hours of training in robotic surgery. Procedures are counted if they are endoscopically guided and are categorized by complexity (low/mid/high).

Several studies [**94**,**95**] demonstrate a direct connection between laparoscopic and robotic-assisted surgical skills, and Hedrick et al. [**96**] showed that machine learning models trained on laparoscopic datasets can assess surgeon performance in RAS. The total experience for each surgeon is summarized in Table 3.4; full details including complexity breakdowns and training hours are available in the CRCD GitHub repository[1].

This information enables the development of models that predict surgeon performance and personalize the robotic system's assistance during surgery, accounting for the relationship between training background and proficiency [**85**, **97**].

[1]`https://github.com/sitleng/CRCD`

3.3. Procedure. The surgeons performed the task following the UIC standardized surgical technique for robotic cholecystectomy [**1**]. The 11 primary steps of the procedure are:

(1) Working area exposure
(2) Gallbladder neck retraction
(3) Calot triangle: anterior peritoneal layer opening
(4) Calot triangle: posterior peritoneal layer opening
(5) Cystic duct isolation
(6) Cystic artery isolation
(7) Cystic duct clipping
(8) Cystic artery clipping *(omitted)*
(9) Cystic duct and artery division *(omitted)*
(10) Detachment of the gallbladder from the liver
(11) Specimen retrieval in an Endobag$^{\text{TM}}$ *(omitted)*

The order of certain steps may vary depending on anatomical considerations, and the steps marked above were omitted to accommodate the *ex vivo* animal model.

4. Preliminary Applications

This section highlights pedal intent recognition as one example application enabled by the CRCD.

4.1. Pedal Intent Recognition. In robotic cholecystectomy, recognizing the surgeon's intent to activate clutch or camera pedals is essential for optimizing procedural efficiency and alleviating cognitive workload. Currently, the surgeon takes full control of the robotic system without assistance. However, a dataset combining kinematics and pedal signals enables the development of assistive systems that could predict when pedal activation is needed and support the surgeon during the procedure [**66**, **67**].

4.1.1. *Data Processing.* The pedal signals ($\sim$230 Hz) and kinematic data ($\sim$100 Hz) were first synchronized by matching timestamps within a threshold of $\epsilon = 0.006$ s. We then generated training samples with a sliding-window approach [**98**], randomly sampling windows

TABLE 3.5. Composition of the pedal dataset, showing the severe class imbalance between pressed and not-pressed states.

Pedal Type	Not Pressed	Pressed
Clutch	1,082,871	4,845
Camera	967,704	31,095

TABLE 3.6. Precision, recall, and F1 scores for the TST pedal intent recognition model on the test set, evaluated across three window sizes for both camera and clutch pedals.

	Camera			Clutch		
Window Size	40	60	80	40	60	80
Precision	0.995	0.991	0.996	0.957	0.940	0.987
Recall	0.967	0.968	0.977	0.992	0.960	0.990
F1 Score	0.981	0.979	0.987	0.974	0.950	0.989

from the synchronized recordings. Each training sample has size $X \in \mathbb{R}^{f \times w}$, where f is the number of kinematic features (arm and manipulator poses) and w is the window size. This preprocessing is reproducible from the public CRCD repository.

4.1.2. *Time Series Transformer.* We trained the Time Series Transformer (TST) [**99**, **100**], which encodes the time-series input to fit the transformer encoder architecture [**101**]. The encoder output is passed to a linear layer, and the model is trained to minimize squared error between predictions and ground truth labels.

The model was trained with window sizes $w \in \{40, 60, 80\}$. The composition of the pedal dataset is shown in Table 3.5. Since the pedals are idle for the majority of the procedure, we reduced the majority class so that the ratio between the two classes was 15, chosen experimentally. After window generation, the data was split in a 7:3 ratio for training and testing. Table 3.6 presents the precision, recall, and F1 score for each trained model.

4.1.3. *Zero-Shot Evaluation.* Figure 3.4 shows the performance of the trained models on data from a surgeon whose recordings were excluded from the original train/test split, providing a separate zero-shot evaluation of cross-surgeon generalization. For this experiment, the TST models predicted pedal states by sliding a window with a step size of 2 samples.

All three window sizes performed similarly for the camera pedal. For the clutch pedal, the model trained with a window size of 80 performed best on the test set, achieving the highest precision (0.987) and F1 score (0.989) in Table 3.6; the shorter windows failed to capture sufficient temporal context for the rarer clutch presses. The difference in performance between camera and clutch prediction stems from the more frequent use of the camera pedal (Table 3.5), which provides more training examples.

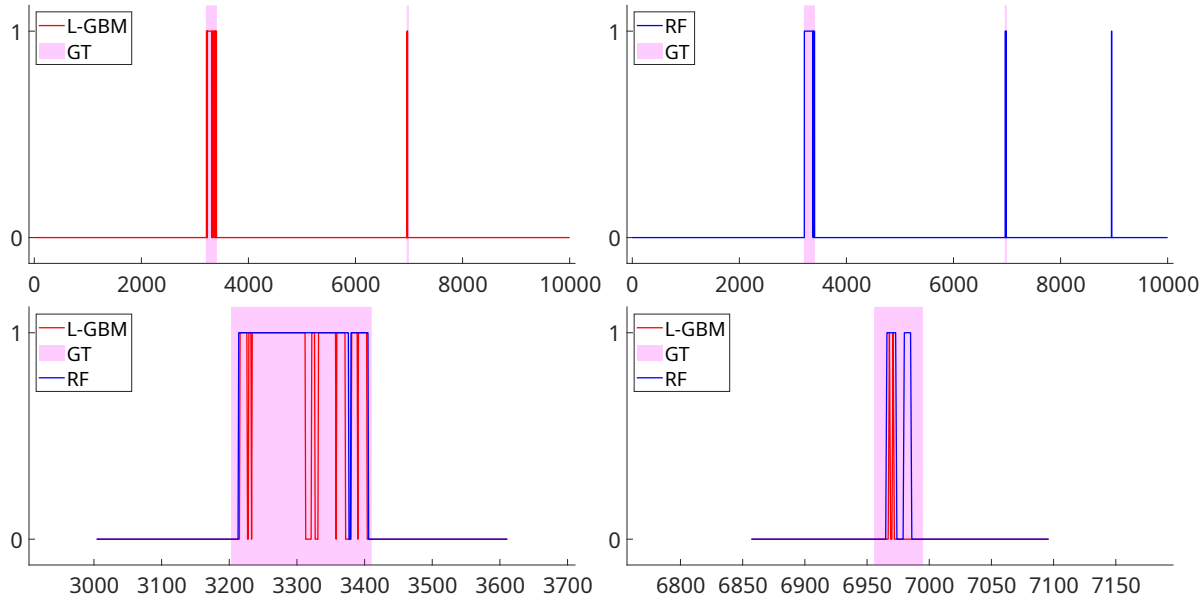


FIGURE 3.4. Zero-shot test of pedal prediction on an unseen surgeon. The models predict clutch (left) and camera (right) pedal states using three different window sizes (40, 60, 80).

Beyond pedal intent recognition, the CRCD also supports the perception and stereo reconstruction studies developed elsewhere in this dissertation. Chapter 4 details the segmentation annotation workflow and full instance segmentation evaluation in Sections 1.2 and 3, while Chapter 2, Section 3 defines the stereo reconstruction pipeline that operates on the released stereo images and calibration parameters shown in Fig. 3.1.

5. Limitations and Discussion

The primary limitation of the CRCD stems from differences between *ex vivo* and *in vivo* surgical environments. The workspace of the robotic arms is less constrained in the *ex vivo*

setting, where the body wall is absent, leading to a broader range of arm configurations than would be encountered *in vivo*. Additionally, the endoscopic view differs: *in vivo* procedures generally produce a brighter surgical field due to light reflections from the body wall. However, this does not substantially affect tissue segmentation or instrument keypoint detection, as these elements are typically centered in the endoscopic image where illumination is adequate. From a surgical perspective, the procedure is comparable: the liver was positioned as it would be in an actual procedure, with the gallbladder covered by the liver and requiring assistance to lift.

Some data quality issues were encountered during collection. Video compression artifacts damaged recordings from certain trials (Surgeon A lost two videos; Surgeon D lost all three), and occasional Arduino shutdowns during high-current energy delivery corrupted the pedal recordings for Surgeon F. These exclusions are documented in Table 3.4.

Despite these limitations, the CRCD provides a multimodal dataset that brings together stereo endoscopic video, complete kinematic data for all da Vinci arms and console manipulators, pedal signals, and dense frame-level annotations for cholecystectomy procedures. The dataset supports the perception studies in Chapter 4, the pedal experiments in this chapter, and the vision-based kinematics prediction work in Chapter 7.

CHAPTER 4

Perception

Automating surgical procedures requires the robot to perceive the surgical scene in real time: it must identify the tissue types, localize the boundaries between them, and track the positions of surgical instruments. Building on the CRCD introduced in Chapter 3, this chapter focuses on the perception pipeline that underpins the autonomous dissection framework. We summarize the evolution of the custom annotations from an initial small-scale collection created with SAM [**44**] (Section 1) to the expanded SAM2-based CRCD annotation release used in the final system. We then detail the three perception model families trained on these datasets—Detectron2 [**38**], MaskDINO [**41**], and YOLO11 [**43**]—covering their architectures and training configurations (Section 2). Finally, we present a complete evaluation of all perception models (Section 3): the Detectron2 baseline on the initial v1 dataset, followed by a comparative analysis of all three model families on the expanded v2 dataset, including their practical impact on autonomous dissection performance. The hardware platform (da Vinci with dVRK, Si endoscope) is described in Chapter 2 and is not repeated here.

1. Dataset Generation

A robust dataset is the foundation for training perception models that generalize across specimens, lighting conditions, and tissue states encountered during surgery. Chapter 3 described the source videos, recording protocol, and final annotation release. This section focuses only on the label design and dataset splits that matter for training the perception models.

1.1. Initial Segmentation Dataset. The initial segmentation dataset was developed to enable automated tissue recognition in endoscopic images of *ex vivo* specimens [**2**]. Because no existing public dataset matched our experimental setting—CholecSeg8k [**34**] and CholecTriplet2021 [**35**] are based on *in vivo* human liver surgeries, whereas this work uses *ex vivo* porcine models—we created a custom dataset from scratch. Differences in anatomy, surgical environment, and imaging characteristics between *in vivo* human and *ex vivo* porcine settings result in significant variations in tissue and instrument appearance under endoscopic cameras.

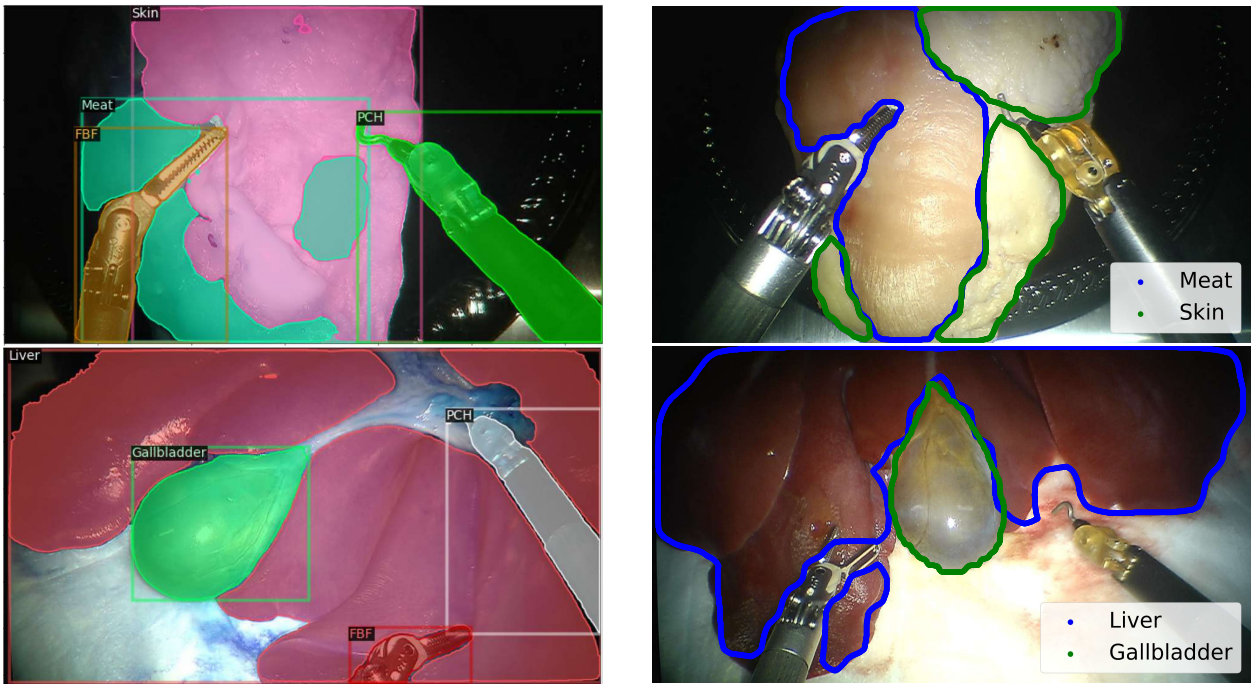
1.1.1. *Annotation Pipeline.* We used the Segment Anything Model (SAM) [**44**], which segments objects in an image given point or bounding-box prompts without task-specific training. For each frame, we provided manually selected positive and negative point prompts to indicate foreground (tissue of interest) and background regions, respectively. SAM produces high-quality segmentation masks but is class-agnostic: it cannot label the segmented regions semantically. We therefore manually assigned a tissue category to each mask to create a labeled training set suitable for downstream models such as Detectron2 [**38**].

1.1.2. *Categories.* The dataset contains four tissue classes:

- **Chicken Meat** and **Chicken Skin**: used in early experiments where the skin–muscle boundary served as a proxy for the gallbladder–liver boundary.
- **Pig Liver** and **Pig Gallbladder**: the actual target tissues for robotic cholecystectomy.

Endoscopic images were recorded while performing motions with both robotic arms. To ensure diversity, the endoscope angle and the position of the *ex vivo* material were varied across recording sessions, yielding a total of 2820 raw images. Table 4.1 details the number of segmentation and keypoint annotations. The dataset is formatted according to Microsoft's COCO format [**4**], ensuring compatibility with standard object detection frameworks. Representative annotations and model predictions are shown in Figure 4.1.

1.1.3. *Limitations.* While this dataset enabled the first end-to-end dissection experiments (Chapter 5), it had several shortcomings that became apparent during evaluation:



(A) SAM annotations (B) DT2-seg predictions

FIGURE 4.1. Initial segmentation dataset (v1). (a) Samples of the manually annotated segmentation dataset created with SAM. (b) Corresponding segmentation predictions from the trained Detectron2 model, demonstrating generalization across chicken and porcine specimens.

TABLE 4.1. Annotation counts for the initial (v1) dataset. Segmentation masks were generated semi-automatically with SAM. Keypoints were annotated manually for the Large Needle Driver (LND), Fenestrated Bipolar Forceps (FBF), and Permanent Cautery Hook (PCH).

Data Type	**Categories**	**Train**	**Test**
Segmentation	Chicken Meat	1390	348
	Chicken Skin	1364	343
	Pig Liver	1430	356
	Pig Gallbladder	1429	359
Keypoints	LND	1188	277
	FBF	471	130
	PCH	1904	477

(1) **Limited specimen variability.** The images were sampled from a small number of specimens, so the model struggled to generalize when tissue color or shape deviated from the training distribution—for example, encountering a yellowish gallbladder instead of the typical dark green.

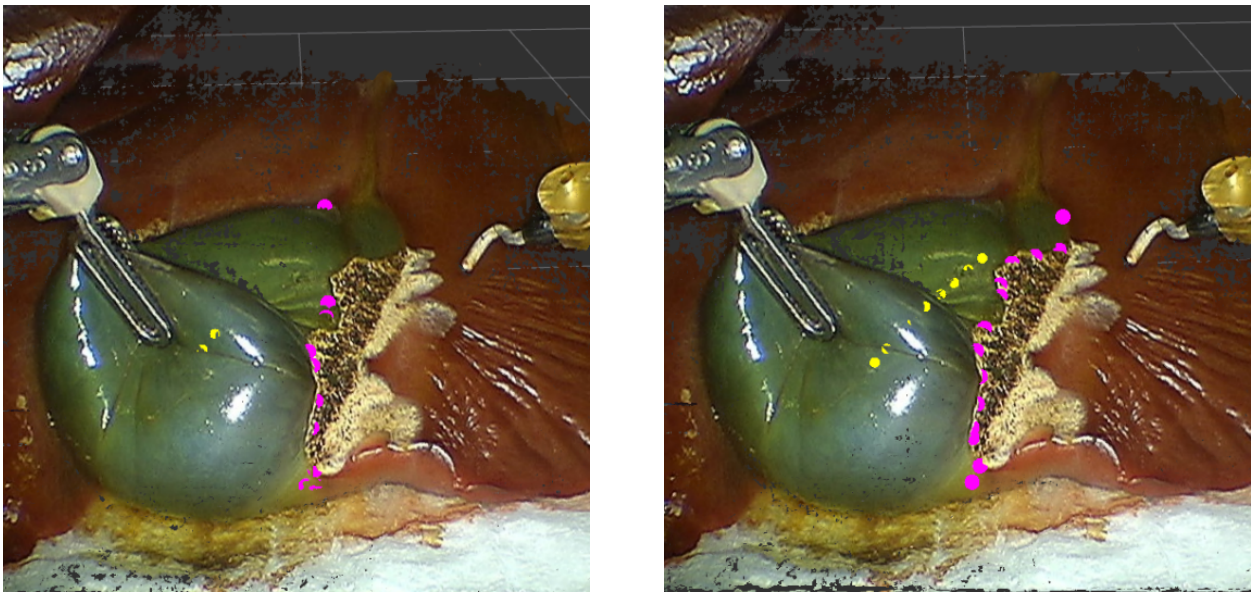
(2) **No representation of dissected tissue.** After monopolar energy is applied, the tissue surface changes in both color and texture. The v1 dataset contained no images of partially dissected tissue, causing the model to fail after even a single round of dissection.

(3) **Missing liver bed class.** Once the gallbladder is partially separated from the liver, the underlying liver surface (the *liver bed*) becomes exposed. The v1 dataset had no label for this region, preventing the model from distinguishing intact liver from the liver bed. This distinction becomes important after repeated rounds of energy delivery and is a prerequisite for future multi-round dissection.

1.2. Expanded Segmentation Dataset. To address the limitations of the initial dataset, we created a substantially expanded segmentation dataset [**66**, **67**] with three key improvements: greater specimen diversity, a new *liver bed* tissue class, and a much larger set of annotated frames.

1.2.1. *Motivation.* Figure 4.2 illustrates the failure mode that motivated the dataset expansion. After several rounds of monopolar energy delivery, the tissue surface is altered and the boundary between gallbladder and liver becomes ambiguous. The v1 model (Figure 4.2a) produces a disconnected boundary and fails to segment the full gallbladder, yielding an incomplete skeleton. This example motivated a model that remains accurate as the tissue appearance evolves, both for the later stages of a dissection cycle and for future extensions to multiple autonomous rounds.

1.2.2. *The Liver Bed Class.* We introduced a new tissue label, *liver bed*, representing the liver surface that was previously in contact with the gallbladder (Figure 4.3). After dissection, this region is visually distinct from the surrounding liver parenchyma: it is lighter in color, smoother in texture, and often bears cauterization marks. Without an explicit label for this class, models confuse the liver bed with either the gallbladder or the intact liver, producing erroneous tissue boundaries. Including the liver bed as a separate class allows the model to correctly delineate the dissection frontier throughout the procedure.



(A) DT2-seg (v1 dataset) [**2**]

(B) YOLO11l-seg (v2 dataset)

FIGURE 4.2. Segmentation performance after repeated rounds of energy delivery during dissection. (a) The v1 model produces a disconnected boundary and an incomplete gallbladder skeleton. (b) A model trained on the expanded v2 dataset accurately detects the boundary and the full gallbladder with a complete skeleton, even after tissue deformation from energy delivery. This comparison is intended as an illustrative failure case rather than a controlled model ablation.

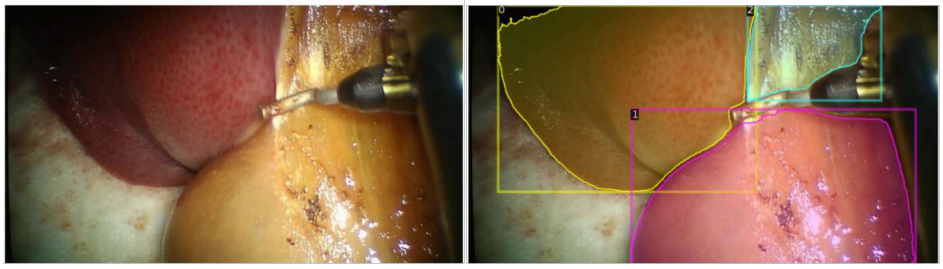


FIGURE 4.3. Annotation example from the expanded segmentation dataset (v2). Three tissue classes are labeled: Liver (orange), Gallbladder (pink), and Liver Bed (green). The liver bed represents the exposed liver surface where the gallbladder was previously attached.

1.2.3. *Annotation Pipeline.* The expanded segmentation dataset is derived from CRCD videos using the SAM2-based annotation workflow summarized in Chapter 3, Section 2.4. For perception training, the key outcome is that masks were propagated across video frames and then manually assigned one of three semantic labels—liver, gallbladder, or liver bed—to produce a structured training dataset.

TABLE 4.2. Comparison of annotation counts between the initial (v1) and expanded (v2) datasets. The v2 dataset expands the porcine segmentation split to 25,988 training frames and 8,690 test frames, adds the liver bed class, and substantially increases the number of keypoint annotations. Both datasets follow the MS COCO format [4].

Data Type	Categories	v1 [2]		v2 [66, 67]	
		Train	Test	Train	Test
Segmentation	Pig Liver	1430	356	25988	8690
	Pig Gallbladder	1429	359	25988	8690
	Liver Bed	—	—	21660	7261
Keypoints	FBF	471	130	5476	1372
	PCH	1904	477	7320	1831

1.2.4. *Dataset Statistics.* Table 4.2 presents the annotation counts for both the initial (v1) and expanded (v2) datasets. The v2 segmentation release contains 34,678 annotated frames and introduces the liver bed class with 21,660 training instances and 7,261 test instances. The chicken categories from v1 were dropped, as all subsequent experiments focus on porcine tissue. The keypoint dataset was similarly expanded (Section 1.3).

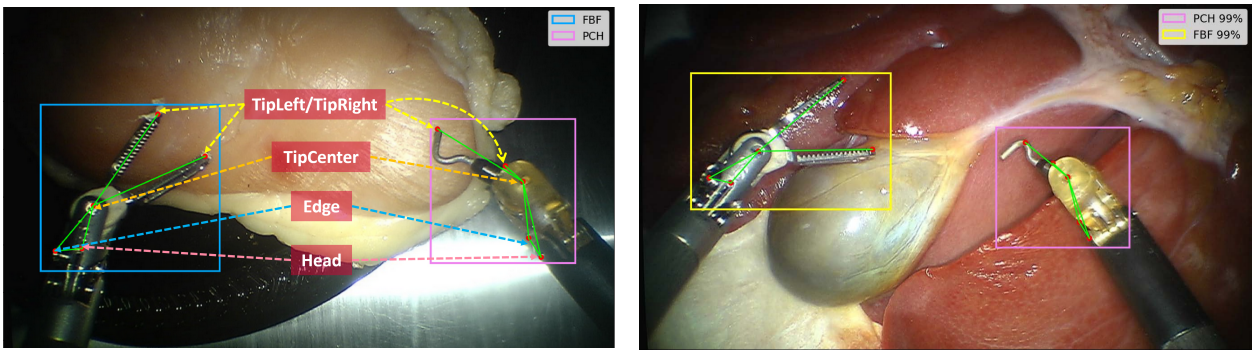
1.3. Keypoint Dataset. Keypoint detection provides the perception system with the 3D pose of each surgical instrument, which is essential for the control algorithms described in Chapter 5. Keypoints are placed on structurally distinctive parts of each instrument—joints, screws, and tips—that exhibit color and geometric contrast for reliable detection.

1.3.1. *Original Keypoint Structure (v1).* In the initial dataset [2], all instruments shared a single five-keypoint structure:

- **TipRight / TipLeft:** the two tips of the gripper (for instruments with grippers) or the two edges of the hook (for the PCH).
- **TipCenter:** the sixth joint of the PSM (the screw that rotates the gripper or hook).
- **Edge:** the left-side screw of the instrument shaft.
- **Head:** the fifth joint (PCH) or the top screw (other instruments).

Three instruments were annotated: the Large Needle Driver (LND), Fenestrated Bipolar Forceps (FBF), and Permanent Cautery Hook (PCH). Annotations were performed manually

using the COCO annotator [**93**] on 2820 endoscopic images recorded during random arm motions. Representative annotations and predictions are shown in Figure 4.4, and instance counts are listed in Table 4.1.



(A) Manual annotations (B) DT2-kpt predictions

FIGURE 4.4. Initial keypoint dataset (v1). (a) Example of manually annotated keypoints using the unified five-point structure shared across all instruments. (b) Corresponding keypoint predictions by the trained Detectron2 model.

1.3.2. *Updated Keypoint Structure (v2).* For the expanded dataset [**66**, **67**], we adopted a different keypoint structure for each instrument to improve robustness against common transformations such as rotation and partial occlusion. Figure 4.5 illustrates the instrument-specific keypoint layouts for the FBF and PCH, which are the two instruments used in the autonomous dissection framework. The LND keypoint structure remained unchanged from v1, as it is not used in the dissection procedure.

The keypoints are strategically placed on instrument regions with distinct colors and edges to optimize detection accuracy. Annotations were again performed manually using the COCO annotator [**93**], and the dataset adheres to the MS COCO format [**4**]. As shown in Table 4.2, the v2 keypoint dataset is substantially larger than v1: FBF annotations increased from 601 to 6848 instances (11.4×), and PCH annotations from 2381 to 9151 instances (3.8×). Both the segmentation and keypoint datasets are publicly available as part of the Expanded CRCD [**66**, **67**].

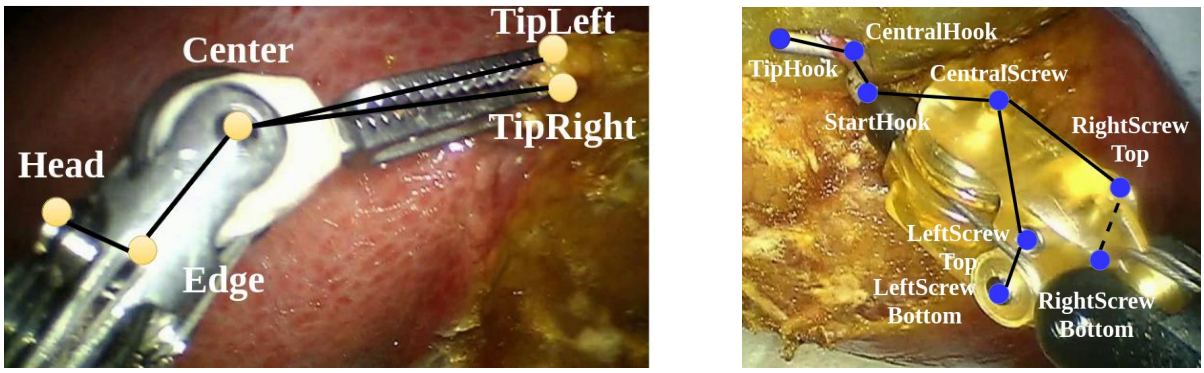


FIGURE 4.5. Updated keypoint structures (v2) for the Fenestrated Bipolar Forceps (FBF, left) and Permanent Cautery Hook (PCH, right). Each instrument has a distinct set of keypoints tailored to its geometry, improving detection robustness compared to the shared structure used in v1.

2. Perception Models

We trained three families of perception models for tissue segmentation and instrument keypoint detection. All models were first pre-trained on large-scale vision datasets and then fine-tuned on our custom datasets described in Section 1.

2.1. Detectron2. Detectron2 [**38**] is Facebook AI Research's modular framework for object detection and segmentation, built on Mask R-CNN [**37**]. We used two independent model heads within Detectron2:

- **DT2-seg**: an instance segmentation model that produces per-pixel masks for each detected tissue region.
- **DT2-kpt**: a keypoint detection model (originally designed for human pose estimation) that localizes a predefined set of keypoints for each detected instrument.

The distinction between DT2-seg and DT2-kpt is important: they are trained independently with different annotation types. DT2-seg learns from segmentation masks (connected pixel regions), while DT2-kpt treats each foreground object as a set of sparse keypoint locations.

Both models share a ResNet-50 [**39**] backbone with a Feature Pyramid Network (FPN) [**40**], pre-trained on ImageNet [**102**]. We selected the R50-FPN configuration for its favorable tradeoff between inference speed and detection accuracy. The training hyperparameters were:

- Initial learning rate: 0.05 with a linear warm-up over 1000 iterations.
- Total training iterations: 9000.
- Learning rate schedule: reduced by 20% every 300 iterations.
- All other parameters: Detectron2 defaults.

The same hyperparameters were used for both DT2-seg and DT2-kpt. When fine-tuned on the v1 dataset, DT2-seg handled four tissue classes (chicken meat, chicken skin, pig liver, pig gallbladder) and DT2-kpt handled three instrument types (LND, FBF, PCH). For the v2 dataset, DT2-seg was retrained on three porcine tissue classes (liver, gallbladder, liver bed) and DT2-kpt on two instruments (FBF, PCH).

2.2. MaskDINO. MaskDINO [**41**] is a unified transformer-based framework for object detection and instance segmentation. It extends the DINO detector [**64**] by adding a mask prediction branch, enabling a single model to perform both bounding-box detection and pixel-level segmentation. Unlike the two-stage Mask R-CNN architecture used in Detectron2, MaskDINO uses a transformer encoder–decoder architecture with deformable attention, which enables it to model long-range dependencies across the image and often yields sharper mask boundaries.

We adopted MaskDINO as a stronger baseline for tissue segmentation on the v2 dataset, motivated by its state-of-the-art performance on standard benchmarks at the time of our experiments [**3**]. The model was fine-tuned on the same three porcine tissue classes (liver, gallbladder, liver bed) using the v2 dataset. MaskDINO was used only for segmentation; keypoint detection continued to rely on the dedicated DT2-kpt or YOLO11-pose models.

2.3. YOLO11. YOLO11 [**43**] is the latest iteration of the YOLO (You Only Look Once) family of real-time object detection models, developed by Ultralytics. It provides several task-specific variants:

- **YOLO11-seg**: instance segmentation, producing per-pixel masks alongside bounding boxes.

- **YOLO11-pose**: keypoint detection (analogous to DT2-kpt), localizing a set of keypoints for each detected object.

YOLO11 is available in multiple sizes (nano, small, medium, large, extra-large). We selected the **YOLO11l** (large) variant for both segmentation and keypoint detection. The large model delivers performance comparable to the extra-large variant (YOLO11x) while offering approximately twice the inference speed [**43**], making it better suited for the real-time requirements of autonomous surgery. Throughout this dissertation, YOLO11l-seg and YOLO11l-pose refer to the large variant fine-tuned for segmentation and keypoint detection, respectively.

As a single-stage detector, YOLO11 processes the entire image in a single forward pass, which yields lower latency than the two-stage Detectron2 pipeline. This architectural advantage is particularly relevant for the autonomous dissection framework (Chapter 5), where perception latency directly affects the control loop frequency.

3. Model Comparison

This section presents the evaluation of all perception models used in this dissertation. We first report the performance of the Detectron2 models on the initial v1 dataset, which served as the baseline for the first autonomous dissection experiments. We then present a comparative evaluation of all three model families on the expanded v2 dataset. All Average Precision (AP) scores follow the COCO evaluation protocol [**4**]: specifically $\text{mAP}_{50\text{-}95}$, the mean AP averaged over IoU thresholds from 0.50 to 0.95 in steps of 0.05.

3.1. Baseline: Detectron2 on the v1 Dataset. Table 4.3 presents the AP scores for the Detectron2 models trained on the initial dataset. The DT2-seg model achieved high bounding box AP across all tissue categories, confirming its ability to localize tissues within the endoscopic image. However, segmentation AP for the pig liver was notably low (37.5), indicating difficulty in precisely delineating liver tissue from the dark background, especially near the outer edges of the endoscopic images. Since the dataset was acquired in an *ex vivo* setting, it lacked the light reflections from body walls typically present in *in*

TABLE 4.3. Average Precision (AP) scores for the Detectron2 models trained on the v1 dataset. Bbox. = Bounding Box, Seg. = Segmentation, Kpt. = Keypoints.

Categories	**AP (Bbox.)**	**AP (Seg.)**	**AP (Kpt.)**
Chicken Meat	91.3	78.6	-
Chicken Skin	86.1	61.3	-
Pig Liver	97.4	37.5	-
Pig Gallbladder	89.2	94.3	-
LND	81.0	-	99.0
FBF	77.1	-	94.6
PCH	74.2	-	98.4

vivo environments, causing the outer regions of the tissues to appear darker and adversely impacting segmentation accuracy.

The DT2-kpt model had lower bounding box AP scores than DT2-seg, likely due to inconsistencies in manual bounding box annotations that served primarily as contextual references for keypoint locations. Despite this, the model achieved high keypoint detection AP scores, with particularly strong performance for the LND (99.0) and PCH (98.4) instruments.

3.2. Tissue Instance Segmentation. Table 4.4 presents the segmentation AP scores for DT2-seg, MaskDINO, and YOLO11l-seg on the three porcine tissue classes. YOLO11l-seg achieved the highest overall performance, with near-perfect bounding-box AP (≥ 99.2) across all categories and the best segmentation AP for liver (98.6) and liver bed (93.0). MaskDINO obtained the highest segmentation AP for gallbladder (99.0), demonstrating particularly strong mask quality for this class.

Both DT2-seg and MaskDINO struggled with the *liver* class. We attribute this to the visual similarity between intact liver and liver bed: both are regions of the liver surface, differing primarily in color and texture due to cauterization. When the model confuses these two classes, the liver bounding box becomes fragmented or incorrectly sized, depressing the bounding-box AP even when the segmentation masks are reasonable. This difficulty is less critical during the initial stages of dissection, when the liver bed is not yet exposed. However,

TABLE 4.4. Average Precision (AP) scores for tissue instance segmentation on the v2 dataset. Bbox. = bounding-box AP; Seg. = segmentation mask AP. Bold indicates the best score per category. YOLO11l-seg achieves the highest overall performance, while MaskDINO excels at gallbladder segmentation.

Model	Categories	AP (Bbox.)	AP (Seg.)
DT2-seg	Liver	61.2	83.9
	Gallbladder	96.2	89.8
	Liver Bed	91.1	75.3
MaskDINO	Liver	55.3	80.0
	Gallbladder	96.7	**99.0**
	Liver Bed	88.9	91.3
YOLO11l-seg	Liver	**99.3**	**98.6**
	Gallbladder	**99.5**	97.7
	Liver Bed	**99.2**	**93.0**

it becomes a significant issue in later stages after the gallbladder has been partially separated and cauterization marks are visible on the liver surface.

YOLO11l-seg's ability to differentiate liver from liver bed—reflected in its markedly higher liver AP (98.6 vs. 83.9 for DT2-seg and 80.0 for MaskDINO)—makes it the preferred model for the later stages of dissection, where the tissue appearance evolves throughout the procedure, and for future multi-round extensions.

3.2.1. *Practical Impact During Dissection.* The practical consequences of these AP differences become apparent as dissection progresses and repeated rounds of energy delivery alter the tissue surface (Chapter 6). The newly exposed liver bed is visually similar to the surrounding intact liver, creating a challenging discrimination task. Figure 4.2 illustrates this challenge: the DT2-seg model, trained on the v1 dataset that lacked a liver bed class, produced a disconnected boundary and failed to detect the full extent of the gallbladder, resulting in an incomplete skeleton. In contrast, YOLO11l-seg, trained on the v2 dataset with the liver bed class, accurately detected the boundary and the entire gallbladder under these more demanding visual conditions. While the liver–liver bed confusion may not

TABLE 4.5. Average Precision (AP) scores for instrument keypoint detection on the v2 dataset. Bbox. = bounding-box AP; Kpt. = keypoint AP. Bold indicates the best score per category. YOLO11l-pose achieves higher keypoint AP for both instruments.

Model	Categories	AP (Bbox.)	AP (Kpt.)
DT2-kpt	FBF	**77.1**	94.6
	PCH	74.2	98.4
YOLO11l-pose	FBF	66.8	**97.2**
	PCH	**85.4**	**98.6**

affect the initial dissection round, it becomes increasingly problematic as the procedure progresses, making YOLO11l-seg the most reliable choice for the current pipeline and a stronger foundation for future multi-round autonomy.

3.3. Instrument Keypoint Detection. Table 4.5 compares DT2-kpt and YOLO11l-pose on the expanded keypoint dataset. YOLO11l-pose outperforms DT2-kpt in keypoint AP for both instruments, achieving 97.2 for FBF (vs. 94.6) and 98.6 for PCH (vs. 98.4). For bounding-box detection, YOLO11l-pose substantially outperforms DT2-kpt on the PCH (85.4 vs. 74.2), while DT2-kpt retains a higher FBF bounding-box AP (77.1 vs. 66.8). The bounding-box variability for DT2-kpt can be attributed to inconsistencies in the annotated bounding boxes, which primarily served as contextual reference for keypoint locations rather than being optimized for detection accuracy.

Figure 4.6 provides a qualitative comparison. DT2-kpt occasionally fails to detect keypoints when the instrument is near the edges of the endoscopic image (highlighted by the red rectangle), where illumination drops off and the instrument is partially out of frame. YOLO11l-pose handles these challenging cases more robustly, producing more consistent keypoint localizations across the field of view. This robustness is particularly important for autonomous dissection, where the instrument frequently operates near the image periphery as it follows the tissue boundary.

3.3.1. *Practical Impact During Dissection.* Accurate keypoint detection is essential for the Position-Based Visual Servoing control loop (Chapter 5): errors in the estimated 3D

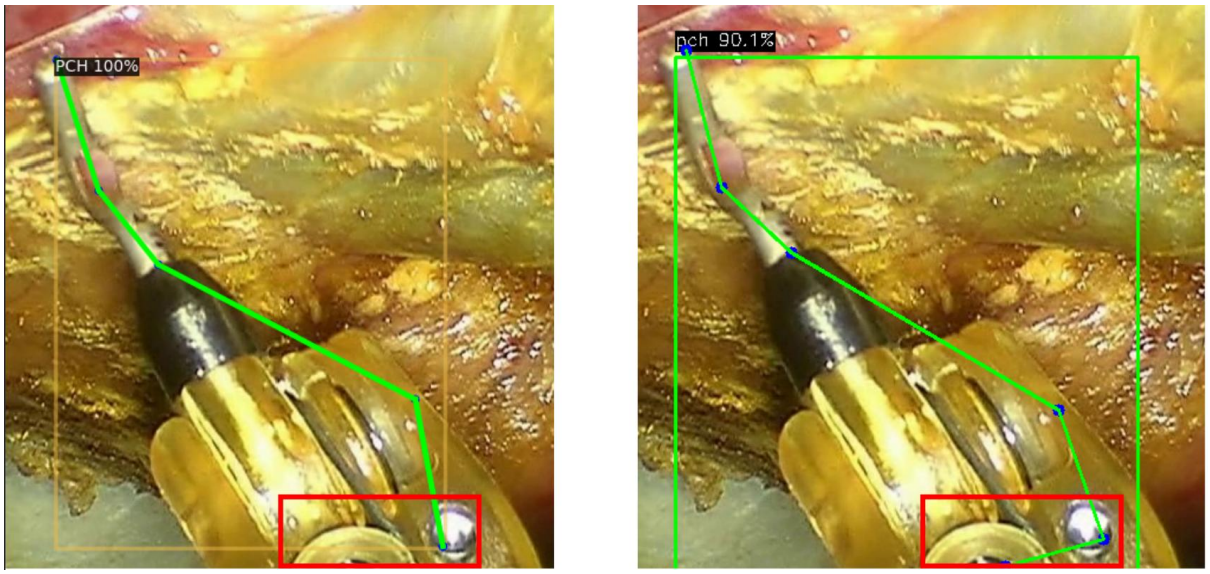


FIGURE 4.6. Qualitative comparison of keypoint detection. DT2-kpt (left) fails to detect certain keypoints when the instrument is near the image edge (red rectangle), while YOLO11l-pose (right) maintains robust keypoint localization across the full field of view.

tip position directly translate into oscillatory movements, increased travel distance, and longer procedure duration. The edge-of-frame issue is inherent to endoscopic cameras, where illumination is concentrated at the center and diminishes toward the periphery. DT2-kpt frequently failed to detect keypoints in these peripheral regions, whereas YOLO11l-pose maintained reliable detection. As detailed in Chapter 6, inaccurate keypoint predictions from DT2-kpt caused oscillatory arm movements that increased both travel distance and procedure duration, even when the segmentation model performed well. Keypoint detection accuracy—particularly in edge cases—is therefore the primary bottleneck for dissection speed and consistency.

3.4. Discussion. The comparative evaluation reveals a clear hierarchy among the three model families on our surgical perception tasks. YOLO11l-seg is the best overall segmentation model, with particularly strong differentiation between visually similar tissue classes (liver vs. liver bed). MaskDINO achieves the highest individual score for gallbladder segmentation but struggles with the liver–liver bed distinction. DT2-seg, while adequate for initial

experiments with the v1 dataset, falls behind both alternatives on the more challenging v2 dataset.

For keypoint detection, YOLO11l-pose provides a modest but consistent improvement over DT2-kpt in keypoint localization accuracy, with the added benefit of substantially better robustness near image boundaries. Since the control algorithms in Chapter 5 rely on accurate instrument tip positions derived from these keypoints, this improved robustness directly translates to more stable robotic motion during dissection.

CHAPTER 5

Autonomous Dissection

This chapter presents the methodology for autonomous dissection along tissue boundaries, showing its evolution from an initial single-arm, offline trajectory-following approach [**2**] to a bimanual online framework with grasping, tissue stretching, and real-time boundary tracking [**3**]. Both versions build on the hardware platform described in Chapter 2 (da Vinci with dVRK, Si endoscope, ESU control, custom arm calibration) and the perception pipeline from Chapter 4 (tissue segmentation and instrument keypoint detection). The experimental evaluation of both versions is presented separately in Chapter 6.

1. Overview

The autonomous dissection pipeline converts stereo endoscopic images into robot commands that deliver energy along the tissue boundary separating the gallbladder from the liver. This section summarizes the two versions of the system architecture and highlights the key differences between them.

1.0.1. *Initial Framework (v1).* In our initial framework [**2**], the system operated with a single PSM arm equipped with a Permanent Cautery Hook (PCH). Figure 5.1 shows the v1 architecture: stereo endoscopic images were processed by Detectron2 to produce tissue segmentation masks and instrument keypoint detections, which were combined with 3D point clouds to extract a dissection trajectory. The trajectory was computed offline before execution, and the PCH followed this fixed set of waypoints using position-based visual servoing while delivering monopolar energy.

This approach had several fundamental limitations: the precomputed trajectory could not adapt to tissue deformation during energy delivery, there was no mechanism for grasping

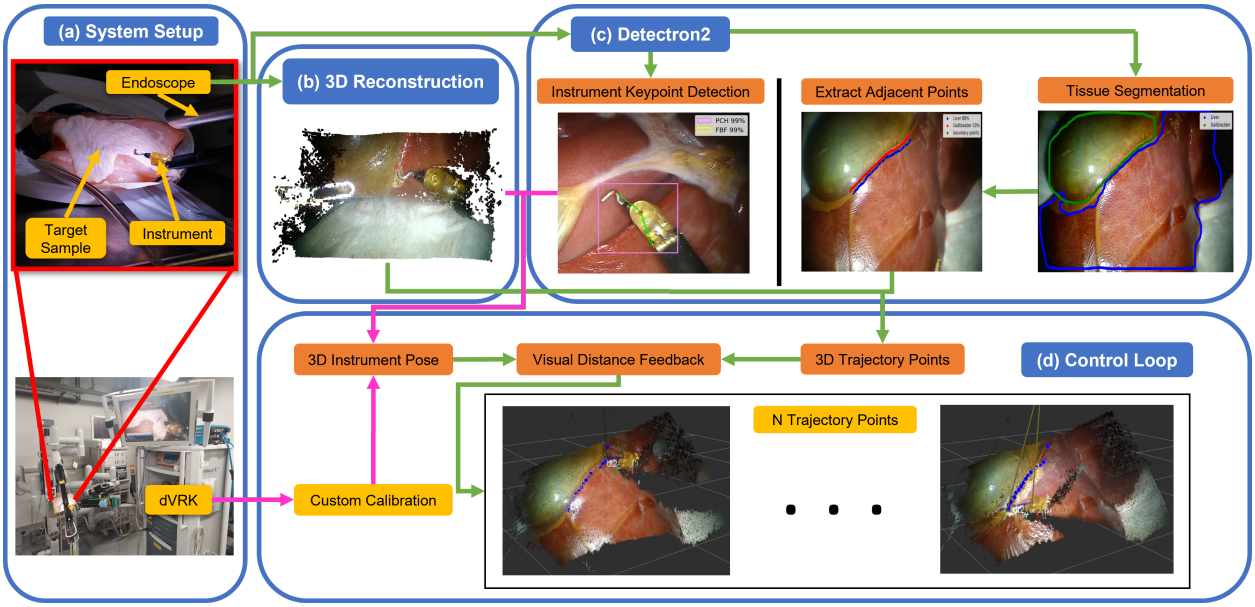


FIGURE 5.1. System architecture of the initial (v1) framework [**2**]: (a) Hardware setup. (b) 3D reconstruction from stereo endoscopic images. (c) Detectron2 outputs for tissue segmentation and instrument keypoint detection. (d) Extracted trajectory points and instrument pose in 3D space, along with inputs to the control system.

or stabilizing the gallbladder, and the single-arm configuration restricted the workspace to one fixed field of view.

1.0.2. *Upgraded Framework (v2).* To address these limitations, we developed an upgraded framework [**3**] that introduces bimanual manipulation, online boundary tracking, and PCA-based instrument alignment. The implementation platform was also upgraded from ROS to ROS2 [**68**]. Figure 5.2 illustrates the v2 architecture, which extends the v1 pipeline with four stages:

(1) **Perception.** Stereo endoscopic images are processed by fine-tuned segmentation and keypoint detection models (Chapter 4) to identify tissue regions and instrument poses. In parallel, the stereo images are used to generate real-time 3D point clouds via the Semi-Global Matching algorithm (Chapter 2).

(2) **Post-processing.** The segmentation masks and point clouds are combined to extract the 3D tissue boundary and compute the gallbladder skeleton, providing the

geometric features needed for instrument alignment and target point selection (Section 2).

(3) **Grasping.** The Fenestrated Bipolar Forceps (FBF) is aligned to the gallbladder surface, grasps the tissue, and pulls it to stretch and stabilize the dissection boundary (Section 3).

(4) **Dissection.** The PCH is aligned to the gallbladder surface and autonomously follows the tissue boundary while delivering monopolar energy to separate the gallbladder from the liver (Section 4).

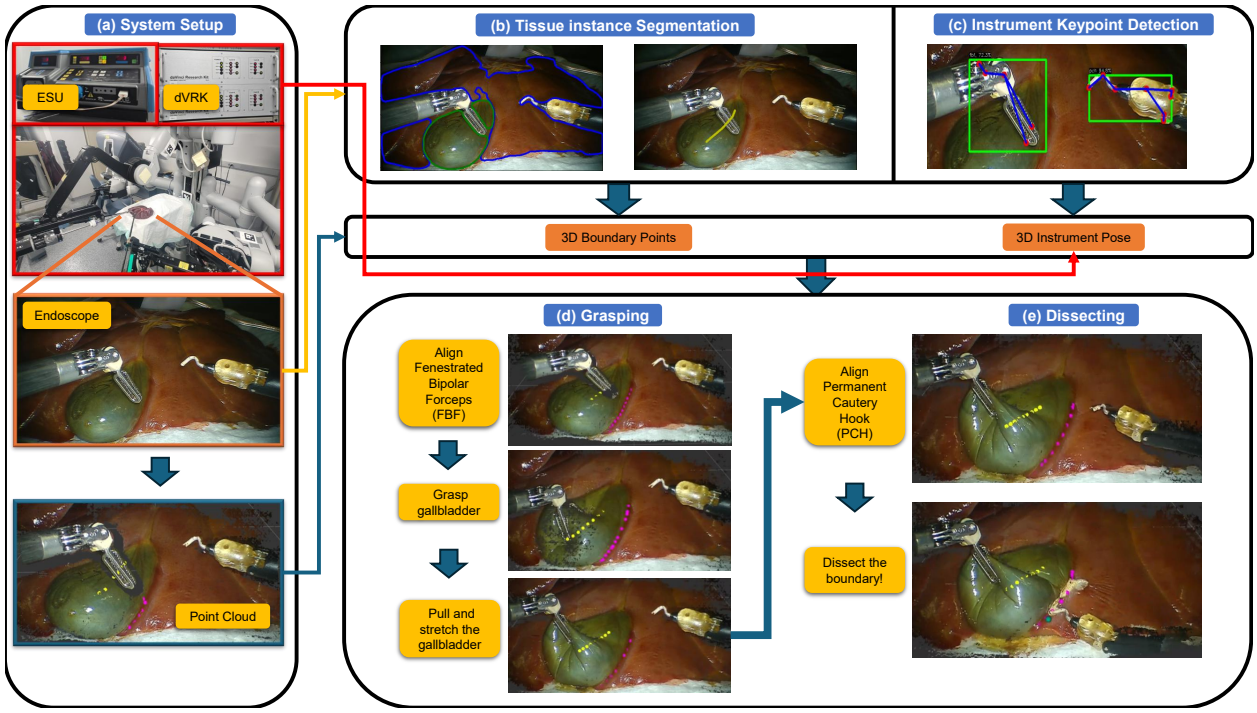


FIGURE 5.2. System architecture of the upgraded (v2) framework [**3**]: (a) Hardware setup including the dVRK and ESU; stereo endoscopic images are used to generate real-time 3D point clouds. (b)–(c) Outputs of the perception models: tissue instance segmentation and instrument keypoint detection, used to compute the 3D dissection boundary and instrument poses. (d) Grasping mechanism using the FBF (Section 3). (e) Dissection mechanism controlling the PCH (Section 4).

2. Image and Point Cloud Post-Processing

The post-processing stage bridges the gap between the raw perception outputs (segmentation masks, point clouds, keypoints) and the geometric features needed for instrument alignment and trajectory planning. This section describes the skeletonization and boundary extraction algorithms that produce these features.

2.1. Skeletonization. To identify the central region of the gallbladder in the current endoscopic view, we apply the classical skeletonization algorithm [**103**] to the binary segmentation mask of the gallbladder. This method iteratively sweeps a fixed-size window over the binary mask, removing boundary pixels at each iteration until the image converges to a one-pixel-wide skeleton.

Because the gallbladder has an approximately oval shape in the endoscopic view, the resulting skeleton effectively represents the medial axis of the visible gallbladder surface. When the 2D skeleton pixels are mapped to their corresponding 3D coordinates via the point cloud, the resulting 3D skeleton points approximate the medial surface of the gallbladder. These skeleton points (denoted as *yellow points* in subsequent figures) serve two purposes: they define the center of the gallbladder for grasping and they provide a reference line for computing the dissection boundary.

Figure 5.3 illustrates the full skeleton extraction pipeline: from the raw segmentation mask, corners are detected and branches are separated, and finally outlier branches are removed and the main branches are merged to produce a clean skeleton.

2.2. Boundary Extraction. The *boundary of interest* for dissection is the portion of the liver–gallbladder interface that is both visible through the endoscope and reachable by the PCH. Because the PCH is mounted on the right-side PSM in our setup, the reachable boundary typically lies on the right side of the skeleton when viewed from the endoscope. This boundary (denoted as *purple points* in the figures) defines where the PCH should deliver energy to separate the two tissues.

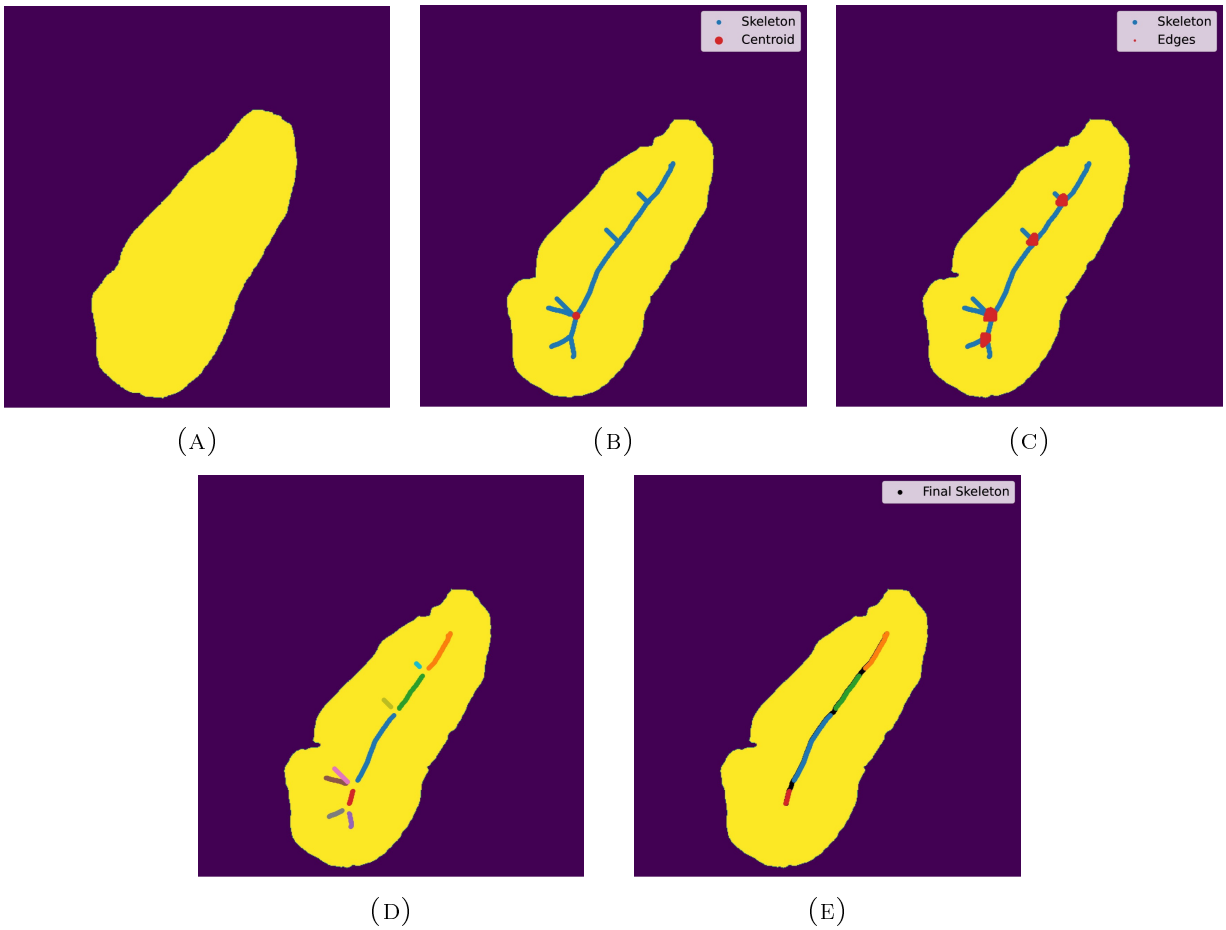


FIGURE 5.3. Skeleton extraction pipeline: (a) Raw gallbladder segmentation mask. (b) Extracted skeleton. (c) Corner detection. (d) Branch separation. (e) Outlier branch removal and merging of main branches to produce the final skeleton.

To determine which side of the skeleton corresponds to the boundary of interest, we apply Principal Component Analysis (PCA) [**104**] to the 3D points in the surface region between the skeleton and the adjacent boundary. The three principal axes of this point set define a local coordinate frame attached to the gallbladder surface:

- The **primary axis** lies along the longest extent of the surface, approximately parallel to the skeleton.
- The **secondary axis** extends from the skeleton toward the boundary points.
- The **normal axis** is perpendicular to the gallbladder surface, directed along the viewing direction of the endoscope.

When necessary, these axes are re-aligned to enforce the above conventions: the secondary axis is adjusted to point from the skeleton to the boundary, and the normal axis is oriented toward the endoscope. By examining the sign of the secondary axis relative to the skeleton center, we identify the "right side" of the skeleton, which corresponds to the dissection boundary. This re-alignment guarantees consistent orientation across different viewpoints and gallbladder positions.

Figure 5.4 visualizes the result of this process: the skeleton (yellow), the boundary of interest (purple), and the surface region between them. The PCA-derived local frame is reused for both instrument alignment (Section 3) and dissection target selection (Section 4.2).

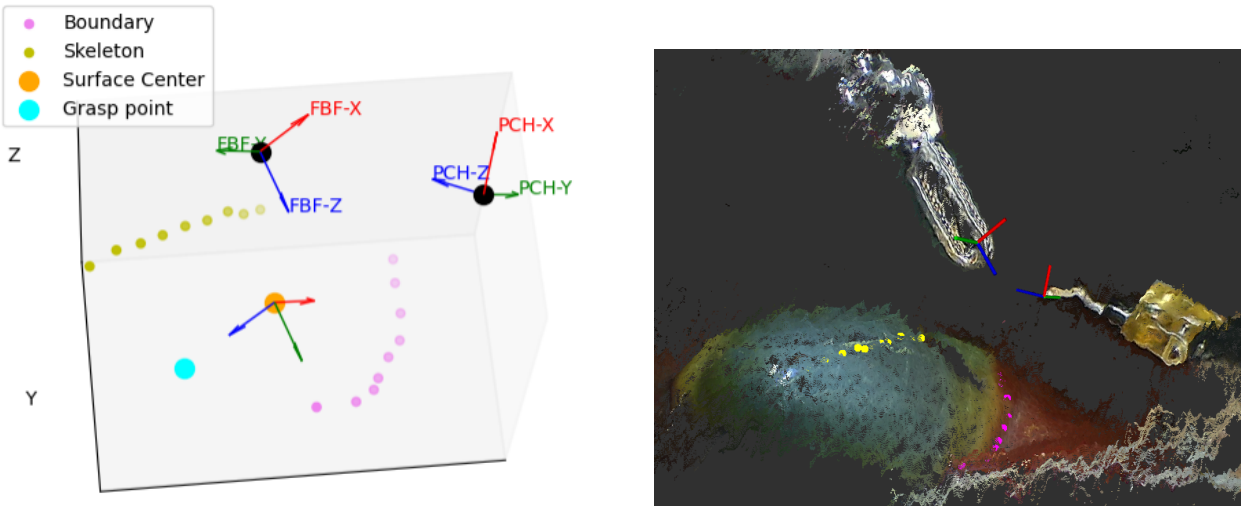


FIGURE 5.4. Instrument alignment and surface geometry after post-processing. Left: the coordinate frames of the FBF and PCH relative to the gallbladder surface. Right: 3D point cloud showing the skeleton (yellow), boundary of interest (purple), and the surface region between them. The three PCA-derived principal axes define a local frame used for instrument alignment.

3. Grasping

In the initial framework [**2**], only a single PSM arm was used, meaning the gallbladder was not stabilized during dissection. This led to tissue deformation during energy delivery, which in turn altered the boundary and degraded the fixed trajectory's accuracy. To address this, we introduced a bimanual grasping strategy [**3**] that uses the FBF to grasp and stretch

the gallbladder before the PCH begins dissection. The grasping procedure consists of three sequential steps: alignment, grasping, and pulling.

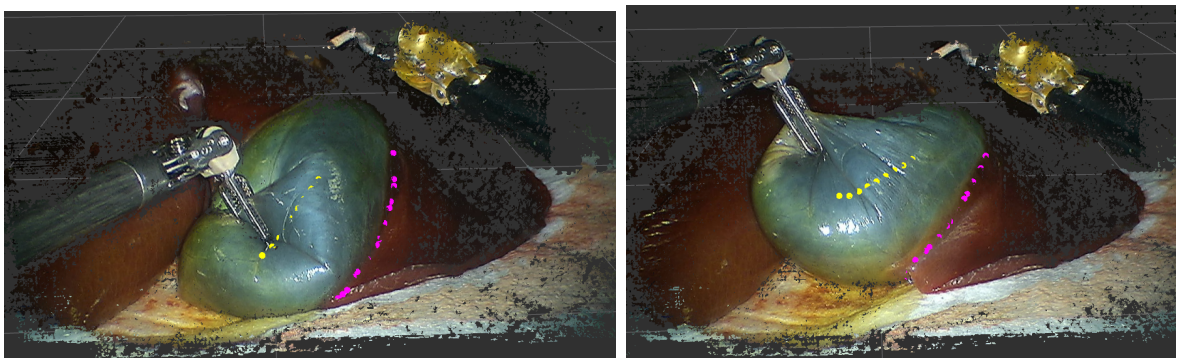
3.1. Alignment. The first step is to align the FBF so that its gripper is parallel to the gallbladder surface in the region between the boundary and the skeleton. The PCA-derived local frame computed during boundary extraction (Section 2.2) provides the alignment pose: the FBF frame is oriented so that its gripper plane matches the primary and secondary axes of the surface, while the approach direction aligns with the normal axis. The resulting configuration is illustrated in Fig. 5.4.

3.2. Grasping. Once aligned, the FBF moves along its negative x-axis—directed through the gallbladder surface—to reach the grasping point. The grasping point is defined along the line segment between the skeleton center and the boundary region and is then shifted inward by approximately 5 mm along the grasping direction so the jaws close on gallbladder tissue rather than directly on the extracted boundary. This placement balances two objectives: grasping deep enough to secure the gallbladder while remaining close enough to the boundary to effectively tension the dissection region when pulled.

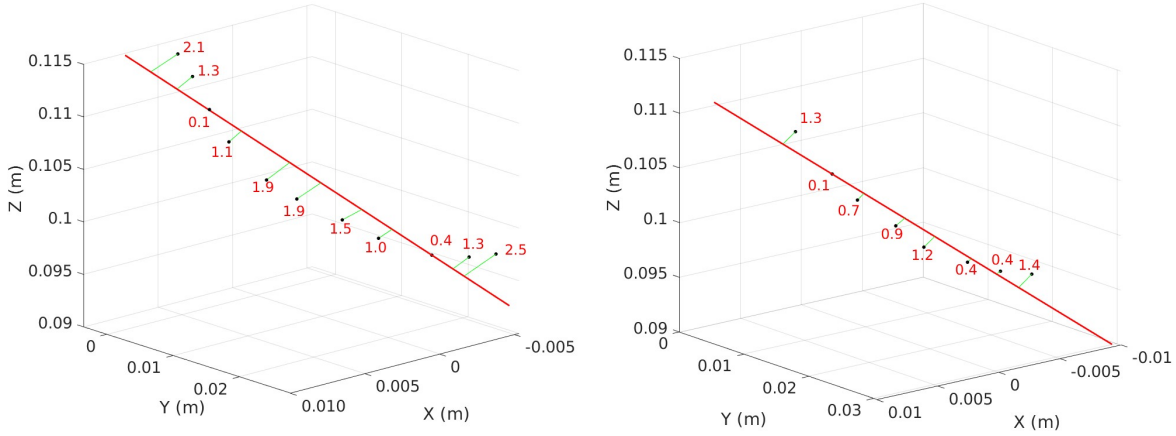
3.3. Pulling. After grasping, the FBF pulls the gallbladder along its negative z-axis, which is tangential to the right-side surface of the gallbladder. The pulling continues until the boundary becomes approximately linear—quantified by a deviation threshold of 0.5 mm between the boundary points and an ideal straight line.

Figure 5.5 illustrates the pulling process. Before pulling (Fig. 5.5a, left), the gallbladder boundary is curved and the tissue is loose. After pulling (Fig. 5.5a, right), the boundary is straightened and the tissue is taut. Fig. 5.5b shows the deviation of boundary points from the ideal boundary (red line) before and after pulling: the deviation decreases substantially, confirming that the tissue is adequately stretched. Figure 5.5c plots the deviation over the pulling trajectory, showing convergence below the 0.5 mm threshold.

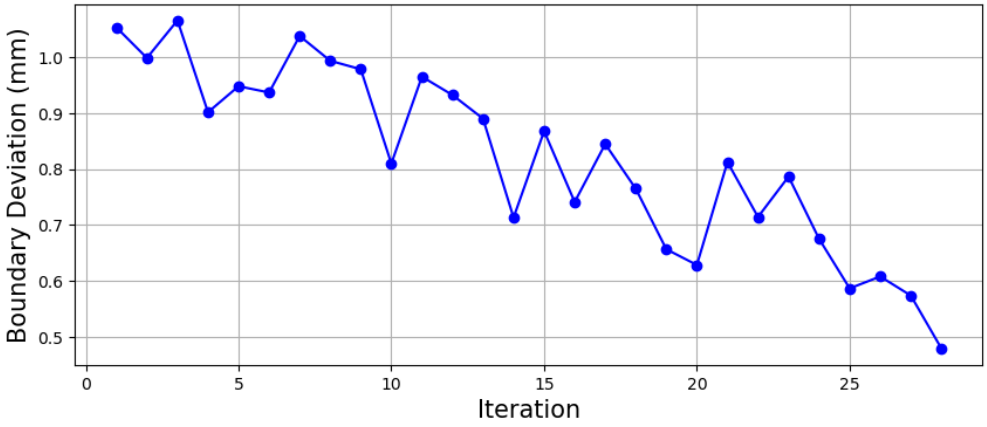
This grasping mechanism stabilizes the gallbladder throughout the dissection task, reducing the impact of anatomical variability and sudden tissue deformation. Once the boundary



(A) Point cloud before (left) and after (right) pulling the gallbladder.



(B) Deviation of boundary points from the ideal boundary (red line) before the pull (left) and after the pull (right).



(C) Boundary deviation over the pulling trajectory. The deviation of the boundary points (purple) converges below the 0.5 mm threshold, indicating the boundary is sufficiently linear.

FIGURE 5.5. Gallbladder pulling process. (a) 3D point cloud before and after pulling shows the boundary becoming linear. (b) Spatial deviation of boundary points from the ideal straight line decreases after pulling. (c) Deviation magnitude over the pulling trajectory converges below the stopping threshold.

is sufficiently stretched, the FBF remains static and the PCH begins the dissection procedure described in Section 4.

4. Dissection

This section presents both versions of the dissection control algorithm: the original offline approach from our initial framework [**2**] and the online boundary-guided approach from the upgraded framework [**3**].

4.1. Offline Trajectory Planning. In our initial framework [**2**], the dissection trajectory was computed offline before the PCH began moving. The pipeline consisted of four stages: boundary detection, trajectory point extraction, trajectory downsampling, and position-based visual servoing.

4.1.1. *Boundary Detection.* The DT2-seg model (Chapter 4) processed the stereo endoscopic images to produce segmentation masks for each tissue class. From these masks, adjacent boundary pixels between the foreground tissue (e.g., gallbladder) and the background tissue (e.g., liver) were identified using the K-Dimensional Tree (KD-Tree) algorithm [**105**]. The KD-Tree efficiently finds nearest-neighbor pairs between the two pixel sets, yielding the set of boundary pixel pairs (visualized as blue and red lines in Fig. 5.1).

4.1.2. *Trajectory Point Extraction.* For each pair of adjacent boundary pixels, we queried the corresponding 3D point cloud to extract the point with the peak disparity value between the two pixels. We use this peak-disparity point as a heuristic estimate of the best-localized 3D position on the tissue boundary. The collection of these peak-disparity points forms the raw boundary trajectory in 3D space.

4.1.3. *Trajectory Downsampling.* The raw boundary trajectory typically contains hundreds of points, many of which are redundant for robot motion. We applied the farthest-first traversal algorithm [**106**] to downsample the trajectory to a manageable number of waypoints while preserving the overall shape of the boundary. Starting from an initial point, farthest-first traversal iteratively selects the point that is farthest from the current set of selected points, producing a well-distributed subset (green points in Fig. 5.1).

4.1.4. *Position-Based Visual Servoing.* The DT2-kpt model detected instrument keypoints in each endoscopic frame, providing the 2D pixel coordinates of the PCH tip. These pixel coordinates were mapped to 3D positions via the point cloud. To improve robustness, a fixed-size 10×10 pixel window was applied around each keypoint, and the 3D coordinates within this window were averaged to filter outliers.

The robot was programmed to move 0.5 mm toward the current trajectory point in each control iteration. The dVRK's built-in position controller was used to execute these incremental motions. At each iteration, the system compared the 3D distance between the PCH tip (from keypoint detection) and the current trajectory point. Once this distance fell below a 0.5 mm threshold, the system advanced to the next trajectory point. This control mechanism is termed Position-Based Visual Servoing (PBVS), as the feedback relies on the 3D positional information derived from the stereo images.

4.1.5. *Limitations.* While this approach achieved submillimeter trajectory-following precision (Chapter 6), it had several fundamental limitations:

(1) **Fixed trajectory.** The entire trajectory was computed once before execution. Any tissue deformation during energy delivery—which is inevitable in practice—caused the precomputed trajectory to diverge from the actual boundary.
(2) **No boundary updates.** The system had no mechanism to re-detect or update the boundary during the procedure, preventing adaptation to the evolving tissue geometry.
(3) **Single arm.** Only one PSM arm (PCH) was used, meaning the gallbladder was not stabilized. Tissue movement during dissection further degraded trajectory accuracy.
(4) **Limited workspace.** Without endoscope motion, the workable volume was restricted to the current field of view, allowing only partial dissection of the tissue boundary.

4.2. Online Boundary-Guided Dissection. To address the limitations of the offline approach, we developed an online dissection algorithm [**3**] that continuously updates the tissue boundary and dynamically selects target points during the procedure. Combined with

the grasping strategy from Section 3, this approach keeps the boundary estimate usable after repeated rounds of energy delivery and provides the core components needed for future multi-round autonomy. The quantitative evaluation in Chapter 6 still measures a single grasp–pull–dissect cycle per trial.

4.2.1. *PCH Alignment.* Before dissection begins, the PCH is aligned to the gallbladder surface using the same PCA-based method described for FBF alignment in Section 3.1. The only difference is that the three principal axes are mapped to the PCH's kinematic frame rather than the FBF's frame (see both instrument frames in Fig. 5.4). This alignment positions the PCH hook orthogonal to the gallbladder surface, minimizing contact between the hook's edge and the liver tissue. This prevents the instrument from catching on tissue during energy delivery—a problem frequently observed in v1 trials.

4.2.2. *Dynamic Target Point Selection.* Unlike the v1 approach, which followed a fixed trajectory of precomputed waypoints, the v2 algorithm selects target points dynamically from the current tissue boundary, which is re-extracted from the segmentation model output at each control iteration. The target selection proceeds as follows:

(1) **Initial target.** The first target point is the first point in the boundary sequence, ordered clockwise from the center of the skeleton. This ensures the PCH begins dissecting at a consistent starting location.
(2) **Subsequent targets.** As the PCH reaches each target, a new target is selected from the updated boundary. The algorithm chooses the point farthest along the current direction of motion, subject to a maximum step of 1 cm. This cap prevents the PCH from attempting to jump across large boundary gaps.
(3) **Alignment maintenance.** At each target update, the PCH maintains its alignment with the gallbladder surface using the PCA-derived frame, ensuring the hook remains properly oriented throughout the procedure.

In the released ROS2/dVRK implementation used for these experiments, this update loop runs with an expected interval of 0.01 s (100 Hz), and each boundary-following update advances the PCH by 0.5 mm toward the current target before the boundary is re-evaluated.

4.2.3. *Termination Criterion.* A dissection round terminates when no boundary point sufficiently distant (greater than 1 mm) from the current PCH position can be found. This indicates that the PCH has traversed the entire accessible boundary segment and further dissection would require repositioning the FBF for another grasping–pulling cycle.

4.2.4. *Contrast with v1.* The key differences between the offline and online approaches are summarized as follows:

- **Trajectory computation:** v1 computes the full trajectory once before execution; v2 selects each target point dynamically from the current boundary.
- **Boundary tracking:** v1 uses a static boundary snapshot; v2 re-extracts the boundary at each control iteration, adapting to tissue deformation from energy delivery.
- **Instrument alignment:** v1 does not align the PCH to the tissue surface; v2 aligns both FBF and PCH using PCA-derived surface frames.
- **Tissue stabilization:** v1 uses a single arm without grasping; v2 uses bimanual manipulation with the FBF holding and stretching the tissue.
- **Repeated-round readiness:** v1 can only execute a single trajectory; v2 maintains a usable boundary estimate after repeated energy delivery within a trial and provides the components needed for future multi-round grasp–pull–dissect autonomy.

These improvements enable the system to handle the dynamic tissue changes inherent in surgical dissection. The quantitative results for the main system configurations are presented in Chapter 6 as system-level comparisons.

CHAPTER 6

Experimental Results

This chapter presents the experimental evaluation of the autonomous dissection framework across both versions of the system [**2**, **3**]. The quantitative and qualitative evaluation of the perception models themselves was presented in Chapter 4; this chapter focuses on the downstream motion performance. Section 1 evaluates the grasping strategy introduced in Chapter 5. Sections 2.1 and 2.2 present the dissection trial results for the initial and upgraded systems, respectively. Finally, Section 3 synthesizes the key findings and identifies remaining limitations.

1. Grasping Performance

The grasping strategy described in Section 3 was evaluated across all trials in the upgraded system. Figures 5.4 and 5.5 illustrate the stages of the grasping process: alignment, grasping, and pulling. The boundary deviation plot (Figure 5.5c) confirms that the pulling procedure consistently reduced the boundary deviation below the 0.5 mm threshold, producing a nearly linear boundary suitable for dissection.

The success of the grasping phase did not depend heavily on the choice of perception model: all three segmentation models (DT2-seg, MaskDINO, YOLO11l-seg) produced sufficiently accurate gallbladder masks for the PCA-based alignment to succeed. The grasping outcome is primarily determined by the 3D geometry of the gallbladder surface rather than fine-grained segmentation accuracy. In all trials reported in Section 2.2, the FBF successfully grasped and stretched the gallbladder before dissection began.

To ensure consistency across trials, the FBF was not repositioned after the initial grasping. This means each trial evaluated a single grasp–pull–dissect cycle. The boundary stability results in Table 6.2 show that the grasping strategy maintained a stable boundary throughout the dissection procedure, which was a key limitation in the initial system [**2**].

2. Dissection Performance

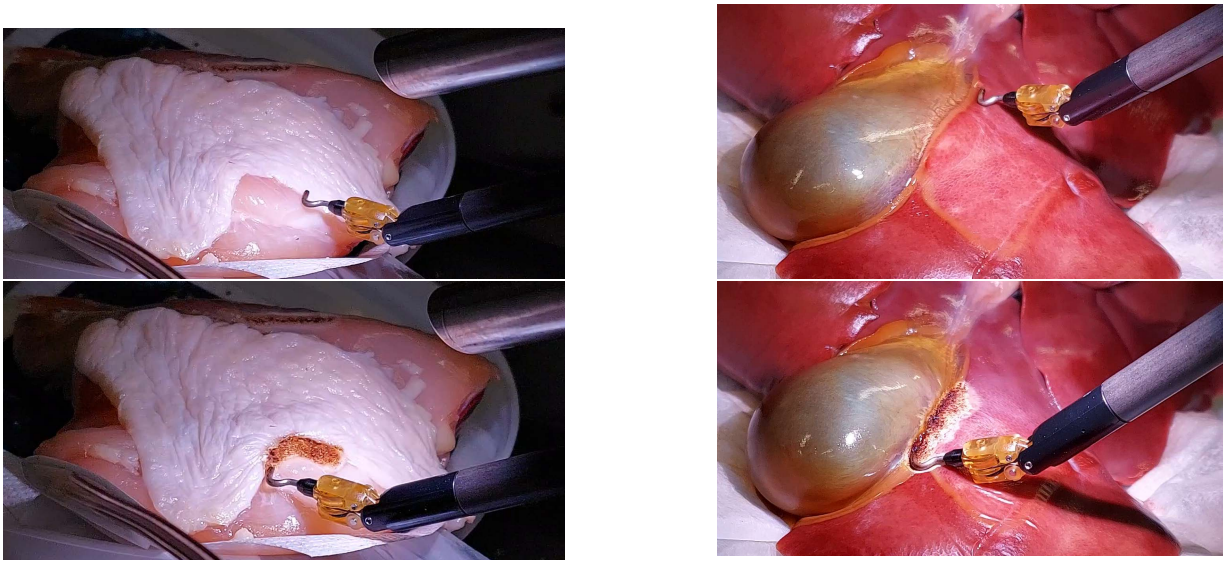
2.1. Initial System Results. The initial system [**2**] used a single PSM arm (PCH) with offline trajectory planning and DT2-seg/DT2-kpt perception models trained on the v1 dataset (evaluated in Section 3.1). We evaluated this system on two specimen types: chicken samples, where the clear boundary between skin and muscle served as a proxy for the gallbladder-liver interface, and porcine liver specimens with attached gallbladders.

2.1.1. *Ex Vivo Dissection Trials.* Table 6.1 presents the results of all 11 dissection trials: 6 on porcine liver specimens and 5 on chicken specimens. For each trial, we measured the distance between each recorded position of the instrument tip and the expected ideal trajectory—defined as a linear path between consecutive trajectory points. The system achieved a weighted mean distance of 0.36 mm with a standard deviation of 0.34 mm across all trials, demonstrating submillimeter trajectory-following precision.

Figure 6.1 shows representative endoscopic frames from the chicken and porcine liver trials, captured when the instrument reached the first and final trajectory points while delivering monopolar energy. Figure 6.2 presents the full trajectory pipeline for four representative trials: the left column shows the final segmentation output with the desired trajectory overlaid on the 2D endoscopic image; the middle column shows the corresponding 3D points extracted from the point cloud; and the right column traces the actual 3D movement trajectory of the instrument tip during the procedure.

2.1.2. *Limitations Observed.* Despite the submillimeter precision, the v1 system exhibited several limitations during the trials:

(1) **Tissue color variation.** Segmentation accuracy degraded when encountering gallbladders whose color (yellow or white) differed from the typical dark green present



(A) (B)

FIGURE 6.1. Endoscopic frames from v1 dissection trials. Top images show the instrument at the first trajectory point; bottom images show the instrument at the final trajectory point. (a) Energy delivery on chicken specimen. (b) Energy delivery on porcine liver.

TABLE 6.1. Mean and standard deviation of distances between the recorded PCH tip position and the optimal trajectory path (i.e., linear movement between consecutive trajectory points) for the v1 system.

Trial		Mean (mm)	Std. dev. (mm)	Duration (s)
Liver	1	0.31	0.23	128
	2	0.37	0.27	138
	3	0.29	0.24	102
	4	0.35	0.28	113
	5	0.30	0.20	117
	6	0.33	0.19	135
Chicken	1	0.28	0.18	75
	2	0.59	0.72	108
	3	0.35	0.25	98
	4	0.41	0.35	153
	5	0.27	0.17	130
Weighted Mean		0.36	0.34	121.6

in the training data. This stemmed from the limited diversity of the v1 dataset, which was sampled from a single video for each tissue type.

(2) **Peripheral keypoint detection.** Keypoint detection failed when the instrument was near the edge of the endoscopic image, as evident in the trajectory deviations when transitioning from point 5 to point 6 in Figure 6.2c and from point 3 to point 5 in Figure 6.2l. This is caused by the characteristic illumination falloff of endoscopic cameras and underrepresentation of peripheral instrument poses in the training data.

(3) **Single arm.** Without a second arm to grasp and stabilize the gallbladder, tissue deformation during energy delivery caused the precomputed trajectory to diverge from the actual boundary.

(4) **No endoscope motion.** The fixed endoscope limited the workable volume, allowing only partial dissection of the tissue boundary rather than the full anatomical structure.

2.2. Upgraded System Results. The upgraded system [**3**] introduced bimanual manipulation, online boundary tracking, PCA-based instrument alignment, and state-of-the-art perception models (Chapter 5). The evaluation provides a structured comparison of the main system configurations. Because the pipeline, training data, and model families changed together between some configurations, these results should be interpreted as system-level comparisons rather than isolated single-factor ablations.

2.2.1. *Experimental Design.* We evaluated four model configurations on porcine liver specimens with attached gallbladders:

(1) **DT2-seg + DT2-kpt (Old):** The original v1 system with offline trajectory planning, no grasping, and models trained on the v1 dataset. This serves as the baseline.

(2) **DT2-seg + DT2-kpt (New):** The upgraded pipeline with online boundary tracking, grasping, and alignment, but using Detectron2 models retrained on the v2 dataset.

(3) **MaskDINO + YOLO11l-pose:** MaskDINO for segmentation and YOLO11l-pose for keypoint detection, with the full upgraded pipeline.

(4) **YOLO11l-seg + YOLO11l-pose:** YOLO11 for both segmentation and keypoint detection, with the full upgraded pipeline.

For configurations 2–4, the FBF first grasped and stretched the gallbladder before the PCH performed online dissection (Chapter 5). For configuration 1, we report previously published results [**2**] obtained with the v1 offline system. Comparing configuration 1 to 2 highlights the effect of moving from the v1 offline single-arm system to the upgraded online bimanual pipeline [**3**], while also reflecting the retraining of Detectron2 on the expanded v2 dataset. Comparing configurations 2, 3, and 4 then shows the effect of the perception model choice within the upgraded pipeline.

2.2.2. *Boundary Stability Analysis.* Figure 6.3 presents the recorded boundary points from one trial each for YOLO11l-seg and DT2-seg. The YOLO11 boundary remained tightly clustered and stable throughout the trial, while the DT2 boundary exhibited greater scatter. The blue curves show cubic splines fitted to the boundary points; the root mean squared error (RMSE) between the recorded points and the fitted curve quantifies boundary consistency.

2.2.3. *Complete Performance Results.* Table 6.2 presents the complete trial-by-trial results for all four model configurations. For each trial, we report the boundary RMSE (where applicable), total PCH travel distance, and procedure duration for one full boundary dissection. Since the boundary points were fixed in the v1 system (Old), RMSE is not applicable for configuration 1.

2.2.4. *Analysis.* The results reveal several clear trends:

(1) **Boundary consistency.** YOLO11l-seg + YOLO11l-pose achieved the lowest mean RMSE (0.49 mm) with the smallest standard deviation (0.02 mm), indicating highly consistent boundary tracking across all five trials. DT2-seg + DT2-kpt (New) exhibited occasional RMSE spikes (e.g., Trial 2: 0.78 mm) even after outlier removal, reflecting the segmentation inconsistencies noted in Section 3.2.1. MaskDINO + YOLO11l-pose showed moderate RMSE (0.53 mm mean) with low variance (0.05 mm), suggesting that while MaskDINO's segmentation was somewhat less accurate than

TABLE 6.2. Performance metrics across all model configurations and trials in the upgraded system. RMSE = root mean squared error of boundary points relative to a fitted cubic spline (Fig. 6.3). Distance = total PCH travel distance during one full boundary dissection. Duration = total time for one boundary dissection. RMSE is not applicable for the Old configuration because boundary points were fixed prior to dissection. Bold values indicate the best (lowest) mean and standard deviation.

Model (Seg. + Kpt.)	**Trial**	**RMSE (mm)**	**Distance (mm)**	**Duration (s)**
DT2-seg + DT2-kpt (Old)	1	–	60.3	104
	2	–	51.3	117
	3	–	28.7	79
	4	–	36.3	92
	5	–	38.3	97
	6	–	45.5	116
Mean		–	43.4	100.8
Std. Dev.		–	11.3	14.6
DT2-seg + DT2-kpt (New)	1	0.42	50.5	29
	2	0.78	51.1	28
	3	0.50	65.2	35
	4	0.43	54.3	30
	5	0.46	82.9	40
Mean		0.51	60.8	32.4
Std. Dev.		0.13	13.7	5.3
MaskDINO + YOLO11l-pose	1	0.51	56.1	60
	2	0.63	46.3	30
	3	0.53	65.5	59
	4	0.49	69.3	64
	5	0.49	62.3	65
Mean		0.53	59.9	55.4
Std. Dev.		0.05	9.0	14.5
YOLO11l-seg + YOLO11l-pose	1	0.47	54.9	31
	2	0.52	47.2	29
	3	0.47	56.0	32
	4	0.50	54.9	31
	5	0.51	48.0	29
Mean		**0.49**	52.2	**30.6**
Std. Dev.		**0.02**	**4.2**	**1.3**

YOLO11's, the YOLO11l-pose keypoint model compensated for much of the variability.

(2) **Travel distance.** YOLO11 demonstrated the most stable travel distance across trials (52.2 ± 4.2 mm), while DT2 (New) showed high variance (60.8 ± 13.7 mm), with Trial 5 reaching 82.9 mm. The elevated travel distances for the DT2 and MaskDINO configurations are consistent with the lower keypoint accuracy of DT2-kpt (Figure 4.6): inaccurate tip localization, together with noise in the point clouds surrounding the instrument, likely produced incorrect 3D tip positions and the oscillatory arm motions observed during those trials.

(3) **Procedure duration.** The dissection speed improved dramatically from the v1 system. The Old DT2 configuration averaged 100.8 ± 14.6 s per boundary dissection, while the YOLO11 configuration averaged only 30.6 ± 1.3 s—a 3.3× speedup with an order-of-magnitude reduction in variance. This difference likely reflects several factors acting together. The DT2 (New) configuration already averaged 32.4 s, suggesting that the updated pipeline and the models retrained on the v2 dataset account for much of the improvement over Old. Within that upgraded pipeline, the PCA-based alignment of the PCH orthogonal to the gallbladder surface (Section 3.1) also appears to reduce unintended contact between the hook and the liver, preventing the instrument from getting stuck at specific locations—a frequent occurrence in v1 trials.

(4) **MaskDINO anomaly.** The MaskDINO + YOLO11l-pose configuration exhibited anomalously long durations (55.4 ± 14.5 s) despite using the same keypoint model as YOLO11. This behavior is consistent with MaskDINO's difficulty distinguishing liver from liver bed (AP: 80.0 vs. 98.6 for YOLO11), which appears to have caused boundary extraction errors that forced the PCH into suboptimal paths and additional corrective movements.

3. Discussion

The experimental results show clear improvements from the initial (v1) to the upgraded (v2) autonomous dissection system [**2**, **3**] in boundary consistency, procedure speed, and robustness to tissue deformation.

Several changes contributed to these improvements. The transition from offline to online boundary tracking likely played a major role: the v1 system's precomputed trajectory assumed a static tissue geometry that could not account for deformation during energy delivery, whereas the v2 system re-extracted the boundary in real time and allowed the PCH to adapt its path as the tissue changed.

The introduction of bimanual grasping also appears to have improved boundary stability. The low RMSE values across all v2 configurations, together with the boundary deviation analysis in Figure 5.5c, suggest that the grasping and pulling strategy reduced the tissue shifts that degraded v1 performance. At the same time, because configurations 1 and 2 differ in both pipeline design and training data, these comparisons should be interpreted as bundled system improvements rather than isolated causal ablations.

Within the upgraded pipeline, the choice of perception model still had a measurable impact on procedure consistency. YOLO11's ability to distinguish liver from liver bed improved performance in the later stages of dissection, while its superior edge-of-frame keypoint detection reduced the oscillatory movements that plagued DT2-based configurations.

3.0.1. *Remaining Limitations.* Despite these advances, several limitations remain:

(1) **Dissection technique.** Through our work with the CRCD (Chapter 3), we observed that surgeons employ different dissection techniques. The current framework follows the boundary after stretching it, but another common technique involves using the PCH tip to hook the boundary and pulling until the tissues detach, applying energy only if the tissue remains too thick. This hooking technique minimizes thermal damage to the liver, reducing the risk of postoperative complications [**107**].

(2) **Keypoint detection in extreme poses.** Although keypoint detection improved substantially with YOLO11l-pose, it still suffers in certain instrument poses or occlusion scenarios during the procedure. Further augmenting the keypoint dataset—similar to how the segmentation dataset was expanded using the CRCD—would address this gap.

(3) **Endoscope tracking.** The current framework uses a fixed endoscope position, limiting the workable volume to the current field of view. An active endoscope that follows the instrument tip using image-based visual servoing would ensure the instrument remains centered in the image, mitigating both occlusion and boundary visibility issues.

(4) **Full gallbladder separation.** The current system performs a single grasp–pull–dissect cycle per trial. Achieving full gallbladder separation requires multiple autonomous rounds of pulling and dissecting, with the FBF repositioning between rounds to expose new boundary segments.

These limitations, along with the need for larger and more diverse training data, were a major motivation for building the Comprehensive Robotic Cholecystectomy Dataset (Chapter 3).

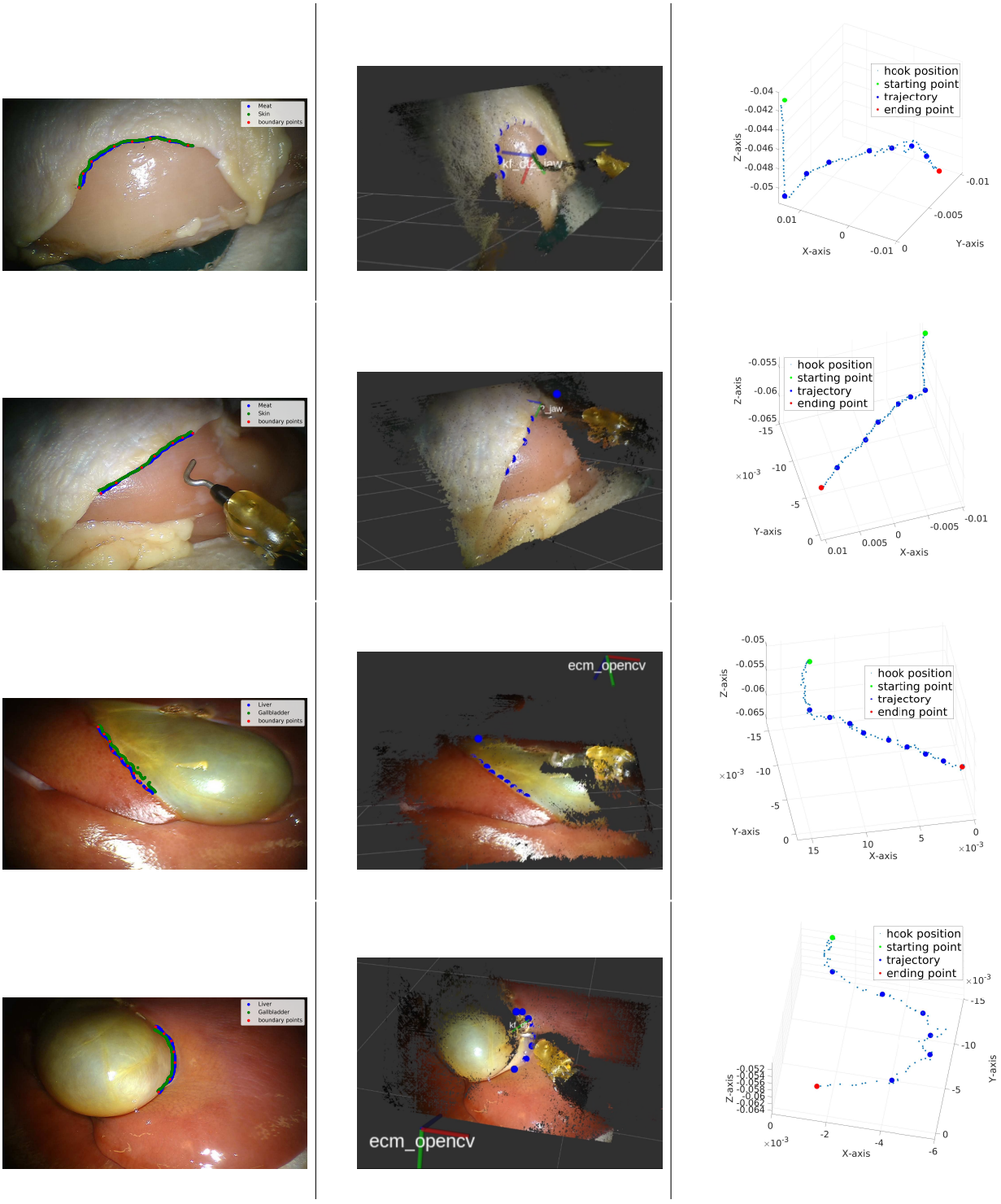
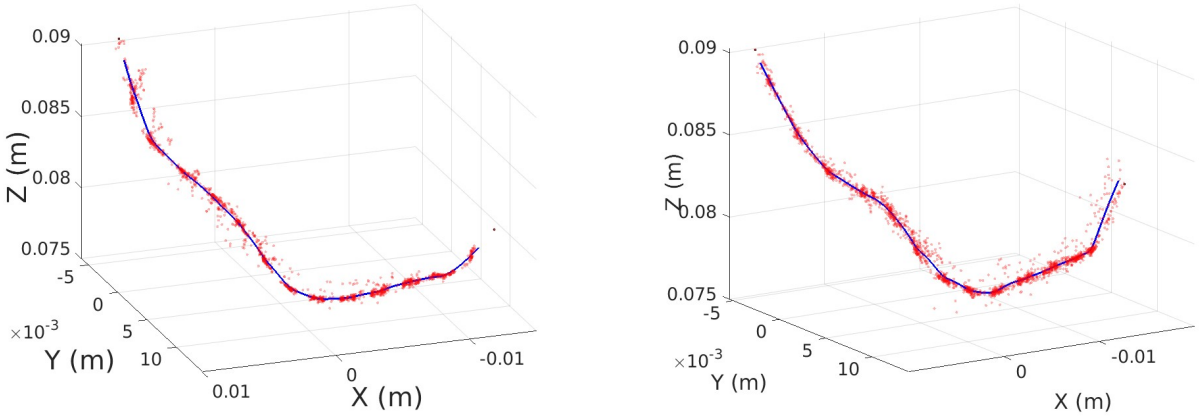


FIGURE 6.2. Trajectory visualization for four representative v1 trials (two chicken, two porcine liver). Left column: final segmentation output with the desired trajectory overlaid on the 2D endoscopic image. Middle column: corresponding 3D trajectory points extracted from the point cloud. Right column: actual 3D movement trajectory of the instrument tip during the procedure. Rows 1–2: chicken trials. Rows 3–4: porcine liver trials.



(A) YOLO11l-seg (Trial 1)

(B) DT2-seg (Trial 2)

FIGURE 6.3. Boundary point samples recorded during a single dissection trial for (a) YOLO11l-seg and (b) DT2-seg. Blue curves show cubic spline fits. The YOLO11 boundary remained stable throughout the procedure, while the DT2 boundary exhibited greater scatter due to segmentation inconsistency.

CHAPTER 7

Kinematics Prediction from Endoscopic Images

The preceding chapters introduced the Comprehensive Robotic Cholecystectomy Dataset (Chapter 3) and the autonomous dissection framework that it enables (Chapters 4–6). A recurring theme throughout these contributions is the reliance on the da Vinci Research Kit's (dVRK) internal joint encoders for instrument localization—either directly through forward kinematics or indirectly through the calibration procedures described in Chapter 2. This chapter takes a fundamentally different approach: rather than trusting the robot's proprioceptive signals, we ask whether the 3D position of each instrument's end-effector can be predicted directly from the stereo endoscopic images that are already available during every procedure.

We present a two-stage pipeline in which per-instrument Vision Transformer (ViT) backbones are first pre-trained on an instrument segmentation task—using masks generated automatically by an instrument-specific YOLO11l-seg model trained for the annotation pipeline described in Section 3.1.1—and then frozen and paired with lightweight stereo pose estimation heads that regress end-effector translation. Evaluated on held-out surgical videos from the CRCD, the models achieve mean Euclidean translation errors of 0.94 cm for PCH and 1.13 cm for FBF, well within the kinematic uncertainty of cable-driven surgical robots. Ablation studies reveal that segmentation pre-training is the dominant contributor to accuracy, suggesting that backbone quality matters more than head architecture. Because the pipeline requires only stereo video at inference time, it can be applied to the large volumes of existing surgical footage that lack kinematic labels, offering a scalable path toward automatic pose annotation.

1. Motivation

Accurate real-time estimation of surgical instrument pose from endoscopic video is essential for automating robot-assisted surgery (RAS). The autonomous dissection pipeline described in Chapters 5 and 6 relies on knowing the 3D position of both the Permanent Cautery Hook (PCH) and the Fenestrated Bipolar Forceps (FBF) in order to plan dissection trajectories, execute grasping maneuvers, and evaluate boundary stability. In that pipeline, instrument localization is achieved through a combination of keypoint detection (Chapter 4), stereo reconstruction (Chapter 2), and the dVRK's forward kinematics. While this approach yields submillimeter dissection precision under controlled conditions, it depends critically on the accuracy of the robot's kinematic chain—a dependency that becomes problematic in practice.

1.1. Limitations of Robot Kinematics. Surgical robotic systems such as the da Vinci record kinematics internally, but these signals are seldom released alongside video data, and when available, they may suffer from calibration drift or cable-driven kinematic errors of up to 5 cm [**28**, **29**]. The dVRK's cable-driven transmission introduces backlash and hysteresis that accumulate over time, and the Setup Joint (SUJ) encoders are particularly unreliable: their potentiometer-based readings drift between procedures and even within a single operation if the patient cart is inadvertently bumped. The fiducial-marker-based calibration described in Chapter 2 addresses this by establishing a camera-to-robot transformation, but it requires a visible ArUco marker and periodic recalibration—steps that are impractical in a fully autonomous setting.

These kinematic inaccuracies have direct consequences for the dissection pipeline. In Chapter 6, we observed that the dissection performance was sensitive to the accuracy of instrument tip localization: errors in 3D tip position caused oscillatory movements and increased travel distances. The keypoint-based approach mitigates this by using visual features rather than raw kinematics, but it still relies on the dVRK's state for the final camera-to-world transformation. A purely vision-based approach to instrument pose estimation would eliminate this dependency entirely.

1.2. Vision-Based Kinematics Estimation. A key observation is that experienced surgeons routinely perceive instrument tip position and orientation from the stereo endoscope view alone, adjusting their controls without consulting kinematic readouts [**108**]. If a human can extract pose from video, a learned model should be able to do the same. Furthermore, while large volumes of surgical video exist across institutions—hours of stereo endoscopic footage captured during routine operations—almost none include ground-truth instrument pose labels and are thus of limited use for advancing RAS technology. A model trained on the few datasets that do pair video with kinematics (such as the CRCD) could subsequently be applied to label—and thereby unlock—the far larger body of surgical footage that exists without kinematic ground truth.

1.3. Connection to the CRCD. The CRCD (Chapter 3) is well suited for this task. Chapter 3 provides the full dataset description; here we rely on its synchronized stereo video and dVRK kinematic recordings for both Patient Side Manipulators (PSMs). The dataset's scale—while modest compared to general-purpose manipulation corpora such as Open X-Embodiment [**56**] (over one million trajectories) or DROID [**57**] (76,000 demonstrations totaling 350 hours)—is sizable for surgical robotics and sufficient to train and evaluate a baseline vision-to-pose model. Community initiatives such as Open-H-Embodiment [**58**] aim to close the data gap by pooling standardized data from multiple surgical platforms, mirroring the Open X-Embodiment paradigm for healthcare robotics.

1.4. Scope of This Work. Pursuing kinematics estimation in full generality—predicting the complete $SE(3)$ pose (translation and rotation) plus jaw angle—introduces complications. Representing rotations as quaternions, Euler angles, or axis-angle vectors creates discontinuities or ambiguities that destabilize gradient-based training [**109**]. Even continuous representations such as the 6-D parameterization of Zhou et al. [**109**] require geodesic losses on $SO(3)$ that interact poorly with translation objectives when combined in a single loss function, and naively weighting rotation against translation often degrades both [**110**]. For these reasons, this work focuses on the *translation* component of instrument pose, which

admits a straightforward Euclidean regression objective and allows us to study visual-to-pose learning without the confounding effects of rotation representation and loss design. Full $SE(3)$ pose remains the subject of future work (Section 5).

2. Related Work

2.1. Surgical Instrument Pose Estimation. Chapter 3 summarizes the broader multimodal dataset landscape for surgical robotics. Here we focus more specifically on pose-estimation benchmarks and model families relevant to vision-based instrument localization. The SurgRIPE challenge [**59**] provides only 2,841 labeled frames for six degree-of-freedom (6-DoF) instrument pose estimation—far too few for training robust deep models. SuPer [**60**] offers a surgical perception framework evaluated on roughly 2,000 frames, and while it demonstrates the feasibility of joint segmentation and pose estimation, its dataset size limits generalization. SurgPose [**61**] contributes approximately 120,000 keypoint-annotated instances but targets joint localization rather than full end-effector Cartesian state, while SurgeoNet [**62**] sidesteps the data scarcity problem by relying entirely on synthetic data generated from 3D instrument models—an approach that inevitably suffers from a domain gap when applied to real surgical scenes.

Beyond pose-specific benchmarks, JIGSAWS [**51**]—the most widely adopted surgical activity dataset pairing kinematics with video—consists of only 120 demonstrations of three elementary bench-top drills (suturing, needle passing, knot tying) performed on dry-lab phantoms by eight operators. Models trained on such normalized, repetitive motions often fail to generalize to the richer instrument interactions of full procedures. More recently, ImitateCholec [**55**] has offered over 18,000 *ex vivo* cholecystectomy demonstrations with dVRK kinematics, but it focuses on the clipping and cutting phase for imitation learning [**28**, **29**] rather than dissection—the core of the procedure—and does not target visual pose estimation.

2.2. Vision Transformers in Surgical Robotics. The Vision Transformer (ViT) [**63**] has become a dominant architecture for visual recognition tasks since its introduction. By

splitting an image into fixed-size patches and processing them through a standard Transformer encoder, ViT captures long-range spatial dependencies that convolutional architectures handle less naturally. Self-supervised pre-training strategies such as DINO [**64**] have shown that ViT encoders develop attention maps that closely correspond to object boundaries, suggesting that Transformer features are inherently well-suited to spatial localization tasks.

In surgical robotics, Transformer-based architectures have been adopted for a variety of tasks. The Surgical Robot Transformer (SRT) [**28**] and its hierarchical extension SRT-H [**29**] use Transformer encoders for imitation learning, mapping visual observations to robot actions. Both build on the Action Chunking Transformer (ACT) framework [**30**] and depend on additional wrist cameras mounted at the instrument tips—a sensing modality that is not part of the standard clinical RAS hardware—and they train a single model for both instruments jointly. These models demonstrate that Transformer features can encode the spatial relationships necessary for surgical task execution, but the joint formulation couples the representations of tools whose motions may be only loosely correlated. Outside the surgical domain, the *Decoupled Interaction Framework* (DIF) of Jiang et al. [**65**]—evaluated on general bimanual manipulation tasks rather than RAS—shows that decoupled, per-arm models can outperform integrated dual-arm control by over 20%. We adopt the same decoupling principle in our per-instrument backbone design, while restricting the input to the standard stereo endoscope already present in clinical RAS.

2.3. Gap Addressed by This Work. Existing approaches to surgical instrument pose estimation are limited by one or more of the following: (1) small datasets that preclude robust training, (2) reliance on synthetic data with limited domain transfer, (3) joint modeling of both instruments that conflates their representations, or (4) focus on activity recognition rather than Cartesian pose estimation. This work addresses these gaps by introducing a pipeline that:

- Leverages the CRCD's large-scale stereo video with synchronized kinematics (443,044 usable stereo pairs after postprocessing).

- Trains per-instrument ViT backbones on a segmentation pre-training task, preventing cross-instrument feature interference and ensuring each encoder specializes to its assigned tool.
- Demonstrates that vision-based translation estimation can achieve mean accuracy near one centimeter on unseen surgical videos, providing a foundation for full $SE(3)$ estimation.

3. Methodology

Our approach decouples visual feature learning from pose regression in a two-stage pipeline (Fig. 7.1). In Stage 1, we train per-instrument ViT backbones on an instrument segmentation task, forcing each encoder to learn spatially grounded, instrument-aware features without requiring any pose annotations. In Stage 2, the frozen backbones are paired with lightweight pose estimation heads that process stereo image pairs and regress end-effector translation. The following subsections detail each component.

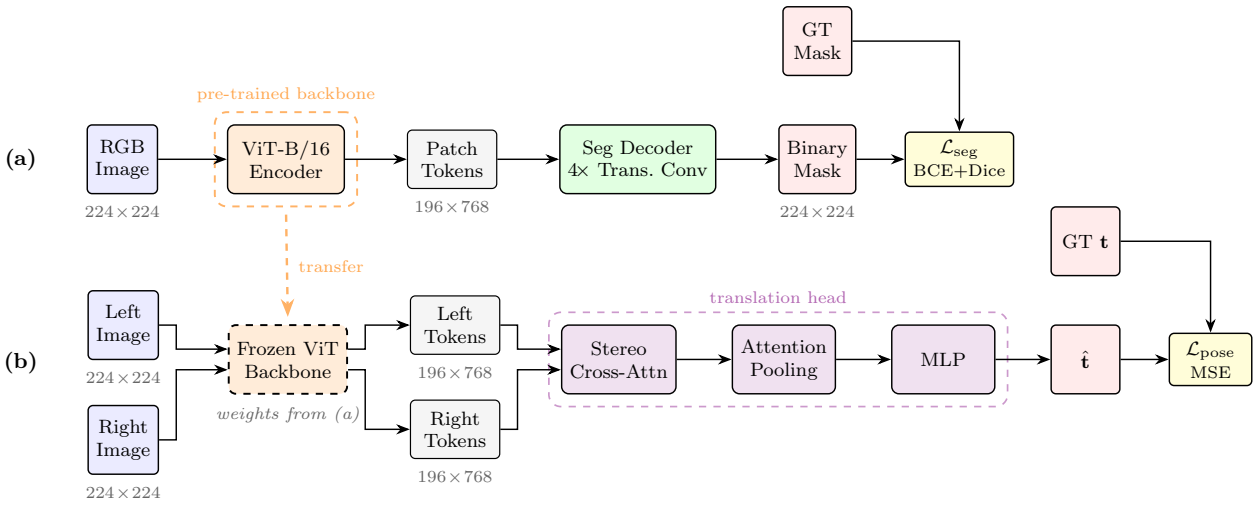


FIGURE 7.1. Two-stage training pipeline. **(a)** Stage 1: a ViT-Base/16 encoder is trained end-to-end with a segmentation decoder to produce instrument-specific spatial features (one model per arm). **(b)** Stage 2: the pre-trained encoder is frozen and paired with a stereo pose estimation head comprising bidirectional cross-attention, attention pooling, and an MLP regressor that predicts the 3D translation of the end-effector.

3.1. Dataset Preparation from the CRCD. The CRCD [**66**, **67**] provides stereo video frames (left and right views) captured from a da Vinci surgical system during robotic cholecystectomy, along with synchronized end-effector state for two PSMs and console pedal signals (see Chapter 3 for a full description). Throughout the CRCD, the PCH is mounted on PSM1 and the FBF on PSM2; the kinematic ground truth for each instrument is therefore taken from the corresponding arm's state record. Each state record includes the Cartesian position, orientation (as a unit quaternion), and jaw angle. The dataset contains over 755,000 frames spanning multiple surgical videos performed by surgeons with varying levels of expertise.

While the CRCD provides kinematic ground truth, it does not include per-frame instrument segmentation masks suitable for pre-training the ViT backbone. To enable segmentation-based representation learning, we produce per-frame instrument masks through a two-step process: manual annotation followed by automated inference.

3.1.1. *Instrument Annotation.* We annotate instrument segmentation masks for a representative subset of CRCD frames using an interactive annotation interface [**111**] built on the Segment Anything Model 2 (SAM2) [**45**]. The annotator allows point-based prompting (positive and negative clicks) on each frame, runs SAM2 inference to produce segmentation masks, and supports batch propagation across video sequences to maintain consistent object tracking. The annotations are exported in YOLO format, forming a training set for the instance segmentation model described below. This annotation pipeline parallels the SAM2-based workflow used for the CRCD tissue annotations (Chapter 3, Section 2.4), but it is adapted specifically for generating per-instrument binary masks rather than multi-class tissue labels.

3.1.2. *Instrument Segmentation Model.* Using the annotated data, we train a YOLO11l-seg instance segmentation model [**43**] to detect and segment the two instrument classes. The model is initialized with pre-trained weights and fine-tuned for 30 epochs using AdamW [**112**] with an initial learning rate of 10^{-3} and a cosine schedule that decays to 10^{-5} [**113**]. We use an input resolution of 640^2, automatic batch scaling (batch fraction of 0.8), dropout of 0.1,

and early stopping with patience 5. Data augmentation includes perspective (10^{-3}), random erasing (0.2), rotation and shear ($\pm 15°$), horizontal and vertical flips (0.2 each), and Mosaic (0.2). Table 7.1 summarizes the complete training configuration.

TABLE 7.1. YOLO11l-seg training settings for instrument mask generation.

Config	**Value**
Model	YOLO11l-seg (pre-trained)
Image size	640^2
Optimizer	AdamW
Base learning rate	10^{-3}
LR schedule	Cosine decay ($\rightarrow 10^{-5}$)
Batch size	Auto-scaled (fraction 0.8)
Training epochs	30
Early stopping	Patience 5
Dropout	0.1
Augmentation	
Perspective	10^{-3}
Random erasing	0.2
Rotation	$\pm 15°$
Shear	$\pm 15°$
Horizontal flip	0.2
Vertical flip	0.2
Mosaic	0.2

Table 7.2 reports validation performance at the best epoch. The trained model achieves a mask mAP_{50} of 89.8% and $\text{mAP}_{50\text{-}95}$ of 77.5%, which we consider sufficiently accurate for downstream training, as confirmed by the pose estimation results in Section 4.2.

TABLE 7.2. YOLO11l-seg validation results at the best epoch.

Task	**Precision**	**Recall**	**mAP$_{50}$**	**mAP$_{50\text{-}95}$**
Box	87.7	87.0	88.4	81.8
Mask	88.5	87.8	89.8	77.5

3.1.3. *Dataset Postprocessing.* We apply the trained YOLO model to the full CRCD to produce a postprocessed dataset suitable for training the ViT pipeline. For each frame, we run inference on both the left and right stereo images. The model outputs a two-channel

instrument mask per image, with each channel corresponding to one instrument class (PCH or FBF). Frames in which no instrument is detected in either view are discarded.

Alongside the masks, we extract the kinematic state for each frame. All poses are expressed in the coordinate frame attached to the endoscope tip (see Chapter 2 for the camera-to-robot calibration), and the translation components are scaled to centimeters ($\times 100$). The resulting state vector for each frame contains, per instrument, the Cartesian translation $\mathbf{t}_i \in \mathbb{R}^3$, the unit-quaternion orientation $\mathbf{q}_i \in \mathbb{R}^4$, and the scalar jaw angle $j_i \in \mathbb{R}$. In this work, we regress only $\mathbf{t}_i$; the remaining components are retained in the dataset for future extension to full $SE(3)$ + jaw-angle estimation.

After discarding frames without a detected instrument and videos lacking usable stereo data, the postprocessed dataset contains **443,044 stereo pairs**, each consisting of left and right instrument masks and the corresponding kinematic state.

3.1.4. *Train–Test Split.* We partition the postprocessed dataset at the *video* level to prevent temporal data leakage between frames of the same surgical procedure. Three videos (C 2, F 1, G 1, using the surgeon labels from [**66**,**67**]) are held out as the test set; all remaining videos form the training set, from which a 10% validation split is drawn. For the segmentation stage, left and right stereo views are treated as independent samples, yielding 667,576 training and 218,512 test samples. For the pose estimation stage, each stereo pair constitutes a single sample, giving 333,788 training and 109,256 test samples. All experiments are conducted on a workstation with an Intel Core i9-12900K CPU, 64 GB RAM, and an NVIDIA GeForce RTX 3090 GPU.

3.2. Stage 1: Segmentation Backbone Pre-Training. Predicting the end-effector pose of a surgical instrument from endoscopic images requires features that are spatially grounded to the instrument. A backbone trained directly on the regression objective may fail to learn such localized representations, especially when both instruments appear in the same frame and the model must disentangle their individual contributions. To address this, we first pre-train a ViT backbone with a segmentation decoder, forcing the encoder

to produce patch-level features that are explicitly instrument-aware; the backbone is then frozen before attaching pose estimation heads.

3.2.1. *Architecture.* The segmentation model couples a ViT-Base/16 encoder [**63**], initialized from the `timm` library [**114**], with a lightweight upsampling decoder. The encoder processes a 224×224 RGB image and produces a 14×14 grid of 196 patch tokens of dimension $d = 768$. The decoder progressively upsamples the token features across four transposed convolutional stages to recover a single-channel binary mask at the input resolution. This architecture is deliberately simple: the encoder carries the representational burden, while the decoder serves only to provide a pixel-level training signal.

3.2.2. *Per-Instrument Training.* Rather than training a single backbone to segment both instruments simultaneously, we train *one segmentation model per instrument*: one for the PCH and one for the FBF. Each model receives the per-instrument binary mask from the postprocessed dataset as its ground-truth target. Training separate backbones allows each encoder to specialize its attention maps on a single instrument—a property we later exploit when attaching pose estimation heads.

Both models are initialized from ImageNet-pretrained weights [**63**] and fine-tuned end-to-end (no layers frozen) for 10 epochs. We use AdamW [**112**] with a learning rate of 3×10^{-4}, a linear warmup for 3 epochs, and cosine annealing to zero [**113**]. The batch size is 4096. The loss function is an equal-weighted combination of Binary Cross-Entropy (BCE) and Dice loss [**115**]:

$$\mathcal{L}_{\text{seg}} = 0.5 \cdot \mathcal{L}_{\text{BCE}} + 0.5 \cdot \mathcal{L}_{\text{Dice}} \tag{7.1}$$

where the Dice loss encourages precise boundary delineation and the BCE loss provides a per-pixel gradient signal. We use a dropout rate of 0.2 and mixed-precision (FP16) training. Both left- and right-stereo views are used as independent training samples, effectively doubling the dataset size. Table 7.3 summarizes the complete training configuration.

3.3. Stage 2: Translation Estimation Head. Given the pre-trained backbones from Stage 1, we attach lightweight regression heads to predict the 3D translation of each end-effector from a stereo image pair.

TABLE 7.3. Segmentation backbone training settings.

Config	**Value**
Architecture	
Encoder	ViT-Base/16 (ImageNet-pretrained)
Decoder hidden dim	128
Image size	224^2
Training	
Optimizer	AdamW
Base learning rate	3×10^{-4}
LR schedule	Cosine decay
Warmup epochs	3
Warmup schedule	Linear
Batch size	4096
Training epochs	10
Dropout	0.2
Loss	$0.5 \cdot \text{BCE} + 0.5 \cdot \text{Dice}$
Mixed precision	FP16

3.3.1. *Architecture.* The pose estimation model reuses the frozen ViT backbone and adds a per-component head for each state variable of interest (in this work, translation only). Each head follows a three-step pipeline:

(1) **Stereo cross-attention.** Given patch tokens from the left and right views, bidirectional cross-attention fuses the two feature sequences. Left tokens attend to right tokens and vice versa, producing depth-aware representations. The attended features are concatenated, projected back to the embedding dimension $d = 768$, and normalized. This mechanism enables the model to implicitly compute stereo correspondences and extract depth cues without explicit disparity estimation.

(2) **Attention pooling.** A single learned query token attends to the fused sequence via cross-attention, compressing the spatial information into a compact vector of dimension $d = 768$. This attention-based pooling allows the model to dynamically weight spatial regions based on their relevance to the pose estimation task, rather than using a fixed spatial average.

(3) **MLP regression.** A two-layer MLP ($768 \to 512 \to 256 \to 3$, with LayerNorm, GELU activation, and dropout at each hidden layer) maps the pooled vector to a 3-dimensional translation prediction $\hat{\mathbf{t}} \in \mathbb{R}^3$ (centimeters).

A single pose estimation model predicts the translation for *one* instrument. We therefore train two models—one per instrument—each inheriting its corresponding segmentation backbone.

3.3.2. *Training.* During head training, the backbone weights are frozen, so gradient updates affect only the stereo fusion, attention pooling, and MLP parameters. This design choice is motivated by two considerations: (1) the segmentation pre-training has already produced features that are spatially grounded to the target instrument, and (2) freezing the backbone prevents catastrophic forgetting of the segmentation features and drastically reduces the number of trainable parameters.

We train both models for 10 epochs using AdamW [**112**] with a learning rate of 3×10^{-4}, a linear warmup for 3 epochs, and cosine decay to zero [**113**]. The batch size is 4096. We inject Gaussian feature noise ($\sigma = 0.05$) on the patch tokens as a regularizer. The loss function is the mean squared error (MSE) between the predicted and ground-truth translation vectors:

$$\mathcal{L}_{\text{pose}} = \frac{1}{N} \sum_{i=1}^{N} \|\hat{\mathbf{t}}_i - \mathbf{t}_i\|^2 \tag{7.2}$$

where $\hat{\mathbf{t}}_i$ and $\mathbf{t}_i$ denote the predicted and ground-truth translations, respectively. Dropout is set to 0.2, and training uses mixed-precision (FP16). Table 7.4 provides the complete training configuration.

4. Experiments and Results

This section presents the experimental evaluation of both stages of the pipeline: the segmentation backbone pre-training and the translation estimation heads. All models are trained and evaluated on the CRCD-derived dataset described in Section 3.1.

TABLE 7.4. Pose estimation head training settings.

Config	Value
Architecture	
Backbone	Frozen segmentation encoder (per-arm)
Cross-attention	4 heads, left $\leftrightarrow$ right views
Attention pooling	4 heads, 1 learned query token
MLP head	$768 \to 512 \to 256 \to 3$
Activation	GELU
Normalization	LayerNorm
Output	$\hat{\mathbf{t}} \in \mathbb{R}^3$ (cm)
Training	
Optimizer	AdamW
Base learning rate	3×10^{-4}
LR schedule	Cosine decay
Warmup epochs	3
Warmup schedule	Linear
Batch size	4096
Training epochs	10
Dropout	0.2
Feature noise	Gaussian, $\sigma = 0.05$
Loss	MSE
Mixed precision	FP16

4.1. Segmentation Backbone Performance. Table 7.5 reports the test-set performance for each per-instrument backbone. The PCH backbone achieves a mean Intersection-over-Union (IoU) [**116**] of 94.0%, while the FBF backbone reaches 88.3%. The lower FBF score is consistent with the retractor role of the forceps: because the FBF is largely stationary once it grasps the gallbladder, much of its shaft lies outside the endoscopic field of view in most frames, leaving only the jaw visible and giving the segmentation model a less distinctive visual signal to learn from.

TABLE 7.5. Segmentation backbone test results at the best epoch.

Backbone	Test Loss	Test IoU	Train IoU
PCH	0.121	94.0%	95.3%
FBF	0.187	88.3%	95.6%

To verify that the encoders learn spatially meaningful features, we visualize Grad-CAM++ [**117**] attention maps from the last Transformer block on test frames (Fig. 7.2). The heatmaps consistently highlight the target instrument while suppressing the background and the other instrument, confirming that each encoder learns to attend exclusively to its assigned tool and produces features well-suited to downstream pose estimation. The PCH attention maps clearly delineate the cautery hook shaft and tip, while the FBF maps focus on the forceps jaws despite the partial occlusion typical of the retractor configuration.

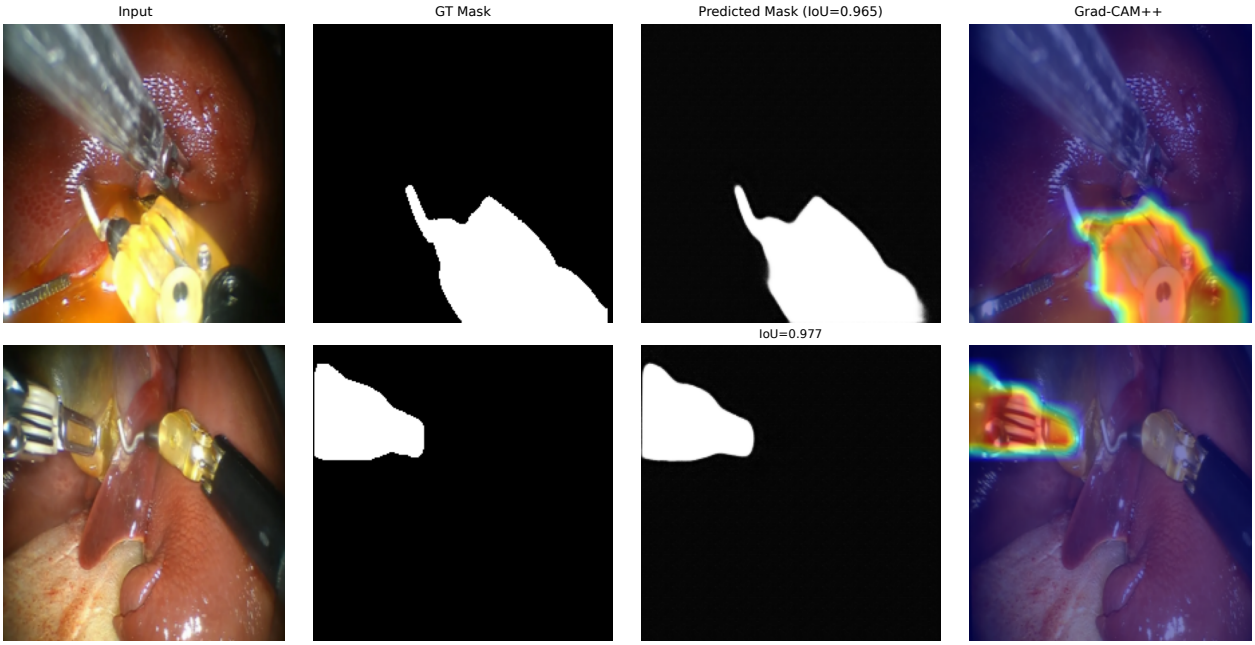


FIGURE 7.2. Grad-CAM++ visualization on test frames for both instrument-specific backbones. Each row uses a different frame and shows (left to right): input image, ground-truth mask for that instrument only (PCH on top, FBF on bottom), predicted segmentation mask (with IoU), and Grad-CAM++ from the last Transformer block. Top: PCH (cautery hook); Bottom: FBF (bipolar forceps).

4.2. Translation Estimation Performance. Table 7.6 reports the MSE on the train, validation, and test splits for both instruments. FBF achieves a lower training MSE than PCH (0.181 vs. 0.254 cm^2), yet its validation and test MSE are substantially higher (0.566 / 0.581 vs. 0.363 / 0.380 cm^2), yielding a generalization gap more than three times larger. This asymmetry reflects the different functional roles of the two instruments during cholecystectomy: the forceps (FBF) acts primarily as a retractor, holding tissue in place and

therefore remaining largely confined to a small region of the workspace, whereas the cautery hook (PCH) is the active manipulator and traverses a broader area. The concentrated spatial distribution makes the FBF training data easy to fit, but the model struggles when the instrument occasionally moves outside its usual region—an effect quantified in Section 4.6. The lower test error of PCH is also consistent with its higher backbone IoU (Table 7.5).

TABLE 7.6. Translation estimation loss (MSE, cm^2).

Instrument	Train MSE	Val MSE	Test MSE
PCH	0.254	0.363	0.380
FBF	0.181	0.566	0.581

To provide an interpretable measure of accuracy, we evaluate both models on the full test set and report the *mean Euclidean translation error* (in centimeters). On the test set, PCH achieves a mean error of **0.94** cm and FBF achieves **1.13** cm. To contextualize these figures: the test-set workspace spans approximately $7.5 \times 6.1 \times 10.0$ cm for PCH and $7.5 \times 6.4 \times 10.2$ cm for FBF, giving bounding-box diagonals of roughly 13.9 cm and 14.2 cm, respectively. The mean errors therefore represent 6.7% (PCH) and 8.0% (FBF) of the workspace diagonal, placing accuracy well below the $\sim$5 cm kinematic drift reported for cable-driven systems [**28**, **29**].

4.3. Per-Axis Error Analysis. Table 7.7 decomposes the error by translation axis. For both instruments, the depth axis (z) incurs the largest error, which is expected: stereo triangulation accuracy degrades with distance from the camera, and the z-axis is most sensitive to stereo disparity. Notably, the cross-attention fusion still recovers depth to within a mean absolute error of 0.51 cm (PCH) and 0.67 cm (FBF), suggesting that the stereo mechanism extracts meaningful disparity cues. The FBF's elevated z-axis error (0.67 cm vs. 0.51 cm for PCH) further supports the spatial coverage hypothesis: the forceps' limited range of motion during training provides fewer depth variations for the model to learn from.

4.4. Error Distribution and Per-Video Analysis. Figure 7.3 shows the per-frame error distributions with density histograms and cumulative distribution functions (CDFs).

TABLE 7.7. Per-axis mean absolute translation error on the test set.

Instrument	Axis	Mean err. (cm)
PCH	x	0.47
	y	0.40
	z	0.51
FBF	x	0.49
	y	0.53
	z	0.67

The tight concentration near zero and steep CDFs indicate that most predictions remain below or near the one-centimeter range, though FBF exhibits a heavier tail consistent with its larger generalization gap. For PCH, the median error (0.79 cm) is notably below the mean (0.94 cm), indicating that the distribution is right-skewed—a small number of difficult frames (typically those where the instrument is partially occluded or at the edge of the field of view) disproportionately affect the mean.

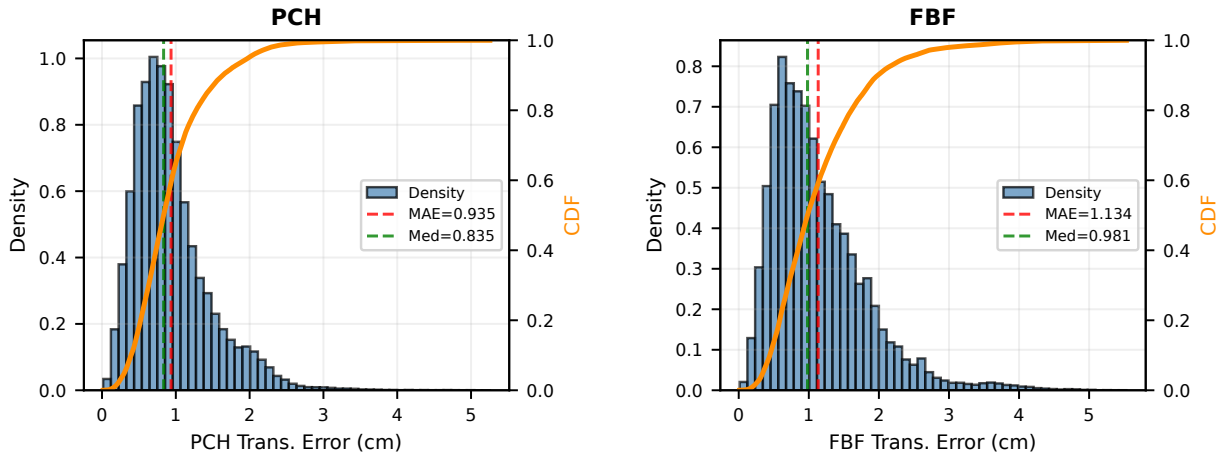


FIGURE 7.3. Mean Euclidean translation error distributions (density and CDF) on the test set for PCH (left) and FBF (right). Dashed red and green lines mark the mean and median error, respectively.

Figure 7.4 breaks down the error by test video. The per-video distributions are broadly similar, with some variation likely reflecting differences in surgical technique and instrument motion patterns across procedures. This consistency across the three held-out surgeons (C, F, and G represent different experience levels) suggests that the model generalizes reasonably

well to different operating styles, though the occasional outlier frames in certain videos indicate room for improvement.

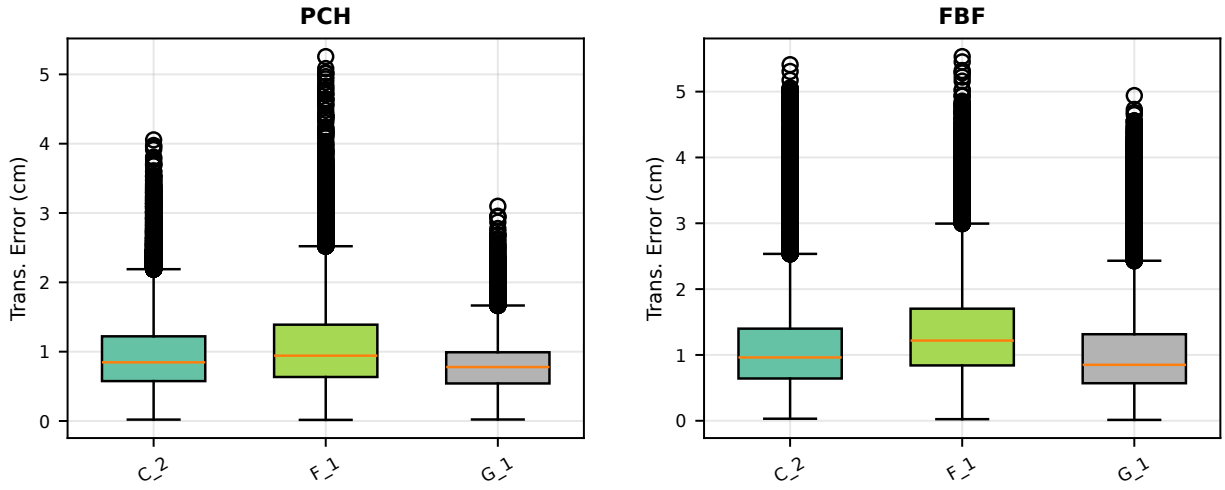


FIGURE 7.4. Per-video translation error breakdown on the test set for PCH (left) and FBF (right). Each box shows the inter-quartile range of the Euclidean translation error (cm) for all frames in one test video; outliers are shown as individual points.

4.5. Trajectory Tracking. Figure 7.5 illustrates the tracking quality on a 5,000-frame segment of test video G 1, selected to minimize the combined error for both instruments. Over this segment, PCH achieves a mean error of 0.64 cm and FBF achieves 0.61 cm. The raw per-frame predictions closely follow the ground truth on all three translation axes. A Savitzky–Golay filter (window = 51 frames, cubic polynomial) further smooths frame-level noise to produce near-continuous trajectories. The filtered predictions virtually overlap with the ground truth, demonstrating that the model captures the overall motion dynamics even when individual frame predictions contain noise.

This trajectory-level accuracy is particularly relevant for the dissection pipeline described in Chapters 5 and 6: the dissection controller operates at a timescale where temporal smoothing is naturally applicable, and the filtered trajectory errors (< 0.65 cm) are well within the tolerance required for boundary-guided dissection.

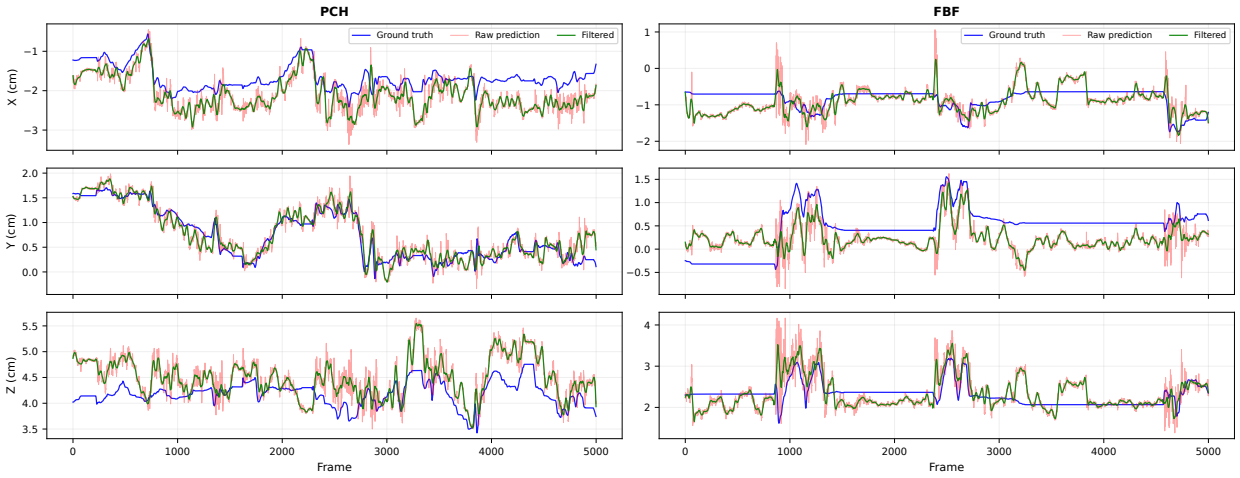


FIGURE 7.5. Translation trajectories on a 5,000-frame segment of test video G 1 for PCH (left, mean error = 0.64 cm) and FBF (right, mean error = 0.61 cm). Each row shows one translation axis (x, y, z), comparing ground truth, raw per-frame predictions, and Savitzky–Golay filtered estimates. Both instruments closely track the ground truth, and the local polynomial filter removes frame-level noise to produce smooth trajectories.

4.6. Spatial Coverage Analysis. To understand the sources of error, we analyze how spatial coverage in the training set relates to prediction accuracy. Figure 7.6 shows training-set spatial frequency heatmaps alongside boxplots of translation error grouped by the fraction of test-frame mask pixels falling in rarely-observed regions (defined as locations with training frequency below 5% of the peak). The FBF heatmap confirms the expected concentration: the forceps occupy a tight region consistent with their role as a stationary retractor. For both instruments, frames with a larger share of pixels outside the well-covered training region exhibit noticeably higher errors, confirming that spatial familiarity is a key determinant of prediction quality.

Figure 7.7 complements this view by showing the difference in test-sample counts between FBF and PCH across the rare-observation bins. FBF has substantially more test samples in the higher out-of-coverage bins, meaning that the forceps more frequently occupies workspace regions that were under-represented during training. This imbalance offers a concrete explanation for FBF's larger generalization gap observed in Table 7.6: when the FBF departs from its habitual retraction position—for example, during instrument exchanges

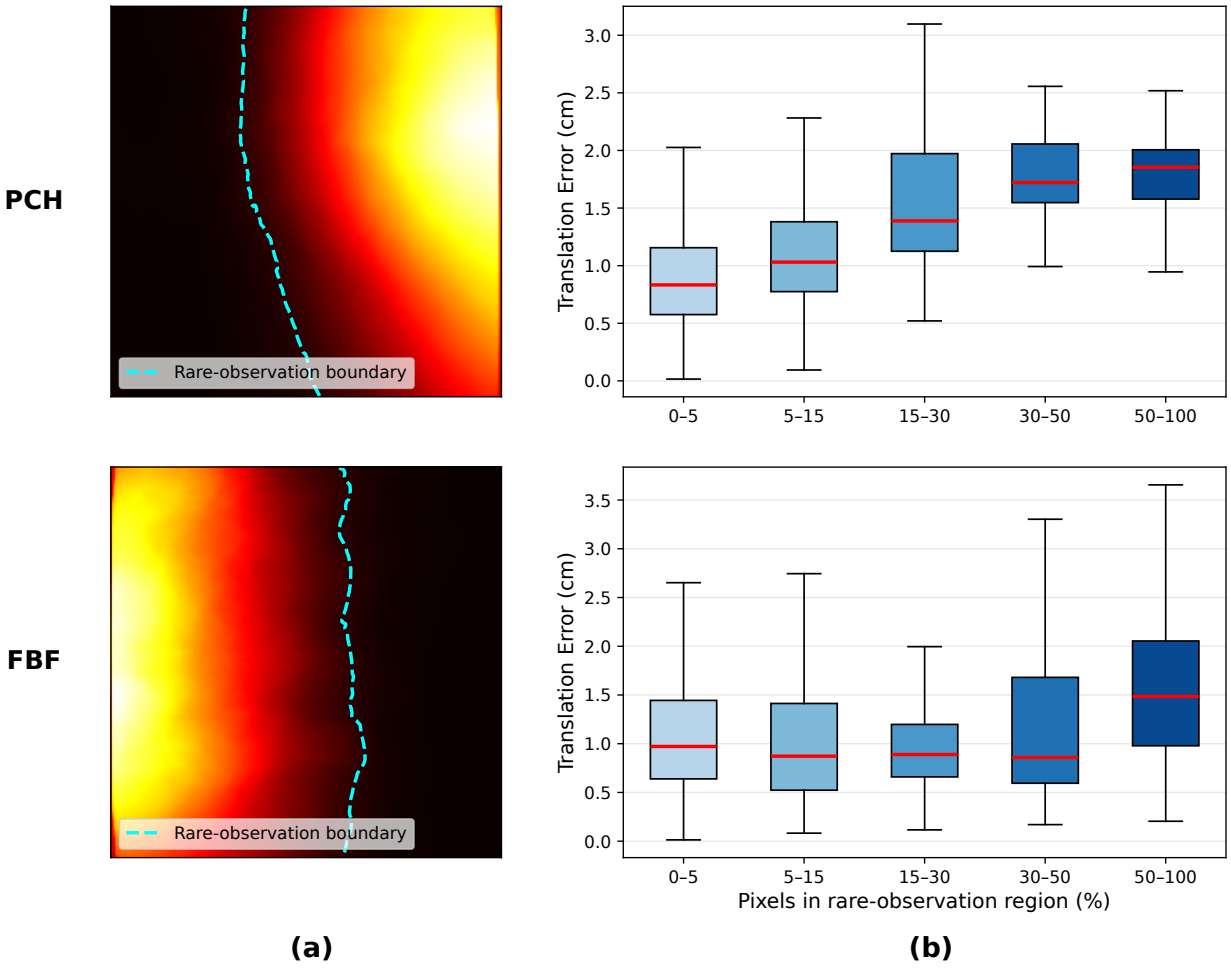


FIGURE 7.6. Spatial coverage analysis for PCH (top) and FBF (bottom). (a) Training spatial frequency heatmaps with the rare-observation boundary (cyan, <5% of peak frequency) overlaid. (b) Translation error grouped by the fraction of test-frame mask pixels falling in the rare-observation region. Higher out-of-coverage fractions are associated with larger prediction errors, indicating that spatial familiarity in the training set is an important factor in pose estimation accuracy.

or repositioning—the model encounters visual configurations it has rarely seen, leading to elevated prediction errors.

4.7. Ablation Study. To isolate the contribution of each design choice, we ablate along three dimensions (Table 7.8): pre-training strategy, backbone freezing, and stereo fusion method. All variants share the same head training hyperparameters from Table 7.4.

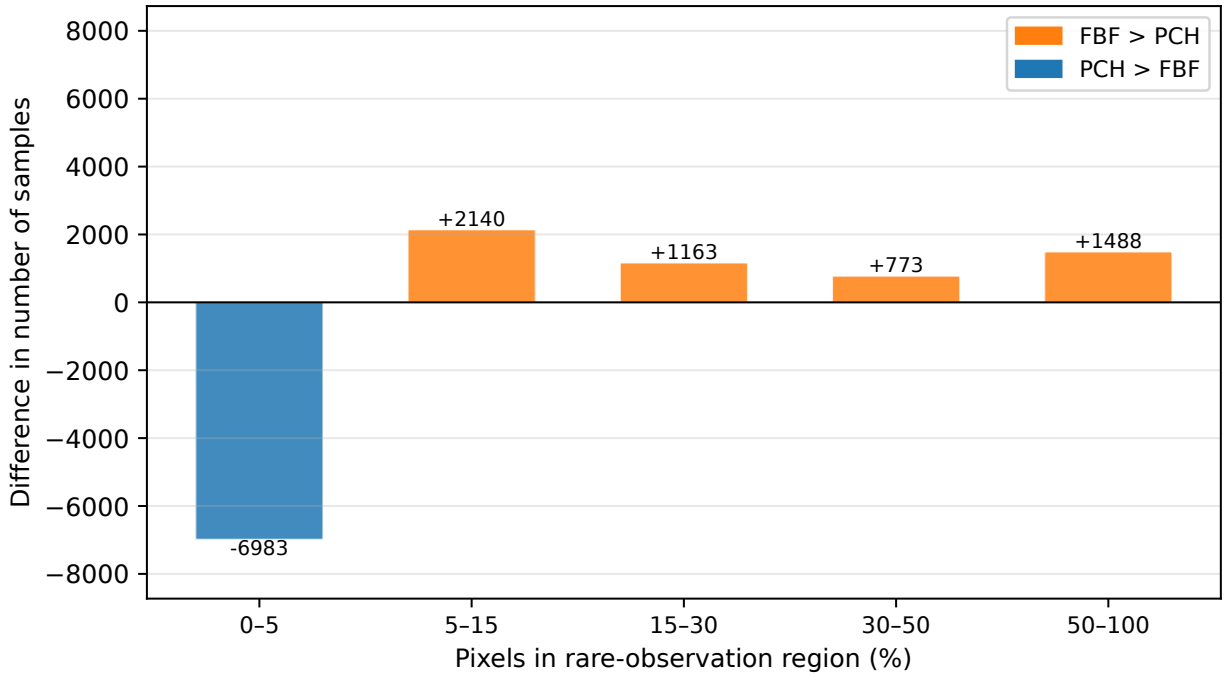


FIGURE 7.7. Difference in test-sample counts between FBF and PCH across rare-observation bins. Positive bars (orange) indicate bins where FBF has more samples; negative bars (blue) indicate PCH dominance. FBF has substantially more samples in the higher out-of-coverage bins, consistent with its larger generalization gap.

TABLE 7.8. Ablation study on the test set. Seg = segmentation pre-training; IN = ImageNet; Err. = mean Euclidean error (cm); MSE in cm^2.

			PCH		FBF	
Pre-train	**Backbone**	**Fusion**	**MSE**	**Err.**	**MSE**	**Err.**
Seg	Frozen	Cross-attn	0.380	0.94	0.581	1.13
Seg	Fine-tuned	Cross-attn	0.394	0.93	0.606	1.15
Seg	Frozen	Concat	0.373	0.91	0.616	1.16
IN	Frozen	Cross-attn	0.808	1.40	0.996	1.54
IN	Fine-tuned	Cross-attn	0.708	1.31	0.934	1.47

4.7.1. *Pre-Training Strategy.* Replacing the segmentation-pretrained backbone with an ImageNet-pretrained ViT-Base/16 increases the mean error by **49%** for PCH (0.94 → 1.40 cm) and **36%** for FBF (1.13 → 1.54 cm). Even when the last Transformer block of the ImageNet backbone is fine-tuned during head training, the error decreases only modestly (1.40 → 1.31 cm for PCH; 1.54 → 1.47 cm for FBF) and remains far above the frozen

segmentation baseline. This confirms that per-instrument segmentation pre-training provides spatial grounding that cannot be recovered by fine-tuning a generic backbone. The instrument-specific segmentation task forces each encoder to develop attention patterns that isolate the target tool from the background and from the other instrument—a capability that a general-purpose ImageNet backbone simply does not possess.

4.7.2. *Backbone Freezing.* Fine-tuning the last Transformer block of the segmentation backbone yields performance nearly identical to the fully frozen variant (0.93 vs. 0.94 cm for PCH; 1.15 vs. 1.13 cm for FBF). This indicates that the segmentation features are already well-suited for pose estimation and further adaptation is unnecessary. The frozen-backbone approach has the additional practical advantage of reducing the number of trainable parameters and preventing catastrophic forgetting of the spatial features learned during segmentation.

4.7.3. *Fusion Method.* Replacing the bidirectional cross-attention with simple feature concatenation yields comparable results (0.91 vs. 0.94 cm for PCH; 1.16 vs. 1.13 cm for FBF). This shows that the segmentation backbone encodes sufficiently discriminative per-view features for the downstream MLP to recover stereo correspondence without explicit cross-view attention. The cross-attention mechanism provides a marginal advantage for FBF (where stereo depth cues are more critical due to the forceps' limited spatial variation), but the effect is small.

4.7.4. *Summary.* Across all three ablation dimensions, the dominant factor in translation accuracy is *backbone quality*, shaped by per-instrument segmentation pre-training. Neither fine-tuning the backbone nor changing the fusion method produces a significant effect. This finding has important practical implications: it suggests that efforts to improve pose estimation accuracy should focus on improving the quality and diversity of the segmentation pre-training data, rather than on designing more sophisticated fusion architectures.

5. Discussion

5.1. Summary of Findings. This chapter presented a two-stage pipeline for estimating instrument end-effector translation from stereo endoscopic images. The key findings are:

(1) **Near-one-centimeter accuracy.** Both instrument models achieve mean Euclidean translation errors near one centimeter on unseen surgical videos (0.94 cm for PCH, 1.13 cm for FBF), and the filtered trajectories track ground truth to within 0.65 cm.

(2) **Segmentation pre-training is critical.** Ablation studies show that removing segmentation pre-training increases error by up to 49%, while changing the fusion mechanism or fine-tuning the backbone has negligible effect. This establishes that backbone quality—not head architecture—is the dominant factor in translation accuracy.

(3) **Spatial coverage drives generalization.** The FBF's larger generalization gap is directly attributable to its limited spatial coverage during training: the retractor occupies a narrow workspace region, leaving the model under-prepared for the occasional repositioning movements.

(4) **Depth is the hardest axis.** The z-axis (depth) consistently incurs the largest per-axis error for both instruments, consistent with the fundamental limitation of stereo triangulation at distance.

5.2. Implications for the Autonomous Dissection Pipeline. The results of this chapter have direct implications for the autonomous dissection framework presented in Chapters 5 and 6. In that framework, the 3D position of the PCH tip is determined through a pipeline of keypoint detection (Chapter 4), stereo reconstruction (Chapter 2), and forward kinematics. Each step introduces potential error: the keypoint detector may mislocalize the tip (particularly at the edge of the frame, as discussed in Section 3.3.1), the stereo reconstruction depends on accurate camera calibration, and the forward kinematics suffer from the cable-driven errors noted in Section 1.1.

The vision-based translation estimation demonstrated here offers a complementary—and potentially simpler—path to the same goal. Instead of detecting keypoints and triangulating their 3D positions, a ViT-based model could directly predict the end-effector translation from the stereo frame pair, bypassing both the keypoint detection and stereo reconstruction steps. The near-one-centimeter accuracy achieved in this work is already competitive with the precision required by the current dissection pipeline, which achieves submillimeter boundary tracking on average but can still exhibit oscillatory motion when keypoints are misdetected (Chapter 6). With further improvements in spatial coverage and the addition of rotation estimation, vision-based pose estimation could serve as either a replacement for or a fusion partner with the keypoint-based pipeline.

5.3. Limitations. Several limitations of the current approach warrant discussion:

- **Translation only.** This work addresses only the translation component. Extending to full $SE(3)$ pose (rotation + translation) and jaw angle is necessary for complete instrument state recovery and will require addressing the rotation representation challenges discussed in Section 1.4.
- **Spatial coverage.** The FBF's elevated test error highlights a fundamental limitation of learning-based approaches: the model can only generalize to configurations that are sufficiently represented in the training data. For instruments with highly constrained roles (such as a retractor), the training distribution may not cover the full range of possible positions.
- ***Ex vivo* data.** All experiments are conducted on *ex vivo* cholecystectomy data from the CRCD. *In vivo* procedures introduce additional challenges—varying lighting, smoke, blood, and a more dynamic surgical field—that may degrade performance.
- **Single procedure type.** The model is trained and evaluated exclusively on cholecystectomy. Generalization to other surgical procedures remains untested.
- **Kinematic ground truth quality.** The dVRK kinematics used as ground truth are themselves imperfect, introducing label noise that places a floor on achievable

accuracy. The reported near-one-centimeter errors must therefore be interpreted relative to a ground truth that may itself be inaccurate by several millimeters.

5.4. Future Directions. This work establishes a foundation for several promising research directions:

- **Full $SE(3)$ + jaw angle estimation.** The natural next step is to extend the pipeline to predict the complete end-effector state—translation, rotation, and jaw angle—using the same frozen backbone with specialized output heads. This will require addressing rotation representation (e.g., the 6-D continuous representation [**109**]) and loss design (e.g., geometric losses [**110**] that properly weight rotation against translation). Multi-task heads that share the same frozen backbone while specializing their output layers may benefit from the shared feature space.
- **Temporal modeling.** The current pipeline processes each stereo pair independently. Incorporating temporal context—through recurrent layers, temporal Transformers, or state-space models—could improve smoothness and reduce outlier errors by leveraging the physical constraints of instrument motion (bounded velocity, smooth trajectories).
- **Scaling to larger datasets.** Community initiatives such as Open-H-Embodiment [**58**] and datasets like ImitateCholec [**55**] are rapidly expanding the pool of surgical video with kinematic labels. Training on pooled, multi-institution data could improve spatial coverage and cross-procedure generalization.
- **Self-supervised and world model approaches.** Beyond supervised regression, the per-instrument ViT backbones could serve as the visual encoder in a surgical world model—a generative model that predicts how the surgical scene evolves in response to robot actions. Such a model would encode not just instrument pose but the full dynamics of the tissue-instrument interaction, enabling model-based planning for autonomous surgery.
- **Integration with the dissection framework.** The vision-based pose estimates could be fused with the keypoint-based localization from Chapter 4 using a Kalman

filter or similar state estimation framework, providing redundancy and improved robustness. Alternatively, the ViT-based model could replace the keypoint detection stage entirely, simplifying the perception pipeline while maintaining accuracy.

The broader vision motivating this work is a transition from relying on potentially inaccurate robot kinematics toward a fully vision-based understanding of the surgical scene. If a model can predict instrument pose from images alone, it can also be applied to the vast archives of surgical video that exist without any kinematic annotations—enabling retrospective analysis, surgical skill assessment, and the construction of large-scale training datasets for autonomous surgery.

CHAPTER 8

Discussion and Future Work

The preceding chapters introduced the Comprehensive Robotic Cholecystectomy Dataset (Chapter 3), used it to train and evaluate the perception and autonomous dissection pipeline (Chapters 4–6), and then extended this line of work to vision-based kinematics prediction (Chapter 7). This chapter steps back from the chapter-by-chapter results to synthesize the broader lessons across those studies, identify the main gaps that remain, and chart the path from task-specific automation toward clinically meaningful autonomy.

1. Cross-Chapter Synthesis

Taken together, the dissertation suggests that progress in robotic cholecystectomy depended on three linked ingredients: online adaptation during dissection, multimodal recordings that connect video to robot and surgeon behavior, and learned visual representations that begin to recover instrument state directly from images. The discussion below focuses on how these ingredients reinforce one another and what they imply for the next stage of the work.

2. Discussion

2.1. Adaptation and Manipulation.

2.1.1. *What the v1-to-v2 Progression Demonstrated.* The evolution from v1 to v2 [**2**, **3**] supports two important conclusions. First, online adaptation appears to be essential for energy-based dissection: the v1 system's precomputed trajectory could not account for the tissue deformation caused by energy delivery, leading to cumulative drift that degraded dissection quality over time. Second, bimanual manipulation appears to be a practical requirement for reliable dissection: grasping and stretching the gallbladder with the FBF

transformed the tissue boundary from a curved, loosely defined region into a taut, approximately linear surface that the PCH could follow more consistently. In the single-cycle trials reported in Chapter 6, the v2 YOLO11 configuration achieved an RMSE of 0.49 ± 0.02 mm, supporting the view that online boundary updates and active tissue stabilization were both important parts of the improvement.

2.1.2. *Comparison with Surgeon Technique.* Through the CRCD recordings (Chapter 3), we observed that experienced surgeons employ dissection strategies that differ in important ways from our current framework. Our system follows the visible boundary between liver and gallbladder, applying energy as the PCH traverses the boundary. Surgeons, however, frequently use a *hooking* technique: the PCH tip hooks under the boundary, and the surgeon pulls upward until the tissues separate, applying energy only when the tissue is too thick to detach mechanically. This hooking approach minimizes thermal damage to the liver and reduces the risk of postoperative complications such as postcholecystectomy syndrome [**107**]. The boundary-following approach implemented in our system is a valid dissection strategy—indeed, some surgeons use it preferentially—but a complete autonomous system will need to support both techniques and select between them based on tissue properties.

2.1.3. *The Gap to Surgeon-Level Autonomy.* Despite the substantial progress from v1 to v2 [**2**,**3**], the current system operates at a level that is far from matching surgeon performance. Surgeons continuously assess tissue thickness, adjust force and energy levels, switch between dissection techniques, reposition both instruments and the endoscope, and make split-second decisions about when to proceed and when to pause. Our system, by contrast, executes a fixed strategy (grasp–pull–follow boundary) with fixed energy settings and no endoscope adjustment. Closing this gap will require not just better perception and control, but higher-level planning and decision-making capabilities—the subject of Section 3.2.

2.2. Multimodal Data as Infrastructure.

2.2.1. *Impact in Context.* Among the public datasets considered in this dissertation, the CRCD is the only one we found that combines stereo video, full robot kinematics, pedal signals, and dense tissue and instrument annotations from cholecystectomy procedures. The

closest comparator, JIGSAWS [**51**], provides kinematics and video but only for elementary bench-top drills, while ImitateCholec [**55**] offers cholecystectomy data but targets the clipping phase rather than dissection and does not provide dense tissue annotations.

2.2.2. *Enabling Downstream Contributions.* The CRCD's impact within this dissertation is concrete and measurable. The expanded segmentation dataset (Chapter 4) was generated by annotating CRCD videos using SAM2, yielding 34,678 annotated segmentation frames across three tissue classes and 15,999 instrument keypoint instances. Compared with the initial porcine segmentation split, the training set grew from 1,430 to 25,988 labeled frames while adding the liver bed class that enabled the v2 perception models. The pedal intent recognition experiments (Chapter 3) demonstrated that the CRCD's synchronized kinematic-pedal data can train models that generalize across surgeons. Most notably, the kinematics prediction work (Chapter 7) would have been impossible without the CRCD's paired video-kinematics data: the 443,044 usable stereo pairs formed the training set for the ViT-based translation estimation models.

2.2.3. *Community Adoption.* The CRCD has been publicly released and cited in subsequent publications. Its COCO-format annotations make it directly compatible with standard computer vision training frameworks, lowering the barrier for adoption. The inclusion of documented surgeon experience levels opens avenues for skill assessment research that were previously inaccessible due to the anonymization of surgeon identity in existing datasets.

2.3. Toward Vision-Based State Estimation.

2.3.1. *What the Results Suggest.* The ViT-based kinematics prediction results suggest that stereo endoscopic images contain sufficient information to recover instrument end-effector translation with mean accuracy near one centimeter—without any access to the robot's internal joint encoders. The finding that segmentation pre-training is the dominant factor (increasing accuracy by up to 49% over ImageNet-only initialization) indicates that learning where the instrument is in the image is the key bottleneck; once the backbone has instrument-specific spatial features, even a simple regression head can recover the 3D position.

The spatial coverage analysis revealed a nuanced picture: accuracy depends heavily on how well the training data covers the instrument's workspace. The FBF, which acts as a stationary retractor during most of the procedure, achieves low training loss but a large generalization gap when the instrument moves outside its habitual region. This finding has important implications for training data collection: uniform spatial coverage may matter more than sheer dataset size.

2.3.2. *Connection to Foundation Models.* This work connects to the broader trend toward foundation models in surgical robotics. Just as general-purpose vision models (e.g., SAM [**44**], DINO [**64**]) provide transferable features for downstream tasks, our per-instrument ViT backbones provide spatially grounded surgical features that transfer from segmentation to pose estimation. The decoupled two-stage design—pre-train a visual encoder on an auxiliary task, then freeze it and attach task-specific heads—mirrors the paradigm that has proven successful in natural language processing and general computer vision. Extending this paradigm to a full surgical foundation model that supports segmentation, pose estimation, phase recognition, and action prediction from a shared backbone is a natural and promising direction.

3. Future Work

The prelim identified short-term needs such as automatic grasping, online boundary tracking, and improved perception models. Those items are now part of the v2 system (Chapter 5) and have been evaluated in Chapter 6. The future directions outlined below therefore move beyond that earlier milestone and focus on what is still missing for repeated dissection rounds, broader scene understanding, and clinical translation.

3.1. Toward Complete Gallbladder Separation.

3.1.1. *Multiple Rounds of Autonomous Pull-and-Dissect.* The current v2 system executes a single grasp–pull–dissect cycle per trial. Full gallbladder separation requires multiple autonomous rounds: after one boundary segment is dissected, the FBF must release, reposition to a new grasping point on an unexposed boundary segment, re-grasp, re-stretch, and initiate

a new dissection cycle. This introduces several new challenges: (a) the system must determine when a dissection round is complete and where to grasp next, requiring a higher-level planning layer; (b) the gallbladder's shape and attachment geometry change progressively as tissue is detached, so the planning layer must reason about partially separated anatomy; and (c) the endoscope may need to be repositioned between rounds to maintain visibility of the active boundary segment.

3.1.2. *Alternative Dissection Techniques.* As discussed in Section 2.1, surgeons frequently use a hooking technique that minimizes thermal damage. Implementing this in the autonomous framework would require the system to (a) position the PCH tip under the boundary rather than along it, (b) apply an upward pulling force and monitor tissue separation, and (c) decide whether to apply energy based on real-time assessment of tissue thickness. Force sensing—either through the dVRK's joint torque estimates or through vision-based force estimation [**26**]—would be valuable for making the energy-application decision. Supporting both boundary-following and hooking techniques, and selecting between them based on local tissue properties, would bring the system closer to the adaptive behavior exhibited by surgeons.

3.1.3. *Endoscope Visual Servoing.* The current framework uses a fixed endoscope position, which limits the workable volume and causes keypoint detection to fail when instruments approach the image boundary. Image-based visual servoing (IBVS) for the ECM would enable the endoscope to actively track the PCH tip, keeping it centered in the field of view throughout the procedure. Prior work on endoscopic servo control for the da Vinci system [**118**] provides a starting point, but integrating it with the dissection framework requires careful coordination: the endoscope must move smoothly to avoid disrupting the stereo reconstruction pipeline, and the boundary extraction algorithm must be robust to the changing viewpoint. An IBVS controller that minimizes the image-space error between the PCH tip and the image center, subject to smoothness and collision-avoidance constraints, would enable the system to operate over the full gallbladder surface rather than the single field of view currently used.

3.2. Self-Supervised Learning and World Models.

3.2.1. *From Kinematics Prediction to Scene Understanding.* Chapter 7 demonstrated that a ViT backbone pre-trained on instrument segmentation can predict instrument translation with mean accuracy near one centimeter. This result suggests a broader possibility: if a model can learn to predict instrument pose from images, it may also be able to learn the dynamics of the surgical scene—how tissue deforms in response to instrument contact, how cauterization changes tissue appearance, and how grasping alters the geometry of the dissection boundary. Such a model would move beyond predicting the current state to predicting *future* states.

3.2.2. *Predicting Future States from Current Observations.* A surgical world model would take the current stereo frame pair and the commanded robot action as input and predict the next frame pair (or its latent representation). Training such a model would be fully self-supervised, requiring only the temporal sequence of stereo frames and the corresponding kinematic commands—exactly the data provided by the CRCD. The key challenge is scale: the CRCD's 755,000 frames, while large by surgical robotics standards, may be insufficient for training a generative model that captures the full complexity of tissue dynamics. Pooling data from multiple institutions through initiatives like Open-H-Embodiment [**58**] could address this.

3.2.3. *Planning and Control from Learned Models.* A world model that accurately predicts the consequences of robot actions would enable model-based planning for autonomous surgery. Instead of the reactive, boundary-following control used in the current framework, the system could plan multi-step action sequences that optimize for task objectives (e.g., maximize tissue separation while minimizing thermal damage). This planning could operate in the model's latent space, avoiding the need for explicit 3D reconstruction, calibration, or keypoint detection. The entire perception-to-action pipeline—from raw stereo images to motor commands—would be learned end-to-end.

3.2.4. *Replacing Explicit Calibration and Keypoint Detection.* A sufficiently capable world model would subsume several components of the current framework. The fiducial-marker-based calibration (Chapter 2) exists because the dVRK's kinematic chain is inaccurate; a model that predicts instrument pose directly from images (Chapter 7) renders this calibration unnecessary at inference time. Similarly, the keypoint detection stage (Chapter 4) exists because the current control algorithm needs to know the 3D position of the instrument tip; a model that maps images directly to actions would bypass this intermediate representation. The progression from explicit, hand-designed pipelines toward end-to-end learned systems represents a fundamental shift in how surgical automation systems are built, and the components developed in this dissertation—the perception models, the dataset, and the ViT-based pose estimator—provide the building blocks for this transition.

3.3. Toward Clinical Translation.

3.3.1. *In Vivo Validation.* All experiments in this dissertation were conducted on *ex vivo* porcine livers. *In vivo* procedures introduce challenges that our *ex vivo* setup does not capture: a brighter and more variable endoscopic view due to body wall reflections, a constrained workspace defined by trocar positions, the presence of blood, smoke, and other fluids that can obscure the surgical field, and live tissue that responds differently to energy delivery than cold tissue. Validating the framework *in vivo* would require (a) retraining the perception models on *in vivo* data (or demonstrating sufficient domain transfer from *ex vivo* training), (b) adapting the control parameters to account for the constrained workspace, and (c) integrating safety monitoring systems that can detect and respond to adverse events.

3.3.2. *Safety Considerations.* Autonomous surgical systems must meet stringent safety requirements before clinical deployment. At a minimum, the system must be able to (a) detect when it is about to cause harm (e.g., approaching a critical structure such as the common bile duct), (b) stop immediately when an anomaly is detected, and (c) hand control back to the surgeon seamlessly. This requires not just accurate perception but also uncertainty quantification: the system must know when it does not know. Bayesian or ensemble-based approaches to uncertainty estimation could be integrated with the YOLO11 or ViT-based

perception models to provide calibrated confidence scores that trigger human intervention when the model's predictions are unreliable.

3.3.3. *Shared Autonomy.* The path to clinical adoption likely passes through shared autonomy rather than full autonomy. In a shared autonomy paradigm, the system handles well-defined subtasks (e.g., boundary-following dissection along a surgeon-specified region) while the surgeon retains high-level decision-making and intervenes when the task exceeds the system's competence. This approach offers immediate clinical value by reducing surgeon fatigue during repetitive subtasks while maintaining the safety guarantees that full autonomy cannot yet provide. The dVRK's pedal interface—already captured in the CRCD—provides a natural mechanism for the surgeon to toggle between manual and autonomous control.

3.4. Dataset Expansion.

3.4.1. *More Surgeons and More Procedures.* The CRCD currently includes data from seven surgeons performing cholecystectomy on porcine livers. Expanding to additional surgeons would improve the diversity of surgical styles and instrument motion patterns in the training data, directly addressing the spatial coverage limitations identified in Chapter 7. Expanding to additional procedure types (e.g., hernia repair, hysterectomy, colectomy) would test whether the perception and pose estimation models generalize beyond cholecystectomy.

3.4.2. *In Vivo Data Collection.* Collecting *in vivo* surgical data with synchronized dVRK kinematics raises both technical and ethical challenges. Technically, the dVRK's setup joints must be carefully calibrated for each procedure, and the kinematic data must be validated against an independent ground truth (e.g., optical tracking). Ethically, patient consent and institutional review board approval are required, and the data must be anonymized and stored securely. Despite these challenges, *in vivo* data is essential for training models that can operate in clinical settings.

3.4.3. *Phase Recognition and Surgical Workflow Analysis.* The CRCD's comprehensive recording of pedal signals, kinematics, and video enables research beyond the scope of this dissertation. Surgical phase recognition—automatically identifying which of the eleven cholecystectomy steps (Section 3.1) is currently being performed—is a prerequisite for higher-level

surgical automation. The pedal intent recognition models in Chapter 3 demonstrated the feasibility of predicting surgeon actions from kinematic data; extending this to full phase recognition using the combined video-kinematics-pedal signals would provide the contextual awareness needed for an autonomous system to know not just *how* to dissect, but *when* to dissect, *when* to clip, and *when* to hand control to the surgeon.

The work presented in this dissertation—a vision-based dissection framework, a multimodal surgical dataset, and a baseline for vision-based kinematics estimation—provides a concrete starting point for these future directions. Considerable work remains before clinically useful surgical autonomy is feasible, but the relevant data, perception, and control components can now be studied within a shared experimental framework.

CHAPTER 9

Conclusion

Autonomous robotic surgery holds the promise of enhanced precision, reduced surgeon fatigue, and improved patient outcomes. Realizing this promise requires advances across multiple fronts: the robot must perceive the surgical scene, plan and execute complex manipulation tasks, and do so using data-driven models trained on sufficiently large and diverse datasets. This dissertation addressed these challenges in the context of robotic cholecystectomy—the surgical removal of the gallbladder—using the da Vinci surgical system with the da Vinci Research Kit.

1. Summary of Contributions

This dissertation made three interconnected contributions.

1.0.1. *Contribution 1: A Vision-Based Autonomous Dissection Framework.* Chapters 4–6 present a vision-based autonomous dissection framework that progressed from an offline single-arm proof of concept to an online bimanual system with grasping, tissue stretching, and real-time boundary updates. In the single-cycle trials reported in Chapter 6, the upgraded system achieved a 3.3× speed improvement and a boundary tracking RMSE of 0.49 ± 0.02 mm with the YOLO11 configuration.

1.0.2. *Contribution 2: The Comprehensive Robotic Cholecystectomy Dataset.* Chapter 3 presents the CRCD, a multimodal cholecystectomy dataset containing stereo video, robot kinematics, pedal signals, tissue segmentation annotations, and instrument keypoint annotations. With over 755,000 stereo frames, 34,678 annotated segmentation frames, and 15,999 instrument keypoint instances, it provides the shared data foundation for the later perception, pedal, and kinematics studies.

1.0.3. *Contribution 3: A Baseline for Predicting Instrument Kinematics from Endoscopic Images.* Chapter 7 introduces a two-stage pipeline in which per-instrument Vision Transformer backbones are pre-trained on instrument segmentation masks and then paired with lightweight stereo pose estimation heads to regress end-effector translation. Evaluated on held-out CRCD videos, the models achieved mean Euclidean translation errors of 0.94 cm for the cautery hook and 1.13 cm for the bipolar forceps, establishing a baseline for future vision-based surgical pose estimation.

2. Significance and Broader Impact

These contributions are most meaningful when considered together rather than as isolated results.

Taken together, the results in Chapters 4–7 suggest that reliable robotic cholecystectomy automation depends on four ingredients: online adaptation to tissue deformation, active tissue stabilization, realistic multimodal data, and vision-based estimates of instrument state. This dissertation does not claim full autonomy, but it shows that these components can be integrated and evaluated within a common experimental framework.

From a broader perspective, the CRCD and the accompanying methods provide a practical basis for follow-on work on multi-step dissection, richer scene understanding, and safer human-robot collaboration in the operating room. In that sense, this dissertation is best read as a foundation for further study rather than as a complete solution.

TABLE A.1. Optimized PSM twist parameters (ζ_i) and linear calibration coefficients (α', β') for PSM1 and PSM2. Each row contains the six components of the twist vector for the corresponding joint, following the notation in Equation (2.1).

	PSM1						PSM2					
ζ_1	0.0002	−0.0385	−0.0967	0.9991	0.0265	−0.0321	−0.0028	0.0704	0.1640	−0.9968	−0.0775	0.0214
ζ_2	−0.1008	0.7068	0.0809	−0.0348	0.1076	−0.9936	−0.0971	−0.6964	−0.0327	−0.0254	0.0518	−0.9983
ζ_3	0.0262	0.9992	0.0286	0.0	0.0	0.0	0.0315	0.9995	−0.0063	0.0	0.0	0.0
ζ_4	−0.0719	0.0745	−0.7082	0.0517	−0.9927	−0.1093	0.0603	0.0391	−0.6942	−0.0458	0.9975	0.0529
ζ_5	0.0662	−0.7091	−0.0787	0.0348	−0.1076	0.9936	−0.0574	−0.6983	−0.0352	−0.0255	0.0518	−0.9983
ζ_6	−0.0002	−0.0425	−0.0495	0.9991	0.0265	−0.0321	0.0034	−0.0719	−0.1141	0.9968	0.0775	−0.0214
α'	−1.0	1.0	−1.0227	1.0	−1.0	−1.0	−1.0	−1.0	−0.9998	−0.9999	−1.0	1.0
β'	0.0047	0.0289	0.0003	0.0	0.0288	−0.0047	0.0031	−0.0306	0.0005	0.0	0.0306	0.0031

TABLE A.2. Optimized ECM twist parameters (ξ_i) and linear calibration coefficients (α', β'), following the notation in Equation (2.2).

	ECM					
ξ_1	0.0019	−0.0862	−0.0753	0.9999	0.0159	0.0019
ξ_2	−0.1008	0.7068	0.0809	−0.0348	0.1076	−0.9936
ξ_3	−0.0051	0.9999	−0.0130	0.0	0.0	0.0
ξ_4	−0.0869	0.0276	0.5999	0.0011	−0.9989	0.0478
α'		1.0491	−1.0134	−1.0077	0.9815	
β'		0.0017	−0.0267	0.0302	−0.0092	

APPENDIX A

Fiducial Marker Based Arm Calibration

This appendix reports the optimized joint twist parameters and linear calibration coefficients obtained from the global optimization procedure described in Chapter 2, Section 2.5. Table A.1 lists the six twist parameter vectors (ζ_1–ζ_6) and the linear mapping coefficients (α', β') for each PSM, as well as the four twist parameter vectors (ξ_1–ξ_4) and their corresponding coefficients for the ECM. These values, combined with Equations (2.1)–(2.2) and the calibrated joint angles from Equation (2.6), define the complete custom forward kinematics used throughout this dissertation.

APPENDIX B

Inverse Kinematics

The numerical inverse kinematics procedure is described in Chapter 2, Section 5. The SLSQP algorithm [**80**] minimizes the weighted distance metric from Equation (2.7), subject to the dVRK joint limits [**16**] reported in Table 2.1.

Bibliography

[1] P. Giulianotti, E. Benedetti, and A. Mangano, *The Foundation and Art of Robotic Surgery*. McGraw-Hill Education, 2024. [Online]. Available: https://books.google.com/books?id=xQ-0zgEACAAJ

[2] K.-H. Oh, L. Borgioli, M. Žefran, L. Chen, and P. C. Giulianotti, "A framework for automated dissection along tissue boundary," in *2024 10th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, 2024, pp. 1427–1433.

[3] K.-H. Oh, L. Borgioli, M. Žefran, V. Valle, and P. C. Giulianotti, "Autonomous dissection in robotic cholecystectomy," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025, pp. 11 240–11 246.

[4] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft COCO: Common objects in context," 2014.

[5] Y. Kwoh, J. Hou, E. Jonckheere, and S. Hayati, "A robot with improved absolute positioning accuracy for CT-guided stereotactic brain surgery," *IEEE Transactions on Biomedical Engineering*, vol. 35, no. 2, pp. 153–160, 1988.

[6] P. Gomes, "Surgical robotics: Reviewing the past, analysing the present, imagining the future," *Robotics and Computer-Integrated Manufacturing*, vol. 27, no. 2, pp. 261–266, 2011, translational Research – Where Engineering Meets Medicine.

[7] D. Pisla, C. Popa, A. Pusca, A. Ciocan, B. Gherman, E. Mois, A.-D. Cailean, C. Vaida, C. Radu, D. Chablat, and N. A. Hajjar, "On the control and validation of the PARA-SILSROB surgical parallel robot," *Applied Sciences*, vol. 14, no. 17, 2024.

[8] R. Taylor, B. Mittelstadt, H. Paul, W. Hanson, P. Kazanzides, J. Zuhars, B. Williamson, B. Musits, E. Glassman, and W. Bargar, "An image-directed robotic system for precise orthopaedic surgery," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 3, pp. 261–275, 1994.

[9] C.-A. O. Nathan, V. Chakradeo, K. Malhotra, H. D'Agostino, and R. Patwardhan, "The voice-controlled robotic assist scope holder AESOP for the endoscopic approach to the sella," *Skull base*, vol. 16, no. 03, pp. 123–131, 2006.

[10] M. Eto and S. Naito, "Robotic surgery assisted by the ZEUS system," *Endourooncology: New Horizons in Endourology*, pp. 39–48, 2005.

[11] A. L. G. Morrell, A. C. Morrell-Junior, A. G. Morrell, J. M. F. Mendes, F. Tustumi, L. G. DE-OLIVEIRA-E-SILVA, and A. Morrell, "The history of robotic surgery and its evolution: when illusion becomes reality," *Revista do Colégio Brasileiro de Cirurgiões*, vol. 48, p. e20202798, 2021.

[12] F. Pugin, P. Bucher, and P. Morel, "History of robotic surgery: From AESOP® and ZEUS® to da Vinci®," *Journal of Visceral Surgery*, vol. 148, no. 5, Supplement, pp. e3–e8, 2011, robotic surgery.

[13] I. Surgical, *da Vinci S Surgical System User Manual*. Intuitive Surgical, 2006.

[14] ——, *da Vinci Si Surgical System User Manual*. Intuitive Surgical, 2012.

[15] ——, *da Vinci Xi Surgical System User Manual*. Intuitive Surgical, 2018.

[16] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio, "An open-source research kit for the da Vinci surgical system," in *IEEE Intl. Conf. on Robotics and Auto. (ICRA)*, Hong Kong, China, 2014, pp. 6434–6439.

[17] G. S. Litynski, "Erich Mühe and the rejection of laparoscopic cholecystectomy (1985): A surgeon ahead of his time," *JSLS: Journal of the Society of Laparoendoscopic Surgeons*, vol. 2, no. 4, p. 341, 1998.

[18] W. Reynolds Jr, "The first laparoscopic cholecystectomy," *JSLS: Journal of the Society of Laparoendoscopic Surgeons*, vol. 5, no. 1, p. 89, 2001.

[19] J. Himpens, G. Leman, and G. B. Cadière, "Telesurgical laparoscopic cholecystectomy," *Surgical Endoscopy*, vol. 12, no. 8, p. 1091, 1998.

[20] L. Michael Brunt, D. J. Deziel, D. A. Telem, S. M. Strasberg, R. Aggarwal, H. Asbun, J. Bonjer, M. McDonald, A. Alseidi, M. Ujiki *et al.*, "Safe cholecystectomy multi-society practice guideline and state-of-the-art consensus conference on prevention of bile duct injury during cholecystectomy," *Surgical endoscopy*, vol. 34, pp. 2827–2855, 2020.

[21] A. Sagitov, T. Tsoy, H. Li, and E. Magid, "Automated open wound suturing: detection and planning algorithm," *Journal of Robotics, Networking and Artificial Life*, vol. 5, pp. 144–148, 2018.

[22] B. Lu, B. Li, W. Chen, Y. Jin, Z. Zhao, Q. Dou, P.-A. Heng, and Y. Liu, "Toward image-guided automated suture grasping under complex environments: A learning-enabled and optimization-based holistic framework," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 4, pp. 3794–3808, 2022.

[23] S. Iyer, T. Looi, and J. Drake, "A single arm, single camera system for automated suturing," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 239–244.

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

[25] R. Jackson, R. Yuan, D. Chow, W. Newman, and M. Çavuşoğlu, "Real-time visual tracking of dynamic surgical suture threads," *IEEE Trans Autom Sci Eng*, vol. 15, no. 3, pp. 1078–1090, Jul 2018, epub 2017 Aug 11. PMID: 29988978; PMCID: PMC6034738.

[26] E. Ayvali, R. A. Srivatsan, L. Wang, R. Roy, N. Simaan, and H. Choset, "Using Bayesian optimization to guide probing of a flexible environment for simultaneous registration and stiffness mapping," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 931–936.

[27] A. Shademan, R. S. Decker, J. D. Opfermann, S. Leonard, A. Krieger, and P. C. W. Kim, "Supervised autonomous robotic soft tissue surgery," *Science Translational Medicine*, vol. 8, no. 337, pp. 337ra64–337ra64, 2016.

[28] J. W. Kim, T. Z. Zhao, S. Schmidgall, A. Deguet, M. Kobilarov, C. Finn, and A. Krieger, "Surgical robot transformer (SRT): Imitation learning for surgical tasks," in *8th Annual Conference on Robot Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=fNBbEgcfwO

[29] J. W. B. Kim, J.-T. Chen, P. Hansen, L. X. Shi, A. Goldenberg, S. Schmidgall, P. M. Scheikl, A. Deguet, B. M. White, D. R. Tsai, R. J. Cha, J. Jopling, C. Finn, and A. Krieger, "SRT-H: A hierarchical framework for autonomous surgery via language-conditioned imitation learning," *Science Robotics*, vol. 10, no. 104, p. eadt5254, 2025.

[30] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware," in *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023.

[31] H. Lijun, H. Wang, Z. Liu, W. Chen, and X. Zhang, "Vision-based cutting control of deformable objects with surface tracking," *IEEE/ASME Transactions on Mechatronics*, vol. PP, pp. 1–1, 10 2020.

[32] Y. Kumazu, N. Kobayashi, N. Kitamura, E. Rayan, P. Neculoiu, T. Misumi, Y. Hojo, T. Nakamura, T. Kumamoto, Y. Kurahashi, Y. Ishida, M. Masuda, and H. Shinohara, "Automated segmentation by deep learning of loose connective tissue fibers to define safe dissection planes in robot-assisted gastrectomy," *Sci Rep*, vol. 11, no. 1, p. 21198, Oct 2021, pMID: 34707141; PMCID: PMC8551298.

[33] A. Attanasio, B. Scaglioni, E. De Momi, P. Fiorini, and P. Valdastri, "Autonomy in surgical robotics," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, no. 1, pp. 651–679, 2021.

[34] W.-Y. Hong, C.-L. Kao, Y.-H. Kuo, J.-R. Wang, W.-L. Chang, and C.-S. Shih, "CholecSeg8k: A semantic segmentation dataset for laparoscopic cholecystectomy based on Cholec80," *arXiv preprint arXiv:2012.12453*, 2020.

[35] C. I. Nwoye, D. Alapatt, T. Yu, A. Vardazaryan, F. Xia, Z. Zhao, T. Xia, F. Jia, Y. Yang, H. Wang *et al.*, "CholecTriplet2021: A benchmark challenge for surgical action triplet recognition," *Medical Image Analysis*, vol. 86, p. 102803, 2023.

[36] A. Murali, D. Alapatt, P. Mascagni *et al.*, "The Endoscapes dataset for surgical scene segmentation, object detection, and critical view of safety assessment," *arXiv preprint arXiv:2312.12429*, 2023.

[37] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2980–2988.

[38] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," https://github.com/facebookresearch/detectron2, 2019.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[40] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[41] F. Li, H. Zhang, H. xu, S. Liu, L. Zhang, L. M. Ni, and H.-Y. Shum, "Mask DINO: Towards a unified transformer-based framework for object detection and segmentation," 2022.

[42] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 1280–1289.

[43] G. Jocher and J. Qiu, "Ultralytics YOLO11," 2024. [Online]. Available: https://github.com/ultralytics/ultralytics

[44] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.

[45] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson *et al.*, "SAM 2: Segment anything in images and videos," *arXiv preprint arXiv:2408.00714*, 2024.

[46] A. P. Twinanda, S. Shehata, D. Mutter, J. Marescaux, M. De Mathelin, and N. Padoy, "EndoNet: A deep architecture for recognition tasks on laparoscopic videos," *IEEE transactions on medical imaging*, vol. 36, no. 1, pp. 86–97, 2016.

[47] D. Bouget, R. Benenson, M. Omran, L. Riffaud, B. Schiele, and P. Jannin, "Detecting surgical tools by modelling local appearance and global shape," *IEEE Transactions on Medical Imaging*, vol. 34, no. 12, pp. 2603–2617, 2015.

[48] T. Ross, A. Reinke, P. M. Full, M. Wagner, H. Kenngott, M. Apitz, H. Hempe, D. M. Filimon, P. Scholz, T. N. Tran, P. Bruno, P. Arbeláez, G.-B. Bian, S. Bodenstedt, J. L. Bolmgren, L. Bravo-Sánchez, H.-B. Chen, C. González, D. Guo, P. Halvorsen, P.-A. Heng, E. Hosgor, Z.-G. Hou, F. Isensee, D. Jha, T. Jiang, Y. Jin, K. Kirtac, S. Kletz, S. Leger, Z. Li, K. H. Maier-Hein, Z.-L. Ni, M. A. Riegler, K. Schoeffmann, R. Shi, S. Speidel, M. Stenzel, I. Twick, G. Wang, J. Wang, L. Wang, L. Wang, Y. Zhang, Y.-J. Zhou, L. Zhu, M. Wiesenfarth, A. Kopp-Schneider, B. P. Müller-Stich, and L. Maier-Hein, "Robust medical instrument segmentation challenge 2019," 2020.

[49] M. Allan, S. Kondo, S. Bodenstedt, S. Leger, R. Kadkhodamohammadi, I. Luengo, F. Fuentes, E. Flouty, A. Mohammed, M. Pedersen, A. Kori, V. Alex, G. Krishnamurthi, D. Rauber, R. Mendel, C. Palm, S. Bano, G. Saibro, C.-S. Shih, H.-A. Chiang, J. Zhuang, J. Yang, V. Iglovikov, A. Dobrenkii, M. Reddiboina, A. Reddy, X. Liu, C. Gao, M. Unberath, M. Kim, C. Kim, C. Kim, H. Kim, G. Lee, I. Ullah, M. Luna, S. H. Park, M. Azizian, D. Stoyanov, L. Maier-Hein, and S. Speidel, "2018 robotic scene segmentation challenge," 2020.

[50] M. Carstens, F. M. Rinner, S. Bodenstedt, A. C. Jenke, J. Weitz, M. Distler, S. Speidel, and F. R. Kolbinger, "The Dresden surgical anatomy dataset for abdominal organ segmentation in surgical data science," *Scientific Data*, vol. 10, no. 1, p. 3, 2023.

[51] N. Ahmidi, L. Tao, S. Sefati, Y. Gao, C. Lea, B. Bejar Haro, L. Zappella, S. Khudanpur, R. Vidal, and G. D. Hager, "A dataset and benchmarks for segmentation and recognition of gestures in robotic surgery," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 9, pp. 2025–2041, 2017.

[52] Y. Gao, S. S. Vedula, C. E. Reiley, N. Ahmidi, B. Varadarajan, H. C. Lin, L. Tao, L. Zappella, B. Béjar, D. D. Yuh *et al.*, "JHU-ISI gesture and skill assessment working set (JIGSAWS): A surgical activity dataset for human motion modeling," in *MICCAI workshop: M2cai*, vol. 3, no. 3, 2014.

[53] I. Rivas-Blanco, C. J. P. Del-Pulgar, A. Mariani, G. Tortora, and A. J. Reina, "A surgical dataset from the da Vinci research kit for task automation and recognition," in *2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, 2023, pp. 1–6.

[54] E. Colleoni, P. Edwards, and D. Stoyanov, "Synthetic and real inputs for tool segmentation in robotic surgery," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2020, pp. 700–710.

[55] P. Hansen, J. W. B. Kim, A. Goldenberg, J. T. Chen, Y. A. Li, A. Deguet, B. White, D. R. Tsai, R. Cha, J. Jopling, P. M. Scheikl, and A. Krieger, "ImitateCholec: A multimodal dataset for long-horizon imitation learning in robotic cholecystectomy," *Scientific Data*, vol. 13, no. 1, p. 210, 2026.

[56] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar *et al.*, "Open X-embodiment: Robotic learning datasets and RT-X models," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 6892–6903.

[57] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis *et al.*, "DROID: A large-scale in-the-wild robot manipulation dataset," *arXiv preprint arXiv:2403.12945*, 2024.

[58] N. Nelson, J.-T. Chen, J. Haworth, X. Chen, L. Zbinden, D. Huang, A. E. Abdelaal, A. Arezzo, A. Acar, F. Alambeigi, C. A. Ammirati, Y. Ao, P. D. Aranda Rodriguez, S. Atar, M. Ballo, N. Barnes, F. Barontini, F. Binkiewicz, P. Black, S. Bodenstedt, L. Borgioli, N. Budjak, B. Calmé, F. Carrillo, N. Cavalcanti, C. Chen, H. Chen, S. Chen, Q. Chen, Z. Chen, Z. Chen, S. S. Cheng, M. Cheng, M. Cheng, Z.-Y. S. Chiu, X. Chu, C. Correa-Gallego, G. Dagnino, A. Deguet, J. Delgado, J. C. DeLong, K. Deng, A. Dimitrakakis, Q. Ding, H. Ding, G. Distefano, D. Donoho, A. Duan, M. Esposito, S. Farritor, J. Fayad, Z. Fayad, M. Ferradosa, F. Filicori, C. Finn, P. Fürnstahl, J. Ge, S. Giannarou, X. Giralt Ludevid, F. Giraud, A. A. Godbole, K. Goldberg, A. Goldenberg, D. Granero Marana, X. Guo, T. Haidegger, E. Hailey, P. Hansen, Z. Hao, K. Hari, K. Hayashi, J. Hawkins, S. Haworth, O. Hellig, S. D. Herrell, Z. Hong, A. Howe, J. Hu, Z. J. Hu, R. Jain, M. Rafiee Javazm, H. Ji, R. Ji, J. Ji, Z. Jiang, D. Jones, J. Jopling, B. Jordan, R. Ju, M. Kam, L. Kang, F. Kang, S. Kapuria, P. Kazanzides, S. Kiehler, E. Kilmer, J. W. B. Kim, P. Korzeniowski, C. Kuchi, N. Kumar, A. Kuntz, F. Lavagno, Y. C. Lee, H.-C. Lee, H. Li, Z. Li, X. Liang, X. Lin, J. Lin, C. Liu, F. Liu, P. Liu, Y.-h. Liu, W. Liuchen, E. Lukács, S. Mann, M. Mannas, B. Marinelli, S. Martyniak, F. Marzola, L. Mazza, X. Mei, M. C. Morais, L. Muratore, C. R. Narayanaswamy, M. Naskręt, D. Navarro-Alarcon, C. Neary, C. K. Ng, C. Nguan, D. Noonan, K. H. Oh, T. C. Olesch, A. M. Okamura, J. Opfermann, M. Pescio, D. X. V. Pham, T. Porras, H. Ren, A. Rodriguez Jimenez, F. Rodriguez y Baena, S. E. Salcudean, A. Sathya, P. Satish, L. Seenivasan, J. Shao, Y. Shen, Y. Sheng, L. X. Shi, Z. Soulé, S. Speidel, M. Su, J. Su, I. Sunmola, K. Takács, Y. Tang, P. Thornycroft, Y. Tian, J. Thompson, M. K. Turkcan, M. Unberath, P. Valdastri, C. Vives, Q. Vuong, M. Wagner, F. Wang, W. Wang, L. Wang, C.-P. Wang, G. Wang, J. Wang, E. Wang, Z. Wang, T. Watts, W. Wein, Y. Wu, Z. Wu, H. Wu, L. Wu, J. Y. Wu, J. Wu, V. Wu, K. Wu, M. Wójcikowski, Y. Xiao, N. Xiao, W. Xie, H. Yang, T. Yang, Y. Yang, M. Ye, R. S. Yeung, N. Yilmaz, C. H. Yin, M. Yip, R. Younis, C. Yu, S. N. Zaman, M. Zefran, H. Zhang, Y. Zhang, Y. Zhang, Y. Zhang, X. Zhang, Y. Zhang, J. Zhang, N. Zhong, P. Zhou, H. Zhou, X. Zuo, N. Navab, M. Azizian, S. D. Huver, and A. Krieger, "Open-H-embodiment: A large-scale dataset for enabling foundation models in medical robotics," 2026. [Online]. Available: https://open-h.github.io

[59] H. Xu, A. Weld, C. Xu, A. Roddan, J. a. Cartucho, M. A. Karaoglu, A. Ladikos, Y. Li, Y. Li, D. Shen, G. Lee, S. Park, J. Shin, L. Fothergill, D. Jones, P. Valdastri, D. Sarikaya, and S. Giannarou, "SurgRIPE challenge: Benchmark of surgical robot instrument pose estimation," *Medical Image Analysis*, vol. 105, p. 103674, 2025.

[60] Y. Li, F. Richter, J. Lu, E. K. Funk, R. K. Orosco, J. Zhu, and M. C. Yip, "SuPer: A surgical perception framework for endoscopic tissue manipulation with surgical robotics," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2294–2301, 2020.

[61] Z. Wu, A. Schmidt, R. Moore, H. Zhou, A. Banks, P. Kazanzides, and S. E. Salcudean, "SurgPose: a dataset for articulated robotic surgical tool pose estimation and tracking," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025, pp. 10 507–10 514.

[62] A. T. Aboukhadra, N. Robertini, J. Malik, A. Elhayek, G. Reis, and D. Stricker, "SurgeoNet: Realtime 3D pose estimation of articulated surgical instruments from stereo images using a synthetically-trained network," in *Pattern Recognition*. Springer Nature Switzerland, 2025, pp. 199–211.

[63] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

[64] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[65] J.-J. Jiang, X.-M. Wu, Y.-X. He, L.-A. Zeng, Y.-L. Wei, D. Zhang, and W.-S. Zheng, "Rethinking bimanual robotic manipulation: Learning with decoupled interaction framework," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025, pp. 12 427–12 437.

[66] K.-H. Oh, L. Borgioli, A. Mangano, V. Valle, M. Di Pangrazio, F. Toti, G. Pozza, L. Ambrosini, A. Ducas, M. Žefran, L. Chen, and P. C. Giulianotti, "Comprehensive robotic cholecystectomy dataset (CRCD): Integrating kinematics, pedal signals, and endoscopic videos," in *2024 International Symposium on Medical Robotics (ISMR)*, 2024, pp. 1–7.

[67] K.-H. Oh, L. Borgioli, A. Mangano, V. Valle, M. D. Pangrazio, F. Toti, G. Pozza, L. Ambrosini, A. Ducas, M. Žefran, L. Chen, and P. C. Giulianotti, "Expanded comprehensive robotic cholecystectomy dataset (CRCD)," 2024.

[68] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm6074, 2022.

[69] R. M. Murray, S. S. Sastry, and L. Zexiang, *A Mathematical Introduction to Robotic Manipulation*, 1st ed. USA: CRC Press, Inc., 1994.

[70] O. Özgüner, T. Shkurti, S. Huang, R. Hao, R. C. Jackson, W. S. Newman, and M. C. Çavuşoğlu, "Camera-robot calibration for the da Vinci robotic surgery system," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 4, pp. 2154–2161, 2020.

[71] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[72] D. Q. Huynh, "Metrics for 3D rotations: Comparison and analysis," *Journal of Mathematical Imaging and Vision*, vol. 35, no. 2, pp. 155–164, 2009.

[73] Z. Ugray, L. Lasdon, J. Plummer, F. Glover, J. Kelly, and R. Martí, "Scatter search and local NLP solvers: A multistart framework for global optimization," *INFORMS Journal on Computing*, vol. 19, no. 3, pp. 328–340, 2007.

[74] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[75] J. Bouguet, "MATLAB camera calibration toolbox," 2000.

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

[77] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, 2008.

[78] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International journal of computer vision*, vol. 47, pp. 7–42, 2002.

[79] P. H. D. Camilo and I. A. Cestari, "A design strategy to control an electrosurgery unit output," in *XXVII Brazilian Congress on Biomedical Engineering*, T. F. Bastos-Filho, E. M. de Oliveira Caldeira, and A. Frizera-Neto, Eds. Cham: Springer International Publishing, 2022, pp. 1003–1007.

[80] D. Kraft, *A Software Package for Sequential Quadratic Programming*, ser. Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt Köln: Forschungsbericht. Wiss. Berichtswesen d. DFVLR, 1988. [Online]. Available: https://books.google.com/books?id=4rKaGwAACAAJ

[81] I. Rivas-Blanco, C. J. Pérez-Del-Pulgar, I. García-Morales, and V. F. Muñoz, "A review on deep learning in minimally invasive surgery," *IEEE Access*, vol. 9, pp. 48 658–48 678, 2021.

[82] T. Rueckert, D. Rueckert, and C. Palm, "Methods and datasets for segmentation of minimally invasive surgical instruments in endoscopic images and videos: A review of the state of the art," *Computers in Biology and Medicine*, vol. 169, p. 107929, 2024.

[83] Y.-H. Su, K. Huang, and B. Hannaford, "Real-time vision-based surgical tool segmentation with robot kinematics prior," in *2018 International Symposium on Medical Robotics (ISMR)*, 2018, pp. 1–6.

[84] C. da Costa Rocha, N. Padoy, and B. Rosa, "Self-supervised surgical tool segmentation using kinematic information," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8720–8726.

[85] A. J. Hung, I. S. Jayaratna, B. Teruya, M. M. Desai, and I. S. Gill, "A comprehensive review of robotic surgery curriculum and training for residents, fellows, and postgraduate surgeons," *Asian journal of endoscopic surgery*, vol. 11, no. 3, pp. 249–257, 2018.

[86] A. Attanasio, B. Scaglioni, M. Leonetti *et al.*, "Autonomous tissue retraction in robotic assisted minimally invasive surgery—a feasibility study," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6528–6535, 2020.

[87] L. C. Garcia-Peraza-Herrera, L. Fidon, C. D'Ettorre, D. Stoyanov, T. Vercauteren, and S. Ourselin, "Image compositing for segmentation of surgical tools without manual annotations," *IEEE transactions on medical imaging*, vol. 40, no. 5, pp. 1450–1460, 2021.

[88] Z. Wang, B. Lu, Y. Long *et al.*, "AutoLaparo: A new dataset of integrated multi-tasks for image-guided surgical automation in laparoscopic hysterectomy," in *MICCAI*. Springer, 2022, pp. 486–496.

[89] A. J. Miao, S. Lin, J. Lu *et al.*, "Hemoset: The first blood segmentation dataset for automation of hemostasis management," in *2024 International Symposium on Medical Robotics (ISMR)*, 2024, pp. 1–7.

[90] Open Source Robotics Foundation, "Robot operating system," 2020. [Online]. Available: https://www.ros.org

[91] R. Smith, "An overview of the Tesseract OCR engine," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, 2007, pp. 629–633.

[92] ITU-T, "H.264: Advanced Video Coding for Generic Audiovisual Services," International Telecommunication Union, Recommendation ITU-T H.264 / ISO/IEC 14496-10, 2003. [Online]. Available: https://www.itu.int/rec/T-REC-H.264

[93] J. Brooks, "COCO Annotator," https://github.com/jsbroks/coco-annotator/, 2019.

[94] J. Heemskerk, W. G. van Gemert, J. de Vries, J. Greve, and N. D. Bouvy, "Learning curves of robot-assisted laparoscopic surgery compared with conventional laparoscopic surgery: an experimental study evaluating skill acquisition of robot-assisted laparoscopic tasks compared with conventional laparoscopic tasks in inexperienced users," *Surgical Laparoscopy Endoscopy & Percutaneous Techniques*, vol. 17, no. 3, pp. 171–174, 2007.

[95] R. Abaza, "The robotic surgery era and the role of laparoscopy training," *Therapeutic advances in urology*, vol. 1, no. 3, pp. 161–165, 2009.

[96] T. L. Hedrick, U. Phatak, T. A. Plerhoples, S. D. Holubar, P. H. Pucher, V. Tam, and C. Brown, "Evaluation of surgeon technical proficiency in robot-assisted surgery compared to traditional laparoscopy using machine learning models," *Surgical Endoscopy*, vol. 33, no. 12, pp. 4102–4110, 2019.

[97] G. Yang, A. D. Menhadji, R. E. Sadun, L. Romero, L.-M. Su, A. K. Tewari, and A. K. Hemal, "Training residents and surgeons for robot-assisted surgery: An expert consensus statement," *Urology*, vol. 140, pp. 4–10, 2020.

[98] S. Tonekaboni, D. Eytan, and A. Goldenberg, "Unsupervised representation learning for time series with temporal neighborhood coding," in *International Conference on Learning Representations*, 2021.

[99] I. Oguiza, "tsai — a state-of-the-art deep learning library for time series and sequential data," Github, 2023. [Online]. Available: https://github.com/timeseriesAI/tsai

[100] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, "A transformer-based framework for multivariate time series representation learning," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. ACM, 2021, pp. 2114–2124.

[101] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 6000–6010.

[102] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[103] T.-C. Lee, R. L. Kashyap, and C.-N. Chu, "Building skeleton models via 3-d medial surface axis thinning algorithms," *CVGIP: graphical models and image processing*, vol. 56, no. 6, pp. 462–478, 1994.

[104] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[105] S. Maneewongvatana and D. M. Mount, "Analysis of approximate nearest neighbor searching with clustered point sets," *CoRR*, vol. cs.CG/9901013, 1999. [Online]. Available: https://arxiv.org/abs/cs/9901013

[106] D. J. Rosenkrantz, R. E. Stearns, and P. M. Lewis, II, "An analysis of several heuristics for the traveling salesman problem," *SIAM journal on computing*, vol. 6, no. 3, pp. 563–581, 1977.

[107] S. Jaunoo, S. Mohandas, and L. Almond, "Postcholecystectomy syndrome (PCS)," *International Journal of Surgery*, vol. 8, no. 1, pp. 15–17, 2010.

[108] R. Bogdanova, P. Boulanger, and B. Zheng, "Depth perception of surgeons in minimally invasive surgery," *Surgical Innovation*, vol. 23, no. 5, pp. 515–524, 2016.

[109] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 5738–5746.

[110] A. Kendall and R. Cipolla, "Geometric loss functions for camera pose regression with deep learning," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6555–6564.

[111] SAM2 GUI Annotator, "SAM2 GUI Annotator," https://github.com/koh43/sam2-gui-annotator.

[112] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=Bkg6RiCqY7

[113] ——, "SGDR: Stochastic gradient descent with warm restarts," in *International Conference on Learning Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=Skq89Scxx

[114] R. Wightman, "PyTorch Image Models," https://github.com/huggingface/pytorch-image-models, 2019.

[115] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *2016 Fourth International Conference on 3D Vision (3DV)*, 2016, pp. 565–571.

[116] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[117] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018, pp. 839–847.

[118] C. Molnár, T. D. Nagy, R. N. Elek, and T. Haidegger, "Visual servoing-based camera control for the da Vinci surgical system," in *2020 IEEE 18th International Symposium on Intelligent Systems and Informatics (SISY)*, 2020, pp. 107–112.

Vita

KI-HWAN OH

EDUCATION

Ph.D. in Electrical and Computer Engineering, University of Illinois Chicago, USA *09.2019 - 2026*

B.S. in Electronic and Electrical Engineering, Sung Kyun Kwan University, South Korea *03.2012 - 02.2018*

EXPERIENCE

Digital Twin Development Intern, Retina Robotics *05.2025 - 08.2025*

Chicago, IL

Developed a real-time warehouse digital twin in Godot with synchronized visualization of tracked items.

Integrated WAM™ with ROS2 to enable real-time multi-client odometry synchronization.

Product Engineering Systems Analyst Intern, Intuitive Surgical, Inc. *05.2024 - 08.2024*

Sunnyvale, CA

Conducted root-cause analysis of encoder signal anomalies across robotic arm subsystems and proposed corrective strategies.

Evaluated and benchmarked alternative sensor configurations to improve arm performance reliability.

Research Assistant, Surgical Innovation and Training Lab *06.2021 - 2026*

Chicago, IL

Released the Comprehensive Robotic Cholecystectomy Dataset and developed vision-based methods for autonomous robotic dissection from endoscopic images.

Developed custom forward and inverse kinematics for the da Vinci arms using fiducial markers.

Created a prototype platform to control the da Vinci arms with sensory gloves.

Research Assistant, UIC Robotics Lab *09.2019 - 2026*

Chicago, IL

Investigated human-human collaborative manipulation, identifying low-level patterns for negotiation and motion execution.

Developed intent recognition methods for collaborative manipulation using force and kinematic signals.
Demonstrated robust intent recognition for human-robot collaboration in physical interaction control.

Teaching Assistant, University of Illinois Chicago *08.2020 - 05.2022*
ECE 451: Principles of Modern Control
ECE 434: Multimedia Systems
ECE 452: Robotics, Algorithms, and Control
ECE 350: Principles of Auto Control

PUBLICATIONS

Journal Publications

K.-H. Oh, L. Borgioli, A. Mangano, V. Valle, M. Di Pangrazio, F. Toti, G. Pozza, L. Ambrosini, A. Ducas, M. Žefran, L. Chen, and P. C. Giulianotti. Expanded Comprehensive Robotic Cholecystectomy Dataset (CRCD). *Journal of Medical Robotics Research*, 2025.

Conference Publications

K.-H. Oh, L. Borgioli, M. Žefran, V. Valle, and P. C. Giulianotti. Autonomous Dissection in Robotic Cholecystectomy. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025.

L. Borgioli, K.-H. Oh, A. Mangano, A. Ducas, L. Ambrosini, F. Pinto, P. A. Lopez, J. Cassiani, M. Žefran, L. Chen, and P. C. Giulianotti. Sensory Glove-Based Surgical Robot User Interface. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

K.-H. Oh, L. Borgioli, M. Žefran, L. Chen, and P. C. Giulianotti. A Framework for Automated Dissection Along Tissue Boundary. In *2024 10th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*, pages 1427–1433, 2024.

K.-H. Oh, L. Borgioli, A. Mangano, V. Valle, M. Di Pangrazio, F. Toti, G. Pozza, L. Ambrosini, A. Ducas, M. Žefran, L. Chen, and P. C. Giulianotti. Comprehensive Robotic Cholecystectomy Dataset (CRCD): Integrating Kinematics, Pedal Signals, and Endoscopic Videos. In *2024 International Symposium on Medical Robotics (ISMR)*, pages 1–7, 2024.

Z. Rysbek, K.-H. Oh, B. Abbasi, M. Žefran, and B. Di Eugenio. Physical Action Primitives for Collaborative Decision Making in Human-Human Manipulation. In *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*, pages 1319–1325, 2021.

Z. Rysbek, K.-H. Oh, and M. Žefran. Recognizing Intent in Collaborative Manipulation. In *International Conference on Multimodal Interaction*, pages 498–506, 2023.

Other Publications

A. M. Shervedani, K.-H. Oh, B. Abbasi, N. Monaikul, Z. Rysbek, B. Di Eugenio, and M. Žefran. Evaluating Multimodal Interaction of Robots Assisting Older Adults. arXiv:2212.10425, 2022.

AWARDS Dean's List, Sung Kyun Kwan University *2015 & 2017*

MEMBERSHIPS IEEE Student Member