POLITECNICO DI TORINO

Master's Degree in Biomedical Engineering

Master's Degree Thesis

Vision-Based Approaches for Surgical Tool Pose Estimation in Minimally Invasive Robotic Surgery

Supervisors
Prof. Kristen Mariko MEIBURGER
Eng. Francesco MARZOLA
Prof. Alberto AREZZO

Candidate
Edoardo GENNARO

Academic year 2024/2025

Acknowledgments - English Version

First and foremost, I would like to thank the MITIC laboratory at Molinette Hospital and everyone involved for welcoming me and supporting me throughout my thesis journey. A heartfelt thanks goes to my supervisors, Professor Kristen M. Meiburger, for her availability, expertise, and guidance, and Professor Alberto Arezzo for creating a cohesive team and for his ability to always motivate each of us to give our best. A special thanks goes to my co-supervisor, Engineer Francesco Marzola, for his dedicated support, for sharing his experience with me, and for teaching me that the quality of an engineer is also measured by the ability to question oneself. I am also grateful to all collaborators who contributed to this work through their suggestions and discussions.

A thought of deep gratitude goes to my family for their unconditional support, trust, and encouragement that I have always received. Thanks to my mother, who has always been cheering for me and who taught me never to stop in front of obstacles, just like when, as a child, I ignored the flags in the water and kept racing. Thanks to my father for always being proud of me, even in moments when I myself struggled to be, and for his always kind and encouraging words. A sincere thanks to my sister Daniela for her advice and her ability to rejoice in my happiness; you are a great source of inspiration to me. Thanks to my brother Andrea, my great friend, who never hesitated to put me first and offer his help: I am proud to be so much like you.

To my girlfriend, Giorgia, goes my sincerest thanks for standing by me even in the toughest moments, with patience, affection, and understanding. You were by my side during every exam session, in disappointments and joys, in waits and uncertainties. Thank you for always being my number one supporter and for backing me in crucial moments. Thank you also for making me feel like a point of reference, for believing in me, and for seeing me as a great engineer from the very beginning.

Thanks to my friends and university mates: Emanuele, Claudia, Enrica, Giuseppe. We shared all the joys and hardships that Politecnico gave us. We learned to work as a team (definitely worth mentioning on the CV), to listen to each other, and to lend a helping hand. Special thanks to Andrea, who shared this journey with me and always managed to bring a smile back to my face, even in moments when I preferred silence.

Thanks to Federico, for the laughter, the closeness, and true friendship.

Thanks to Stella, who gave me glimmers of light, listened to me, supported me, and always reminded me of my worth.

Thanks to Chiara, Berl, Marta, and Agnese for being my friends since before I even knew who I was, and for never missing a chance to share a laugh together.
To Salvo, for the listening, understanding, and stories that helped me disconnect in the most intense moments.
To Alessandra, "Gary," who believed in me from day one: thank you for lightening difficult moments and for seeing me for who I am even when I couldn't see it myself. A few years ago, we dreamed of working together, and today I still think it would be a great privilege, both for your brilliant mind and especially for your beautiful heart. Thanks to Robi, for every exam we faced together, for constant updates on our theses, for attentive listening, advice, and all the new perspectives you offered me. Thank you for being as passionate about the subjects we studied as I am. I hope our professional paths will continue to cross.
Finally, thanks to all my friends, even those who have taken different paths: each of you has played an important role in my personal growth and academic journey. Thanks to myself, for every time I fell, went down to the deepest basements, reached the center of the earth, and always found a way to get back up and rise even higher. Thanks to everyone who has been there. To all of you, I extend my most sincere and deepest gratitude.

Acknowledgments - Italian Version

Desidero innanzitutto ringraziare il laboratorio MITIC dell'Ospedale Molinette e tutte le persone che ne fanno parte per avermi accolto e accompagnato durante il mio percorso di tesi. Un sentito grazie va alla mia relatrice, professoressa Kristen M. Meiburger, per la sua disponibilità, competenza e guida. Ringrazio il professor Alberto Arezzo per aver creato un gruppo di lavoro affiatato e per la sua capacità di motivare sempre ciascuno di noi a dare il massimo. Un ringraziamento speciale va al mio co-relatore, ingegnere Francesco Marzola, per avermi seguito con dedizione, per aver condiviso con me la sua esperienza e per avermi insegnato che la qualità di un ingegnere si misura anche nella capacità di mettersi in discussione. Ringrazio inoltre tutti i collaboratori che, con suggerimenti e confronti, hanno contribuito alla crescita di questo lavoro.

Un pensiero di profonda gratitudine va alla mia famiglia, per il sostegno incondizionato, la fiducia e l'incoraggiamento che non mi hanno mai fatto mancare. Un grazie a mia mamma, che tifa per me da sempre e che mi ha insegnato a non fermarmi di fronte agli ostacoli, proprio come quando, da piccolo, ignoravo le bandierine in acqua e continuavo la gara. Grazie a mio papà, per essere sempre stato orgoglioso di me, anche nei momenti in cui io stesso facevo fatica a esserlo, e per le sue parole sempre dolci e incoraggianti. Un grazie sentito a mia sorella Daniela, per i consigli e per la sua capacità di gioire delle mie felicità; sei per me una grande fonte di ispirazione. Grazie a mio fratello Andrea, mio grande amico, che non ha mai esitato a mettermi al primo posto e a offrirmi il suo aiuto: sono fiero di essere così simile a te.

Alla mia ragazza, Giorgia, va il mio più sincero ringraziamento per avermi accompagnato anche nei momenti più difficili, con pazienza, affetto e comprensione. Sei stata al mio fianco durante ogni sessione d'esame, nelle delusioni e nelle gioie, nelle attese e nelle incertezze. Grazie per essere sempre la mia prima tifosa e per avermi sostenuto nei momenti decisivi. Grazie anche per avermi fatto sentire un punto di riferimento, per aver creduto in me e avermi visto come un grande ingegnere sin dall'inizio.

Un grazie ai miei amici e compagni di università: Emanuele, Claudia, Enrica, Giuseppe. Abbiamo condiviso tutte le gioie e i dolori che il Politecnico ci ha dato. Abbiamo imparato a lavorare in team (da segnare sul CV), ad ascoltarci e a dare una mano in più. Un grazie particolare ad Andrea, che ha condiviso con me questo percorso e che ha sempre saputo farmi tornare il sorriso anche nei momenti in cui avrei preferito restare in silenzio.

Grazie a Federico, per le risate, la vicinanza e la vera amicizia.

Grazie a Stella, che mi ha dato degli spiragli di luce, mi ha ascoltato e sostenuto e

mi ha sempre ricordato il mio valore.
Grazie a Chiara, Berl, Marta e Agnese per essere mie amiche da quando io non ero ancora chi sono, e per non perdere un'occasione per fare una risata assieme. A Salvo, per l'ascolto, la comprensione e le storie con cui sei riuscito a farmi staccare la mente nei momenti più intensi.
Ad Alessandra, "Gary", che ha creduto in me fin dal primo giorno: grazie per aver reso più leggeri momenti difficili, per avermi visto per ciò che sono anche quando io non riuscivo a farlo. Qualche anno fa sognavamo di lavorare insieme, e oggi continuo a pensare che sarebbe un grande privilegio, per la tua mente brillante, ma soprattutto per il tuo bel cuore.
Grazie a Robi, per ogni esame affrontato insieme, per gli aggiornamenti costanti sulle nostre tesi, per l'ascolto attento, i consigli e tutte le prospettive nuove che hai saputo offrirmi. Grazie per essere appassionato alle materie che abbiamo studiato, tanto quanto lo sono io. Spero che il nostro futuro lavorativo possa continuare ad intrecciarsi.
Un ringraziamento infine va a tutti i miei amici, anche a quelli che hanno preso strade diverse: ognuno di voi ha avuto un ruolo importante nella mia crescita personale e nel mio percorso accademico. Un grazie a me stesso, per tutte le volte che sono caduto, andato nei seminterrati più profondi, arrivato nel centro della terra e ho sempre trovato il modo rialzarmi e andare ancora più su. Grazie a chiunque c'è stato. A tutti voi, va la mia più sincera e profonda gratitudine.

Declaration

I hereby declare that the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Turin, 10/07/2025

Abstract

In recent years, Robotic-Assisted Minimally Invasive Surgery (RMIS) has led to significant improvements in surgical precision and patient safety. Accurate 6D pose estimation of surgical tools is a fundamental enabler for several critical capabilities that enhance both human-in-the-loop and semi-autonomous interventions. The da Vinci Research Kit (dVRK) offers an open-source platform to study these tasks in both simulated and real environments. As part of an ongoing research, this thesis focuses on pose estimation as a key component in the automation of tasks such as suturing using the dVRK. This work investigates and compares two strategies for 6D pose estimation of dVRK instruments: a Marker-based approach and a Model-based, markerless solution based on Deep Learning. The Marker-based method uses a printable cylindrical marker and the EPnP algorithm to compute 6D pose from 2D–3D correspondences. It was applied in both simulated and real scenarios, delivering robust and accurate results. In addition to serving as a stand-alone solution, it also provided ground truth (GT) annotations to evaluate the learning-based approach. The Marker-Less method is *FoundationPose*, a framework that compares a cropped RGB image of the tool with its CAD model to regress its 6D pose. It uses two networks: a pose refinement module that generates pose candidates from image-model alignment, and a pose selection module that ranks and selects the best hypothesis. In this study three dataset were used: two simulated ones (one per tool—Needle Driver and Cadiere Forceps) and a real world one.

In simulation, 150 frames per tool were generated using Unity, with the application of realistic textures and anatomical backgrounds derived from real surgical scenarios. GT was obtained via the Marker-based method. Unity also provided *FoundationPose* requirements, including segmentation masks, absolute depth maps, and camera intrinsics.

The real-world dataset consists of 150 frames of the Needle Driver, acquired in a laboratory setting, using the dVRK system. GT was again computed using the Marker-based approach, with a printed marker mounted along the tool's shaft. Since automated generation of segmentation and depth data is not available in this setting, masks were manually created using Roboflow, while depth maps were generated using Depth Anything and then converted into absolute values. Results show that the Marker-based method provided consistent and accurate pose estimates in both domains, serving as a solid reference throughout the study. *FoundationPose* showed promising performance in simulation, with rotational estimates corresponding to cosine similarity values between 0.83–0.87 ($\sim 9 - 21$ rotational errors), and positional errors of 1–1.5 cm on the X and Y axes. Depth estimation was less accurate, with

errors of 6–7 cm. In real settings, performance declined moderately with a cosine similarity of 0.78–0.81 ($\sim 21 - 27$ rotational errors), positional errors increasing to 2–3 cm and depth errors of 16 cm, likely due to the approximated depth data. In conclusion, the Marker-based method proved to be an effective solution for both pose estimation and GT annotation. *FoundationPose* showed potential for markerLess estimation but also revealed limitations, especially in real-world use. The small size and fine structure of the instruments increase sensitivity to error, while symmetric geometries can lead to rotational ambiguities. Nevertheless, targeted fine-tuning on domain-specific data, and integration with geometric approaches or kinematic data could improve performance.

Contents

1 Introduction **11**

1.1 Minimally Invasive Surgery (MIS) . . . 11

1.1.1 Advantages and disadvantages of MIS . . . 12

1.2 Robotic assisted minimally invasive surgery (RMIS) . . . 14

1.2.1 Role of Robotics in MIS . . . 14

1.2.2 Evolution of robotics for MIS . . . 17

1.2.3 The Da Vinci Research Kit (dVRK) system . . . 18

1.3 Pose Estimation of dVRK tools . . . 22

1.3.1 Current applications of tools tracking and pose estimation in the surgery context . . . 22

1.3.2 Future clinical integrations of Pose Estimation of dVRK tools 23

1.4 Overview of existing approaches for Pose Estimation . . . 24

2 State of The Art **25**

2.1 Marker-based approaches for Pose Estimation . . . 25

2.1.1 Fiducial Markers . . . 26

2.1.2 Custom Markers . . . 29

2.2 Markerless Approaches for Pose Estimation . . . 34

2.2.1 Direct Regression . . . 35

2.2.2 Indirect Methods . . . 36

2.2.3 Dense Correspondence Methods . . . 37

3 Materials and Methods **38**

3.1 Marker-based method . . . 40

3.1.1 Marker Design . . . 40

3.1.2 Methodology . . . 41

3.1.3 EPnP Solver . . . 43

3.2 Model-based method: *FoundationPose* . . . 47

3.2.1 What is a Foundation model? . . . 47

3.2.2 Architecture . . . 48

3.2.3 Methodology . . . 52

3.2.4 Dataset generation in simulation . . . 53

3.2.5 Generation of a simulated surgical scene with Unity . . . 53

3.2.6 Ground Truth Generation . . . 56

3.3 Dataset generation in the Real World . . . 58

3.3.1 DVRK Camera Calibration . . . 59

3.3.2 Data Acquisition 60
3.4 Metrics 65

4 Results **66**
4.1 Pose Estimation in Simulation 66
4.2 Pose Estimation in Real Environment 70
4.3 Discussion 71
4.3.1 Result Analysis 71
4.3.2 Current Limitations 72
4.3.3 Potential Improvements 73

5 Conclusion **75**

1 Introduction

Minimally Invasive Surgery (MIS) has revolutionised the surgical field and has become the standard in many countries. An overview of MIS and how robotic assistance can improve it is provided in this section. It examines how robotics can enhance surgical accuracy and patient outcomes, traces the evolution of robotic systems in MIS over time, and presents the Da Vinci Research Kit (dVRK), a widely used research platform that has been used in this study. The significance of the dVRK tool's pose estimation for automation, safety, and surgical guidance is then highlighted in this section. It gives a summary of current pose estimation methods and discusses current and future surgical applications.

1.1 Minimally Invasive Surgery (MIS)

Minimally Invasive Surgery (MIS) is a modern approach that focuses on reducing the physical trauma associated with traditional open surgery, thanks to smaller incisions with specialized instruments. Laparoscopy was one of the first MIS procedures. Essential components are the laparoscope, with a camera that provides real-time visualization of the surgical site on a monitor, and other precision instruments.

Figure 1: Illustration of Minimally Invasive Surgery (MIS).

The starting point of MIS can be traced back to the '60s with the Hopkins rod-lens endoscope. This laid the groundwork for several developments in the surgery field, in particular the first laparoscopic cholecystectomy, performed by Erich Mühe in 1985. This is the formal beginning of the MIS era: since then, MIS techniques have been widely adopted and have become the standard approach over traditional open surgery in a wide range of procedures. In general surgery, it is routinely employed for procedures such as cholecystectomies and hernia repairs. In gynecology, it plays a central role in operations like hysterectomies and ovarian cyst removals. The field of urology benefits from MIS in interventions such as prostatectomies and nephrectomies, while orthopedic surgeons commonly use arthroscopy for joint-related treatments. Even complex cardiothoracic operations, including valve repairs and certain coronary interventions, have increasingly adopted minimally invasive approaches. The versatility and patient-centered advantages of MIS continue to drive its integration into modern surgical practice, reshaping operative standards across disciplines. [1] One important development in the ongoing progress of MIS has been the introduction of robotic surgical systems. Technologies such as the da Vinci Surgical System have revolutionized the field by offering surgeons enhanced visualization, increased dexterity through articulated instruments, and improved precision in confined anatomical spaces. These systems mitigate many of the limitations of conventional laparoscopy, such as restricted range of motion and poor ergonomics, allowing for greater control and reducing surgeon fatigue during long or complex procedures. Robotic-assisted surgery has now become a standard option in many domains, including robotic prostatectomies in urology, robotic hysterectomies in gynecology, and increasingly in colorectal, thoracic, and cardiac surgeries. As robotic technologies continue to evolve, their role in enhancing the safety, efficiency, and personalization of MIS is expected to grow. Together, MIS and robotic systems represent a transformative shift in surgical practice, reshaping how procedures are performed and improving patient outcomes.

1.1.1 Advantages and disadvantages of MIS

The main goal of MIS is to reduce both the number and size of incisions, thereby minimizing the damage to soft tissues caused by large incisions. This approach offers numerous advantages, including:

- **Less pain and blood loss:** analyses in colorectal cancer show laparoscopic patients bleed on average $\sim 90\,\text{mL}$ less than open cases [2]. This reduced trauma to tissues results in less postoperative pain, which often translates to a reduced

need for opioids.

- **Shorter hospital stay and faster recovery:** smaller incisions heal faster and cause less stress on the body. This means that MIS patients tend to leave the hospital sooner and return to normal activities more quickly [2, 3].

- **Fewer complications:** by minimizing wound size, MIS dramatically lowers wound-related complications. Reviews report fewer wound complications and a reduced risk of surgical-site infection in laparoscopy compared to open approaches [3]. Limiting tissue trauma also reduces formation of adhesions and the risk of incisional hernias. Laparoscopic patients tend to have lower rates of pneumonia and thromboembolism, likely due to earlier postoperative ambulation.

- **Enhanced quality of life and satisfaction:** less pain and scarring, coupled with a quicker return to daily activities, translates into a better patient experience.

- **Improved cosmetic outcome:** MIS incisions are much smaller (often a few millimeters instead of many centimeters). In one prospective comparison of hernia surgery, patients undergoing laparoscopic repair reported a superior cosmetic satisfaction versus open surgery [4].

However, MIS also presents some challenges and limitations that must be considered:

- **Challenging learning process:** minimally invasive techniques, particularly laparoscopic and robotic procedures, require new technical skills. Surgeons must learn new techniques to operate through small incisions while viewing the procedure on a screen, which requires specific training and practice.

- **Limited tactile feedback:** unlike open surgery, where surgeons can directly feel tissue texture and resistance, minimally invasive procedures provide only limited tactile feedback through instruments. Researchers are working on developing tools that can sense and give feedback about the force applied during surgery, to help solve this problem.

- **Technical constraints:** visual limitations can restrict the surgeon's dexterity [3]. Robotic systems limit some of these issues with their articulated instruments and 3D vision.

1.2 Robotic assisted minimally invasive surgery (RMIS)

1.2.1 Role of Robotics in MIS

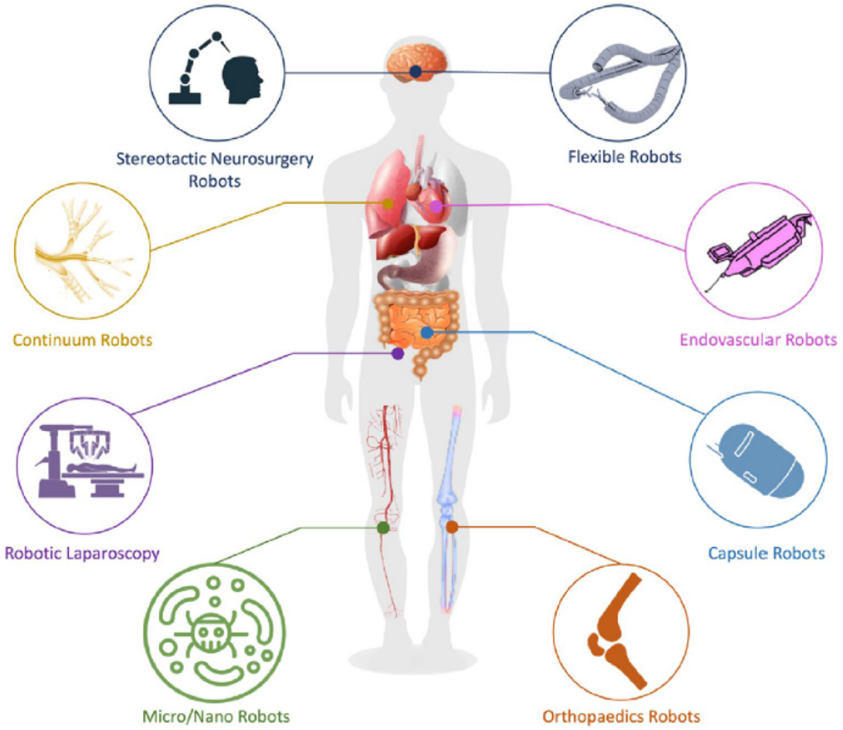


Figure 2: Examples of robotic MIS applications. [4]

The field of surgery is under constant evolution. Surgeons have steadily pursued innovative methods to improve patient outcomes, focusing on making surgeries safer, less invasive, and more efficient. This pursuit has been ongoing for many generations, with early breakthroughs occurring in the 1860s with Lister's seminal work on antiseptic surgery. In traditional MIS, visual-motor alignment is obtained by a two-dimensional display and the use of rigid instruments, which limit haptics, dexterity, and coordination [4, 5].

Computer-enhanced surgery has already made significant progress and is set to evolve rapidly in the future. Computers play a significant role by processing data, offering image guidance, and providing real-time access to expert advice in different situations. Additionally, they actively participate in surgical procedures through robotic systems. The introduction of robotic-assisted surgery in the late 90s marked

a significant milestone in the history of MIS. The surgeon skills combined with the robot's precision can address many challenges, representing a significant evolution in surgical practice [5].

3D visualization, available in many robotic platforms, has enhanced depth perception and spatial awareness, making it easier to perform precise manoeuvres. Studies show that task performance consistently exceeds the ones achieved with 2D visualization, regardless of the level of surgeon experience, or the complexity of the task [6]. Robotics also enhances dexterity, as robotic arms offer a broader range of motion than human hands, enabling more intricate movements even in confined anatomical spaces. Moreover, robotic systems help reduce tremors, ensuring steadier and more precise movements.

To overcome the lack of haptic feedback, one of the primary drawbacks of traditional MIS, innovative solutions are being developed. For instance, Titan Medical Inc. (Toronto, Ontario, Canada) is integrating force sensing directly at the tips of surgical instruments. Another notable advancement in the field of telesurgery is the RAVEN system (Fig. 3), developed by the Biorobotics Laboratory at the University of Washington (Seattle, WA, USA) [7]. This platform features lightweight, cable-driven instrument arms with seven degrees of freedom (DoF), which can be mounted directly onto the operative table. While the arms are rigid, the imaging probe is designed to be flexible, enhancing visibility during procedures. Multiple copies of RAVEN are being used to create a research network internationally to investigate new approaches to robotically assisted MIS.

Figure 3: On the left: close-up photo of two RAVEN mechanisms. On the right: surgeons manipulating conventional RAVEN tools. [7]

In traditional laparoscopy, surgeons must deal with the fulcrum effect, where hand movements result in the opposite motion of the instrument tip due to the pivot at the entry point. This reversed control can be unintuitive and difficult to master. Robotic systems eliminate this issue by translating the surgeon's hand movements into direct, mirrored motions of the instruments, thanks to a master-slave configuration that replaces the mechanical setup, allowing instrument movements to directly match the surgeon's intentions for more intuitive control. Advances in imaging and digital vision technologies, such as high-definition stereoscopic displays and augmented reality, further enhance the functionality of the surgeon's console. The most widely recognized master-slave robotic system is the Da Vinci Surgical System (Fig. 4) by Intuitive Surgical Inc. (Sunnyvale, CA, USA) [8]. In addition to the benefits mentioned, its primary advantage lies in restoring wrist articulation, which is lost in conventional laparoscopy, providing surgeons with greater flexibility and precision [5].

Figure 4: Refiguration of the Da Vinci System. On the left a surgeon using the surgeon's console of the system, on the right the patient-side and the vision cart.

1.2.2 Evolution of robotics for MIS

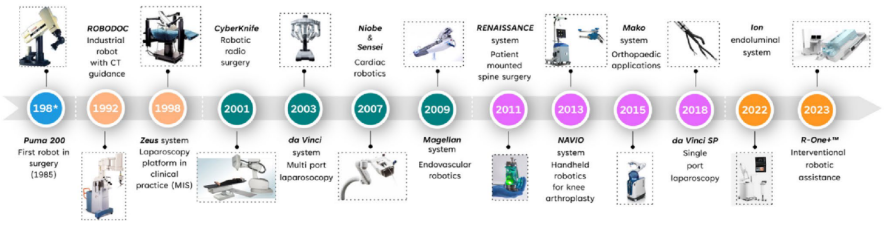


Figure 5: Timeline of Minimally Invasive Robotic Surgery. [1]

In 1985, the PUMA 200, a conventional industrial robot developed by Unimation in Danbury, CT, was experimentally utilized for a needle insertion procedure, marking the first recorded use of a robotic system in surgery [9]. Since then, an ever-increasing number of platforms, developed by both commercial and research entities, have been created and successfully applied across a growing number of surgical fields, including neurosurgery, ear-nose-throat (ENT), orthopedics, laparoscopy, and endoluminal procedures. The 80s saw the development of the first generation of surgical robots, designed primarily for procedures in neurosurgery and orthopedics. Two examples are the Neuromate system (Renishaw, UK) was used for procedures like neuro-endoscopy, biopsies, electrode placement [10] and Robodoc (Curexo Technology, USA), created to enhance hip replacement [11].

The 90s saw a second generation of surgical robots designed for MIS. These systems featured multiple rigid arms controlled by the surgeon through a remote console. This approach was first introduced with the Zeus platform (Computer Motion, USA) and later advanced with the da Vinci system after Computer Motion merged with Intuitive Surgical in 2003. Since then, various versions of the da Vinci system have been developed. These include the daVinci Xi, which became the most widely used multiport robotic surgery system in the world, and the da Vinci SP, a single-port system that uses three flexible, multi-jointed instruments along with a fully articulated endoscope.

In the early 2000s, the innovations included compact and steerable tools like robotic catheters, endoscopes, and snake-like robots, which are flexible enough to reach and operate in tight areas of the human body that rigid laparoscopic instruments couldn't access before. These devices have been used in procedures such as endovascular interventions, abdominal surgeries, and bronchoscopies. However, one major drawback of these flexible robots is their limited ability to apply strong forces to

tissues due to their structural design. To address this issue, concentric tube robots (also known as active cannulas) were introduced around 2005 through the work of researchers such as Furusho et al. [12], Sears and Dupont [13], and Webster [14]. These robots are composed of nested, pre-curved elastic tubes (often made of materials like Nitinol) that bend and deform when rotated and moved relative to each other, providing greater stiffness and strength [5].
During the 2000s, a novel generation of robots emerged, focusing on untethered microrobots designed to improve the intraluminal navigation in the human body, enhancing diagnostics and treatment. One notable example is capsule endoscopy, which enables fewer risks, discomfort, and pain than traditional flexible endoscopy. The success of capsule endoscopy led to its commercialization, with systems like the ENDOCAPSULE 10 (Olympus, Tokyo, Japan), which offers advanced imaging and 3D tracking capabilities. Beyond this, various research groups have explored alternative designs and movement techniques for these devices, including legged systems [15] and worm-like locomotion [16].
Currently, untethered micro-/nano-surgical devices are emerging [17, 18]. Current surgical platforms in clinical use still face significant challenges, such as the size and rigidity of mechanical components, which limit their ability to access and treat small, early-stage lesions or areas of the human body that were previously unreachable. Advancements in miniaturized, flexible robots, just a few micrometers in size, capable of navigating throughout the human body, hold the potential to enable highly precise and localized (at the cellular level) treatments.

1.2.3 The Da Vinci Research Kit (dVRK) system

To help overcome the main limitations that have historically made minimally invasive techniques difficult to apply in complex surgeries, Intuitive Surgical Inc. (Mountain View, CA) developed the da Vinci System. The da Vinci System combines high-resolution stereo visualization with a direct hand-to-instrument control interface, allowing the surgeon's movements to be mirrored precisely at the instrument tips inside the patient's body [19]. This alignment between the surgeon's hands and the visual field restores natural hand-eye coordination, thanks to an optical system that overlays the 3D image of the surgical site on top of the surgeon's hands. The system also adapts the instrument movements to match the camera's perspective, enhancing spatial coherence and giving the surgeon the sensation of working directly within the operative field. As mentioned before, it also overcomes the limitations of

traditional laparoscopy by adding a wrist mechanism with three degrees of freedom at each instrument tip, allowing for a total of seven degrees of motion (translation, rotation, and grip) which enhances dexterity and precision [19]. In addition, the system filters out hand tremors and allows for motion scaling, so that large movements at the console can be translated into smaller, more controlled movements at the surgical site. For example, a 3:1 scaling ratio would convert a 3 cm movement by the surgeon into a 1 cm movement by the robotic instrument. When combined with high-resolution close-up views, this feature makes delicate procedures significantly easier to perform [20]. The da Vinci Surgical System is structured around a master–slave configuration, composed primarily of two interconnected components: the surgeon's console (master) and the patient-side robotic system (slave). There is also a vision cart that displays the surgical scene in real time. All the components are shown in Fig. 6. At the surgeon's console, the operator is seated in an ergonomic position and interacts with the system through two hand controllers known as masters. These are serial-link manipulators that capture the surgeon's hand, wrist, and finger movements, including grip commands. At the console, two medical-grade Cathode Ray Tube (CRT) monitors, each dedicated to one eye, create a stereo image with depth perception, shown in a display. Additionally, the user interface has control buttons and foot pedals that let the surgeon change modes, move the endoscope and the instruments, change focus, and do other tasks all without leaving the console. Supporting this functionality is a custom-designed electronic controller, built for speed, reliability, and safety. The patient-side cart houses the slave robotic manipulators, which are responsible for translating the surgeon's commands into movements at the surgical site. This system typically includes three robotic arms: two PSM (Patient Side Manipulators) for the surgical instruments and one for the stereo endoscope ECM (Endoscopic Camera Manipulator), which provides real-time visual feedback. The manipulators are positioned via multi-link arms mounted to a stable base, allowing flexible setup around the patient. Each surgical instrument offers four active degrees of freedom, including grip actuation, and attaches to the robot via quick-connect mechanisms. These instruments are fully sterilizable using FDA- and EU-approved protocols. To ensure immersive visualization, the system uses a high-resolution stereo endoscope with dual optical channels, each equipped with a three-chip Charge-Coupled Device (CCD) camera, enabling accurate depth perception and clear imaging inside confined anatomical spaces. In some versions of the surgical systems, only a monocular camera is used, which provides two-dimensional imaging and consequently reduces depth perception, making precise spatial perception more challenging for the surgeon [8].

Figure 6: DVRK principal components.

In 2014, the da Vinci Research Kit (dVRK) was formally introduced as an open-source platform to facilitate advanced research in the field of robotic-assisted surgery [21, 22]. The publication presented the development of an open-source mechatronics system composed of modular hardware, custom electronics, firmware, and software (Fig. 7), designed to interface with components of the first-generation da Vinci Surgical System. The electronic system utilizes a Field-Programmable Gate Array (FPGA) to support a centralized computation framework with a distributed input/output (I/O) architecture [21].

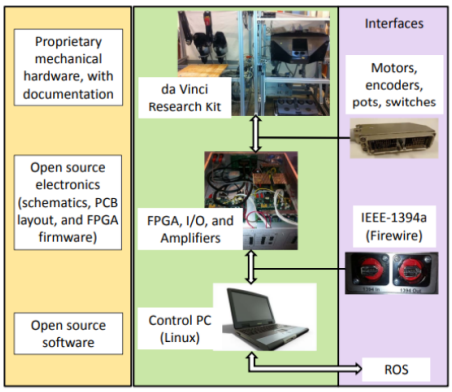


Figure 7: Overview of telerobotic research platform. [8]

The authors designed a series of robotic tasks, including tissue grasping, palpation, and incision, which were performed by expert surgeons, medical residents, and non-surgeons, both with and without haptic feedback. Although experienced robotic surgeons are often thought to compensate for the absence of tactile sensation through enhanced visual cues and extensive training, the study showed that force reflection made a statistically significant difference, enhancing surgical outcomes, particularly in tasks that place a high cognitive demand on the surgeon. Another

study [23] confirmed the importance of haptic feedback by using a four-channel teleoperation system with disturbance observers and sensorless force estimation enhanced through learning-based dynamic compensation. The results showed that even without physical sensors, estimated haptic feedback can improve tissue palpation accuracy and significantly reduce unintended interaction forces during manipulation. The dVRK is also widely used for image-guided and simulation research. For example, in robotic-assisted partial nephrectomy (RAPN)—a kidney-sparing procedure for renal cancer—ultrasound (US) imaging is employed intraoperatively to delineate tumor margins and locate blood vessels. Since then, in the standard da Vinci setup, the US probe has always been inserted through an auxiliary port and manipulated by a robotic tool over the kidney surface. However, since the US images are displayed separately from the main surgical view, the surgeon is required to mentally integrate the ultrasound and surgical views, which adds cognitive load and increases the complexity of the procedure. The dVRK has been used as a research platform to address these limitations. The dVRK can be equipped with stereo endoscopes and advanced imaging tools, such as 3D US probes, enabling research into augmented-reality overlays and real-time image-guided interventions that enhance spatial integration and procedural efficiency. For instance, a robotic rail was implemented to automate intraoperative US acquisition in a RAPN study [24]. The system enabled semi-autonomous placement and scanning of the US probe using the dVRK setup, which included a stereo laparoscope to enable co-registration of US and surgical views. Common experimental setups include bench-top rigs with synthetic or animal tissue phantoms (porcine organs, biopsy simulators, etc.) and simulation models where virtual instruments mimic dVRK motion. In these setups, researchers simulate laparoscopic procedures such as suturing vessels, manipulating soft tissue, or drilling bone. In summary, the dVRK serves as a versatile platform for research and clinical simulation: it supports studies of surgical dexterity (skills and training), development of semi-autonomous surgical subtasks, experimentation with haptic interfaces, and integration of imaging into robotic workflow. Regarding imaging, several datasets have been developed using the dVRK to support research in robotic surgery [25, 26]. The use of large datasets is essential in surgical robotics to advance in the field of recognition and automation of surgical tasks. Furthermore, public datasets allow the comparison of different algorithms and methods to evaluate their performance [26]. These resources support a wide range of applications, including tool tracking, 3D pose estimation, force inference, and the analysis of surgical workflows, while also providing training data for machine learning models and validating new algorithmic approaches.

1.3 Pose Estimation of dVRK tools

1.3.1 Current applications of tools tracking and pose estimation in the surgery context

In the context of robotic-assisted surgery, accurate pose estimation of surgical tools is a fundamental enabler for several critical capabilities that enhance both human-in-the-loop and semi-autonomous interventions. The knowledge of a tool's 6-DoF pose relative to the camera allows the system to interpret and interact with the surgical environment more safely. Maintaining the surgical instruments within the camera's field of view is fundamental for patient safety in robotic-assisted MIS. Medical errors can be decreased by employing visual servoing techniques based on endoscopic images to automate the movement of the endoscope holder arm. A study developed a marker-based visual servoing system on the Da Vinci robot. In this work, the pose of the instruments (obtained using ArUco markers on the PSMs) was used as 3D feedback: the software calculates the spatial position of the instrument and autonomously moves the camera arm (ECM) to keep the instruments continuously in view. [26] Safety and collision avoidance are other domains where pose tracking plays a key role. By continuously estimating the spatial configuration of multiple instruments and anatomical models, the system can prevent unintended tool-to-tool or tool-to-tissue collisions, which is particularly important in confined workspaces such as those in minimally invasive procedures. In some studies, pose tracking is used as a safety layer to prevent collisions between instruments and with sensitive tissue. A common approach is to impose dynamic virtual fixtures (forbidden zones) around tools and anatomy. Moccia et al. (2020) [27] implemented vision-based forbidden-region virtual fixtures on the dVRK: they segment each tool and fuse the visual tip position with the robot's kinematics via an Extended Kalman Filter (EKF). This yields a real-time tool pose estimate that drives a repulsive force whenever instruments go against each other or into forbidden regions. The system "pushes back" against dangerous motions, so that even if the surgeon moves the controls into a collision, the robot stops or changes direction. Pose tracking is also applied to assess surgeon skill in context. For example, Pan et al. (2023) introduced a framework that fuses video-based tracking of the instrument tip with robot kinematics to extract each tip's motion trajectory during a dVRK task. They then classify these motion signals with a deep network to predict surgeon skill level (novice vs. expert). [28]

1.3.2 Future clinical integrations of Pose Estimation of dVRK tools

One of the primary future applications of pose estimation is in the automation of surgical sub-tasks. Within the research group where this thesis is conducted, there is a strong focus on the automation of suture—a complex and clinically relevant challenge that involves several interdependent modules. These include, but are not limited to, tool pose estimation (the focus of this work), instrument and parts segmentation, action recognition, and surgical step classification. Accurate pose estimation of the robotic instruments is essential in this context, as it enables the planning and execution of needle trajectories that comply with anatomical constraints and aim to minimize tissue deformation: autonomous suturing requires real-time knowledge of the needle driver's pose to compute motion paths that ensure safe and effective manipulation. While current systems often rely on semi-autonomous functionalities, assisting the surgeon in critical subtasks, the long-term goal is to achieve full autonomy for specific procedures. The transition from surgeon-assisted systems to fully automated execution represents a major advancement, as it has the potential to significantly reduce human error, enhance consistency, and improve surgical outcomes, particularly in high-precision tasks such as suturing. Another potential future application of pose estimation lies in enhancing surgical action and phase recognition. The use of instrument position information has already been explored in some studies [29], where the presence and trajectories of instruments are extracted from stereo videos and combined with CNN-based features. This approach has led to improved recognition of surgical phases, as incorporating kinematic data (such as instrument motion and interaction) significantly enhances system performance. However, in this case, pose information is derived from segmentation and therefore lacks depth accuracy. The use of actual 3D pose data could further increase the effectiveness of these systems.

Moreover, the pose of the instrument can be used to maintain alignment with moving anatomical structures, such as in heart surgery or lung interventions, ensuring that the tool stays on trajectory despite physiological movements.

Furthermore, the need for reliable methods to assess a surgeon's technical performance is growing as robotic surgery continues to expand. Standardized pose-based metrics can be useful for evaluating robotic skill levels and providing feedback during training. Over time, such pose-based evaluations could help establish fair certification standards across different institutions.

Lastly, with the growing adoption of dataset-driven methodologies in surgical robotics,

real-time pose tracking plays a critical role in the automatic generation of large-scale annotated datasets within clinical environments. Accurate 3D pose data of instruments, captured in real time, can provide reliable ground truth (GT) annotations for tasks such as tool segmentation, action recognition, and workflow analysis.

1.4 Overview of existing approaches for Pose Estimation

This section provides a general overview of the main categories of pose estimation techniques used in surgical robotics, focusing on marker-based and markerless approaches. A more detailed analysis of the most effective methods, covering both traditional and learning-based solutions, will be presented in the subsequent *State of the Art* chapter. Existing approaches for estimating the pose of surgical instruments can be broadly categorized into marker-based and markerless methods. These techniques differ in terms of precision, practicality, and suitability for deployment in real surgical environments. Marker-based methods rely on the use of artificial fiducial markers—such as AprilTags [21, 30], or ArUco [31]—attached directly to the surgical tools. These markers have well-defined visual patterns that can be detected and processed using geometric algorithms to recover the tool's 3D pose. These methods are generally chosen for their high accuracy and low computational requirements, making them well-suited also for real-time applications. However, in surgical contexts, marker-based systems face critical limitations: markers can compromise the sterility of instruments, become occluded by tissue or fluids, or interfere with tool manipulation in constrained spaces. Their effectiveness also depends on maintaining continuous visual contact with the marker, which is not always guaranteed during surgery. To address these challenges, markerless methods have been developed, eliminating the need for physical modifications of the instruments. Traditional markerless approaches typically rely on known 3D models of the instruments and attempt to match these models to visual cues extracted from the scene, such as edges, silhouettes, or point clouds obtained from RGB-D cameras. More recently, machine learning and deep learning have become increasingly popular in the field due to their ability to learn complex patterns from data and enhance the accuracy of pose estimation across various surgical scenarios. Convolutional Neural Networks (CNNs), sometimes combined with temporal or depth information, can learn to predict the pose of surgical tools directly from RGB or RGB-D images. These methods have shown better generalization across different tool types and can

be integrated with other perception tasks such as segmentation or action recognition. However, these methods are sensitive to partial occlusions, lighting changes, specular reflections, and the complex backgrounds often encountered in surgical settings. Nevertheless, they require large, annotated datasets for training, which are scarce in surgical robotics. The cylindrical shape of the instruments can also lead to pose ambiguities, such as 180-degree errors around the tool axis. Furthermore, deep learning models are often challenging to interpret when they fail, as the reasoning behind their predictions is not always transparent. Finally, deploying these models in real-time scenarios typically demands powerful hardware and significant computational resources. In summary, the choice between marker-based and markerless pose estimation methods involves trade-offs between accuracy, operational feasibility, and generalization. While marker-based approaches offer precision in controlled environments, markerless solutions represent a promising direction for real-world surgical applications, where adaptability and minimal hardware modification are essential.

2 State of The Art

Building on the categorization introduced above, this chapter provides a detailed review of the most representative and effective pose estimation methods applied—though not limited—to surgical robotics. The analysis highlights not only the principles, strengths and limitations of each approach, but also innovations and improvements that have driven progress in the field.

2.1 Marker-based approaches for Pose Estimation

Marker-based pose estimation approaches provide great accuracy under controlled environmental conditions. Thus, they are used in many fields such as robotics or biomedical applications but are primarily implemented through classical approaches, which require lots of heuristics and parameter tuning for reliable performance under different environments [32].

2.1.1 Fiducial Markers

Fiducial markers —artificially designed patterns placed in the environment— are useful in robotics, computer vision, medical imaging, and related fields for 6D pose estimation, mapping, localization, and other pose-related tasks. Tracking an object's pose by attaching a marker to it and using a vision system can be highly accurate while still being a low-cost alternative when compared to other strategies [33]. These markers are typically designed with high-contrast, easily distinguishable patterns, such as black-and-white squares or circles, that facilitate fast and robust detection by computer vision algorithms. Factors such as imaging noise and subtle changes in illumination induce jitter on the estimated pose that impairs robustness in vision and robotics applications [34].Unlike natural features, which depend on the environment's textures or lighting conditions, fiducial markers are artificial and engineered specifically to ensure reliable detection under a wide range of imaging conditions. They can encode information and support accurate localization even in the presence of noise, partial occlusion, or challenging viewpoints. Fiducial markers also provide better-defined features than the ones naturally available in the scene. For this reason, they are widely utilized in computer vision applications where reliable pose estimation is required. To manage fiducial systems, it is common to use a marker package, which is a comprehensive software toolkit that bundles the marker designs with the routines needed to generate, detect, and decode them. ARTag, AprilTag, ArUco, and STag represent the state-of-the-art and most widely used fiducial markers.

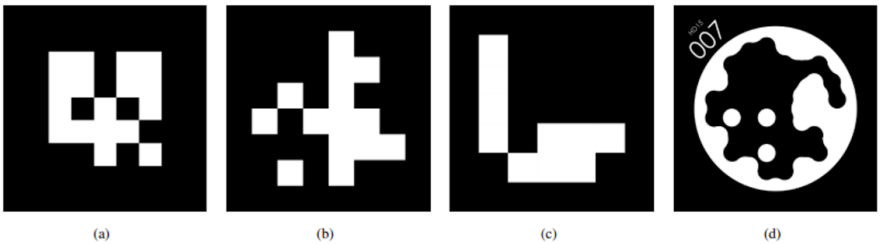


Figure 8: (a) ARTag, (b) AprilTag, (c) ArUco, (d) Stag

ARTag (Fig. 8(a)), created by Fiala [34, 35], uses an array of black and white squares that are interpreted as 0 and 1 by the detection algorithm. Although it was originally designed for augmented reality, it can also be effectively used for accurate 6D pose estimation of a camera or an object relative to the marker. ARTag provides high contrast and precise corner localization, making it suitable for computer vision systems that require reliable pose tracking. Although its popularity

has decreased compared to more modern systems like AprilTag or ArUco, it laid important groundwork for marker-based pose estimation and continues to be cited in different works in the field. ARTag is built on ARToolkit but enhances the internal marker pattern using concepts from digital coding theory. It represents the inside of each square marker as a 6x6 grid of bits, resulting in a unique 36-bit identifier for every marker. The set of markers is designed to maintain a minimum Hamming distance between each codeword to reduce recognition errors [36]. Hamming distance measures how many bits differ between two codes; a higher minimum distance helps avoid confusion between similar markers. Furthermore, a gradient-based method is used to detect lines that are later grouped into quadrilaterals of candidate markers. These improvements led to increased tag detection reliability and enabled detections under partial occlusions. AprilTag (Fig. 8(b)) [37, 30] builds on the framework established by ARTag, introducing several enhancements. It includes a graph-based image segmentation technique that examines gradient patterns to accurately detect lines, along with a quad extraction process capable of recognizing non-intersecting edges as potential candidates. Additionally, it adopts a new coding scheme to overcome challenges associated with the 2D barcode system, such as sensitivity to rotation and susceptibility to false positives in outdoor environments. These enhancements make this marker more resilient to occlusions and distortions, while also reducing the rate of incorrect detections [36]. In the context of surgical instrument tracking, AprilTag can be attached to instruments to facilitate real-time 6D pose estimation using monocular cameras. This approach offers a low-cost and flexible alternative to more complex tracking systems. However, it's important to note that certain limitations, such as sensitivity to viewing angles and lighting conditions, can affect accuracy. AprilTag has also been investigated in a study [38] in which the authors propose a multi-scale strategy to accelerate marker detection in video sequences. Their approach selects the most suitable scale at each stage (detection, identification, and corner refinement), significantly reducing computational overhead. Experimental results demonstrate that this method achieves up to a 40× speed-up compared to state-of-the-art techniques, without compromising detection accuracy. Researchers have proposed further enhancements, including geometric corrections and probabilistic error models, to improve precision [39]. ArUco (Fig. 8(c)) is a marker system based on ARTag and ARToolkit. It was initially developed by Garrido-Jurado et al. in 2014 [40] and further extended in 2016 [41]. Similar to AprilTag, ArUco (Fig. 8(c)) uses markers represented by a square shape, with information encoded in black and white patterns. One of the most notable contributions of ArUco is that it allows users to create configurable libraries of

markers. Instead of including all possible markers in a standard predefined library, users can generate a library specific to their application needs. The generated library will contain only the specified number of markers that share the greatest possible Hamming distance between their codewords. This reduces the probability of misidentification and improves robustness. Additionally, the smaller size of these custom libraries contributes to reduced computing time, which is beneficial for real-time applications. Another advantage is that ArUco supports the generation of markers of various sizes depending on the needs of the application. Since smaller markers contain less information, this can lead to improved performance of the detection algorithm, especially in terms of speed and efficiency. Furthermore, the ArUco system has an open-source detection algorithm, which has not only facilitated its widespread adoption but also encouraged continued development. The ArUco marker system has been widely used for tracking and pose estimation in surgical robotics and other medical applications. For instance, J. Birch et al. [42] presented a trocar localization method using a micro camera mounted on a vitreoretinal surgical gripper to track two ArUco markers placed on either side of the trocar. In another work, D. Tsui et al. [43] developed a low-cost stereoscopic optical tracking system for computer-assisted surgery using ArUco markers. Moreover, ArUco has been used in multimodal interactive systems for surgical robots, where marker-based pose estimation is employed for fine adjustments of the manipulator's position and orientation, enabling the execution of 12 different operational commands by rotating and moving the marker in different directions [44]. To mitigate occlusion, some systems use multiple planar tags arranged in 3D. For instance, a custom multi-face ArUco board with 21 small markers on different faces of a probe was used in hybrid tracking, resulting in improved performance [45]. In one study [46],two methods are proposed for tracking square fiducial markers in challenging conditions. The first relies on Discriminative Correlation Filters (DCFs), which are used to track visual targets by learning filters that discriminate between the target and the background; this approach is combined with adaptive scale selection and a corner refinement strategy to enhance robustness and accuracy. The second method addresses efficient camera pose estimation using marker maps by continuously tracking visible markers and predicting the position of those about to enter the scene, enabling robust and low-cost pose estimation in dynamic environments. Another study [47] introduces an improved method for enhancing the accuracy of pose estimation in mobile robotics by utilizing multiple fiducial markers simultaneously. While traditional single-marker systems such as ArUco are easy to implement, their reliability can degrade under challenging environmental conditions or in the presence of mea-

surement noise. To overcome these limitations, the authors propose an algorithm that combines two core strategies: spatial consistency checks to identify and reject erroneous markers within a marker array, and temporal stability analysis to filter out outlier measurements over time. By averaging pose estimates from consistent and reliable markers, the method significantly improves overall estimation accuracy while ensuring that no single marker disproportionately influences the final result. STag [34] (Fig. 8(d))is a recently introduced fiducial marker system designed with a primary focus on enhancing the stability and reliability of pose estimation results. The main difference between STag and the other marker systems discussed lies in the use of a circular pattern at the center of each STag marker. Once line segmentation and quadrilateral detection are performed, an initial homography is estimated for the detected marker. This estimate is then refined by identifying the central circular pattern and applying elliptical fitting, which has shown a greater accuracy in the localization. This refinement step enhances the overall stability of the pose estimation measurements [36]. Given its focus on stability and the availability of open-source ROS integrations, STag can be considered a valuable option for accurate and robust pose estimation in robotic surgical scenarios.

2.1.2 Custom Markers

Besides standard fiducial markers, it is important to highlight systems that employ customized markers specifically designed for laparoscopic tools. These custom markers are often useful to address the challenges of the surgical environment, such as tool curvature, limited space, strong specular reflections, and frequent partial occlusions. Gadwe et al. [48] proposed a novel intracorporeal endoscopic tracking approach using a printable and wrapable marker for tip pose estimation of cylindrical surgical devices. The marker consists of a green band at the tip of the instrument and a planar square containing four white and one black circle. Twelve modules are wrapped in a spiral fashion along the instrument's length, ensuring that at least one module remains visible from any camera orientation (Fig. 9). The pose estimation pipeline begins with the detection of these markers in RGB images using a combination of geometric constraints and intensity-based heuristics(Fig. 10). Initially, candidate quadrilateral regions are extracted through edge detection followed by contour extraction and polygonal approximation using the Douglas-Peucker algorithm. Only contours approximated to four-sided polygons with near-right angles and convexity are retained. To further suppress false positives, each quadrilateral is analyzed to verify the presence of a white circular region inscribed within a black square back-

ground. This contrast-rich circular pattern not only facilitates reliable verification through intensity thresholding and ellipse fitting but also improves corner localization accuracy under noise, motion blur, or defocus. Once a valid marker is identified, the four corners of the square are used as 2D-3D correspondences for solving the PnP problem, which estimates the position and orientation of a calibrated camera given a set of known 3D points and their 2D projections, and will be discussed in greater detail in Section 3.1.3. The corresponding 3D coordinates of the corners are predefined in the marker's local coordinate system, assuming the marker lies on a planar surface. Given the intrinsic camera calibration matrix, the rotation and translation between the camera and marker are estimated using OpenCV's solvePnP function. The estimated pose represents the transformation from the marker coordinate frame to the camera frame.

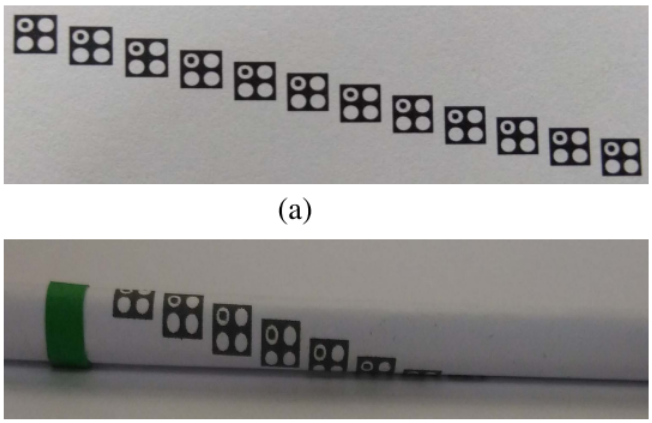


Figure 9: (a) Planar printable marker view. (b) A front view of the tool with the marker wrapped around it. [48]

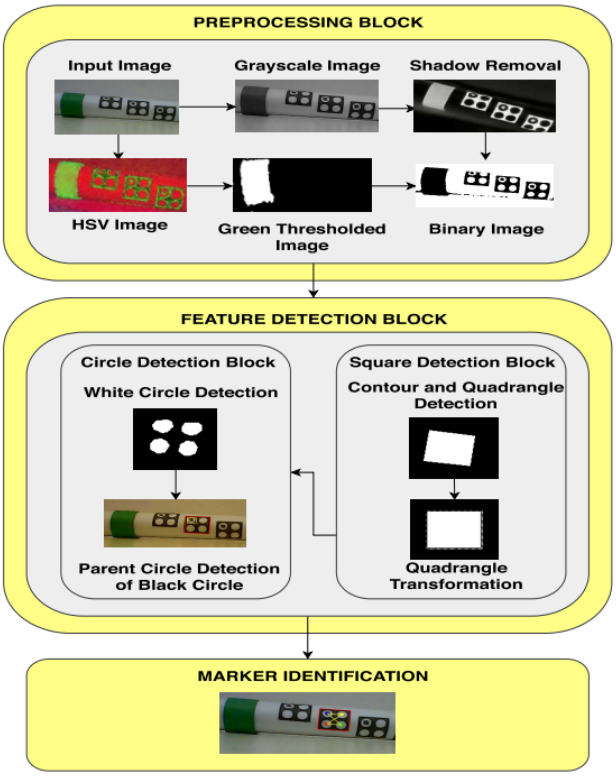


Figure 10: Marker detection algorithm flow diagram: In preprocessing block input image is converted into a binary image, which is further passed to feature detection, where a square module is detected. Finally, in the marker identification block, the marker is detected with proper orientation. [48]

An alternative design is a sparse array of dots. Pratt et al. [49] used an asymmetric circular dot marker, specifically the KeyDot® pattern. While the system is designed for a US probe, its principles are general and extendable to other surgical tools. The marker (Fig. 11) consists of a grid of circular dots arranged asymmetrically (3 rows × 7 columns), with a known spacing and diameter. This geometry was intentionally chosen due to its robustness against image degradation effects (blooming or smudging), which commonly affect traditional square markers. The use of circles also allows detection without relying on edge intersections, making tracking more resilient to noise. For detection, the system employs OpenCV's findCirclesGrid() combined with a SimpleBlobDetector [50]. This method converts the input image to multiple binary thresholded images, extracts contours, and identifies blobs based on their geometric properties (area, inertia, convexity). The centers are then aggregated across thresholds and refined through a weighted mean. To maintain real-time performance in HD video, a rOnce the pattern is initially detected, a rectangular region surrounding it is computed and used to crop subsequent video frames.

This significantly reduces the number of pixels to process, thereby improving performance. The size of the cropping area is controlled by a scaling factor S. However, this approach can fail if the marker moves too far between frames, causing the crop region to miss part or all of the pattern. To address this, the system estimates the marker's velocity by computing the displacement between the last two cropping rectangles (Fig. 13). This velocity is scaled and used to predict the next cropping region's position, increasing robustness to motion. Once the 2D dot positions are identified, a standard PnP algorithm is applied using the known 3D layout of the pattern and the camera's intrinsic parameters to recover the probe's pose.

Figure 11: Keydot marker. [49]

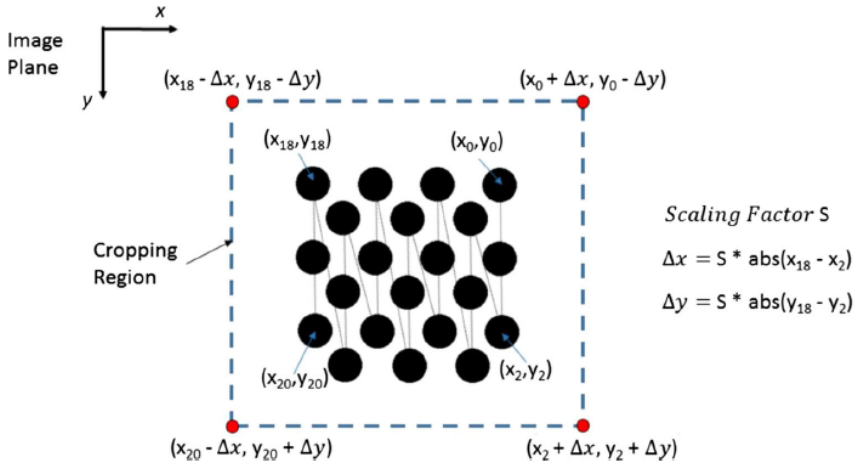


Figure 12: Calculating the appropriate cropping area around the circular dot pattern. [49]

Jayarathne et al. proposed another type of marker [51]. Although originally developed for tracking a laparoscopic ultrasound (US) probe, the proposed system is generalizable and can be adapted for 6DOF pose estimation of other surgical instruments, such as those used with the da Vinci Research Kit (dVRK). Its reliance on

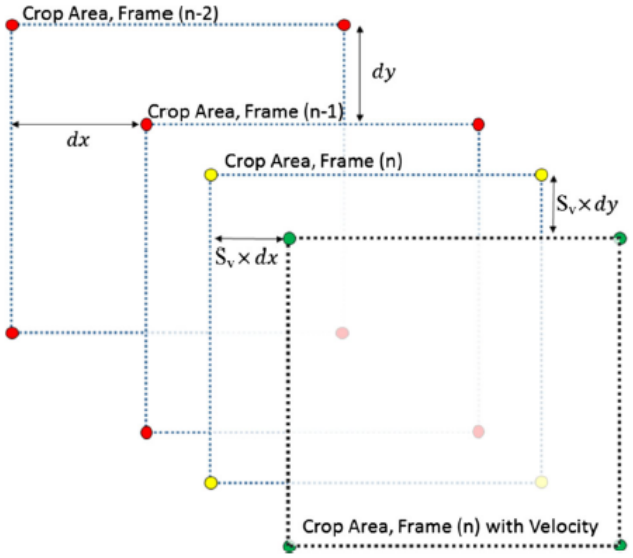
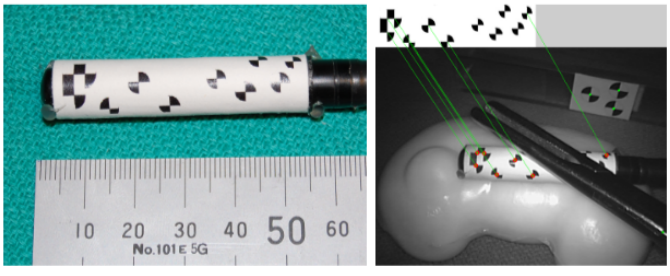


Figure 13: Cropping rectangles with velocity estimation and weighted accumulation. [49]

a monocular endoscopic camera and vision-based methods makes it suitable for integration into a wide range of robotic-assisted surgical systems. The paper presents a purely vision-based method that uses a standard monocular laparoscopic camera to track a custom-designed 3D marker attached to the instrument. This marker (Fig. 14)consists of multiple "X-corners" created from the intersection of black and white squares arranged on a curved 3D surface that conforms to the shape of the instrument. These corners are detected efficiently and with high precision using a dedicated algorithm, then refined to subpixel accuracy.



(a) (b)

Figure 14: (a) a 3D X-corner pattern axed to a linear US probe, (b) the proposed method is able to establish the 2D to 3D correspondence even in the presence of spurious (top right) and missing features (those occluded by surgical grasper). [51]

To compute the pose, the system solves the Perspective-n-Point (PnP) problem based on 2D-3D correspondences. However, to address practical challenges like occlusions and visual noise, the method employs a Gaussian Mixture Model (GMM) trained offline on a plausible range of poses. During inference, the most likely pose

hypothesis is selected from this GMM. An Extended Kalman Filter (EKF) is then used to project this hypothesis into image space and guide the search for matching features. The process is iterative: each matched point updates the pose estimate and informs the location of the next expected corner, making the search progressively more efficient and accurate. After obtaining at least four correspondences, a globally convergent PnP algorithm is used to refine the pose estimate. The result is passed forward as a prior for the next frame, enabling sequential real-time tracking with uncertainty propagation handled by the EKF.

2.2 Markerless Approaches for Pose Estimation

In recent years, the problem of 6-DoF pose estimation for robotic surgical tools has advanced in the research community, due to its importance in enabling accurate tracking, autonomous control, and context-aware assistance in the operating room. With the growing adoption of deep learning techniques, traditional geometric and feature-based approaches have increasingly been replaced or enhanced by data-driven models that can handle occlusions, reflections, and complex anatomical environments. markerless pose estimation approaches for articulated instruments can be broadly categorized into three families:

- **Direct methods**, which regress pose parameters end-to-end;
- **Indirect methods**, which first detect keypoints and then infer pose geometrically;
- **Dense Correspondence methods**, which establish fine-grained pixel-to-model mappings.

Each one of them offers strengths and trade-offs, particularly in the context of surgical robotics.

2.2.1 Direct Regression

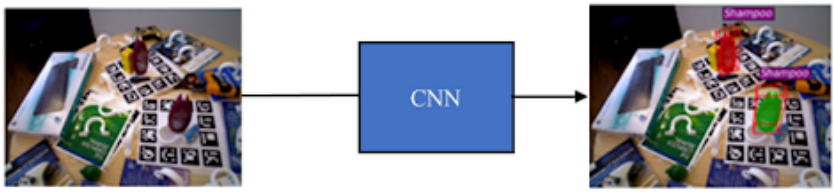


Figure 15: Direct Pose Estimation Process [52]

Direct approaches to 6D pose estimation use deep learning models to predict an object's 3D rotation and translation directly from image input, bypassing conventional stages like feature extraction and correspondence matching. These methods typically rely on convolutional neural networks (CNNs) to learn rich image representations in an end-to-end fashion. The predicted pose parameters are refined by minimizing the reprojection error between the estimated and actual object projections. This strategy is effective even in challenging scenarios involving cluttered scenes and partial occlusions. [52] One of the early works in this category is by Kendall et al. [53], who demonstrated that a CNN could regress camera pose directly from a single RGB image, achieving robust real-time performance in different environments. Building upon this idea, Do et al. [54] introduced the *Deep-6D Pose* framework, which integrates 6D pose estimation into the *Mask R-CNN* architecture. This method simultaneously detects and segments objects, while a new branch is added to directly regress pose parameters. The pose is decomposed into translation and rotation, with the latter represented using Lie algebra to maintain differentiability. The training is driven by a multi-task *Loss* combining classification, bounding box regression, mask prediction, and pose estimation, enabling the model to learn all tasks jointly in an end-to-end manner.

Although many direct 6D pose estimation methods have been primarily developed and evaluated on generic objects, their core principles and network architectures can be effectively adapted for pose estimation of surgical robotic tools, such as those employed in the *daVinci* system. Among the earliest approaches in this category is the *SSD-6D* method [55], notable for its discretization-based formulation of pose estimation as a classification problem and its effective use of synthetic training data. This approach treats the pose estimation task as a classification problem. It extends single-shot detection networks (based on InceptionV4) to predict object class, 2D bounding box, discrete viewpoints, and in-plane rotations. Each image is processed through multiple feature scales to capture different object sizes, and pose hypotheses

are generated from the combination of predicted views and rotation bins.
Another recent advancement in direct 6D pose estimation methods is represented by *FoundationPose* [56], which leverages large pretrained vision models and CAD model information to predict the 6D pose of novel objects without requiring retraining on each new object. *FoundationPose* operates in an end-to-end manner, directly regressing or classifying pose parameters from RGB or RGB-D input. This approach benefits from the power of foundation models to generalize to unseen objects and reduce the need for extensive labeled training data. This is the method used in this work and will be described in detail later.
Some approaches have combined the use of neural networks with geometric modeling and optimization techniques. Kamrul Hasan et al. proposed *ART-Net* [57], which uses CNNs to predict geometric primitives such as shaft contours, midlines, and tool tips directly from input images. These primitives are then used to initialize a simplified 3D geometric model of the instrument, with the final 6D pose refined by minimizing reprojection errors through Levenberg–Marquardt optimization.

2.2.2 Indirect Methods

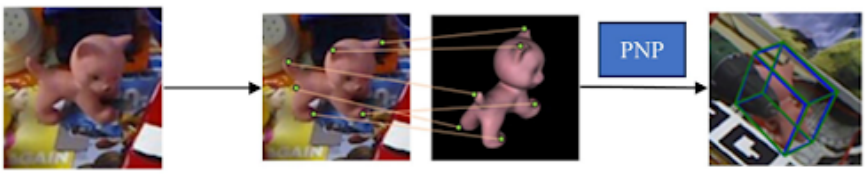


Figure 16: Indirect method for Pose Estimation using keypoints. [52]

Keypoint-based methods infer 6D pose by first detecting a sparse set of distinctive 2D points—such as tool tips or defined anatomical landmarks—in RGB images, and then computing the 3D pose through PnP algorithms using known correspondences to a CAD model. This two-stage pipeline balances modularity and interpretability while leveraging geometric constraints to constrain the pose, making it well-suited for surgical tools that can be modeled accurately. Among the generalized approaches widely used for rigid objects are *PVNet* [58] and *HybridPose* [59], which predict pixel-wise vector fields pointing toward keypoints and employ RANSAC-based voting to robustly localize them, even under occlusion.
In addition to general-purpose approaches, several keypoint-based methods have been specifically developed for the pose estimation of surgical tools. For example, Doughty et al. introduced an end-to-end CNN pipeline tailored for surgical drills,

including integration with mixed-reality headsets [60]. Their method directly estimates the 6-DoF pose from monocular RGB input and achieves a 3D vertex error of approximately 11 mm in real-world scenarios, demonstrating its viability for real-time instrument tracking. Xu et al. [61] developed a technique that leverages a high-resolution *HRNet* backbone to predict unit-vector fields from each visible pixel toward sampled keypoints on the CAD model. These vectors are processed through a RANSAC-PnP pipeline to estimate the 6D pose, exhibiting strong robustness to heavy occlusions. On benchmark datasets of surgical tools, the method achieves translation errors under 1 mm and rotation errors around 1°.

2.2.3 Dense Correspondence Methods

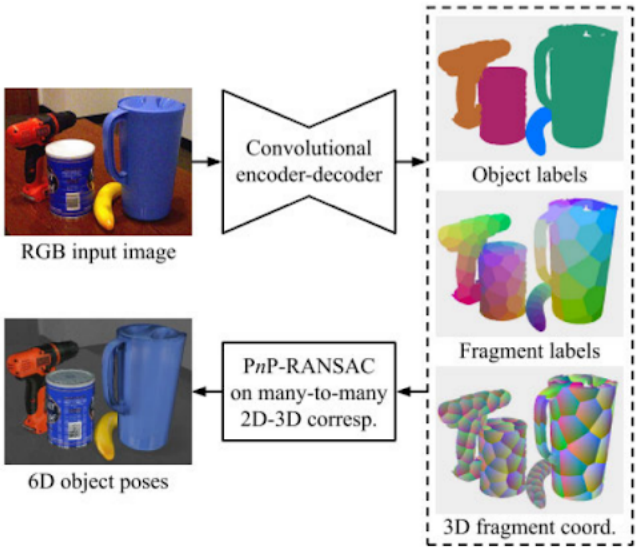


Figure 17: Dense Correspondence method for pose estimation [52]

Dense correspondence-based methods approach 6D pose estimation by predicting dense 2D–3D correspondences between image pixels and the surface of a known 3D object model. Typically, these methods operate on RGB images and recover the pose using PnP with RANSAC, leveraging the geometric consistency between predicted correspondences and the model.

Li et al. introduced the Coordinate-based Disentangled Pose Network (CDPN) [62], which separates the prediction of object rotation and translation. This separation enhances robustness, particularly for textureless or partially occluded objects. Park et al. presented *Pix2Pose* [63], a model that directly regresses dense object coordinates from RGB images without relying on texture models. The approach demonstrates strong resilience to occlusion and generalizes well to unseen poses.

Zakharov et al. developed *DPOD* (Dense Pose Object Detector) [64], which infers object ID masks and dense 2D–3D correspondences from RGB input. Final pose estimation is achieved via a RANSAC-based PnP algorithm. Ausserlechner et al. proposed *ZS6D* (Zero-Shot 6D Pose Estimation) [65], which eliminates the need for per-object training by aligning 2D image features with precomputed colored object coordinate templates. Other methods, such as those in [66], recover the pose by establishing surface-level correspondences, often using deep networks to learn local 2D–3D mappings.

Although most dense correspondence-based approaches were developed on generic object datasets, they can be adapted for surgical robotic tool pose estimation. Otherwise, some approaches have been specifically designed for surgical tools. In 2024, Barragan et al. proposed a method specifically designed for the dVRK [67]. The approach processes a Region of Interest (ROI) of the input RGB simulated image, containing the instrument. After that, it generates three geometric feature maps: a visible mask, a dense correspondence map that links 2D pixels to 3D model points, and a surface region attention map. These maps are then combined and passed through a fully differentiable Patch-PnP module, which regresses the final 6D pose. Haugaard & Buch introduced *SurfEmb* [68], which learns continuous surface embeddings to represent object geometry. SurfEmb trains a CNN to output dense 2D descriptors (from the image) and corresponding 3D descriptors (on the CAD surface) for each pixel. Pose is then found by matching descriptors: many 2D–3D correspondences are sampled and fed into a multi-hypothesis PnP solver. In the recent *SurgRIPE challenge (2025)* [69], the winning IGTUM team applied SurfEmb to *da Vinci* tool images (Large Needle Driver, Bipolar Forceps) and achieved the best accuracy.

3 Materials and Methods

This section presents the methods analyzed for 6DoF (position and orientation) pose estimation of surgical tools of the da Vinci Research Kit (dVRK), both in a simulation scenario and in the real world. In particular, the following methods are discussed:

1. **Marker-based method** designed for cylindrical objects [70].
2. ***FoundationPose***, a unified foundation model for generic objects [71].

In the simulated environment, pose estimation was performed on two instruments:

the *Needle Driver* and the *Cadiere Forceps*. In the real-world case, however, the pose estimation process was carried out only on the *Needle Driver*. This choice was made both to reduce the time required for data acquisition and because the simulation results showed comparable performance across the two instruments.
Section 3.1 introduces a pose estimation method specifically designed for cylindrical tools, which is used to generate the ground truth (GT) poses of the surgical instruments.
To evaluate the foundation model, it is essential to have GT data to compare performance and establish a reliable reference. For this reason, the marker-based method was chosen, as it provides a robust and accurate solution for pose annotation. This method was selected for GT generation in both the simulated surgical scene and in the real-world scenario.
The foundation model *FoundationPose* is introduced in Section 3.2, with an overview of its architecture and the methodology utilized. *FoundationPose* is based on transformer architecture and is trained on a large-scale synthetic dataset to predict object poses from RGB-D images, given a CAD model of the object. Sections 3.3 and 3.4 describe the generation of annotated surgical image datasets in simulation and in the real world.
To satisfy the input requirements of *FoundationPose*, the *Unity Perception* package was used in the simulated environment, while *Roboflow* [**?**] and *DepthAnything* [**?**] were used in the real-world setting.
As mentioned earlier, GT pose annotations in both cases were obtained using the marker-based method, due to its robustness and reliability. In particular, Section 3.3.2 highlights the GT generation process for the simulated case, which was carried out using *Blender*, an open-source 3D creation suite that supports the full pipeline of modeling, texturing, and rendering. This platform facilitated the correct placement and rendering of the marker texture on the instrument surface.
An essential aspect of ensuring consistency between the 3D annotations and the 2D image data is camera calibration, along with the acquisition of distortion parameters. In the simulated scene, this calibration process was performed within Unity, while for the real-world setup, Zhang's method [72] was employed, as described in Section 3.4.1.
Ultimately, the evaluation metrics used to assess the performance of the model-based foundation model are defined.

3.1 Marker-based method

This section describes the marker-based method used for estimating the pose of dVRK surgical instruments, which serves as the basis for GT generation in both cases. It introduces the marker design, which enables reliable detection of the instrument, and details the full pipeline leading to pose estimation. In particular, Section 3.1.3 explains the method used to solve the Perspective-n-Point (PnP) problem, which involves estimating the pose of an object relative to a camera, given a set of known 3D points on the object (in object coordinates), their corresponding 2D projections in the image, and the camera intrinsic parameters.

3.1.1 Marker Design

This approach uses a marker that contains a binary pattern made up of black features placed on a green background. The green colour is chosen because it enhances the segmentation process, which is a pre-step for pose estimation, as the green color stands out clearly from most backgrounds. Segmentation is performed in the HSV (Hue, Saturation, Value) color space, ensuring that only the marker's features are used in the pose estimation process. The marker pattern is characterized by features that encode a binary pattern with two classes (0 and 1). The features on the marker had to remain reliable under difficult surgical tool poses. For example, when the tool is farther from the camera, its features appear smaller in the image, and when viewed at a sharp angle, those features become more distorted. For this reason, the shape of the features has been designed to ensure they can be easily classified in binary terms and remain visually consistent even in challenging viewing conditions. Elliptical blobs oriented in opposite directions have been chosen, providing a robust solution since they were consistently detectable and distinguishable, even when the tool was viewed from extreme angles or distances. The introduced pattern has a total of 16 binary sequences. Each sequence corresponds to a row on the pattern and is formed by 8 features, resulting in a total of 128 features in the marker. Each of the 128 features is assigned an ID and a 3D coordinate (X, Y, Z) relative to the marker's coordinate frame. This 3D coordinate is calculated using the radius of the tool and the position of the feature's centroid in the marker. The IDs and 3D coordinates of the marker features are used as input to a PnP solver to estimate the pose of the marker.

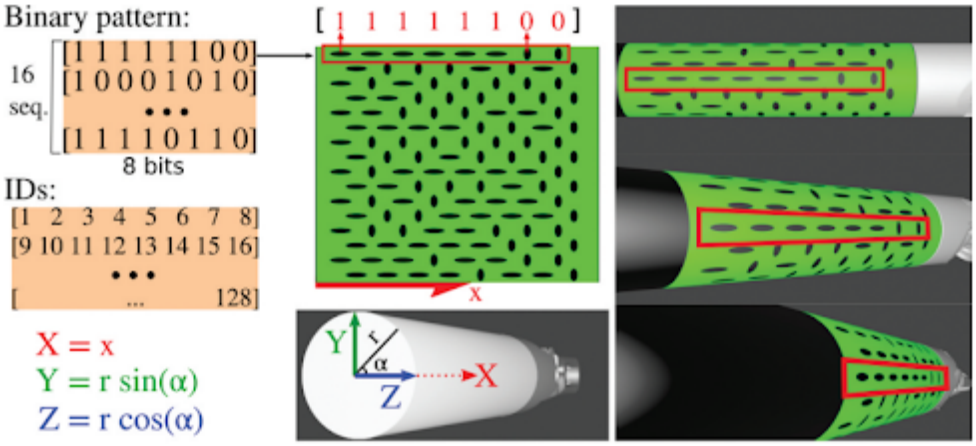


Figure 18: Marker design with binary pattern. [73]

3.1.2 Methodology

Starting from raw endoscopic images, the pipeline corrects lens distortion, segments and identifies the marker's features, and finally establishes 3D-to-2D correspondences that enable pose estimation. The method relies on geometric features of the marker and computer vision techniques. Each step of the pipeline is shown and detailed below. The first step is the undistortion of the input laparoscopic image

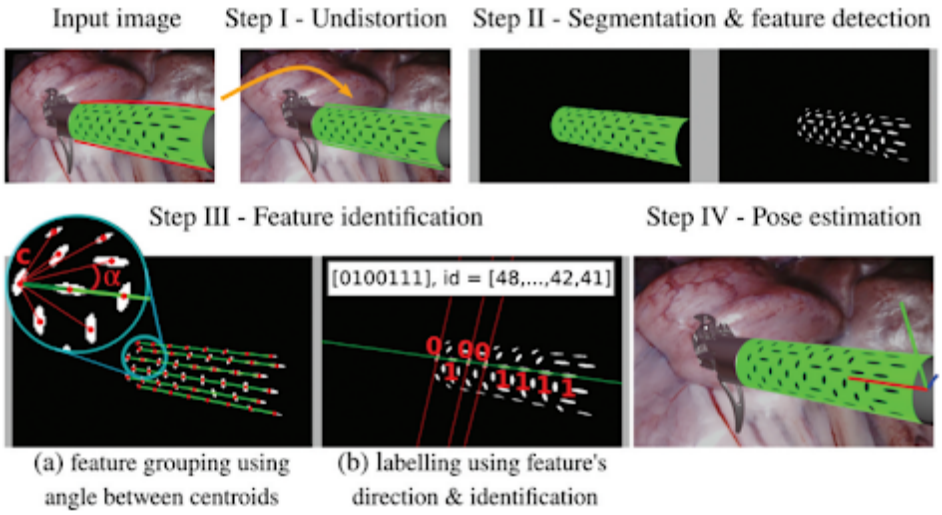


Figure 19: Surgical tool pose estimation using cylindrical marker [73]

using the tangential and radial camera's distortion parameters. This step is essential to ensure that the centroids of the marker features within the same sequence align along a straight line. Then, the undistorted image is converted into the HSV colour space to segment the marker and detect features on its pattern. Firstly, the marker is

segmented using its green colour, which is done by thresholding the Hue, Saturation, and Value channels. After segmenting the green marker, the black elliptical features inside it are also segmented. The V channel alone is enough to differentiate black from green, as black pixels exhibit much lower V intensity values compared to the surrounding green pixels. All pixels within the marker area are classified as either green or black. The black pixels are then clustered using a standard connected-component labelling approach, where each resulting connected region represents an individual detected feature. Once the features are detected, they need to be identified, which means knowing their 3D position relative to the marker's coordinate frame. The features are identified in groups of 8, which are located on a straight line. To find which features lie on a straight line, the pixel centroid of each detected feature is calculated using the image moments of the contours of each feature. Then, using all the centroids, the features are grouped into sequences by finding 8 centroids that lie on the same line. Given one of those centroids c, the angles that the other centroids form with c are calculated as the inverse tangent of the difference of their image coordinates. Then, the angle values are sorted, and if there are 7 adjacent angles in the sorted order that have approximately the same value, then those centroids are grouped into the same sequence as c, forming a group of 8 features. After grouping the centroids into sequences, each feature is classified as 1 or 0. To do that, a 2D line is fitted to the shape of the feature, using a least-squares method, to determine the feature's direction. If the feature's direction is aligned with the line formed by the centroids belonging to the sequence, it is classified as 1; otherwise, as 0. After classifying all the features in a group, the features can finally be identified by assigning an ID value from 1 to 128. A minimum of three sequences need to be identified to estimate the marker's pose. For each identified feature, a 3D-to-2D correspondence of a feature's centroid is obtained. The 3D corresponds to the position of the centroid in the marker's coordinate frame, and the 2D corresponds to the pixel coordinates of the detected feature's centroid in the image plane. A 2D to 3D conversion is used to calculate each marker's 3D coordinates on the cylindrical tool surface. The horizontal coordinate in the 2D image corresponds directly to the position along the tool's longitudinal axis and is assigned as the X coordinate in 3D. The angular position around the cylindrical surface is represented by the vertical coordinate in the 2D map. First, this vertical value is scaled proportionately to the entire circumference to obtain an angle alpha. Specifically, it is computed as:

$$\alpha = \frac{y}{H} \cdot 2 \tag{1}$$

where

- y = vertical position,
- H = marker height.

Given the radius r of the tool, the Y and Z coordinates in 3D space are then computed using standard Cylindrical-to-Cartesian conversion:

$$Y = r\sin(\alpha) \tag{2}$$

$$Z = r\cos(\alpha) \tag{3}$$

This process allows the 2D centroid of each marker in the flat layout to be accurately mapped to its corresponding 3D position on the curved surface of the tool. To estimate the marker's pose, all the identified features and all the 3D-to-2D correspondences are used as input to a RANSAC implementation of an EPnP solver [Lepetit et al. 2009]. After this step, the tracking algorithm is repeated for the next input image.

3.1.3 EPnP Solver

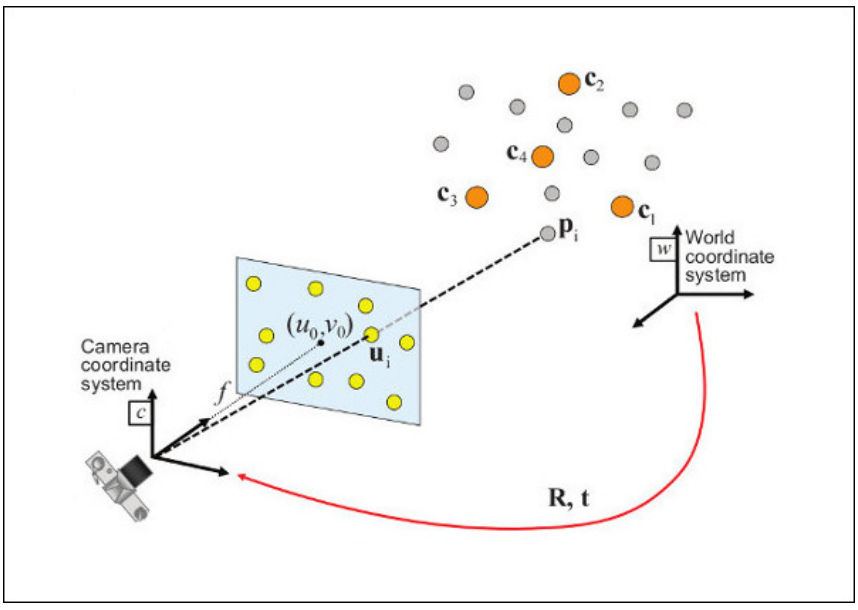


Figure 20: Perspective-n-Point (PnP) pose computation. [74]

The Efficient Perspective-n-Point (EPnP) Solver algorithm [75] provides a fast and accurate solution for retrieving the object's pose given the 2D-3D correspondences

and the camera intrinsics, and is well-suited for this application due to its computational efficiency. The goal is to find the rotation R and translation t that align the 3D points with their 2D projections when passed through the camera projection model. EPnP, proposed by Lepetit et al. [71], is a non-iterative and highly efficient method for solving PnP. This method is based on the notion that each of the n points (which are called reference points) can be expressed as a weighted sum of four virtual control points. Thus, the coordinates of these control points become the unknowns of the problem. It is from these control points that the final pose of the camera is solved for. Each of the n 3D reference points in the world frame, $\mathbf{p}_i^w$, and their corresponding 3D image points, $\mathbf{p}_i^c$, are weighted sums of the four control points, $\mathbf{c}_j^w$ and $\mathbf{c}_j^c$, respectively, and the weights are normalized per reference point as shown below. All points are expressed in homogeneous form.

$$p_i^w = \sum_{j=1}^{4} a_{ij}\, c_j^w \tag{4}$$

$$p_i^c = \sum_{j=1}^{4} a_{ij}\, c_j^c \tag{5}$$

$$\sum_{j=1}^{4} a_{ij} = 1 \tag{6}$$

$$\sum_{j=1}^{4} a_{ij} f_x x_j^c + \alpha_{ij}(u_0 - u_i) z_j^c = 0 \tag{7}$$

$$\sum_{j=1}^{4} a_{ij} f_y y_j^c + \alpha_{ij}(v_0 - v_i) z_j^c = 0 \tag{8}$$

Where:

- f_x, f_y are the focal lengths in pixels,
- (u_0, v_0) are the 2D coordinates of the optical center,
- x_j^c, y_j^c, z_j^c are the coordinates of the control point j in the camera frame,
- (u_i, v_i) are the observed 2D coordinates of the point in the image.

Using these two equations for each of the n reference points, the system $M\mathbf{x} = 0$ can be formed, where:

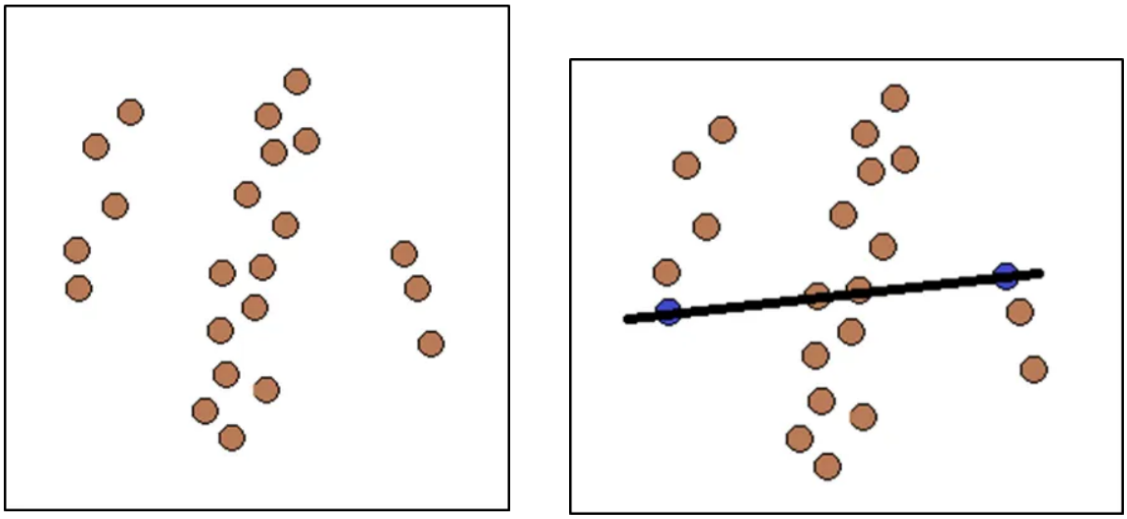
$$\mathbf{x} = \begin{bmatrix} c_1^{c^T} & c_2^{c^T} & c_3^{c^T} & c_4^{c^T} \end{bmatrix}^T$$

The solution is expressed as:

$$\mathbf{x} = \sum_{i=1}^{N} \beta_i v_i = 1 \tag{9}$$

where N is the number of null singular values in M, and each v_i is the corresponding right singular vector of M. N can range from 0 to 4. After calculating the initial coefficients β_i, the Gauss-Newton algorithm is used to refine them. The R and T matrices that minimize the reprojection error of the world reference points $\mathbf{p}_i^w$ and their corresponding actual image points $\mathbf{p}_i^c$ are then calculated.

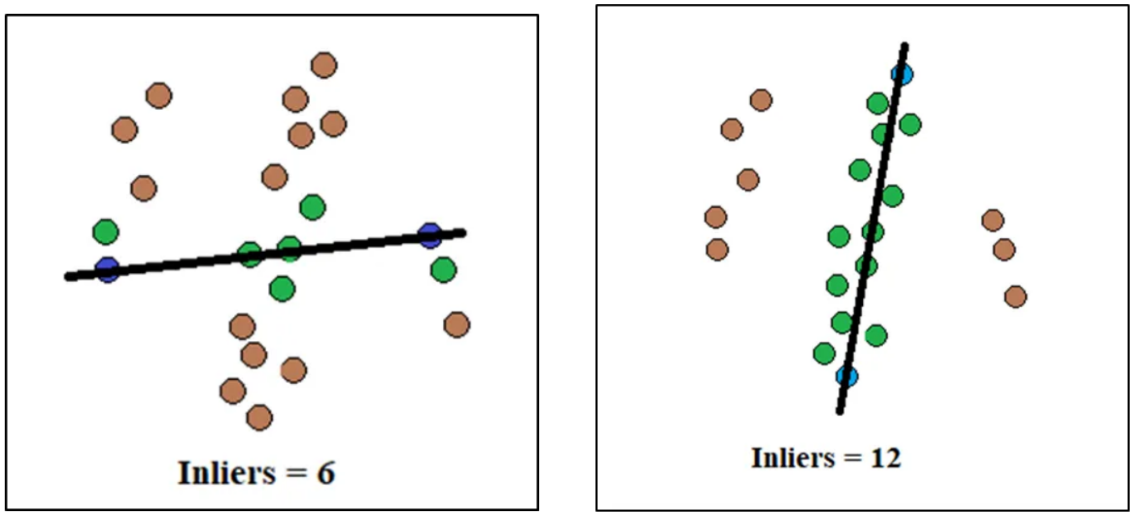
Standard EPnP takes all the given 2D-3D correspondences and estimates the pose assuming that all the points are accurate. It is fast, with linear complexity, but it is not robust to outliers: just one wrong point can significantly affect the result. In this method, EPnP is used with RANSAC implementation. It randomly selects small subsets of points, estimates the pose using EPnP, and then checks how many of the total points agree with that solution by measuring the reprojection error. This process is repeated multiple times, and the pose with the most consistent inliers is selected. Although this approach is slower, it is much more robust to outliers. The structure and explanation in this part are inspired and partially adapted from an online source [74]. To understand how RANSAC works, we consider an example in which the objective is to fit a line to a set of 2D data points (Fig. 21a). This is done by randomly selecting two data points from the dataset, which represent the minimum number of points needed to estimate the model. This subset is referred to as the Minimal Sample Set (Fig. 21b). Using more than the minimum number of points to generate a model is generally inefficient, as it decreases the probability that all selected points are inliers and lead to a good initial estimate. In fact, the larger the sample, the lower the chance that it contains only inliers. For this reason, relying on a minimal sample set increases the probability of selecting inliers, which improves the quality of the initial model.



(a) Set of 2D data points.

(b) Straight line fitting in a set of 2D data points.

Once the model is estimated, the next step is to check how many of the remaining data points—excluding the initial two—are consistent with the fitted line.



(a) Case 1

(b) Case 2

Figure 22: Evaluation of points that lie on or close to the same line (Inliers).[76]

In the first case (Fig. 22a), six data points fall on or close to the estimated line, indicating their consistency with the model. The rest, which deviate significantly from the line, are considered outliers. As a result, the model receives a score of 6. The procedure is then repeated by choosing a new pair of data points and evaluating the corresponding score for the newly fitted model.
In the second case (Fig. 22b), the score would be 12. This process is repeated a predetermined number of times, and the model that achieves the highest score is ultimately selected. This approach lies at the heart of the sampling algorithm.

In summary, RANSAC follows a three-step procedure:

- Sample: Select a subset of data points and treat them as potential inliers.
- Model Estimation: Use the selected inliers to compute the model and its parameters.
- Scoring: Evaluate the model by determining how many of the remaining data points align with or support the model.

To determine how many samples to choose, the following formula is used:

$$T = \frac{\log(1-p)}{\log(1-(1-e)^s)} \tag{10}$$

Where:

- e = probability that a point is an outlier (i.e., the outlier ratio in the dataset)
- s = number of points in a sample
- p = desired probability of selecting at least one good sample
- T = number of trials required to succeed with probability p

3.2 Model-based method: *FoundationPose*

This section presents the *FoundationPose* method. It begins by introducing the concept of foundation models and explaining why such an approach was chosen for the pose estimation of surgical instruments. The model architecture is then analyzed in detail, with particular focus on the use of transformers and the role of contrastive learning. The functioning of the two networks used for pose estimation is described: a Pose Refinement network and a Pose Ranking network, along with the corresponding loss functions employed by the authors in the training process.

3.2.1 What is a Foundation model?

A foundation model is an AI neural network trained on mountains of raw data, generally with unsupervised learning or self-supervised learning, that can be adapted to accomplish a broad range of tasks [72].
In unsupervised learning the model is trained on data without any labels: instead of being told what to look for, the model tries to find hidden patterns, structures,

or relationships within the data. Self-supervised learning is a form of unsupervised learning where the model learns by solving a pretext task with labels automatically generated from the data itself.
Initially, foundation models appeared in the context of natural language processing [77, 78]. In the field of computer vision, these models outperform or match supervised models [72, 79, 80, 81, 82, 83, 84]. For example, DINOv2 [82], based on the Vision Transformer architecture [85], encodes both spatial information about object parts and semantic information regarding object categories [86]. Additionally, it has been successfully applied in zero-shot settings, enabling semantic correspondences without any training [86, 87, 88, 89].
In the context of pose estimation, a foundation model is a model trained to understand and predict object poses across various categories, environments, and viewpoints. It captures general geometric, visual, and spatial patterns, making it more robust and versatile compared to traditional task-specific models.
Using a foundation model for dVRK pose estimation can bring several benefits compared to training a specific model from scratch. Because foundation models are trained on large and diverse datasets, they can generalize better to unseen scenarios, making them especially useful in surgical contexts where the appearance of tools and scenes can vary a lot. This makes them transferable: they can be adapted to different tools or camera setups. Another advantage is that foundation models reduce the amount of annotated data needed, which is valuable in medical applications where labelling is often not available.

3.2.2 Architecture

FoundationPose is a unified foundation model for 6D object pose estimation and tracking. It can be tested on a novel object without unnecessary fine-tuning, as long as its CAD model is given. Large-scale synthetic training, transformer-based architecture, and contrastive learning led to the obtainment of strong generalizability. A transformer-based architecture relies on the self-attention mechanism, which allows each element in the input (image patch) to interact with all others, capturing both local and global relationships. Each element is mapped into query, key, and value vectors. The model computes attention scores by comparing queries and keys, and uses these to produce weighted combinations of the values, resulting in context-aware representations.
This is done across multiple attention heads (multi-head attention), allowing the model to learn different types of interactions in parallel. Since transformers don't

have an inherent sense of order, positional encodings are added to preserve spatial or sequential structure. In computer vision, images are split into patches treated like a sequence of tokens. The transformer processes this sequence, enabling long-range dependencies to be captured more efficiently than CNNs. This makes transformers especially powerful for tasks like pose estimation, where understanding both fine details and overall structure is fundamental.

Contrastive learning is a self-supervised approach that trains a model to distinguish between similar ("Positive") and dissimilar ("Negative") pairs of examples, without requiring manual labels. During training, the network embeds data points (images or image patches) into a feature space and is rewarded when embeddings of two augmented views of the same sample are pulled closer together, while embeddings of different samples are pushed apart. By optimizing this objective, the model learns representations that naturally cluster semantically similar inputs and separate dissimilar ones, improving its ability to generalize to new poses, tools, or backgrounds.

The Pose Hypothesis Generation module (Fig. 23) is responsible for producing an initial, refined estimate of the object's pose. The process starts by rendering a coarse pose from a globally sampled distribution and extracting a corresponding crop from the input RGB-D image. Both the rendered view and the input crop are encoded using shared weights, meaning that the two encoders learn to extract features that are comparable in the same semantic space.

These features are processed through convolutional layers and subsequently divided into patches, each assigned positional embeddings to retain spatial information. This step is fundamental, as transformers lack an inherent understanding of spatial structure. The resulting patches are then passed through two separate transformer encoders: one predicts rotation updates ΔR and the other predicts translation updates Δt.

In this setup, Δt describes how much the object's position should shift in the camera frame, while ΔR describes how its orientation should change in the camera frame. Starting from the coarse pose $[R, t]$, the refined pose $[R^+, t^+]$ is computed by

$$t^+ = t + \Delta t \tag{11}$$

$$R^+ = \Delta R \cdot R \tag{12}$$

where "$\cdot$" indicates composition of rotations. Separating translation and rotation updates in this way keeps everything expressed in the camera coordinate frame and avoids having to apply translation in a rotated frame, which simplifies learning. These updates are applied iteratively to refine the pose. During training, the network

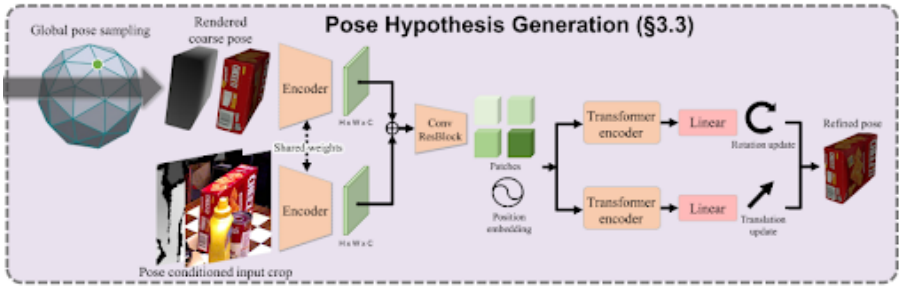


Figure 23: Pose Hypothesis Generation module of *FoundationPose*. [56]

is supervised with an L_2 loss:

$$L_{\text{refine}} = w_1 \|t - t_{\text{gt}}\|^2 + w_2 \|R - R_{\text{gt}}\|^2 \tag{13}$$

where t_{gt} and R_{gt} are the ground-truth updates, and the weights w_1 and w_2 are both set to 1.

Using an L_2 loss to supervise both translation and rotation updates is beneficial in this context because it provides a direct and smooth optimization objective that encourages the predicted updates to closely match the GT corrections. This formulation is simple, stable, and effective for guiding the network to minimize the difference between the predicted pose and the actual pose, leading to improved accuracy in iterative pose refinement.

The Pose Ranking module (Fig. 24) evaluates and selects the most plausible pose among all the poses previously generated.

First, for each pose hypothesis, the rendered image previously generated is compared against the cropped input observation, using the pose-conditioned cropping operation. The comparison is carried out using a pose ranking encoder that shares the same backbone architecture for feature extraction as the refinement network. The extracted features are concatenated, transformed into tokens, and passed through a multi-head self-attention module, allowing the model to enhance the comparison by taking into account the complete visual context of the image. The pose ranking encoder performs average pooling to output a feature embedding $\mathbf{F} \in \mathbb{R}^{512}$ describing the alignment quality between the rendering and the image.

A second level of comparison across the K pose hypotheses is introduced to exploit their global context and support a more informed decision.

Multi-head self-attention is performed on the concatenated feature embedding $\mathbf{F} = [\mathbf{F}_0, ..., \mathbf{F}_{K-1}] \in \mathbb{R}^{K \times 512}$, which encodes the pose alignment information from all poses, ensuring a global comparison. Treating $\mathbf{F}$ as a sequence allows the generalization to varying lengths of K. The attended feature is then linearly projected

to the scores $\mathbf{S} \in \mathbb{R}^K$ to be assigned to the pose hypotheses. After computing the scores for all pose hypotheses using the ranking network, the final pose is chosen as the one with the highest score.

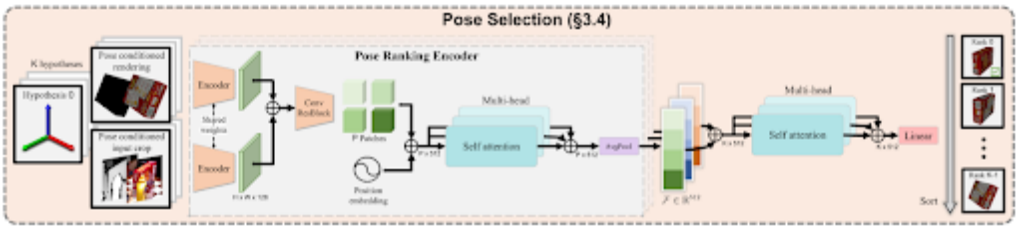


Figure 24: Direct Pose Estimation Process [56]

The training of the pose ranking network is guided by a pose-conditioned triplet loss:

$$L(i^+, i^-) = \max\left(S(i^-) - S(i^+) + \alpha,\ 0\right) \tag{14}$$

where α denotes the contrastive margin; i^+ and i^- represent the positive and negative pose samples, respectively, which are determined by computing the ADD metric [90] using GT data. While this loss can be computed over each pair in the list, the comparison becomes ambiguous when both poses are far from the GT. Therefore, to make the comparison meaningful, the pose pairs kept are only those whose positive sample is close enough to the GT.

$$V^+ = \{i : D(R_i, R) < d\} \tag{15}$$

$$V^- = \{0, 1, 2, \ldots, K-1\} \tag{16}$$

$$L_{\text{rank}} = \sum_{i^+ \in V^+,\, i^- \in V^-,\, i^+ \neq i^-} L(i^+, i^-) \tag{17}$$

where

- $i^+ \in V^+$, $i^- \in V^-$, $i^+ \neq i^-$;
- R_i and R are the rotations of the hypothesis and ground truth, respectively;
- $D(\cdot)$ is the geodesic distance between rotations;
- d is a predefined threshold.

3.2.3 Methodology

Given the RGBD image, the object is detected using an off-the-shelf method such as *Mask RCNN* [91] or *CNOS* [92]. The translation is initialized using the 3D point located at the median depth within the detected 2D bounding box. To initialize rotations, Ns viewpoints of the object are uniformly sampled from an icosphere centered on the object with the camera facing the center. These camera poses are further augmented with Ni discretized in-plane rotations, resulting in Ns · Ni global pose initializations. To enable this, for each of these poses, the CAD model is rendered using a rasterization process, which converts the 3D geometry into pixel-based images from the sampled viewpoints, forming the set of coarse pose hypotheses.

Since the coarse pose initializations from the previous step are often quite noisy, the refinement module is needed to improve the pose quality. Specifically, the pose refinement network takes as input the rendering of the object conditioned on the coarse pose and a crop of the input observation from the camera; the output is a pose update that improves the pose quality.

For the input observation, a pose-conditioned cropping strategy is performed. Concretely, the object origin is projected to the image space to determine the crop center. Then the object diameter (the maximum distance between any pair of points on the object surface) is slightly enlarged and projected to determine the crop size that encloses the object and the nearby context around the pose hypothesis. This crop is thus conditioned on the coarse pose and encourages the network to update the translation to make the crop better aligned with the observation. The refinement process can be repeated multiple times by feeding the latest updated pose as input to the next inference, to iteratively improve the pose quality.

Given a list of refined pose hypotheses, a hierarchical pose ranking network is used to compute their scores. As mentioned before, the network adopts a two-stage comparison strategy to evaluate and select the best pose hypothesis. First, for each pose, the pose-conditioned rendering is compared to the cropped input image using a pose ranking encoder, which shares the same backbone as the refinement network. The resulting features are tokenized and passed through a multi-head self-attention module to capture global context. The pose ranking encoder performs average pooling to output a feature embedding describing the alignment quality between the rendering and the observation. The second level of comparison is conducted among all the K pose hypotheses. Multi-head self-attention is used to analyze their mutual relationships, enabling relative scoring, and the features are then projected into final

scores for ranking the poses. The pose with the highest score is selected as the final estimate.

3.2.4 Dataset generation in simulation

Real-world data collection in endoscopic and surgical settings presents several challenges. Strict sterility regulations, restricted access to operating rooms, ethical concerns, and the acquisition of precise GT data for pose estimation are a few examples. To overcome these limitations, the model was evaluated on data generated in a simulation environment. This enables precise annotation, but also repeatability, and the possibility to test under a wide range of controlled conditions. This section describes the steps involved in dataset generation in the simulated scenario. Custom scripts and the Unity *Perception Package* are used to obtain *FoundationPose* requirements, including the camera calibration matrix, segmentation masks, and absolute depth maps. An established method involving a cylindrical marker is used to generate GT annotations.

3.2.5 Generation of a simulated surgical scene with Unity

Unity was selected as the simulation environment because it combines a high-fidelity real-time renderer with an extensible scripting API (Application Programming Interface), allowing fast and reproducible simulation of surgical scenes. To maximize visual realism, the project was built on Unity's High Definition Render Pipeline (HDRP), which provides advanced lighting features such as physical light units, volumetric fog, screen-space reflections, and shadows. The CAD models of two da Vinci Research Kit instruments (Needle Driver and Cadiere Forceps) were imported into Unity's asset pipeline. Each model was assigned a Physically Based Rendering (PBR) material and high-resolution textures to capture metallic glints and realistic color variations. To simulate a more realistic scenario, a background was added, and a texture extracted from real surgical operation images was applied to it. Then, to mimic the lighting conditions of an operating scene, multiple setups were experimented including the adjusting of directional lights, ambient probes, and point lights. A custom C script iterates over 300 uniformly sampled poses for each tool, constraining rotations and translations within different ranges.

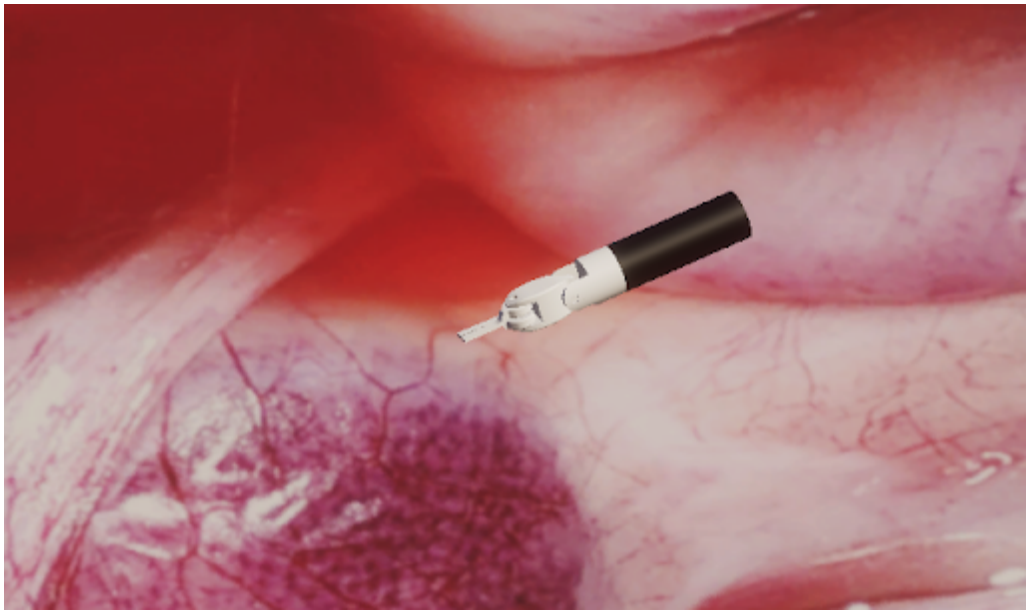


Figure 25: Simulation of Needle Driver with real surgical scene background.

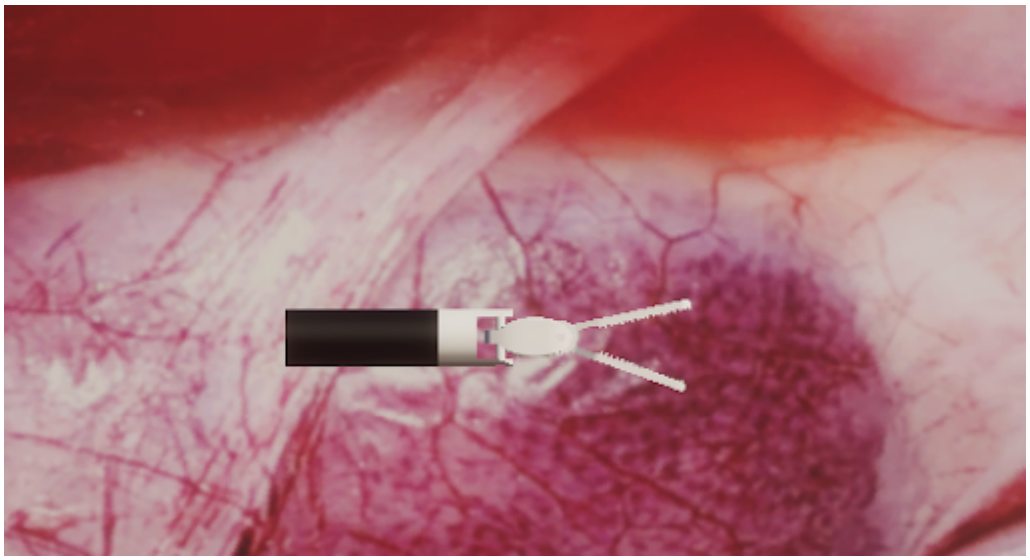


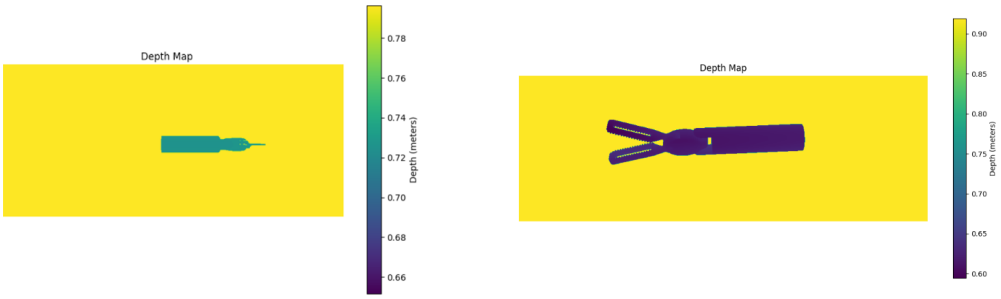
Figure 26: Simulation of Cadiere Forceps with real surgical scene background.

Unity's *Perception Package* was fundamental as it provides a variety of tools for generating synthetic datasets intended for use in perception-based machine learning tasks such as object detection, semantic segmentation, pose estimation, and so on. These datasets are in the form of frames captured using simulated sensors. These frames are annotated and are thus ready to be used for training and validating machine learning models. It provides a set of Camera Labelers that can be attached to the Perception Camera, each responsible for generating a specific type of annotation. For instance, the Semantic Segmentation Labeler outputs segmentation images in which each labeled object is rendered in a user-definable color (white,

in this case) and non-labeled objects and the background are rendered black. The BoundingBox2DLabeler produces 2D bounding boxes for each visible object with a defined label. This configuration acts as a mapping between string labels and object classes (currently colors or integers), deciding which labels in the scene (and thus which objects) should be tracked by the labeler, and what color (or integer ID) they should have in the captured frames. Two configuration files are needed: one for semantic segmentation labels and one for all other annotations, to automatically generate per-frame outputs including the coordinates of instrument segmentation masks and absolute depth maps.

(a) (b)

Figure 27: Segmentation masks of (a) the Needle Driver, (b) the Cadiere Forceps



(a) (b)

Figure 28: Absolute depth maps of the recreated scene, with (a) the Needle Driver, (b) the Cadiere Forceps

Beyond segmentation masks and depth maps, the *Perception Package* also logs pixel counts, unique label IDs, and visibility fractions for each instrument, metrics that are useful in case of retraining. In parallel, a separate calibration script computes the camera intrinsic matrix K. It first retrieves the camera's parameters, such as:

- f_{ov}: vertical field of view
- a: image aspect ratio
- W, H: image width and height (resolution)

Using these values, it calculates the intrinsic matrix, which contains the focal lengths (f_{x} , f_{y}) and the optical center (c_{x} , c_{y}).

$$f_y = \frac{H}{2 \cdot \tan\left(\frac{f_{\mathrm{ov}} \cdot \pi}{360}\right)} \tag{18}$$

$$f_x = f_y \cdot a \tag{19}$$

$$c_x = \frac{W}{2}, \quad c_y = \frac{H}{2} \tag{20}$$

The resulting intrinsic matrix K is:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{21}$$

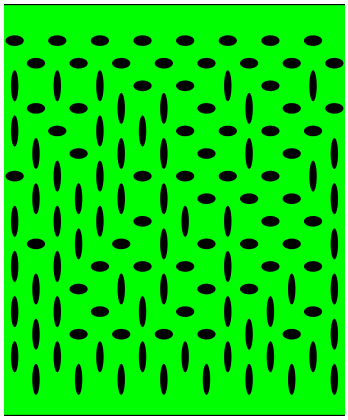
By moving objects instead of the camera, data management is simplified: every image shares identical camera parameters, and each pose record corresponds directly to an object transformation, ensuring consistency. A Python script was used to extract geometric properties, such as diameter and symmetries, that are used by *FoundationPose* as part of its pose refinement process.

3.2.6 Ground Truth Generation

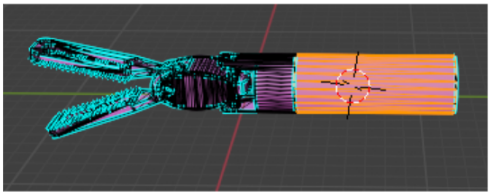
Ground Truth (GT) data, in this study, consists of the pose (position and orientation) of the dVRK tools. It is used to evaluate the performance of the pose estimation of the *FoundationPose* model and to provide a reference for comparison. Initially, GT was generated using a script directly in Unity. However, it was later observed that the axis configuration did not match the one expected by the *FoundationPose* model. Several attempts were made to align Unity's reference frame with that of the model, but no consistent result was achieved. As a result, a robust and well-tested method was adopted to define the GT. Consequently, it was also decided to estimate the pose of the dVRK tools one at a time. The chosen method relies on a cylindrical marker. To use this approach, the marker had to be adapted to the dVRK tool.

To guarantee proper 2D–3D correspondences, a marker with a new pattern was created, instead of the one provided, due to the scale discrepancy between the CAD model and the actual surgical instruments. Blender was used to make additional adjustments since applying the marker texture directly in Unity caused distortion,

which would have compromised the marker-based model's functionality. The mesh that corresponded to the instrument's handle was isolated in Blender so that the marker could be applied to that area only.



(a)



(b)

Figure 29: (a) Marker pattern. (b) Isolation of the instrument's handle on Blender to guarantee correct application of the marker.

A material and the corresponding texture for the newly created marker were generated. The texture is mapped onto the object using UV coordinates, allowing the image to follow the surface geometry. The model was then exported in .fbx format to be imported into Unity and so to generate the dataset with the marker applied.

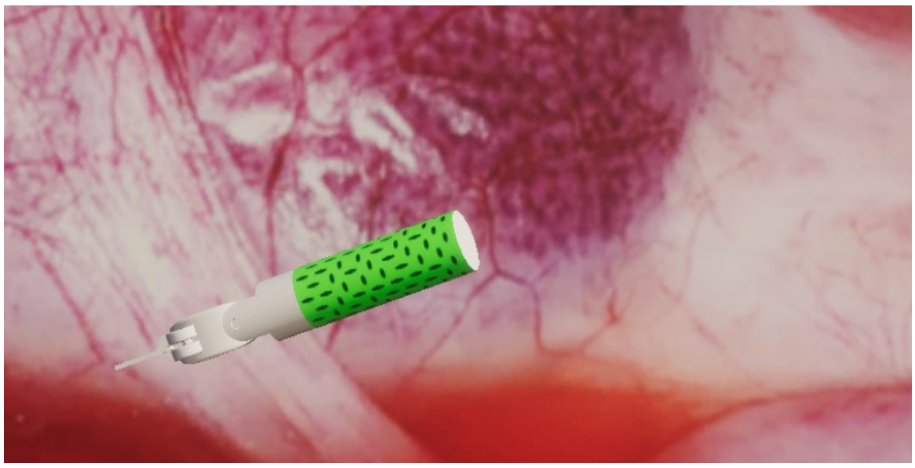


Figure 30: Application of the marker in simulation scenarios on the Needle Driver.

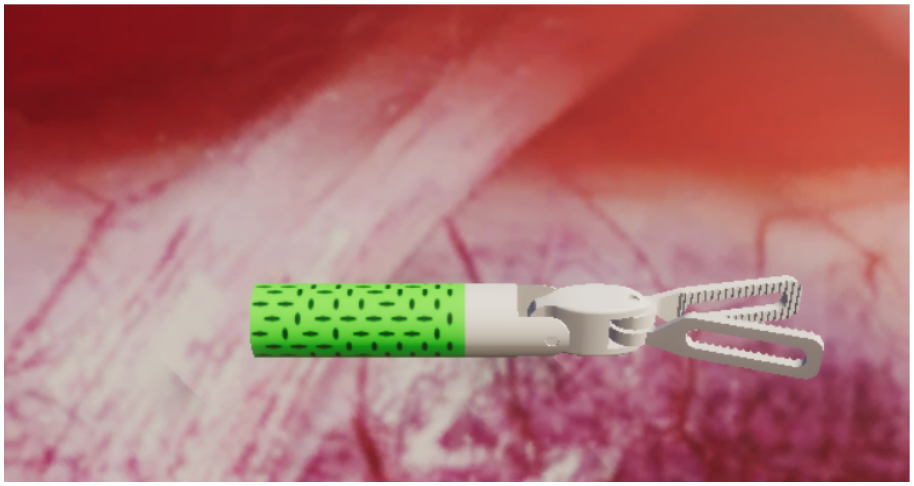


Figure 31: Application of the marker in simulation scenarios on the Cadiere Forceps.

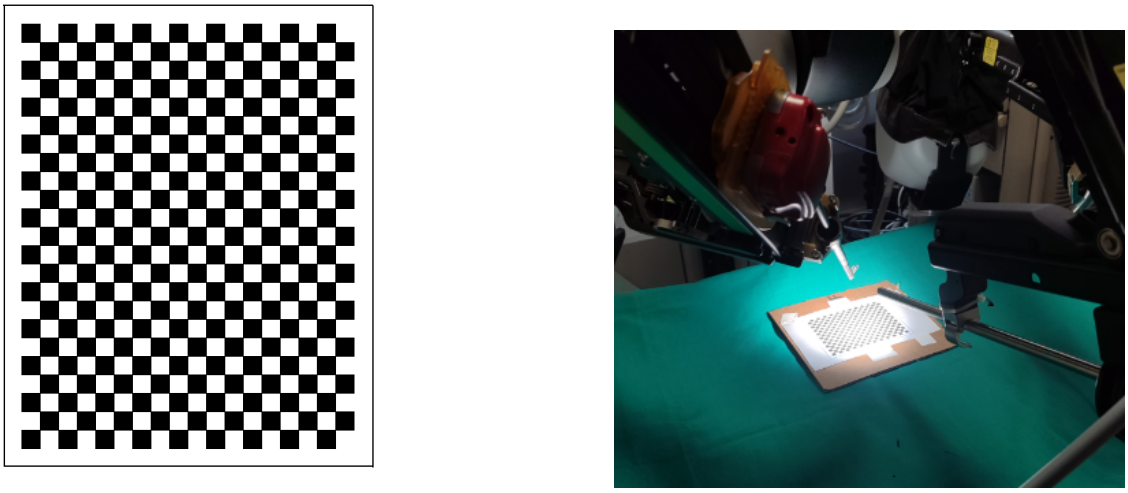
3.3 Dataset generation in the Real World

A real-world dataset was generated to further evaluate the model's performance under non-simulated conditions. However, this data was not acquired during actual surgical procedures. Instead, it was generated in a controlled laboratory setting using the dVRK, where the robotic system was manually positioned through specific commands. This setup allows for realistic data acquisition while avoiding the complexities and restrictions mentioned before. Moreover, obtaining accurate GT data in real surgical settings is particularly challenging, and such datasets are rarely available, which further motivates the use of controlled, reproducible setups. In the real-world case, the pose estimation process is conducted only on the Needle Driver, since experiments in the simulated scene show comparable performance across tools, and this choice reduced the processing time. This section outlines the various steps involved in real-world dataset generation. The first step is the calibration of the camera and the acquisition of its distortion parameters. This is followed by the data acquisition phase, which is further divided into four sub-steps. First, the RGB images are acquired and collected. Since *FoundationPose* requires both segmentation masks of the instrument and depth maps of the scene, these are obtained respectively using a widely adopted web-based platform for computer vision tasks and a foundation model. Finally, to evaluate performance, the GT is generated using the marker-based method used in the simulation case.

3.3.1 DVRK Camera Calibration

This step is essential for acquiring the intrinsic parameters of the camera on the dVRK, specifically the intrinsic matrix K and the distortion parameters. The K matrix includes the focal lengths (f_x, f_y) and the optical center (c_x, c_y), which are essential for converting between 3D world coordinates and 2D image coordinates. The distortion parameters describe how the camera lens affects the image. Unlike the simulation case, in this scenario, it is necessary to calibrate the camera manually. Zhang's method [70] is a widely used and robust approach for camera calibration. It relies on capturing multiple images of a planar checkerboard (calibration pattern) from different viewpoints. In each image, the 2D positions of the checkerboard corners are detected, and since their 3D locations on the board are known, the algorithm estimates how the camera projects 3D points into the 2D image plane. In this work, the intrinsic calibration of the endoscopic camera is performed using Zhang's approach, through a pipeline based on ROS (Robot Operating System) that processes multiple checkerboard images and uses the appropriate OpenCV calibration function. First of all, the camera calibration pattern (100x125) mm was printed. The pattern was then fixed to a rigid, flat surface to prevent any bending, and placed approximately 20 cm away from the camera. The camera was focused at this working distance to ensure accurate calibration. Images are recorded from the camera in a ROS environment. The camera feed is continuously displayed and updated, allowing the user to manually save frames when the checkerboard is correctly detected. 50 pictures of the calibration pattern in different poses were taken, while keeping the distance between the camera and the pattern around 20 cm and adjusting the brightness of the light source, so that the pattern is well detected. The images had to be without reflections and not too dark. The rectangular line had to be inside the field of view of the camera, and the pictures had to be taken with the pattern inclined with respect to the image plane (less than 45 °).

It's important to note that the images are acquired with a 1920x1080 resolution, but due to the high computational costs of *FoundationPose*, it was necessary to reduce the resolution to 640×360, to preserve the original aspect ratio, avoiding any distortion of the image. When applying resizing, the intrinsic camera matrix's focal lengths and optical center values need to be scaled by the same resizing factor. Consequently, a scaled version of the original calibration matrix is obtained. The distortion parameters, however, remain unchanged since they are independent of the image resolution.

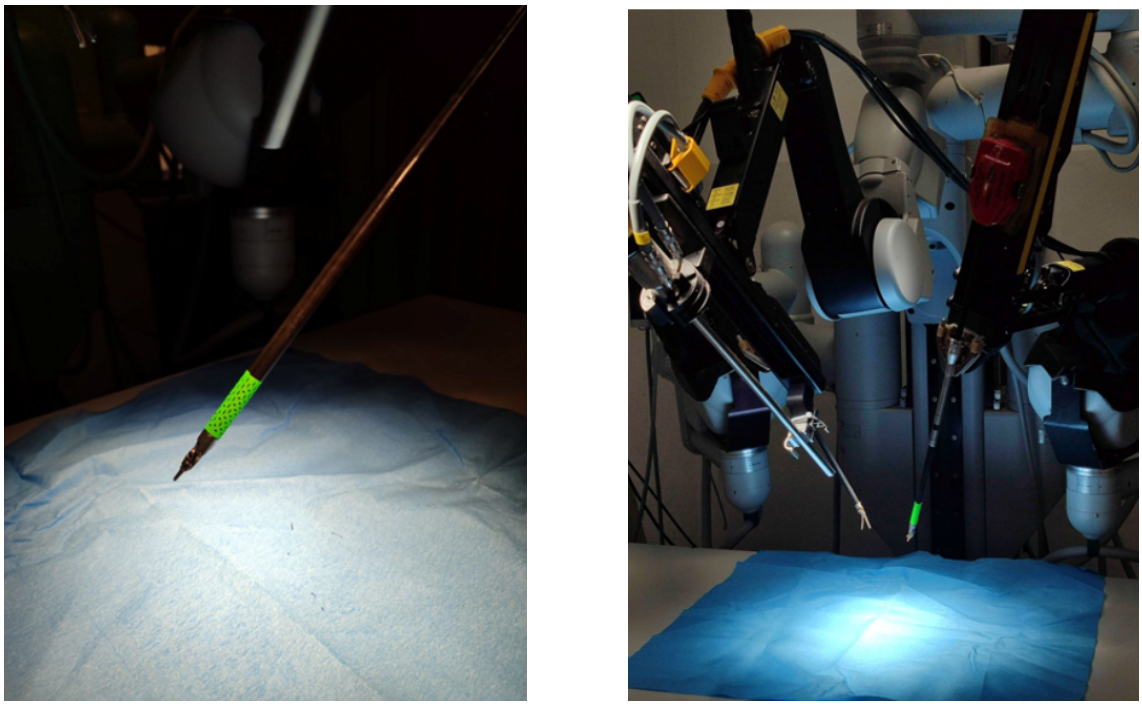


(a) (b)

Figure 32: (a) Camera Calibration Pattern (b) Camera Calibration Process

3.3.2 Data Acquisition

To acquire the data, the marker was printed on green paper and attached to the handle of the Needle Driver. With the camera on, the instrument was moved through multiple poses, and each pose was recorded, resulting in a total of 15 acquisitions. The marker was kept close to the camera throughout the process, at an approximate distance of 10 cm. RGB images and the corresponding joint states of the robot are recorded.



(a) (b)

Figure 33: (a) Application of the marker on the Needle Driver in real-world settings. (b) Acquisition of the images in the real world.

At initialization, configuration parameters are loaded, including the ROS topics for the camera and the robot's joint states. The image and the corresponding joint angles, indexed by a unique pose ID, are stored. This allows for the creation of a dataset where each recorded image has a matching robot configuration. After saving the recorded poses, another script is used to extend the dataset. It begins by loading previously saved joint states and interpolates a fixed number of intermediate steps between each consecutive pair of poses. This interpolation is applied only to the first three joints, while the wrist joints are kept fixed to avoid unwanted rotations of the end-effector. During execution, the robot is commanded to move to each interpolated pose sequentially. After each movement, the system waits for the robot to stabilize and then records a synchronized RGB image from the camera and the corresponding joint values and saves them. The approach allows for the generation of a large number of diverse viewpoints of the tool and the attached marker. By interpolating the motion, it also ensures smooth transitions and avoids sudden changes that could introduce motion blur or inaccuracies in the data collection process. At the end of the process, 150 unique poses are acquired.

To evaluate *FoundationPose* on real data, a preliminary data preparation step is required, which involves generating both segmentation masks of the instrument and

depth maps of the scene, for each frame. The images are segmented using RoboFlow [93], a platform that simplifies and speeds up the creation and management of computer vision datasets. It supports image annotation, dataset organization, and export in various formats, making it especially useful for tasks like object detection and segmentation. Through its interface, it was possible to manually select and label the relevant regions of interest within each image. RoboFlow then automatically converted these annotations into precise segmentation masks. This platform significantly streamlined the segmentation of the instrument, offering an efficient and faster alternative to other annotation methods.

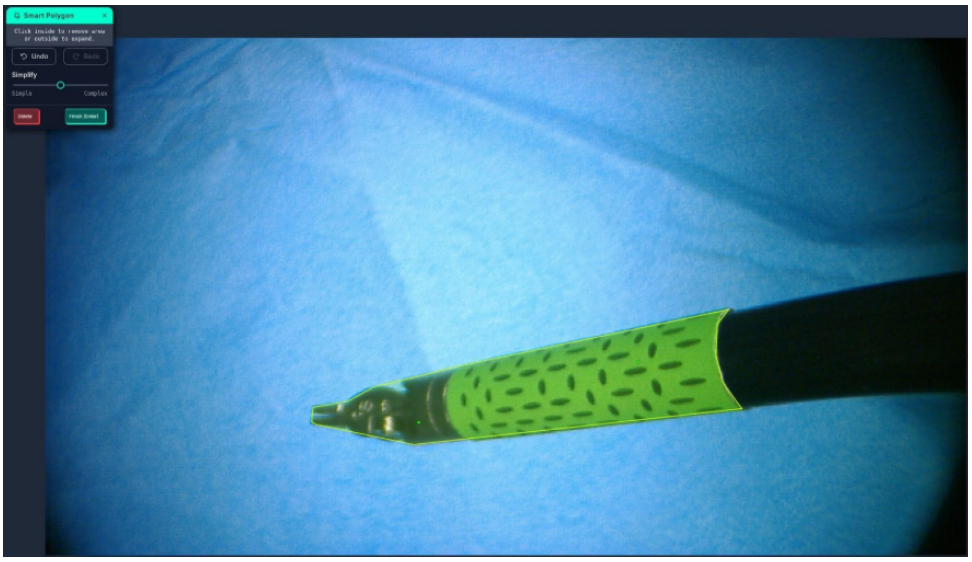


Figure 34: Selection of the area that represents the Needle Driver.

The segmentation masks are generated using a separate script, setting the background to black (RGB value [0, 0, 0]) and the instrument to white (RGB value [255, 255, 255])

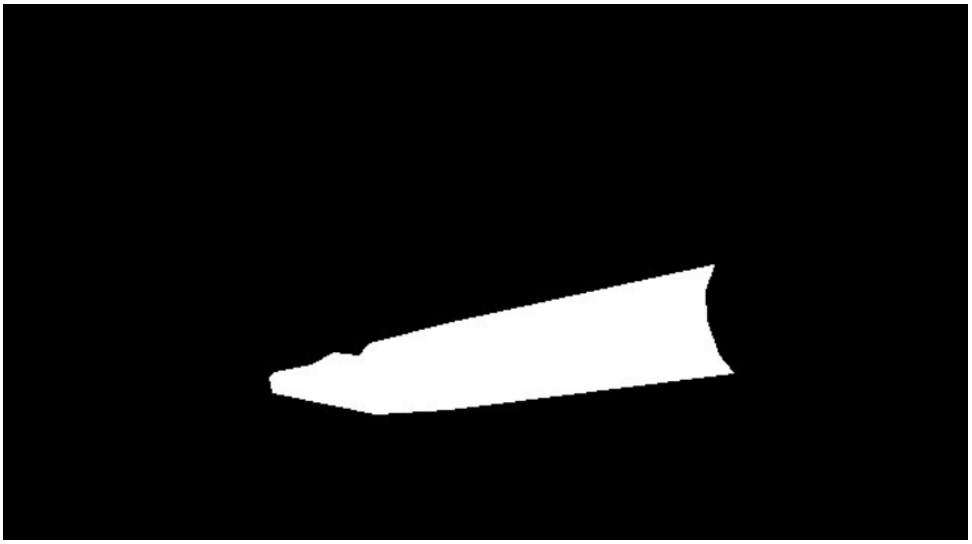


Figure 35: Segmentation mask of the Needle Driver in real world settings.

For the generation of depth maps, the *DepthAnything* model [94] was employed. *DepthAnything* is a deep learning model designed to estimate depth information from images. It uses a single RGB image to predict the relative depth of each pixel. The model is trained to recognize depth cues from the image, such as object size, texture, and perspective, and generates a depth map where each pixel represents the relative distance of objects from the camera. *DepthAnything* is designed as a foundation model for monocular depth estimation (MDE). It is trained on labeled and 62M unlabeled images to enhance the dataset. It uses a pre-trained *DINO* model as an image encoder to inherit its existing rich semantic priors, and *DPT* as the decoder. *DINO* (Distillation with No Labels) is a foundation model, based on transformer architecture, that extracts features for applications such as image classification and depth estimation. It learns visual representations from unlabeled images, avoiding in this way the need for manual annotations. *DPT* (Dense Prediction Transformer) is built on the same architecture but is specifically optimized for dense prediction tasks (tasks in which the prediction is made for every pixel in the image). Thanks to these two models, *DepthAnything* is able to recognize depth cues from the image, such as object size, texture, and perspective, and generate a depth map where each pixel represents the relative distance of objects from the camera. However, *FoundationPose* requires absolute depth values. While the depth information provided is highly useful, the values are relative and scaled to the scene, meaning they do not correspond to absolute distances unless additional processing is performed. To convert the relative maps into absolute ones, the distance from the camera to a known point in the scene was manually measured and used as a reference. The corresponding relative depth value at that pixel was then matched

to the real-world distance, allowing all other values in the depth map to be scaled accordingly and converted into metric units.

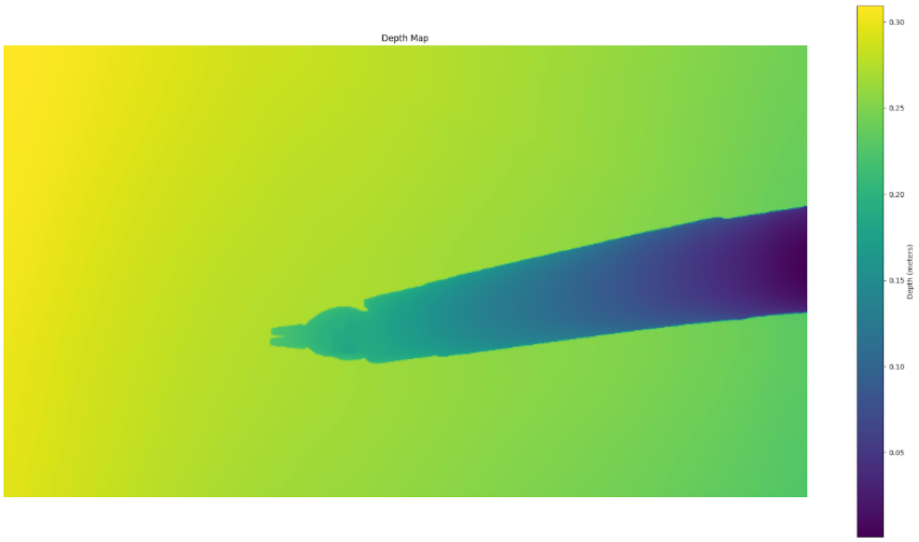


Figure 36: Absolute Depth Map of the Needle Driver in real world settings.

The marker-based approach was employed for GT generation, as in the simulated scenario. This approach requires only the captured images, the camera calibration matrix, and the distortion parameters previously obtained. To guarantee precise marker detection, the HSV values were manually modified considering the obtained images. The method returned the transformation matrix representing the pose of the Needle Driver relative to the camera.

Figure 37: Ground Truth (GT) pose generation of the Needle Driver in real world settings.

3.4 Metrics

Positional and orientational accuracy are assessed using different metrics.
Position Estimation is evaluated through the Mean Translation Error, measured in centimeters, representing the average Euclidean distance between the predicted and ground truth positions over N frames:

$$\text{Mean Translation Error} = \frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{t}^{\text{i}}_{\text{pred}} - \mathbf{t}^{\text{i}}_{\text{gt}}\right\|^2 \tag{22}$$

For orientation, the Mean Absolute Cosine Similarity is used to compute the angular error between predicted and ground truth axes.
Given two normalized vectors u and v, the Absolute Cosine Similarity is defined as:

$$\text{Absolute Cosine Similarity} = \left|\frac{\mathbf{u}\cdot\mathbf{v}}{\|\mathbf{u}\|\|\mathbf{v}\|}\right| = |\cos(\theta)| \tag{23}$$

- A value of 1 indicates perfect alignment (parallel or antiparallel).
- A value of 0 indicates orthogonality (perpendicular directions).

Due to possible axis permutations in the predicted coordinate frame, the cosine similarity is computed between each GT axis and all predicted axes. The axis with the highest similarity is selected as the match.
The Mean Absolute Cosine Similarity across all frames is computed as:

$$\text{Mean Absolute Cosine Similarity} = \frac{1}{N}\sum_{i=1}^{N}|\cos(\theta_i)| \tag{24}$$

To complement this, the Mean Rotation Error is calculated as the average angular deviation between aligned predicted and GT axes:

$$\theta_i = \arccos(\cos(\theta_i)) \tag{25}$$

The average over all frames provides the final orientation error:

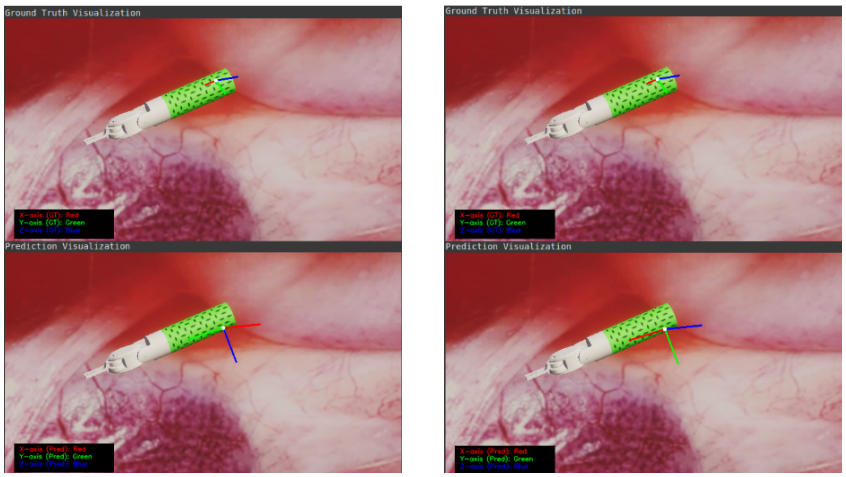
$$\text{Mean Rotation Error} = \frac{1}{N}\sum_{i=1}^{N}\theta_i \tag{26}$$

4 Results

This section presents both the qualitative and quantitative results of the thesis work. It includes comparative visualizations of the estimated poses of the dVRK tools in both simulation and real-world settings. A quantitative evaluation of the performance is provided for the model-based approach *FoundationPose* using the marker-based method as ground truth (GT), as mentioned in the Methods chapter. The marker-based method is already solid and has been used in previous studies. However, to provide additional visual confirmation of its reliability, the instrument was placed in a known position in the simulation scenario, generating 50 identical frames. The method proved to be robust and consistent, showing the same pose across all frames, visually matching the one that was set. The current limitations and challenges of the two approaches are then discussed, together with possible future improvements.

4.1 Pose Estimation in Simulation

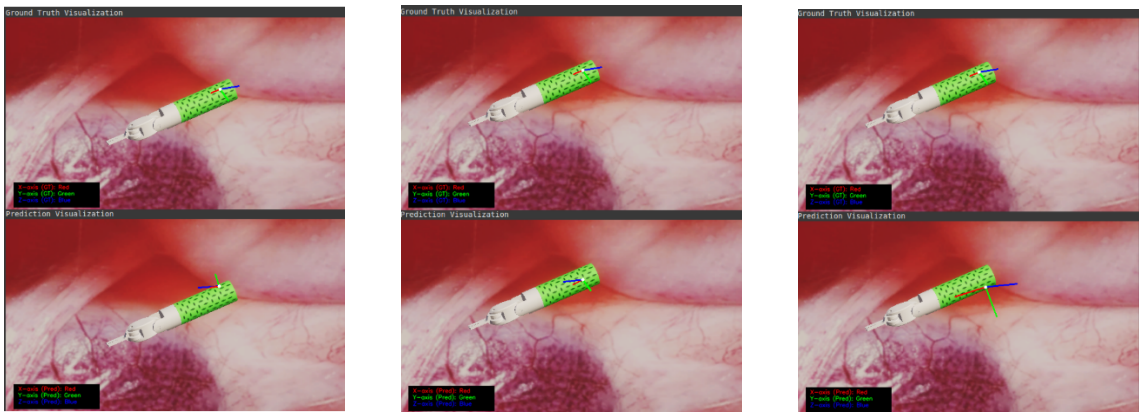
Several tests were conducted using different datasets, varying in the number of images and the orientation of the CAD model's reference frame. A recurring issue was observed in *FoundationPose*'s estimations: the model irregularly swaps axes with respect to the GT. Fig. 38a shows a visualization of the pose estimated by *FoundationPose* for a single frame, and Fig. 38b shows the same visualization after axis swapping according to GT. Since the axis permutation is not consistent across frames, applying a unified post-processing step becomes unfeasible.



(a) (b)

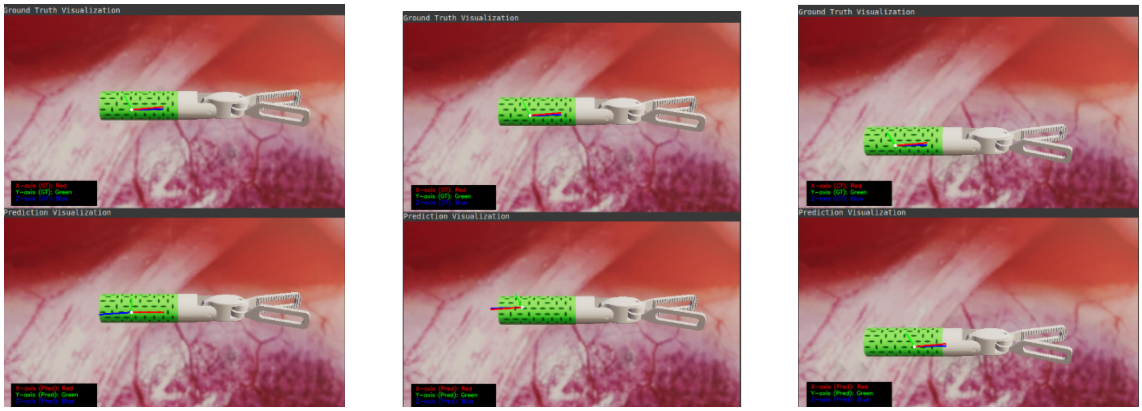
Figure 38: (a) Visualization of *FoundationPose* estimation w.r.t. GT in simulation scenario. (b) Same visualization with *FoundationPose* estimated axes swapped according to GT.

Another issue encountered was the ambiguity in the predicted axis signs. In several frames, the predicted orientation appeared visually similar to the ground truth in terms of direction, but the sign of one or more axes were reversed, as shown in Fig. 39 and Fig. 40 (a) and (b). In (c) it's shown a good orientation prediction.



(a) (b) (c)

Figure 39: Visualization of *FoundationPose* Estimation of the Needle Driver with respect to GT in Simulation Scenarios.



(a) (b) (c)

Figure 40: Visualization of *FoundationPose* Estimation of the Cadiere Forceps with respect to GT in Simulation Scenarios.

Considering the issues encountered, absolute cosine similarity was chosen to quantify orientation. The predicted axis with the highest cosine similarity to GT was selected, as it represents the best alignment between axes. Initially, the comparison focused only on the longitudinal axis, since it appeared to be the most accurately predicted based on visual inspection. However, the evaluation was later extended to all three axes. To resolve the axis sign ambiguity, the absolute value of the cosine similarity was used, allowing to focus on the predicted direction regardless of sign. Cosine similarity provides a general indication of the quality of orientation estimation. However, for a more precise evaluation, the Mean Rotation Error across N frames is computed and reported. Given the considerable standard deviations observed in the results, the average cosine similarity alone may not fully capture the performance of the orientation estimation. Therefore, Fig. 41 and 42 show the Cosine Similarity distribution and the per-frame Cosine Similarity values for the three axes for the Needle Driver and the Cadiere Forceps, respectively, providing a more detailed evaluation of the predictions. As previously mentioned, the Mean Translation Error across N frames was used to evaluate position estimation.

Table 1: **Results of position estimation evaluation for dVRK tools in Simulation Scenario**

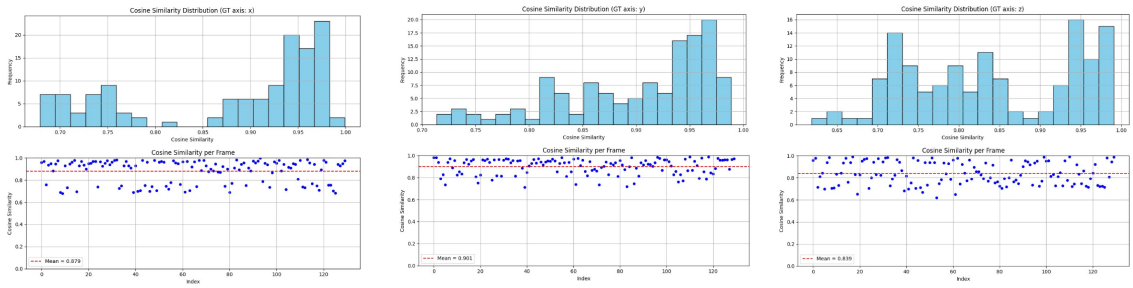
	Mean Translation Error (cm) ± STD		
	X	**Y**	**Z**
Needle Driver	0.92 ± 0.77	1.58 ± 0.75	7.21 ± 1.44
Cadiere Forceps	0.94 ± 0.91	1.44 ± 1.27	6.98 ± 1.35

Table 2: **Results of position estimation evaluation for dVRK tools in Simulation Scenario, in terms of Mean Absolute Cosine Similarity.**

	Mean Cosine Similarity ± STD		
	X - axis	**Y - axis**	**Z - axis**
Needle Driver	0.88 ± 0.10	0.90 ± 0.07	0.84 ± 0.10
Cadiere Forceps	0.89 ± 0.11	0.87 ± 0.09	0.83 ± 0.10

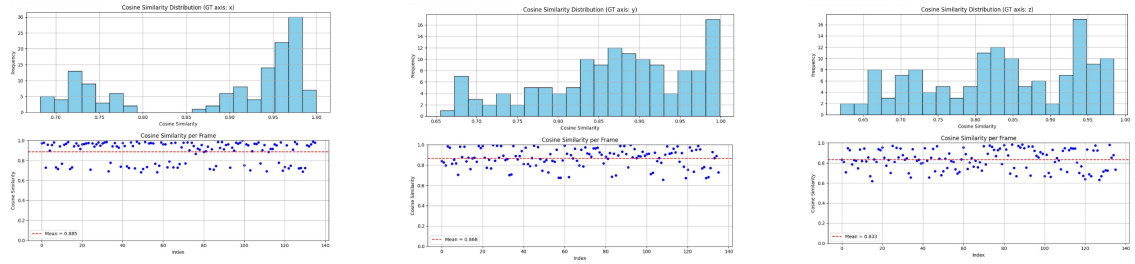
Table 3: **Results of position estimation evaluation for dVRK tools in Simulation Scenario, in terms of Mean Orientation Error.**

	Mean Orientation Error (°) ± STD		
	Roll	**Pitch**	**Yaw**
Needle Driver	12.22 ± 9.15	9.90 ± 8.05	18.43 ± 10.58
Cadiere Forceps	11.05 ± 8.27	14.09 ± 13.23	21.65 ± 12.08



(a) X-axis (b) Y-axis (c) Z-axis

Figure 41: Visualization of Cosine Similarity Distribution and Cosine Similarity per Frame in simulation scenarios, for each axis of the Needle Driver.

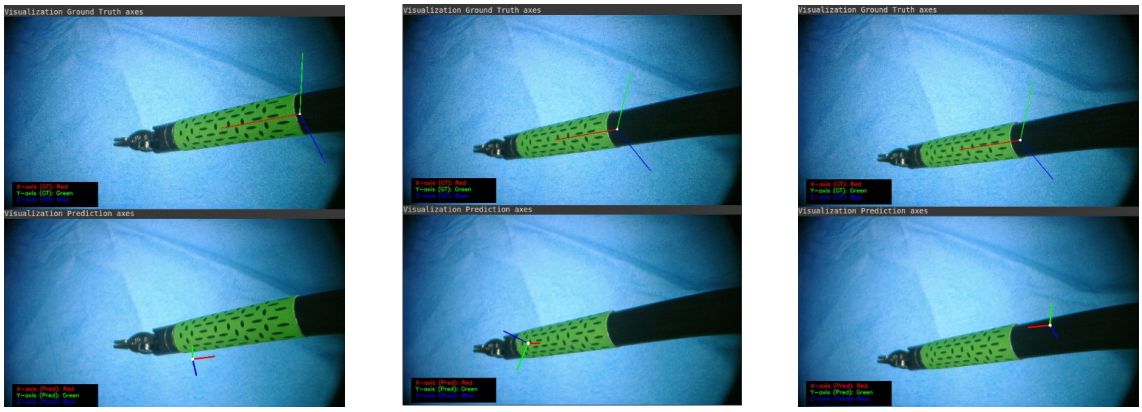


(a) X-axis (b) Y-axis (c) Z-axis

Figure 42: Visualization of Cosine Similarity Distribution and Cosine Similarity per Frame in Simulation Scenarios, for each axis of the Cadiere Forceps.

4.2 Pose Estimation in Real Environment

The same issues observed in the testing on the simulation dataset were also found in the real-world pose estimation. As shown in Fig. 43 (a) and (b), the model often predicts the correct axis direction, but with the opposite sign. 43 (c) shows a good axes alignment. In the figure, axis swapping has been already applied to account for the permutation issues affecting the model.



(a) (b) (c)

Figure 43: Visualization of *FoundationPose* Estimation of the Needle Driver in real world settings w.r.t GT.

The metrics used for the quantitative evaluation are the same of the ones used for the Simulation case and are reported above.

Table 4: **Results of position estimation evaluation for dVRK tool Needle Driver in Real World setting.**

Mean Translation Error (cm) ± STD			
	X	**Y**	**Z**
Needle Driver	3.24 ± 2.42	2.48 ± 1.96	16.86 ± 6.09

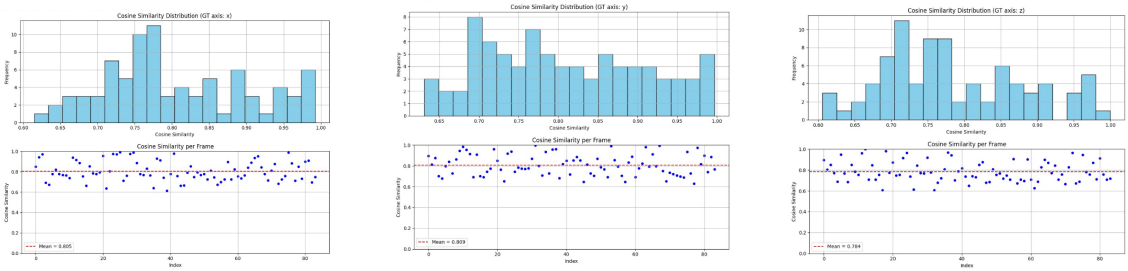
Table 5: **Results of orientation estimation evaluation for dVRK tool Needle Driver in Real World setting, in terms of Mean Cosine Similarity.**

Mean Absolute Cosine Similarity ± STD			
	X-axis	**Y-axis**	**Z-axis**
Needle Driver	0.80 ± 0.09	0.81 ± 0.10	0.78 ± 0.10

Table 6: **Results of orientation estimation evaluation for dVRK tool Needle Driver in Real World setting, in terms of Mean Orientation Error.**

	Mean Orientation Error (°) ± STD		
	Roll	**Pitch**	**Yaw**
Needle Driver	23.68 ± 13.01	21.88 ± 19.59	27.91 ± 13.92

As in the Simulation Case, visual representations of Cosine Similarity Distribution and per-frame Cosine Similarity are shown in Fig. 44.



(a) (b) (c)

Figure 44: Visualization of Cosine Similarity Distribution and Cosine Similarity per Frame in real world settings, for each axis, for the Needle Driver.

4.3 Discussion

4.3.1 Result Analysis

Regarding the tests of *FoundationPose* on Simulation scenarios, position errors on the X and Y axes ranged from 0.9 to 1.6 cm across both tools, with the Needle Driver showing slightly better performance overall. These results are promising; however, in surgical scenarios, even small errors can be critical. dVRK tools are very small (about 10 cm in length with an approximate radius of 8 mm), so while such errors may be acceptable in general object pose estimation, higher accuracy is required in this context. Depth estimation is the least accurate, with errors around 7 cm for both tools. Considering the orientation estimation, both tools present similar rotational errors. Specifically, errors of approximately 14° (with cosine similarity scores between 0.88 and 0.89) were observed for Roll, corresponding to rotation around the longitudinal X-axis; around 10° (cosine similarity between 0.87 and 0.90) for Pitch, rotation around the Y-axis—which appears to be the most accurately predicted;

and about 20° (cosine similarity between 0.83 and 0.84) for Yaw, rotation around the Z-axis. The standard deviations indicate that the model occasionally performs very well, with errors of 1°, but also shows peak errors of 30°. In particular, for the Needle Driver, the cosine similarity distribution for the X and Y axes is more concentrated toward values close to 1. Specifically, the highest frequency is observed at a cosine similarity of 0.97 for both axes. In contrast, the distribution for the Z axis is more uniform, although it still peaks at 0.97. Overall, the visual distribution suggests that the model generally performs well, but with some variability, as confirmed by the spread of the individual frames. Some outliers deviate from the average predictions. For the Cadiere Forceps, the cosine similarity distributions appear slightly different, with a frequency peak at 0.97 for the X axis. The Y-axis distribution is more uniform, but in this case, the peak reaches 1, which indicates perfect alignment between the predicted and ground truth axes. The same holds for the Z axis, although its peak is lower, at 0.94. Outliers are also visible here when looking at the per-frame distribution, particularly for the Y and Z axes.
In the real world setting, the performance in both position and orientation estimation is slightly worse. Specifically, translation mean errors reach 3.24 cm and 2.48 cm for the X and Y axes, respectively, and 16.86 cm for depth prediction. This significant drop in depth accuracy is most likely due to the approximate method used to generate depth maps in the real setup, given the lack of access to absolute depth data. Moreover, the Orientation Errors are higher for each axis, ranging in values of 21-27° (cosine similarity between 0.78 and 0.81). The Cosine Similarity distributions are more uniform across all axes compared to the simulation tests, with peaks that are shifted to the left, indicating lower values (0.78 for the X-axis, 0.70 for the Y-axis, and 0.73 for the Z-axis). Outliers are more evident in the real-world case, deviating more from the average cosine similarity compared to the simulation setup.

4.3.2 Current Limitations

The performance gap between the simulation and real world scenarios can be attributed to several factors. First, the image quality in the simulation is generally higher, with more visual detail compared to the real-world data. Additionally, due to the high computational cost of *FoundationPose*, it was not feasible to use high-resolution images, which further affected the real world performance. In the simulated dataset, the images only included the handle and the end-effector of the tool, while the real-world images contained the entire instrument, introducing more vari-

ability and visual complexity. In fact, in the real-world setup, the tool typically enters the frame from one corner or one side, and is never fully isolated or centered in the image. Instead, it is attached to the long insertion shaft of the dVRK, which remains visible in the scene. This may negatively affect pose estimation, as the model may struggle to correctly crop the image and therefore fail to focus on the actual geometry of the tool. The performance is promising, but there is certainly room for improvement. Even a 'small' error by the model can lead to a significant prediction error, given the very small scale involved. Although the model was originally designed to avoid the need for retraining, fine-tuning it specifically for surgical instruments and robotic tools could greatly enhance its performance, allowing it to better adapt to the small size of the dVRK instruments.

One major challenge in estimating the pose of surgical robotic tools lies in their geometry. These instruments have symmetries that can mislead the model, particularly in orientation estimation. As previously mentioned, the model often predicts the correct axis direction but with the opposite sign. In addition to this, axis permutation represents one of the main limitations of *FoundationPose*. These issues are likely due to the model's nature: it is not supervised, but instead works by aligning the CAD model to the image. As a result, especially with cylindrical objects, the model may consider the alignment successful because the cylindrical part of the tool in the image overlaps well with the CAD, even when the end-effector is misaligned, particularly at angles where the end-effector is not clearly visible. This issue is further amplified in the real-world case, where segmentation masks are manually generated, although assisted by Roboflow, and may include imperfections around the end-effector area. This often leads the model to rely on clearer features, typically the cylindrical handle, for alignment. Moreover, the Loss functions used during training do not explicitly penalize axis permutations, leading to inconsistent behavior over frames.

4.3.3 Potential Improvements

Applying general-purpose models like *FoundationPose* in surgical robotics is still a relatively unexplored but promising research direction. With some improvements—such as a better definition of the reference frame used for prediction and ensuring consistency across frames, image undistortion, and especially fine-tuning with images from a surgical robotic context—the model's performance could be greatly enhanced. Additionally, using higher-resolution images, supported by more powerful hardware, could further improve accuracy. Future improvements also in-

clude the use of more suitable techniques for generating absolute depth maps, such as leveraging stereo cameras (not used in this work due to incompatibility with the marker-based model used for GT generation), possibly combined with optical flow methods for depth estimation. Since the initial pose hypothesis is also generated from the depth map before being refined, any early-stage error can propagate through the pipeline, leading to an inaccurate final estimate. In the context of simulation, it's easier to obtain RGBD images (RGB + Depth) of Robotic-Assisted MIS scenes, although they are less generalizable. The marker-based approach remains the most accurate among the evaluated methods, but it also presents typical limitations of the use of markers, such as becoming dirty during surgery or partially occluded in some frames. A possible future direction could involve combining strengths from both approaches. For example, the image undistortion step used in the marker-based pipeline could be adopted in the *FoundationPose* workflow to enable more effective cropping, since it is based on the tool's diameter extracted from the image. There are already hybrid methods that combine direct regression approaches (such as *FoundationPose*) with indirect regression (used in the marker-based method). A similar strategy could be explored here, potentially incorporating keypoint detection (using keypoints from the instrument itself without the use of a marker) to help address or mitigate the issues caused by object symmetries, by better aligning the CAD models with the images. Other hybrid approaches could also be used, such as integrating the robot's kinematic data to establish relationships between different parts of the tools and to constrain the pose estimation within physically plausible ranges of motion. It may be interesting to explore the use of real-time pose estimation. As mentioned in the introduction section, the common goal of my research group is the automation of suturing, and in this case, real-time pose estimation can be very useful in the actual surgical procedure. In a robot training phase, the dVRK tools could be trained to reach a specific position and orientation based on data collected from archives of various suturing procedures, allowing the method to generalize, so in this part, it is not essential. However, real-time pose estimation can be useful during the suture, providing feedback and helping to reduce tissue damage. In the context of future full automation of surgical tasks, pose estimation could be used not only for guidance but also as a way to verify whether a task or sub-task has been completed. This feedback could then be leveraged to iteratively teach and improve the robot's behavior.

5 Conclusion

In this thesis, two approaches were used for pose estimation of surgical robotic tools in Minimally Invasive Surgery (MIS), specifically the da Vinci Research Kit (dVRK instruments), but the approaches can be extended to other surgical robots. The task involves predicting the position and orientation of the tools relative to the camera, starting from image frames, in simulation scenarios and in real world settings, in which the instruments are visible. The output is a transformation matrix, for each frame, that includes the translation vector indicating the tool's position, and the rotation matrix indicating the orientation. In the simulation scenario, two dVRK tools were used, in particular Needle Driver and Cadiere Forceps. In the real world, only the Needle Driver was used. The first approach is a marker-based method that estimates the tool's pose by keypoint detection and PnP solving. The second method is a markerless foundation model that uses two AI networks to generate, update and rank the pose hypothesis. 150 frames in simulation were generated in Unity, for each tool, with the help of Blender for the application of the marker. In Unity, essential requirements such as depth maps, segmentation masks, and camera intrinsics were also generated. In the real-world settings, the dVRK was used for the acquisition of 150 images, depth maps were obtained using DepthAnything, and segmentation masks were generated using Roboflow. Calibration was fundamental to obtain camera intrinsics such as the camera intrinsic matrix and undistortion parameters. The marker-based method was employed as the ground truth poses, both in simulation and real world, for evaluating the second approach, given its higher reliability and accuracy. The second model, a markerless foundation model, was applied to the same set of images, and the performances were evaluated with metrics such as Mean Translation Error and Mean Absolute Cosine Similarity, with an indication of the corresponding Mean Rotational Errors to provide a clearer understanding. In simulation, *FoundationPose* achieved reasonable accuracy, with translation errors on the X and Y axes between 0.8 and 1.6 cm, around 7 cm on the Z axis, and rotational errors with cosine similarity between 0.83 and 0.90 (approximately 10–20°). In the real-world case, performance decreased, with translation errors of 2.5–3.2 cm on the X and Y axes, around 16 cm on the Z axis, and rotational errors with cosine similarity between 0.78 and 0.91 (approximately 21–27°). With improvements, fine-tuning, or the integration of *FoundationPose* with other methods based on keypoints or kinematic data, performance could be further enhanced. This work on pose estimation of robotic tools in MIS can be applied to various surgical tasks to provide

feedback and reduce tissue damage, and in the future, it could be used for the full automation of surgical procedures.

AI use Disclosure

In the writing process of this thesis the following tools have been used to improve readability and language:

- DeepL Write (DeepL SE)
- ChatGPT (OpenAI)

Turin, 10/07/2025

References

[1] C. L. Wang, G. Qu, and H. W. Xu. The short- and long-term outcomes of laparoscopic versus open surgery for colorectal cancer: a meta-analysis. *International Journal of Colorectal Disease*, 29(3):309–320, 2014.

[2] R. Watrowski, S. Kostov, and I. Alkatout. Complications in laparoscopic and robotic-assisted surgery: definitions, classifications, incidence and risk factors - an up-to-date review. *Wideochirurgia i Inne Techniki Małoinwazyjne*, 16(3):501–525, 2021.

[3] D. M. Mandi, F. A. Grama, A. Popa, D. E. Giuvara, R. C. Turluianu, A. C. Ilie-Petrov, C. Andrei, R. Scaunasu, T. Burcos, and D. A. Cristian. Aesthetic outcomes and patient satisfaction in laparoscopic vs. open incisional hernia repair: Have we asked the patients? *Chirurgia (Bucur)*, 119(3):260–271, 2024.

[4] G. Dagnino and D. Kundrat. Robot-assistive minimally invasive surgery: trends and future directions. *International Journal of Intelligent Robotics and Applications*, 8:812–826, 2024.

[5] V. Vitiello, S. L. Lee, T. P. Cundy, and G. Z. Yang. Emerging robotic platforms for minimally invasive surgery. *IEEE Reviews in Biomedical Engineering*, 6:111–126, 2013.

[6] O. J. Wagner, M. Hagen, A. Kurmann, S. Horgan, D. Candinas, and S. A. Vorburger. Three-dimensional vision enhances task performance independently of the surgical method. *Surgical Endoscopy*, 26(10):2961–2968, 2012.

[7] M. J. H. Lum, D. C. W. Friedman, G. Sankaranarayanan, H. King, K. Fodero, R. Leuschke, and et al. The raven: Design and validation of a telesurgery system. *International Journal of Robotics Research*, 28:1183–1197, 2009.

[8] G. S. Guthart and J. K. Salisbury. The intuitivetm telesurgery system: Overview and application. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 618–621, 2000.

[9] Y. S. Kwoh, J. Hou, E. A. Jonckheere, and S. Hayati. A robot with improved absolute positioning accuracy for ct guided stereotactic brain surgery. *IEEE Transactions on Biomedical Engineering*, 35(2):153–160, 1988.

[10] S. Lavallee, J. Troccaz, L. Gaborit, P. Cinquin, A. L. Benabid, and D. Hoffmann. Image guided operating robot: a clinical application in stereotactic neurosurgery. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, volume 1, pages 618–624, 1992.

[11] B. Mittelstadt, H. A. Paul, P. Kazanzides, J. Zuhars, B. Williamson, R. Pettitt, P. Cain, D. Kloth, L. Rose, and B. Musits. Development of a surgical robot for cementless total hip replacement. *Robotica*, 11:553–560, 1993.

[12] Junji FURUSHO. Development of a curved multi-tube(cmt) catheter for percutaneous umbilical blood sampling and control methods of cmt catheters for solid organs. *2005 IEEE International Conference on Mechatronics and Automation ICMA2005, 2005. 7.29-8.1*, pages 410–415, 2005.

[13] Patrick Sears and Pierre Dupont. A steerable needle technology using curved concentric tubes. pages 2850–2856, 10 2006.

[14] Robert J Webster III. *Design and mechanics of continuum robots for surgery.* The Johns Hopkins University, 2008.

[15] Zisos Mitros, S.M.Hadi Sadati, Ross Henry, Lyndon Cruz, and Christos Bergeles. From theoretical work to clinical translation: Progress in concentric tube robots. *Annual Review of Control Robotics and Autonomous Systems*, 5, 01 2021.

[16] S. Gorini, M. Quirini, A. Menciassi, G. Pernorio, C. Stefanini, and P. Dario. A novel sma-based actuator for a legged endoscopic capsule. In *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006. BioRob 2006.*, pages 443–449, 2006.

[17] Huaijuan Zhou, Carmen Mayorga Martinez, Salvador Pané, Li Zhang, and Martin Pumera. Magnetically driven micro and nanorobots. *Chemical Reviews*, 121, 03 2021.

[18] Chengzhi Hu, Salvador Pané, and Brad Nelson. Soft micro- and nanorobotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 1, 05 2018.

[19] J. K. Salisbury. The heart of microsurgery. *Mechanical Engineering Magazine, ASME International*, 120(12):47–51, December 1998.

[20] Volkmar Falk, J. McLoughlin, Gary Guthart, John Jr, Thomas Walther, Jan Gummert, and F. Mohr. Dexterity enhancement in endoscopic surgery by

a computer-controlled mechanical wrist. *Minimally Invasive Therapy Allied Technologies*, 8:235–242, 07 2009.

[21] Peter Kazanzidesf, Zihan Chen, Anton Deguet, Gregory Fischer, Russell Taylor, and Simon Dimaio. An open-source research kit for the da vinci® surgical system. pages 6434–6439, 05 2014.

[22] Johns Hopkins University. jhu-dvrk github repository. `https://github.com/jhu-dvrk`, 2025. Accessed: 2025-07-15.

[23] Nural Yilmaz, Brendan Burkhart, Anton Deguet, Peter Kazanzides, and Ugur Tumerdem. Enhancing robotic telesurgery with sensorless haptic feedback. *International journal of computer assisted radiology and surgery*, 19, 04 2024.

[24] Claudia D'Ettorre, Agostino Stilli, George Dwyer, Maxine Tran, and Danail Stoyanov. Autonomous pick-and-place using the dvrk. *International Journal of Computer Assisted Radiology and Surgery*, 16, 05 2021.

[25] Irene Rivas-Blanco, Carlos Jesús Pérez-del-Pulgar, Andrea Mariani, Claudio Quaglia, Giuseppe Tortora, Arianna Menciassi, and Victor F. Muñoz. A surgical dataset from the da vinci research kit for task automation and recognition. *CoRR*, abs/2102.03643, 2021.

[26] Cecilia Molnar, Tamas D. Nagy, Renata Nagyne Elek, and Tamas Haidegger. Visual servoing-based camera control for the da Vinci Surgical System. In *2020 IEEE 18th International Symposium on Intelligent Systems and Informatics (SISY)*. IEEE, 2020.

[27] Rocco Moccia, Cristina Iacono, Bruno Siciliano, and Fanny Ficuciello. Vision-based dynamic virtual fixtures for tools collision avoidance in robotic surgery. *IEEE Robotics and Automation Letters*, PP:1–1, 01 2020.

[28] Mingzhang Pan, Shuo Wang, Jingao Li, Jing Li, Xiuze Yang, and Ke Liang. An automated skill assessment framework based on visual motion signals and a deep neural network in robot-assisted minimally invasive surgery. *Sensors*, 23:4496, 05 2023.

[29] Bogyu Park, Hyeongyu Chi, Bokyung Park, Jiwon Lee, Sunghyun Park, Woo Jin Hyung, and Min-Kook Choi. *Visual Modalities Based Multimodal Fusion for Surgical Phase Recognition*, pages 11–23. 10 2022.

[30] John Wang and Edwin Olson. Apriltag 2: Efficient and robust fiducial detection. pages 4193–4198, 10 2016.

[31] OpenCV. Aruco marker detection — opencv documentation. `https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html`, 2025. Accessed: 2025-07-15.

[32] Jhacson Meza, Lenny Romero, and Andrés Marrugo. Markerpose: Robust real-time planar target tracking for accurate stereo pose estimation. In *CVPR Workshops*, pages 1282–1290, 2021.

[33] Alisa JV Brown, Ali Uneri, Tharindu S De Silva, Amir Manbachi, and Jeffrey H Siewerdsen. Design and validation of an open-source library of dynamic reference frames for research and education in optical tracking. *Journal of Medical Imaging*, 5(2):021215, 2018.

[34] Burak Benligiray, Cihan Topal, and Cuneyt Akinlar. Stag: A stable fiducial marker system. *arXiv preprint*, 2017.

[35] Mark Fiala. Artag, a fiducial marker system using digital techniques. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 590–596, 2005.

[36] M. Kalaitzakis, S. Carroll, A. Ambrosi, C. Whitehead, and N. Vitzilaios. Experimental comparison of fiducial markers for pose estimation. In *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 781–789, 2020.

[37] Chengzhi Hu, Salvador Pané, and Brad Nelson. Soft micro- and nanorobotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 1, 05 2018.

[38] Francisco Romero-Ramirez, Rafael Muñoz-Salinas, and Rafael Medina-Carnicer. Speeded up detection of squared fiducial markers. *Image and Vision Computing*, 76, 2018.

[39] S.M. Abbas, S. Aslam, K. Berns, and A. Muhammad. Analysis and improvements in apriltag based state estimation. *Sensors*, 19(24):5480, 2019.

[40] 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*, 47(6):2280–2292, 2014.

[41] S. Garrido-Jurado, R. Muñoz-Salinas, F.J. Madrid-Cuevas, and R. Medina-Carnicer. Generation of fiducial marker dictionaries using mixed integer linear programming. *Pattern Recognition*, 51:481–491, 2016.

[42] J. Birch, L. Da Cruz, K. Rhode, and C. Bergeles. Trocar localisation for robot-assisted vitreoretinal surgery. *Int J Comput Assist Radiol Surg*, 19(2):191–198, 2024.

[43] Darin Tsui, Capalina Melentyev, Ananya Rajan, Rohan Kumar, and F.E. Talke. An optical tracking approach to computer-assisted surgical navigation via stereoscopic vision. 09 2023.

[44] X. Zhang, J. Wang, X. Dai, S. Shen, and X. Chen. A non-contact interactive system for multimodal surgical robots based on leapmotion and visual tags. *Front Neurosci*, 17:1287053, 2023.

[45] X. Liu, W. Plishker, and R. Shekhar. Hybrid electromagnetic-aruco tracking of laparoscopic ultrasound transducer in laparoscopic video. *J Med Imaging (Bellingham)*, 8(1):015001, 2021.

[46] Francisco Romero-Ramirez, Rafael Muñoz-Salinas, and Rafael Medina-Carnicer. Tracking fiducial markers with discriminative correlation filters. *arXiv preprint*, 2020.

[47] D. Kim, J.H. Bong, and S. Jeong. Enhancing pose estimation using multiple graphical markers with spatial and temporal outlier detection. *Appl. Sci.*, 14:10225, 2024.

[48] A. Gadwe and H. Ren. Real-time 6dof pose estimation of endoscopic instruments using printable markers. *IEEE Sensors Journal*, 19(6):2338–2346, 2019.

[49] P. Pratt, A. Jaeger, A. Hughes-Hallett, E. Mayer, J. Vale, A. Darzi, T. Peters, and G.Z. Yang. Robust ultrasound probe tracking: initial clinical experiences during robot-assisted partial nephrectomy. *Int J Comput Assist Radiol Surg*, 10(12):1905–1913, 2015.

[50] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, Inc., 2008.

[51] Uditha Jayarathne, A. McLeod, Terry Peters, and Elvis Chen. Robust intraoperative us probe tracking using a monocular endoscopic camera. In *MICCAI*, volume 16, pages 363–370, 2013.

[52] C. Yuanwei, M. Hairi Mohd Zaman, and M. Faisal Ibrahim. A review on six degrees of freedom (6d) pose estimation for robotic applications. *IEEE Access*, 12:161002–161017, 2024.

[53] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 2938–2946, Santiago, Chile, 2015.

[54] T. T. Do, M. Cai, T. Pham, and I. Reid. Deep-6dpose: Recovering 6d object pose from a single rgb image. *arXiv preprint arXiv:1802.10367*, 2018.

[55] W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab. Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, pages 1530–1538, Venice, Italy, 2017.

[56] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects, 2024.

[57] Md. Kamrul Hasan, Lilian Calvet, Navid Rabbani, and Adrien Bartoli. Detection, segmentation, and 3d pose estimation of surgical tools using convolutional neural networks and algebraic geometry. *Medical Image Analysis*, 70:101994, 2021.

[58] M. Oberweger, M. Rad, and V. Lepetit. Making deep heatmaps robust to partial occlusions for 3d object pose estimation. In *Proc. Eur. Conf. Comput. Vis. (ECCV)*, pages 119–134, 2018.

[59] C. Song, J. Song, and Q. Huang. Hybridpose: 6d object pose estimation under hybrid representations. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 428–437, Seattle, WA, USA, 2020.

[60] Mitchell Doughty and Nilesh Ghugre. Hmd-egopose: Head-mounted display-based egocentric marker-less tool and hand pose estimation for augmented surgical guidance. *arXiv preprint arXiv:2202.11891*, 2022.

[61] H. Xu and S. Giannarou. Occlusion-robust markerless surgical instrument pose estimation. *Healthc Technol Lett*, 11(6):327–335, 2024.

[62] Zhigang li, Gu Wang, and Xiangyang Ji. Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation. 10 2019.

[63] K. Park, T. Patten, and M. Vincze. Pix2pose: Pixel-wise coordinate regression of objects for 6d pose estimation. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 7667–7676, Seoul, Korea (South), 2019.

[64] S. Zakharov, I. Shugurov, and S. Ilic. Dpod: 6d pose object detector and refiner. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, pages 1941–1950, Seoul, Korea (South), 2019.

[65] P. Ausserlechner, D. Haberger, S. Thalhammer, J.-B. Weibel, and M. Vincze. Zs6d: Zero-shot 6d object pose estimation using vision transformers. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 463–469, 2024.

[66] T. Hodan, D. Baráth, and J. Matas. Epos: Estimating 6d pose of objects with symmetries. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 11700–11709, Seattle, WA, USA, 2020.

[67] J. A. Barragan, J. Zhang, H. Zhou, A. Munawar, and P. Kazanzides. Realistic data generation for 6d pose estimation of surgical instruments. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13347–13353, Yokohama, Japan, 2024.

[68] Rasmus Haugaard and Anders Buch. Surfemb: Dense and continuous correspondence distributions for object pose estimation with learnt surface embeddings. In *Proc. CVPR*, pages 6739–6748, 2022.

[69] Haozheng Xu, Alistair Weld, Chi Xu, Alfie Roddan, Joao Cartucho, Mert Karaoglu, Alexander Ladikos, Yangke Li, Yiping Li, Daiyun Shen, Geonhee Lee, Seyeon Park, Jongho Shin, Lucy Fothergill, Dominic Jones, Pietro Valdastri, Duygu Sarikaya, and Stamatia Giannarou. Surgripe challenge: Benchmark of surgical robot instrument pose estimation. *Medical Image Analysis*, page 103674, 2025.

[70] Zhengyou Zhang. A flexible new technique for camera calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22:1330 – 1334, 12 2000.

[71] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o(n) solution to the pnp problem. *International Journal of Computer Vision*, 81, 02 2009.

[72] Rishi Bommasani, Drew Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney Arx, Michael Bernstein, Jeannette Bohg, Antoine Bosselut, Emma

Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Davis, Dora Demszky, and Percy Liang. On the opportunities and risks of foundation models. 08 2021.

[73] Joao Cartucho, Chiyu Wang, Baoru Huang, Daniel Elson, Ara Darzi, and Stamatia Giannarou. An enhanced marker pattern that achieves improved accuracy in surgical tool tracking. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging Visualization*, 10:1–9, 11 2021.

[74] Abdul Abdulhaq. Solvepnpransac and optimization. `https://medium.com/@abdulhaq.ah/solvepnpransac-and-optimization-47a0683227b1`, 2023. Accessed: 2025-07-15.

[75] CVLab - EPFL. Epnp github repository. `https://github.com/cvlab-epfl/EPnP`, 2023. Accessed: 2025-07-12.

[76] Abdulhaq Ahmad. solvepnpransac and optimization. `https://medium.com/@abdulhaq.ah/solvepnpransac-and-optimization-47a0683227b1`, 2023. Accessed: 2025-07-12.

[77] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2019.

[78] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[79] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[80] Mehdi Cherti, Richard Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Carl Gordon, Claudius Schuhmann, Lukas Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023.

[81] Van-Nam Nguyen, Thibault Groueix, Mathieu Salzmann, and Vincent Lepetit. Gigapose: Fast and robust novel object pose estimation via one correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[82] Maxime Oquab, Thomas Darcet, Tarek Moutakanni, Huy Vo, Maciej Szafraniec, Vadim Khalidov, Pablo Fernandez, David Haziza, Francisco Massa, Ahmed El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[83] Eren P. Örnek, Anirudh K. Krishnan, Sumanth Gayaka, Chih-Hong Kuo, Ankan Sen, Nassir Navab, and Federico Tombari. Supergb-d: Zero-shot instance segmentation in cluttered indoor environments. *IEEE Robotics and Automation Letters (RA-L)*, 2023.

[84] Yilun Wang, Xin Shen, Shuang Hu, Yuchen Yuan, James Crowley, and Didier Vaufreydaz. Self-supervised transformers for unsupervised object discovery using normalized cut. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[85] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021.

[86] Shai Amir, Yedid Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. In *European Conference on Computer Vision Workshops (ECCVW)*, 2022.

[87] Will Goodwin, Shubham Vaze, Ioannis Havoutis, and Ian Posner. Zero-shot category-level object pose estimation. In *European Conference on Computer Vision (ECCV)*, 2022.

[88] Yujun Liu, Ming Zhu, Huan Li, Hao Chen, Xin Wang, and Chunhua Shen. Matcher: Segment anything with one shot using all-purpose feature matching. *arXiv preprint arXiv:2305.13310*, 2023.

[89] Jian Zhang, Christoph Herrmann, Jaesik Hur, Luis P. Cabrera, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. A tale of two features: Stable diffusion complements dino for zero-shot semantic correspondence. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[90] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. 06 2018.

[91] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2961–2969, 2017.

[92] Van Nguyen Nguyen, Thibault Groueix, Georgy Ponimatkin, Vincent Lepetit, and Tomas Hodan. Cnos: A strong baseline for cad-based novel object segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2134–2140, 2023.

[93] Roboflow. Roboflow github repository. `https://github.com/roboflow`, 2023. Accessed: 2025-07-12.

[94] DepthAnything. Depth anything v2. `https://github.com/DepthAnything/Depth-Anything-V2`, 2024. Accessed: 2025-07-12.