POLITECNICO DI TORINO

Master's Degree in Biomedical Engineering

Master's Degree Thesis

Autonomous Endoscopic Camera Navigation with KUKA LBR Med robot

Supervisors

Prof. Kristen M. MEIBURGER

Prof. Alberto AREZZO

Eng. Francesco MARZOLA

Dr. Federica BARONTINI

Dr. Matteo PESCIO

Candidate

LUDOVICA ADAMO

July 2026

Declaration

I hereby declare that the content 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.

This thesis is licensed under a Creative Commons License, Attribution - Noncommercial - NoDerivative works 4.0 International: see www.creativecommons.org. The text may be reproduced for non-commercial purposes, provided that credit is given to the original author.

Turin, 20/07/2026

AI Use Disclosure

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

- Gemini (Google)
- Perplexity
- ChatGPT Plus
- Deep L

Turin, 20/07/2026

Abstract

Minimally invasive surgery (MIS) has become the clinical gold standard for many procedures due to its ability to reduce tissue trauma, postoperative pain, and recovery time. In MIS, endoscopic imaging provides the primary visual interface with the surgical field and is traditionally controlled by a human camera assistant. While this approach remains the standard for surgical visualization, manual camera control can introduce ergonomic fatigue, workflow interruptions, and visual instability, ultimately affecting surgical efficiency and team coordination.

This thesis presents a deterministic, autonomous tracking method based on the KUKA LBR Med collaborative robot. The objective is to autonomously orient the endoscopic camera and provide a stable, responsive field of view of the surgical instrument while strictly preserving the trocar geometric constraint. The main contribution of this work is the integration of classical computer vision and robotic control to enable safe and reliable camera tracking in MIS.

The proposed framework consists of a complete pipeline for autonomous endoscope tracking in minimally invasive surgery. First, surgical tools are localized in the endoscopic image stream. The estimated tool poses are then transmitted to a robotic controller, which generates endoscope motions while enforcing the Remote Center of Motion (RCM) constraint around the entry point of the trocar. Finally, the integrated system is validated in a simulated surgical environment to assess communication performance, visual tracking responsiveness, and geometric safety.

For tool localization, geometric solvers and HSV colour segmentation are employed to estimate the 3D pose of the instruments. The extracted poses are transmitted via UDP to a Java-based robotic controller operating in real time. For robotic camera control, the system keeps the camera view steady while ensuring that the instrument pivots safely around the entry point in the patient's body. A motion control method allows the endoscope to move smoothly, and a rule-based decision making system helps manage tracking changes in a stable way, preventing rapid back-and-forth movements.

Experimental validation was performed using an anatomical phantom and a custom endoscope support. The communication module achieved an average UDP Round-Trip Time (RTT) of 2.48 ± 1.42 ms. During surgical tracking tasks, the integrated system maintained an average RCM spatial deviation of 3.10 ± 1.41 mm, demonstrating accurate constraint preservation while tracking the surgical instruments.

Some limitations remain. The reliance on HSV colour segmentation assumes relatively controlled lighting conditions, and the static geometric definition of the RCM does not dynamically account for severe tissue deformation or patient

respiration. Nevertheless, the results demonstrate that the proposed framework provides a stable, responsive, and anatomically safe camera tracking solution for minimally invasive surgery.

Table of Contents

List of Tables IX

List of Figures X

1 Introduction 1
- 1.1 Context . . . 1
  - 1.1.1 Advantages and disadvantages of MIS . . . 2
  - 1.1.2 Limitations of manual assistance . . . 3
- 1.2 Robotic assistance for laparoscopic camera control . . . 3
  - 1.2.1 From robotic camera holders to autonomous visual tracking 3
  - 1.2.2 Level of autonomy of the proposed system . . . 4
- 1.3 Motivation of the thesis . . . 6
- 1.4 Aim of the thesis . . . 7
- 1.5 Contribution of the thesis . . . 8

2 Background 9
- 2.1 Automation of endoscopic tracking . . . 10
  - 2.1.1 Strategies for managing the field of view and target selection 10
  - 2.1.2 Markerless tracking based on deep learning . . . 12
  - 2.1.3 Deterministic and marker-based geometric tracking . . . 14
- 2.2 RCM constraint in robotic control of the endoscopic camera . . . 16
  - 2.2.1 Kinematics and geometric constraints . . . 17
  - 2.2.2 Redundancy and hierarchical control strategies . . . 20
  - 2.2.3 Admittance control and dynamic RCM . . . 22
- 2.3 Critical summary and methodological gaps . . . 23

3 Materials and methods 26
- 3.1 Robot-camera system configuration . . . 26
  - 3.1.1 KUKA LBR Med robotic manipulator and TCP definition . 26
  - 3.1.2 Functional flow of the developed pipeline . . . 27
- 3.2 Deterministic vision and tracking system . . . 28

3.2.1 Focus adjustment, camera calibration and distortion model . 29
3.2.2 Geometric colour segmentation in HSV colour space 31
3.2.3 Three-dimensional pose estimation 35
3.3 Communication protocol 38
3.3.1 UDP architecture between the vision module and the robot controller 38
3.3.2 Latency management and data synchronisation 39
3.4 Control law and RCM constraint 40
3.4.1 Geometric formulation of the RCM constraint 41
3.4.2 State machine and non-linear tracking logic 44
3.4.3 Dynamic filtering and vibration mitigation 47
3.5 Setup and experimental validation 49
3.5.1 Simulation of the surgical environment using an medical phantom 49
3.5.2 Performance evaluation metrics 52

4 Results 55
4.1 Vision pipeline cycle time 55
4.2 Round-Trip time of the UDP communication 56
4.3 RCM error 58
4.4 Summary of quantitative results 59

5 Discussion 60
5.1 Overall assessment of the integrated system 60
5.2 Vision pipeline performance 61
5.3 The role of UDP communication 62
5.4 Discussion of the RCM error 63
5.5 The influence of the experimental mechanical setup 64
5.6 The role of the state machine and filtering 65
5.7 Limitations of the proposed work 65
5.8 Future work 66

6 Conclusion 68

Bibliography 71

List of Tables

4.1 Summary of experimental results. 59

List of Figures

1.1 Minimally Invasive Surgery (MIS) 2

1.2 Classification of the levels of autonomy for medical robotic systems. Adapted from Yang et al. [19]. 5

2.1 Division of the endoscopic field of view into target, transition and violation zones. The transition zone implements hysteresis: the controller response depends on the previous motion state, reducing rapid switching when the visual target fluctuates close to a threshold. Image adapted from [6]. 10

2.2 Markerless localisation pipeline for surgical instruments: from mask segmentation to skeleton extraction and graph node definition. Image taken from Gruijthuijsen et al. [6]. 13

2.3 Relationship between the outputs defined in the image plane and the quantities expressed in the camera's reference frame during instrument localisation. In the image plane, O_I denotes the image origin and (u, v) represent the pixel coordinates. The white outline represents the segmented mask of the instrument, the blue line indicates the extracted skeleton or centreline, whilst the blue dot identifies the 2D target used for target localisation, for example a centroid, a tip of the skeleton or the projection of the instrument tip. In the camera reference frame F_C, however, the estimation of the 3D pose requires the reconstruction of both the position $p = [x_C, y_C, z_C]^T$ and the orientation R of the tool. The 2D target or the estimated 3D pose can then be used by the robotic controller to generate camera motion whilst satisfying the Remote Centre of Motion constraint. 15

2.4 *Kinematic interpretation of RCM-constrained laparoscope motion.* The trocar defines a Remote Centre of Motion about which the laparoscope can rotate. Admissible motions include pan, tilt, insertion or retraction, and axial roll. A lateral displacement generates an RCM error, defined as the minimum distance between the trocar centre and the laparoscope axis. The image error detected in the endoscopic view must therefore be converted into a camera motion that respects this geometric constraint. 18

3.1 An overview of the main reference systems used. 27
3.2 Functional diagram of the pipeline developed. 28
3.3 Manual focus adjustment for the camera. 29
3.4 Examples of images that are useful for camera calibration. 30
3.5 A cylindrical marker used for visual tracking. It features a green surface, which is useful for segmentation in HSV colour space, and an internal binary pattern for identifying characteristic points. . . . 32
3.6 An example of a binary mask obtained in HSV colour space. The identified areas are candidate regions for the presence of the marker 33
3.7 An example of a foreground mask derived from the candidate region. 34
3.8 Examples of tracking cylindrical markers attached to instruments. (a) Detection of a single marker. (b) Detection of two separate markers, processed independently by the pipeline. 36
3.9 An example of two instruments overlapping. In this situation, the detected area is treated as a single region. 37
3.10 Illustration of the initial configuration used to define the RCM constraint. 42
3.11 Operating modes of the state machine. 45
3.12 Medical phantom used for simulation. 50
3.13 Surgical instruments used during the experimental trials. 51
3.14 Overall configuration of the experimental setup. 52
3.15 Geometric interpretation of the RCM error. 53

4.1 Temporal trend of the vision cycle time. 56
4.2 Distribution of vision cycle times. 57
4.3 Time trend of the UDP Round-Trip time measured between the Python gateway and the KUKA controller. 57
4.4 Time-dependent trend of the RCM error. 58
4.5 Box plot of the RCM error calculated from the 327 valid samples, after excluding the outlier sample of 11.69 mm. 59

Chapter 1

Introduction

Over the last few decades, minimally invasive surgery (MIS) techniques have undergone significant development. They have gradually established themselves as an alternative to conventional open surgical procedures and have become the standard of care in numerous clinical fields. This chapter introduces the clinical and technological context of minimally invasive surgery. It clarifies the motivation for autonomous endoscopic camera control, defines the level of autonomy of the proposed system and outlines the aims and contributions of the thesis.

1.1 Context

Minimally invasive surgery encompasses a range of techniques designed to reduce tissue trauma compared with open surgery. Depending on the procedure, access to the anatomical site may be gained via small incisions, as in laparoscopy, or via natural orifices, as in certain endoscopic procedures.

Direct observation of the surgical field is therefore replaced by an indirect view via a monitor. The endoscopic camera is the main source of visual information and must be continuously oriented to maintain a clear, stable and clinically useful view of the area of interest.

The quality of visualisation is therefore a key factor in minimally invasive procedures. The camera's field of view is limited and may require frequent repositioning during the procedure, raising the issue of endoscopic camera handling. The advantages and limitations of MIS, the challenges of manual assistance and, finally, the role of robotic systems will be discussed.

[1] `https://www.milanomedica.it/area-specialistica_chirurgia-mininvasiva/`

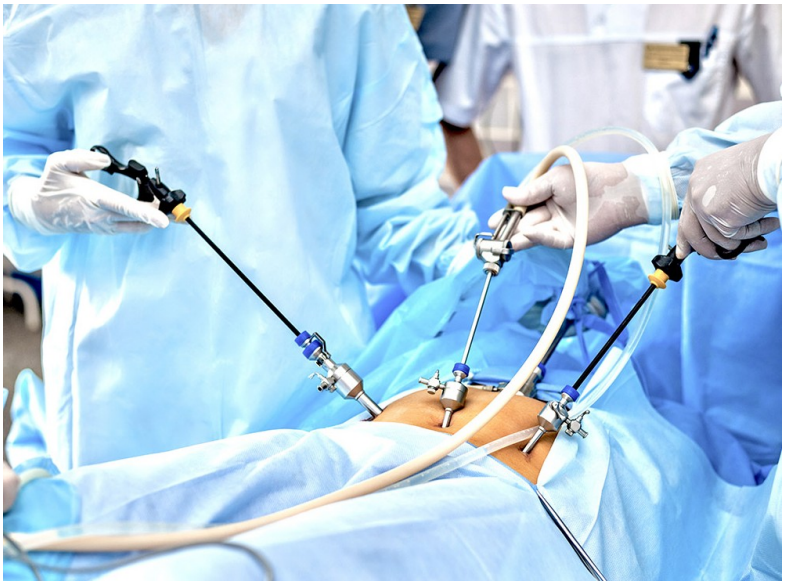


Figure 1.1: Minimally Invasive Surgery (MIS)[1]

1.1.1 Advantages and disadvantages of MIS

In recent years, minimally invasive surgery has become established in numerous clinical fields thanks to its perioperative advantages over open surgery. In general, the use of smaller incisions is associated with reduced surgical trauma, a shorter hospital stay and a quicker return to daily activities [1]. Some studies also report benefits in terms of reduced blood loss and an improved post-operative patient experience, although long-term clinical outcomes may be comparable to those of traditional surgery [1][2][3].

However, these benefits are accompanied by significant technical and ergonomic limitations for the surgical team. The transition from an open surgical approach to a laparoscopic one substantially alters the surgeon's interaction with the surgical field: direct vision is replaced by a monitor display, whilst tissue manipulation is carried out using rigid instruments inserted through dedicated access ports [4].

Laparoscopic instruments and cameras are constrained by the access port, and their movement cannot be planned as if they were moving freely in space. This is the fundamental constraint that robotic camera control must adhere to [5].

From a visual perspective, the most significant limitation for the present work is the loss of direct vision of the surgical site. In laparoscopic procedures, the surgeon relies on the endoscopic camera to obtain information about the surgical field. However, the laparoscope provides a limited field of view and must be correctly oriented to keep both the instruments and the anatomical region of interest visible.

As the surgeon is occupied with manipulating the surgical instruments, control of the camera is normally entrusted to a dedicated assistant [6].

For this reason, the quality of the procedure depends not only on the surgeon's technical skills, but also on the stability and adequacy of the endoscopic view.

1.1.2 Limitations of manual assistance

In traditional laparoscopic practice, the positioning of the camera is often entrusted to an assistant who orients the laparoscope based on verbal instructions from the surgeon. Although this approach is widely used, it introduces several operational limitations directly linked to the quality of the surgical view. The movement of the camera depends on communication between the surgeon and the assistant, which can lead to delays, misunderstandings and camera movements that do not always meet the surgeon's visual requirements [7].

A further limitation concerns image stability. Manual handling of the laparoscope can be affected by physiological tremor, fatigue, distractions or accidental movements, resulting in a deterioration in the continuity of the field of view [8][9][10].

Added to these considerations is the ergonomic burden on the assistant. During lengthy procedures, holding the camera in static or awkward positions can cause physical fatigue and sub-optimal postures [11][12]. Furthermore, as the camera assistant is often a qualified clinical professional, their deployment in a role that primarily involves maintaining and adjusting the field of view may also represent an organisational and economic constraint [13].

These challenges have prompted the development of robotic solutions for assisted or autonomous control of the endoscopic camera.

1.2 Robotic assistance for laparoscopic camera control

1.2.1 From robotic camera holders to autonomous visual tracking

The introduction of robotics into minimally invasive surgery was driven by the need to overcome certain operational limitations of traditional laparoscopy, including reduced stability of the gesture, difficulty in maintaining an adequate view of the surgical field, and reliance on manual assistance during the procedure [4]. The focus

is on systems developed to support or automate the handling of the laparoscopic camera.

A particularly significant contribution was made by early robotic systems designed for camera positioning. AESOP allowed the surgeon to control the positioning of the endoscope using foot pedals and, in later versions, via voice commands [14]. The main aim was to reduce reliance on a human assistant, improving the stability of the view and minimising the effects of tremor and fatigue.

Subsequently, systems such as EndoAssist and ViKY have explored further methods of camera control, based, for example, on tracking the surgeon's head movements or on dedicated motorised controls [15]. These solutions demonstrated the value of robotic camera positioning, but their operation still requires explicit verbal or physical commands from the surgeon and does not allow the system to interpret the surgical scene autonomously.

From this perspective, the most significant development can be described as a shift from systems based on explicit commands from the surgeon to systems capable of automatically tracking the instruments visually. In this context, the robot does not merely maintain a position set by the operator, but uses visual information to update the camera's orientation and keep the surgical instruments within a useful region of the field of view.

At the same time, remote-controlled platforms represent a different paradigm. The surgeon controls the instruments directly from a console rather than relying on the camera to reposition itself autonomously [16][17].

It is against this backdrop that interest in collaborative robots has emerged. Cobots are manipulators designed to share the workspace with the operator, under appropriate safety conditions, and are therefore well-suited to set-ups where the robot operates close to the operating table [18]. In this context, the KUKA LBR Med provides a collaborative robotic platform suitable for investigating constrained endoscopic camera motion.

The aim is to use this platform to autonomously orient the camera according to the position of the instruments, whilst respecting the geometric constraint imposed by the trocar.

1.2.2 Level of autonomy of the proposed system

To describe a robotic system in the medical field accurately, it is not enough to consider its technical performance; it is also necessary to clarify the degree of autonomy with which it operates and the role played by the human operator. In this regard, the taxonomy proposed by Yang et al. provides a useful framework for

distinguishing between systems that merely execute commands issued by the user and those capable of performing specific tasks with increasing levels of operational independence [19].

At the lower end of the classification are systems that are entirely operator-controlled, in which the robot does not make autonomous decisions but translates human intent into movement. At the intermediate levels, however, the system may contribute to the execution of a specific task or autonomously perform a function initiated and supervised by the operator. At the highest levels, autonomy may extend to the planning of operational strategies or the management of increasingly large parts of the clinical procedure [19]. The precise meaning of each level, however, depends on the taxonomy adopted and must be interpreted in relation to the task performed, the degree of human supervision and the associated clinical risk.

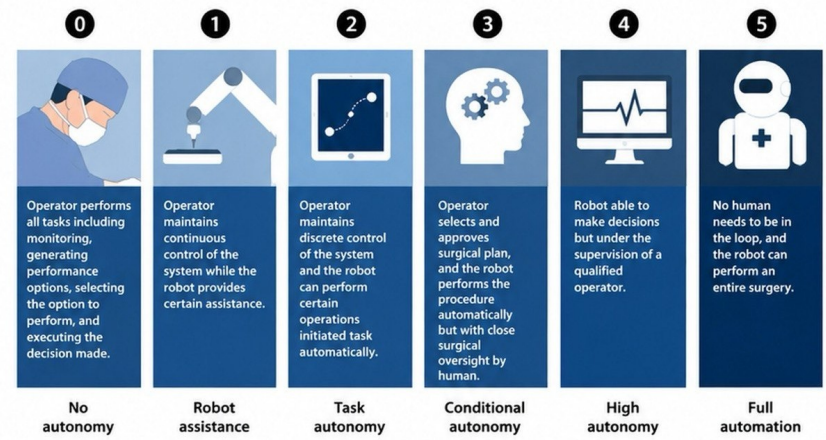


Figure 1.2: Classification of the levels of autonomy for medical robotic systems. Adapted from Yang et al. [19].

In a surgical context, this distinction is particularly important because autonomy must not be interpreted as the system being completely independent of the surgeon. Even when a robot performs a task automatically, the definition of the clinical objective, supervision of the procedure and decision-making responsibility remain with the surgeon. For this reason, any assessment of autonomy must always specify which function is automated and which operational limits are maintained [19].

In light of the classification shown in Figure 1.2, the system developed in this thesis can be classified as a *Level 2* system, corresponding to *task autonomy*. The robot autonomously performs the task of orienting the endoscopic camera based on the tracking of surgical instruments. It determines when to move and how to move, but its autonomy is limited to managing the field of view. The system does

not make clinical decisions, does not perform surgical actions on tissue and does not alter the surgical strategy.

The surgeon therefore retains oversight of the system and control of the procedure. The robot's role is to support the continuity of the endoscopic view, keeping the instruments within a useful region of the field of view and generating movements that are compatible with the geometric constraints imposed by the trocar. In this way, the autonomy introduced relates exclusively to the control of the camera and not to the autonomous performance of the surgical procedure.

1.3 Motivation of the thesis

The main motivation for this thesis stems from the limitations associated with the manual control of the endoscopic camera during laparoscopic procedures. In traditional clinical practice, control of the view is often entrusted to an assistant, who manoeuvres the endoscope based on verbal instructions from the surgeon. This approach can lead to delays between the request and the actual movement of the camera, potential misunderstandings in communication, and a constant reliance on the interaction between the lead surgeon and the assistant [7]. Furthermore, manual control of the laparoscope can be affected by physical fatigue, tremors, distractions or uncomfortable postures, resulting in image instability and reduced continuity of the surgical view [13][12].

The first robotic systems for camera control addressed some of these challenges by introducing motorised mounts controlled by the surgeon, for example via foot pedals or voice commands [15][14]. Although such solutions can improve the stability of the view and reduce reliance on a human assistant, they still require explicit commands from the surgeon. This can disrupt the surgical workflow and increase the cognitive load, particularly during phases when attention should remain focused on instrument manipulation and the progress of the procedure. This has led to interest in systems capable of autonomously orienting the camera based on the tracking of surgical instruments [6].

A second key aspect concerns the compatibility of the camera's movement with the constraint imposed by the trocar. During a laparoscopic procedure, the laparoscope cannot be repositioned freely in space, but must move in a manner compatible with the point of access to the cavity. In robotics, this constraint is described as the *Remote Centre of Motion* (RCM) and requires the instrument's axis to rotate around a fixed point, avoiding unwanted lateral movements relative to the trocar [5]. For an autonomous camera control system, the challenge therefore lies not only in tracking the instruments within the image, but also in generating robotic movements that are consistent with this geometric constraint.

A third factor concerns the stability of the movement and the predictability of the system's behaviour. The visual reference may vary from frame to frame due to instrument movement, measurement noise or minor fluctuations in segmentation. If these variations are directly translated into robotic commands, the result may be an overly reactive movement. This can lead to continuous camera corrections and a possible deterioration in the quality of the visual feed. Furthermore, computationally intensive perception methods can introduce latency, which must be characterised before integration into a real-time robotic control loop.

In light of these considerations, this thesis aims to develop a robotic system for the autonomous orientation of the laparoscopic camera that combines instrument tracking, the generation of movements compatible with the RCM constraint, and control stabilisation strategies. The aim is to reduce reliance on continuous explicit commands, keep the instruments within a useful region of the field of view, and achieve behaviour that is sufficiently stable and predictable for integration into a controlled experimental setup.

1.4 Aim of the thesis

The aim of this thesis is to develop and experimentally evaluate a robotic system for the autonomous orientation of the laparoscopic camera based on the tracking of surgical instruments.

The proposed platform is based on the use of the KUKA LBR Med collaborative manipulator to orient the camera towards a visual target, while maintaining compatibility with the trocar-related Remote Centre of Motion constraint.

In particular, this work aims to achieve the following objectives:

- to localise surgical instruments within the image using a computer vision pipeline;
- to define a control logic for stable camera repositioning;
- to generate movement commands compatible with the RCM constraint imposed by the trocar;
- to introduce a motion-stabilisation strategy to reduce unwanted oscillations and abrupt corrections during tracking;
- to characterise the system's performance in terms of latency, motion stability and geometric deviation from the RCM constraint.

1.5 Contribution of the thesis

The main contribution of this thesis lies in the development and experimental validation of an architecture for controlling the endoscopic camera in minimally invasive surgery. The proposed system integrates visual instrument tracking, communication with the robotic controller and camera movement. The movement must be compatible with the RCM constraint within a single functional pipeline implemented on the KUKA LBR Med platform.

This is a deterministic 'perception-to-control' architecture. The position of the surgical instruments is estimated from the camera image and used as a visual reference for the autonomous orientation of the camera. Tracking is of the geometric type and utilises markers. This yields an interpretable measurement that can be directly integrated into the robotic control loop.

A further contribution is the implementation of a camera motion strategy that takes into account the geometric constraint imposed by the trocar. Indeed, the repositioning of the camera cannot be treated as a free movement in space. This allows the endoscopic view to be updated whilst preserving the fundamental geometric requirement of laparoscopic access.

The thesis also introduces a stabilisation logic aimed at improving the continuity of the camera's movement. The system does not react to every minor variation in the visual target, but modulates the robotic response in order to reduce unnecessary corrections, abrupt movements and oscillatory behaviour. This aspect is particularly relevant when visual measurements are updated discretely and may be influenced by small frame-to-frame fluctuations.

Finally, the architecture was evaluated experimentally in a simulated laparoscopic environment. Visual processing time, UDP communication latency and residual deviation from the RCM constraint were assessed.

Chapter 2

Background

This chapter analyses the main solutions proposed in the literature for the automation of endoscopic tracking in minimally invasive surgery, with a particular focus on the integration of visual perception, robotic control and the Remote Centre of Motion constraint.

This literature review is organised to follow the logical flow of an autonomous tracking system. It first analyses how the visual target is selected within the endoscopic field of view, and then examines the main strategies for localising surgical instruments. A distinction is made between deep learning-based methods and deterministic geometric approaches. The discussion then turns to how to incorporate the constraint imposed by the trocar into robotic control. Kinematic formulations, redundancy-based strategies and admittance-based controls will be considered.

The aim of this chapter is not merely to describe the individual approaches found in the literature, but to highlight the difficulty of integrating them into a complete chain, from perception to robotic movement. In an autonomous system, indeed, the visual localisation of the instrument must be transformed into a stable, smooth command that is compatible with the RCM constraint and with implementation on a commercial surgical cobot.

For this reason, the chapter is organised according to the functional workflow that an autonomous endoscopic tracking system must carry out:

$$\text{image acquisition} \rightarrow \text{instrument localisation} \rightarrow \text{target selection} \rightarrow$$
$$\rightarrow \text{RCM-constrained motion} \rightarrow \text{camera repositioning}$$

2.1 Automation of endoscopic tracking

2.1.1 Strategies for managing the field of view and target selection

One of the most critical aspects of autonomous tracking of the endoscopic camera is image stability. If the system reacts too sensitively to every slight movement of the instrument, the field of view may fluctuate constantly, making it more difficult for the surgeon to interpret what is happening.

To mitigate this problem, several studies have proposed strategies based on Regions of Interest (ROI) or tolerance zones [20]. In these approaches, the field of view is divided into concentric regions: a central zone, or target zone, a transition zone and a violation zone, as shown in Figure 2.1 [6]. The significance of the transition zone is linked to hysteresis logic: the controller's response depends not only on the instantaneous position of the instrument, but also on the system's previous state. Consequently, if the camera was stationary, it may remain so as long as the target remains within a certain threshold; if, on the other hand, it was already in motion, it may continue until the target returns within a more internal threshold. The use of different thresholds for entering and exiting motion avoids rapid switching when the target oscillates near the boundary between two regions.

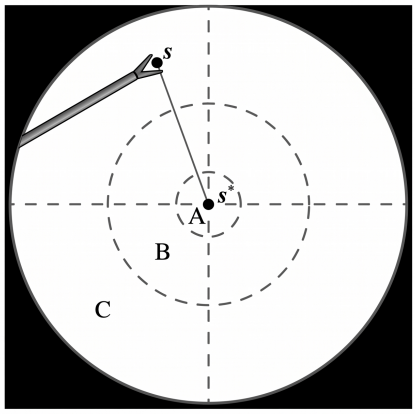


Figure 2.1: Division of the endoscopic field of view into target, transition and violation zones. The transition zone implements hysteresis: the controller response depends on the previous motion state, reducing rapid switching when the visual target fluctuates close to a threshold. Image adapted from [6].

Managing the field of view becomes more complex when there are several surgical instruments in the same scene. In this case, the camera does not necessarily have to track a single instrument, but can be controlled relative to various possible

visual reference points. The target can be defined, for example, as the midpoint between two instruments, as the overall centroid of the visible instruments, or as a working region that must remain within the field of view [21]. In other cases, the target may be chosen based more on the surgeon's preference or the relevance of the anatomical area being observed [22].

The simultaneous presence of multiple instruments also raises an issue relating to the field of view. Some systems incorporate a form of dynamic zoom, in which the camera's insertion or retraction is adjusted according to the apparent distance between the instruments. When the instruments move further apart, the field of view can be widened; when they move closer together, the camera can instead maintain a closer framing. In this way, the scene can be adapted to the configuration of the instruments, keeping them visible whilst the surgical task is being performed [21].

However, the geometric distribution of the instruments does not always coincide with the area of greatest surgical interest. For this reason, some approaches do not select the target solely on the basis of the centre of mass or the distance between the instruments, but also take into account the position and size of the object or region of interest [22]. This makes visual control more consistent with surgical practice, because the aim is not merely to centre the image, but to keep the area most relevant to the surgeon visible.

The architectures analysed show that the management of the field of view cannot be regarded as a purely perceptual problem. The choice of which target to track is, in fact, closely linked to the robotic platform used and the way in which it enables the camera to be controlled. This is evident, for example, in the approaches developed using the da Vinci Research Kit, in which endoscope control is formulated in terms of the kinematics, reference frames and control interfaces specific to the dVRK platform [21][22].

This observation does not diminish the validity of such strategies, but highlights a limitation in terms of practical transferability. A logic based on ROI, midpoint, centroid, dynamic zoom or the surgeon's preferences cannot be directly transferred to a different robotic platform without redefining reference frames, kinematic constraints, control modes and safety requirements. The transition from a platform such as the dVRK to a commercial surgical cobot therefore requires a new integration between visual perception and robotic control.

It is within this context that the system developed in this thesis is situated. The aim is not to propose a strategy that is universally independent of the robot, but to study a solution that can be implemented on a commercial surgical cobot, in which the visual target can be transformed into an autonomous camera movement that is compatible with the Remote Centre of Motion constraint.

Having defined the general principles for selecting the visual target within the field of view, it is necessary to analyse how surgical instruments are localised in the image. Localisation strategies can be categorised into two main groups: deep learning-based approaches and deterministic geometric methods. The former offer high performance in instrument detection and segmentation, but present critical issues relating to dependence on training data, the predictability of computation time and integration into a real-time robotic control loop.

2.1.2 Markerless tracking based on deep learning

Deep neural networks are widely used for the markerless detection and segmentation of surgical instruments. Architectures such as YOLO-v8 [23] or U-Net [6] can provide useful visual information for identifying the instrument within the image. It is, however, important to distinguish between different levels of information: detection identifies the instrument using a bounding box; segmentation reconstructs its mask on a pixel-by-pixel basis; tracking maintains the temporal association between successive frames; whilst target localisation consists of identifying the point actually used by the controller. 3D pose estimation, on the other hand, requires information on the instrument's position and orientation in space and is not automatically provided by a simple detection or segmentation network.

The point to be tracked can be derived from the segmented mask using various strategies. A simple solution is to calculate the centroid of the mask. More structured approaches, on the other hand, extract a skeleton graph of the instrument, from which it is possible to estimate the orientation of the visible portion and identify characteristic points of the surgical structure, as proposed by Gruijthuijsen et al. [6] and illustrated in Figure 2.2. Once the visual target has been identified, the problem concerns not only its position in a single frame, but also its temporal continuity as the camera and instruments move.

To improve temporal continuity, some studies introduce memory-based architectures or advanced segmentation techniques, such as XMem and SAM [24]. Other approaches, by contrast, act directly on the geometric representation of the instrument in the image, for example using Oriented Bounding Boxes optimised with genetic algorithms [22]. These strategies highlight how the markerless problem is not limited to instrument segmentation, but also requires a stable transformation of visual information into a target that can be utilised by robotic control.

Deep learning-based approaches therefore represent a very promising direction for the automation of endoscopic vision. However, when the perceptual module is integrated into a real-time robotic control loop, certain trade-offs arise. Firstly, performance depends on the availability and quality of the training data. The robustness of the system is therefore linked to the dataset's ability to represent

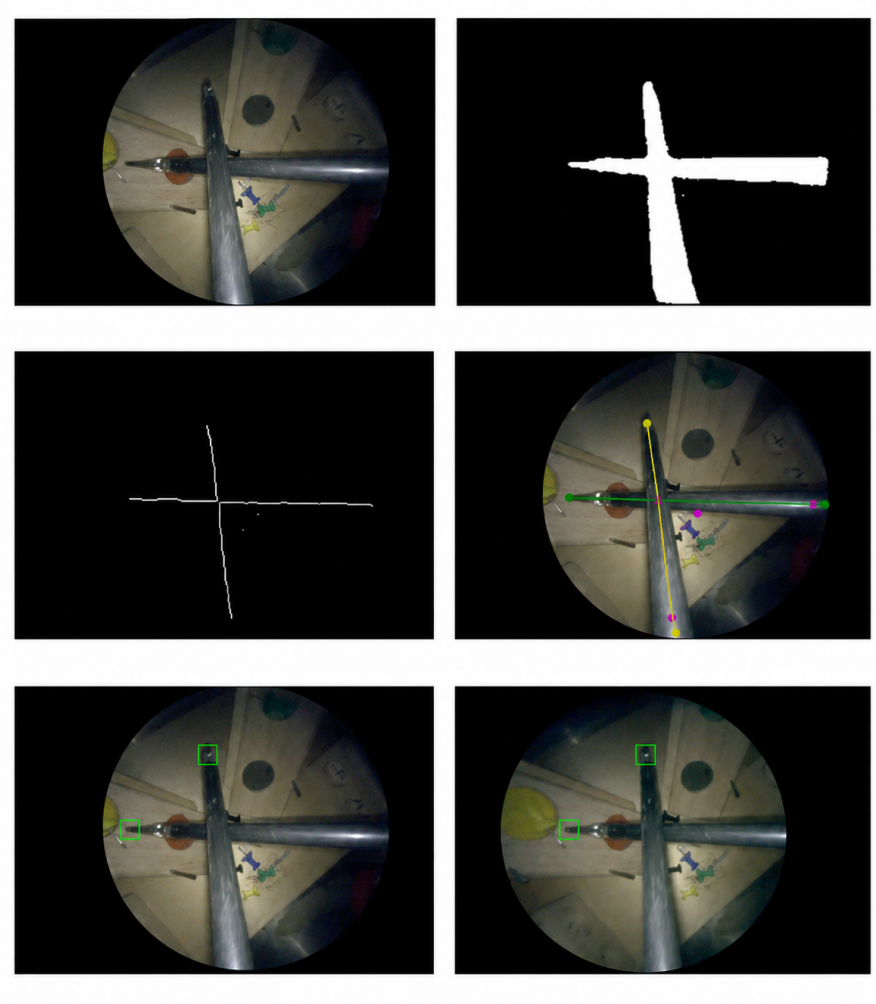


Figure 2.2: Markerless localisation pipeline for surgical instruments: from mask segmentation to skeleton extraction and graph node definition. Image taken from Gruijthuijsen et al. [6].

realistic operating conditions, such as variations in lighting, reflections, the presence of blood, surgical smoke or partial occlusions of the instrument. Under such conditions, even a seemingly accurate segmentation can produce incomplete or unstable masks, altering the target localisation point and propagating the error to the controller.

A second aspect concerns the type of output provided by the perception module. Networks such as YOLO, U-Net or SAM do not necessarily produce a measurement that can be used directly by the robot. They provide bounding boxes, masks or regions of interest, from which the point or quantity relevant to camera control

must subsequently be derived. This transformation introduces a dependency on the quality of the visual output: distorted, incomplete or partially occluded masks can alter the estimated position of the target and generate unwanted corrections in the control loop.

The distinction between visual information in the image plane and geometric measurements in camera space is illustrated in Figure 2.3. The outputs produced by the perceptual module initially belong to the image plane, where the target is expressed in pixel coordinates (u,v). The estimation of 3D pose, on the other hand, requires the reconstruction of the instrument's position and orientation relative to the camera's reference frame.

From a temporal perspective, the latency of neural models cannot be regarded as a fixed property, but depends on the chosen architecture, the available hardware, the image resolution and the software implementation. This aspect becomes particularly relevant when perception is not used for offline analysis or decision support, but directly feeds a robotic controller that must produce continuous, oscillation-free movements. For the controlled experimental setup of this thesis, a deterministic and geometrically interpretable method was therefore chosen, with the aim of obtaining a repeatable measurement of the visual target and a more direct integration with robotic control. This choice does not completely eliminate the problem of processing time, but it makes the perception chain more predictable than approaches based on complex neural models.

2.1.3 Deterministic and marker-based geometric tracking

In systems where visual measurement needs to be integrated into a robotic control loop, geometric methods continue to represent a significant alternative to purely neural approaches. Their value does not necessarily lie in greater perceptual accuracy under all conditions, but in the ability to obtain measurements that are more interpretable and more directly linked to the quantities required by the controller.

One category consists of markerless geometric methods. In these approaches, the instrument is localised without the use of physical markers, instead utilising geometric features of the image or the instrument model. One example is the LINE method, which estimates the instrument's pose by first extracting image gradients, which are useful for identifying the edges separating the instrument from the background [25]. The outline of a three-dimensional CAD model is then projected onto the image and updated iteratively until the match with the observed edges is maximised. In the setup described by the authors, however, convergence times are of the order of seconds per frame, making it difficult to use this specific solution directly for continuous camera control.

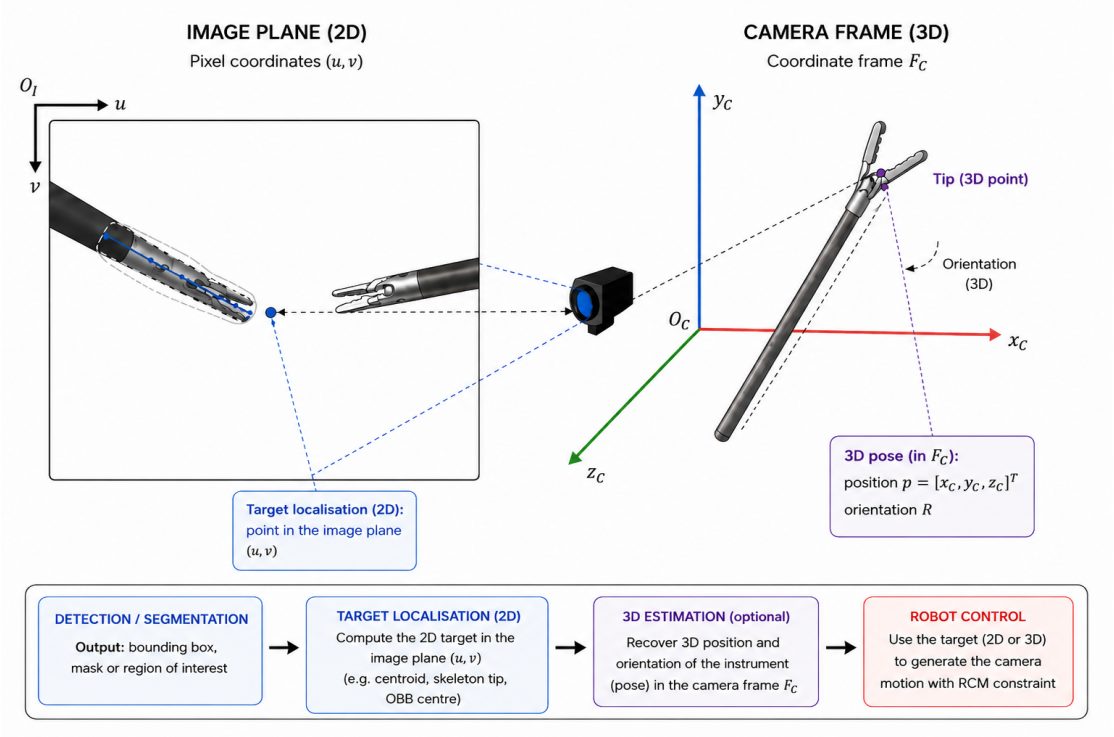


Figure 2.3: Relationship between the outputs defined in the image plane and the quantities expressed in the camera's reference frame during instrument localisation. In the image plane, O_I denotes the image origin and (u, v) represent the pixel coordinates. The white outline represents the segmented mask of the instrument, the blue line indicates the extracted skeleton or centreline, whilst the blue dot identifies the 2D target used for target localisation, for example a centroid, a tip of the skeleton or the projection of the instrument tip. In the camera reference frame F_C, however, the estimation of the 3D pose requires the reconstruction of both the position $p = [x_C, y_C, z_C]^T$ and the orientation R of the tool. The 2D target or the estimated 3D pose can then be used by the robotic controller to generate camera motion whilst satisfying the Remote Centre of Motion constraint.

A second option is to use external tracking systems. In this case, the instrument's position is estimated using physical markers observed by dedicated cameras or sensors, separate from the endoscopic vision system [26]. This solution can provide reliable spatial measurements, but it introduces greater experimental complexity, as it requires additional hardware, calibrations between different reference systems and a less compact configuration than a system based solely on endoscopic imagery.

Between these two extremes lie image-based marker-based methods. In such approaches, visual markers attached to the instrument are identified directly within

the camera's field of view, for example through colour segmentation. In the systems proposed by Ma et al., coloured markers applied near the tip of surgical instruments are detected via colour segmentation; the resulting mask is then analysed to identify contiguous regions of pixels belonging to the marker, filter out spurious detections and calculate a centroid that can be used as a reference for the endoscope's visual servoing [27]. In a subsequent stereo extension, the same principle is employed not only to centre the instruments within the field of view, but also to adjust the insertion and rotation of the endoscopic view [28].

The main advantage of marker-based methods is therefore the availability of a simple, repeatable and geometrically interpretable visual measurement. Compared with markerless methods based on iterative search or CAD models, they reduce the complexity of localisation and facilitate the transition from image to control. However, this introduces certain limitations, as the measurement depends on the visibility of the marker, lighting conditions, the presence of reflections, any contamination from blood or fluids, and possible occlusions of the instrument; furthermore, the generalisability is reduced compared with fully markerless approaches.

However, such limitations may be acceptable if the primary objective is to obtain a measurement that is repeatable and compatible with a robotic control loop. For this reason, marker-based methods represent a useful engineering compromise when the aim is to prioritise the geometric interpretability of the measurement and its direct integration with the controller, whilst acknowledging the dependence on the marker's visibility conditions.

However, there remains a further crucial step: reliable visual localisation of the instrument is not, on its own, sufficient to generate a movement that is kinematically compatible with the laparoscopic procedure. In laparoscopic systems, in fact, the movement of the camera is constrained by the access point in the patient's body, which must be maintained to avoid unwanted lateral translations at the trocar site. Visual localisation must therefore be linked to a robotic control strategy capable of explicitly incorporating the Remote Centre of Motion constraint.

2.2 RCM constraint in robotic control of the endoscopic camera

The Remote Centre of Motion represents the fundamental geometric constraint in robotic systems for minimally invasive surgery. In the case of a rigid laparoscope, the access point through the trocar acts as a pivot. The axis of the instrument may change orientation, but it must continue to pass through a fixed point, ideally coinciding with the centre of the trocar.

The laparoscope must therefore only perform movements that are compatible with this constraint. Permitted movements include rotations around the trocar - commonly referred to as 'pan' and 'tilt' - insertion or retraction along the instrument's axis, and axial rotation, known as 'roll'. Conversely, lateral translations at the trocar must be limited, as they would displace the instrument's axis relative to the access point.

For this reason, the RCM constraint acts as the link between the visual localisation of the instrument and the robotic movement of the camera. A target identified within the field of view cannot be converted into an arbitrary robot movement, but must generate a movement compatible with the pivot point imposed by the trocar. The following section therefore analyses the main strategies used in the literature to incorporate this constraint into robotic control, distinguishing between kinematic formulations, redundancy-based strategies and more adaptive control approaches.

2.2.1 Kinematics and geometric constraints

The kinematic formulations of the RCM constraint can be categorised into three main strategies: the geometric definition of the error relative to the trocar, the integration of the constraint into the visual servoing law, and the use of correction strategies based on inverse kinematics or learning.

In the simplest case, the RCM is treated as a geometric constraint. The idea is that the longitudinal axis of the laparoscope must pass through the centre of the trocar. Consequently, the RCM error can be interpreted as the minimum distance between the centre of the trocar and the axis of the instrument. If all coordinates are expressed in the same robotic reference frame, the axis of the laparoscope can be described by a point on the instrument, denoted as p_{pre}, and by a direction vector $\hat{p}_s$. Let p_r denote the vector connecting p_{pre} to the centre of the trocar; the point on the axis closest to the trocar can be calculated via projection [29]:

$$p_{rcm} = p_{pre} + (p_r^T \hat{p}_s)\hat{p}_s \tag{2.1}$$

The point p_{rcm} therefore represents the virtual point on the laparoscope's axis that best approximates the centre of rotation defined by the trocar. The residual distance between this point and the centre of the trocar constitutes the geometric error to be minimised. In this type of approach, the task error is then mapped onto joint velocity commands via a Jacobian dedicated to the RCM task, denoted as J_{rcm} [29]. In other words, the Jacobian does not convert the error into 'degrees of rotation', but describes the local relationship between joint velocities and the variation of the RCM error in the operational space.

This geometric interpretation is summarised in Figure 2.4, which highlights the role of the trocar as a remote centre of rotation, as well as the movements that are compatible with the constraint and those that are not permitted.

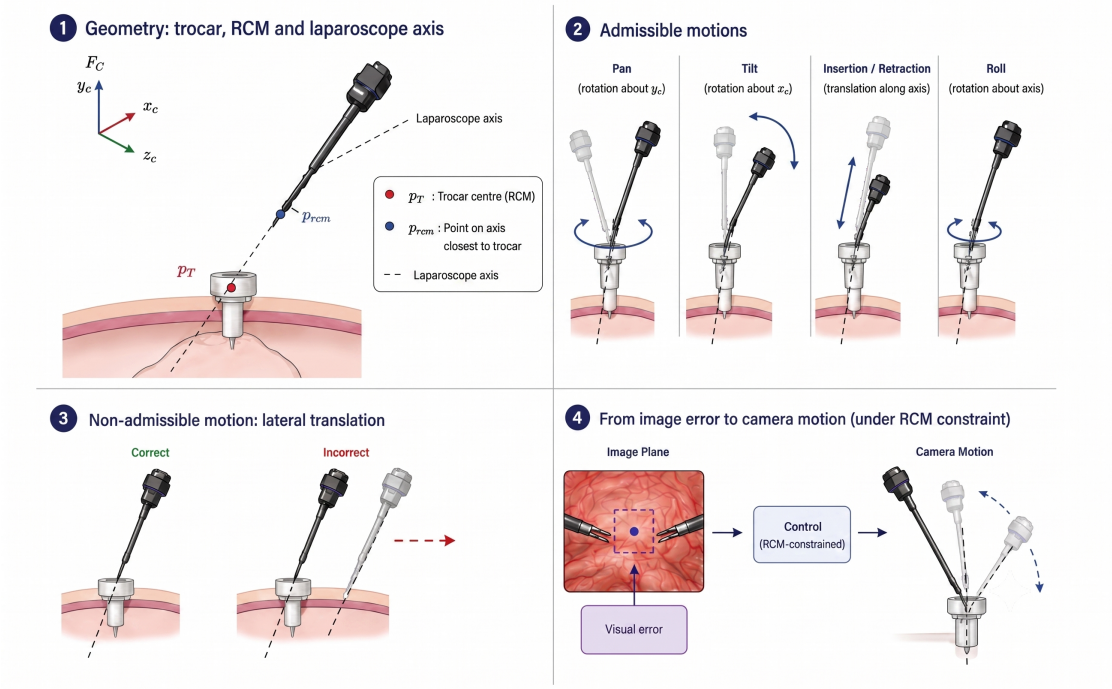


Figure 2.4: *Kinematic interpretation of RCM-constrained laparoscope motion.* The trocar defines a Remote Centre of Motion about which the laparoscope can rotate. Admissible motions include pan, tilt, insertion or retraction, and axial roll. A lateral displacement generates an RCM error, defined as the minimum distance between the trocar centre and the laparoscope axis. The image error detected in the endoscopic view must therefore be converted into a camera motion that respects this geometric constraint.

A similar principle is also adopted in other frameworks [30], based on the kinematic formulation originally proposed by Sandoval et al. [31]. In these cases, the constraint can be expressed by introducing a virtual point along the instrument's longitudinal axis, which must be kept aligned with the trocar. The main difference from the previous formulation does not concern the physical meaning of the constraint, but the way in which it is incorporated into the robot's kinematic model. Instead of calculating a new centre of rotation as the point of minimum distance, the control system modifies the kinematic relationship between the end-effector and the operational space, so that the camera's movement respects the pivot point imposed by the trocar [30].

These formulations highlight an important point: maintaining the RCM depends on the quality of the instrument's geometric model and the correct definition of the reference systems. Minor deflections of the laparoscope, assembly errors, misalignments between the optical axis and the mechanical axis, or calibration uncertainties can alter the actual distance between the instrument's axis and the trocar. To reduce dependence on the ideal geometry of the tool, some variants estimate the position of the constraint by utilising the robot's internal kinematic information, such as joint encoders [20].

A second group of approaches, on the other hand, integrates the RCM constraint directly into the visual servoing law. In some architectures validated on commercial cobots, such as the KUKA LBR Med 7 R800, the control is not based on the explicit reconstruction of the tool's three-dimensional position, but on the comparison between the current image and a desired view using homography [32]. The homography matrix describes the perspective transformation between the two views and enables the estimation of the movement required to reposition the camera to the desired visual configuration. The RCM constraint is incorporated into the same formulation via a composite Jacobian, which combines the visual error and the geometric constraint imposed by the trocar.

A similar logic is found in approaches where the trocar constraint is incorporated directly into the visual interaction matrix [6]. In standard visual servoing, this matrix translates the error observed in the image into a camera movement command. If this transformation were carried out without taking the RCM into account, the controller might generate lateral translations that are incompatible with the pivot point imposed by the trocar. For this reason, the interaction matrix can be modified by incorporating the geometry of the access point a priori, so that the generated command corrects the camera framing whilst respecting the kinematic constraint.

These studies show that the RCM constraint can be incorporated directly into the control formulation. However, in methods based on the instantaneous calculation of velocities, the robotic command is generated as a sequence of local corrections. In the presence of measurement noise, temporal discretisation or uncertainties in the kinematic model, such corrections may affect the continuity of the motion and the stability with which the constraint is maintained throughout the trajectory.

Some approaches tackle this problem using inverse kinematics strategies. The Dual Step IK method, proposed by Berthet-Rayne et al. [33], virtually divides the manipulator into two kinematic chains. The first relates to the section upstream of the RCM, whilst the second is centred on the trocar. The fulcrum constraint is treated more explicitly than the rest of the joint movements. However, this approach has been developed primarily on flexible and teleoperated platforms,

where mechatronic complexity introduces further considerations.

A different approach is represented by learning-based methods. In a study conducted on a KUKA LWR 4 collaborative manipulator, the camera control is managed using a Random Forest model trained on manoeuvres performed manually by an operator [34]. In this case, the trocar constraint is not imposed via an explicit geometric formulation, but through the learning of a workspace deemed permissible. The system's adaptability increases, but this requires training data, external tracking and specific validation of the model's ability to generalise to different procedures and configurations.

The solutions analysed demonstrate that the RCM constraint can be addressed through geometric, visual, inverse kinematic or learning-based formulations. The key challenge, therefore, is the realisation of a complete, deterministic perception-to-motion chain that can be implemented on a commercial surgical cobot. Furthermore, visual localisation, command continuity and RCM deviation control must all be maintained simultaneously. In redundant manipulators, the RCM constraint must often coexist with other control objectives. For this reason, the following section analyses the role of kinematic redundancy and hierarchical strategies.

2.2.2 Redundancy and hierarchical control strategies

In robotic manipulators equipped with more degrees of freedom than are strictly necessary to perform the main task, this is referred to as kinematic redundancy. In robotic systems for minimally invasive surgery, this redundancy can be utilised to maintain the RCM constraint and coordinate other control objectives. The residual degrees of freedom can be used for secondary tasks, such as visual tracking, optimising the robot's posture or avoiding undesirable configurations. The fundamental condition, however, is that these tasks must not interfere with maintaining the centre of rotation on the trocar. This principle is often formalised via the null space of the Jacobian associated with the primary task [29].

In this context, the null space represents the set of joint velocities which, according to the kinematic model adopted, do not alter the primary task. The hierarchical logic therefore consists of assigning the highest priority to maintaining the RCM constraint, whilst the secondary task is projected into the remaining available space. In concise form, the total joint velocity can be expressed as the sum of the contribution required for the RCM task and the contribution of the secondary task, filtered by its projection into null space [29]:

$$\dot{q} = \dot{q}_{\text{RCM}} + N_{\text{RCM}}\dot{q}_{\text{secondary}} \tag{2.2}$$

In this context, $\dot{q}_{\text{RCM}}$ represents the joint command associated with maintaining

the Remote Centre of Motion constraint, whilst $\dot{q}_{\text{secondary}}$ represents the command relating to the secondary task. The matrix N_{RCM} projects this contribution into the null space of the primary task to ideally prevent the secondary task from altering the RCM constraint. However, this is not an absolute guarantee, as its validity depends on the correct calibration of the system, the definition of the reference frames, the joint limits and the distance from singular configurations.

Some studies implement this logic within more general hierarchical control structures, such as Hierarchical Quadratic Programming. In such approaches, the various tasks are organised according to priority levels and solved using constrained optimisation problems. In the work by Colan et al., for example, the average time reported for solving the optimisation problem is 372 μs in the setup considered by the authors [29]. This figure demonstrates the computational compatibility of that specific implementation with real-time control; however, it cannot be automatically generalised to other architectures, hardware or experimental conditions.

A similar approach, but one formulated more directly from a spatial perspective, consists of limiting the manipulator's control a priori to those degrees of freedom that are compatible with the laparoscopic pivot point. In some frameworks, the movement is constrained to four degrees of freedom defined relative to the trocar reference frame: pan, tilt, insertion or retraction, and roll [24]. In this way, lateral translations incompatible with the RCM constraint are excluded at the stage of defining the control space, rather than being corrected subsequently.

Redundancy can also be used to improve the mechanical configuration of the manipulator. In the hierarchical framework proposed by Su et al., validated on a KUKA LWR4+ architecture, the priority level is dedicated to maintaining the RCM constraint. The remaining degrees of freedom are instead used to optimise the robot's manipulability [35]. The manipulability index describes the manipulator's ability to move effectively within its operational space and is often used to avoid kinematically unfavourable configurations, such as singularities. In this case, posture optimisation does not replace the primary task, but is treated as a secondary objective compatible with maintaining the fulcrum.

Null-space management can also be useful in physical interactions between robots and operators. In the work by Berthet-Rayne et al., for example, the repositioning of intermediate joints allows the system to adapt the manipulator's configuration in the presence of external interactions, whilst maintaining the main task associated with the RCM constraint [33]. Here too, the ability to modify the robot's posture depends on the correct separation between the primary task and secondary tasks.

Not all architectures, however, strictly separate tasks into hierarchical levels. In some approaches, the various kinematic constraints are integrated within a single

formulation. In the work by Nasiri et al., for example, the system simultaneously considers both the insertion or retraction movement of the instrument and the maintenance of the rotation point via an augmented Jacobian [36]. The centre of rotation is not treated as a rigidly fixed point, but is parameterised along the instrument's axis. The control system is allowed to adapt the position of the virtual pivot point according to the manipulator's configuration.

Overall, strategies based on null space, hierarchical control and augmented Jacobians demonstrate how kinematic redundancy can be utilised for a dual purpose. Firstly, to maintain the RCM constraint and, secondly, to coordinate secondary objectives such as visual tracking, posture optimisation and workspace management. In hierarchical approaches, maintaining the fulcrum is treated as the priority task, whilst the other objectives are projected into the residual space. In more integrated approaches, however, the constraints are combined into a single kinematic formulation.

The main challenge lies in translating these formulations into a continuous robotic command. Projection into null space, priority management and the calculation of Jacobians require considerable care. There must be a precise definition of the reference frames, joint limits and possible critical configurations of the manipulator. Furthermore, when the visual reference changes frame by frame, even small fluctuations in the target can result in frequent movement corrections. This aspect introduces the topic of more compliant and adaptive control strategies, discussed in the following section.

2.2.3 Admittance control and dynamic RCM

A rigid RCM constraint is useful from a geometric point of view, but represents a simplification compared with a real surgical environment. Factors such as breathing, tissue deformation and small involuntary movements by the patient can alter the actual position of the access point, meaning that keeping a trocar perfectly static is not always realistic. In such cases, overly rigid control can generate unwanted forces between the instrument, the trocar and the tissues.

To address this problem, some studies propose admittance control strategies, in which the robot modifies its behaviour in response to the measured forces. In the architecture proposed by Nasiri et al., for example, the physical interactions detected at the trocar are used to correct the instrument's orientation, making the RCM constraint more flexible and adaptive [36]. In this way, the system does not merely impose a rigid geometric relationship, but seeks to compensate for the lateral forces generated by interaction with the tissues.

This work also combines a reactive response to forces with a predictive feedforward component. The aim is to prevent compensation from occurring only after a

contact force has arisen, thereby reducing the response delay during movement. The resulting control therefore combines a component calculated from the kinematic geometry and a component generated from the forces measured at the trocar [36]. This type of approach allows for a more dynamic representation of an RCM, but requires a reliable measurement of forces and an accurate modelling of the physical interaction.

The concept of compliance can also be extended to the protection of larger anatomical or spatial regions. In the work by Kastritsi and Doulgeri, for example, the approach to constrained zones is managed through a passive admittance control with variable damping, so as to increase the robot's resistance when the tool approaches undesired regions [37]. Similarly, Sandoval et al. describe strategies in which compliance relates not only to the point of entry but also to the manipulator's posture, utilising the null space to make certain movements of the robotic structure more compliant [26].

These architectures are significant because they take into account physical phenomena that a static RCM does not model, such as tissue deformations, displacements of the access point and interactions distributed along the instrument. However, they introduce greater complexity in terms of experimentation and validation. The use of external force sensors may be limited by sterilisation requirements, costs and integration within the operating theatre; furthermore, the measurement of forces at the trocar may be affected by noise, robot dynamics and non-localised contacts. In the case of Nasiri et al., the authors themselves indicate that the validation was conducted primarily in simulation and that noise from the force sensors was not fully taken into account [36].

In light of these considerations, adaptive and admittance-based control methods represent an important avenue for bringing RCM control closer to real surgical conditions. However, the system developed in this thesis is deliberately situated within a more controlled and deterministic context. Since the main objective is to study visual localisation, command continuity and geometric deviation from the RCM constraint in a phantom setup, a static virtual RCM is adopted as an explicit experimental hypothesis. This choice is not intended to describe the full complexity of tissue-instrument interaction, but rather to isolate the problem of the camera's autonomous orientation under a known and repeatable geometric constraint.

2.3 Critical summary and methodological gaps

An analysis of the state of the art shows that the automation of endoscopic tracking cannot be reduced to a single vision problem or a single control problem. Various solutions described in the literature address individual aspects of the process chain,

such as instrument localisation, the selection of the visual target, the maintenance of the RCM constraint, or the generation of robotic motion. The main challenge, however, lies in integrating these into a complete pipeline capable of transforming a visual measurement into a continuous, stable robotic command that is compatible with the constraints imposed by the trocar.

From a perceptual perspective, deep learning-based approaches offer high performance in instrument detection and segmentation, but may introduce dependence on training data, greater implementation complexity and processing times linked to the architecture. Geometric and marker-based methods, on the other hand, provide measurements that are more directly interpretable by the controller, but remain sensitive to marker visibility, lighting conditions, occlusions and the experimental setup.

From a control perspective, the kinematic formulations of the Remote Centre of Motion allow for a rigorous description of the constraint imposed by the trocar, but do not automatically resolve the problem of the continuity of the generated command. The visual reference is updated frame by frame; there may be perceptual noise, and there are decision thresholds, temporal discretisation and communication latency which can result in frequent corrections, micro-movements or oscillations in the camera view. Strategies based on redundancy, null space and hierarchical control allow for the coordination of multiple objectives, but require a precise definition of the reference systems, joint limits and the manipulator's kinematic model.

Approaches based on admittance control and adaptive RCM address the physical interaction between the instrument, the trocar and the tissues more directly, taking into account phenomena such as respiration, deformation and small displacements of the access point. However, they introduce greater complexity in modelling, measurement and validation, particularly when reliable estimates of force or torque are required in a real surgical environment.

The methodological gap addressed in this thesis therefore lies in the need to evaluate a complete perception-to-motion chain on a commercial medical cobot, taking into account visual localisation, command continuity, communication latency and geometric deviation from the RCM constraint. The design choices adopted are a response to certain limitations identified in the literature. Oscillations of the visual target and frequent changes in movement are addressed through the introduction of a deadband and hysteresis logic. The need to obtain a visual measurement that is interpretable and directly usable by the controller, on the other hand, motivates the adoption of deterministic, marker-based tracking. The kinematic constraint imposed by the trocar is managed by explicitly defining a static virtual RCM, consistent with the controlled experimental setup. To limit overly frequent or

abrupt robotic corrections, motion generation is organised into control states with differentiated gains, distinguishing between slow and fast responses.

Based on these considerations, the following chapter describes the proposed experimental architecture, from the vision pipeline to communication with the robot, up to the control logic adopted to autonomously orient the laparoscopic camera whilst respecting the Remote Centre of Motion constraint.

Chapter 3

Materials and methods

This chapter describes the experimental architecture developed for the autonomous tracking of an endoscopic camera using a KUKA LBR Med manipulator. The methodology is organised according to the system's functional workflow: first, the robot-camera setup and the main reference systems used are presented; subsequently, the vision pipeline, responsible for marker detection and target pose estimation, is described. The UDP communication protocol is then introduced between the Python module and the robot controller, the RCM constraint-based control law, and the strategies adopted to make the motion more stable and smooth. Finally, we present the experimental setup used to validate the system and define the metrics employed for performance analysis.

3.1 Robot-camera system configuration

3.1.1 KUKA LBR Med robotic manipulator and TCP definition

The robotic platform used in this work is a KUKA LBR Med collaborative manipulator, specifically certified for integration into medical applications. The key feature that makes this platform particularly well-suited to the role of assistant in endoscopic procedures is its redundant kinematics with 7 degrees of freedom (DoF). As seen previously, this property makes it suitable for applications in which the movement of the end-effector (the end-effector) must be generated whilst adhering to additional geometric constraints [38].

To aid understanding of the following sections, it is useful to briefly review the main reference frames used. The robot's base frame is fixed to the base of the manipulator and represents the Cartesian reference frame against which the robot's absolute poses are expressed and its movements are planned. At the end

of the last joint, there is the flange frame, which constitutes the manipulator's terminal interface. A rigid mount has been fixed to the flange, at the end of which the camera used in the system is mounted. A tool reference frame (tool frame) is defined starting from the flange, with the Tool Centre Point (TCP) as its operating point. Figure 3.1 summarises the above. During Cartesian control, the tool pose is therefore described in terms of the position of the TCP and the orientation of the axes of the associated tool frame. In this specific case, as the camera is mounted at the end of the support, the TCP has been defined at the optical centre of the camera, and is therefore offset relative to the flange along the direction of the support.

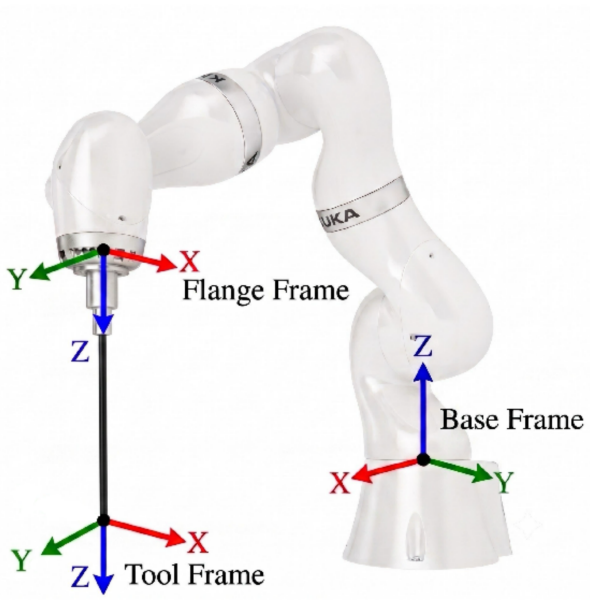


Figure 3.1: An overview of the main reference systems used.

3.1.2 Functional flow of the developed pipeline

Having defined the configuration of the robot-camera system and the reference frames used, it is possible to briefly describe the functional workflow of the architecture developed. The entire system can be viewed as a processing chain in which the information acquired by the camera is progressively transformed into a geometric reference, transmitted to the robotic controller and finally used to generate a movement compatible with the RCM constraint.

The pipeline is organised into three main blocks. The first block concerns the vision system, from which the three-dimensional pose of the target is estimated.

The second block concerns communication between the vision module and the robot controller, implemented via the UDP protocol. The third block consists of the control law, which uses the visual reference received to calculate a new Cartesian pose for the camera, whilst maintaining the geometric constraint imposed by the virtual trocar.

Figure 3.2 summarises this organisation. The diagram provides an overview of the process followed: from image acquisition to the generation of the robotic movement.

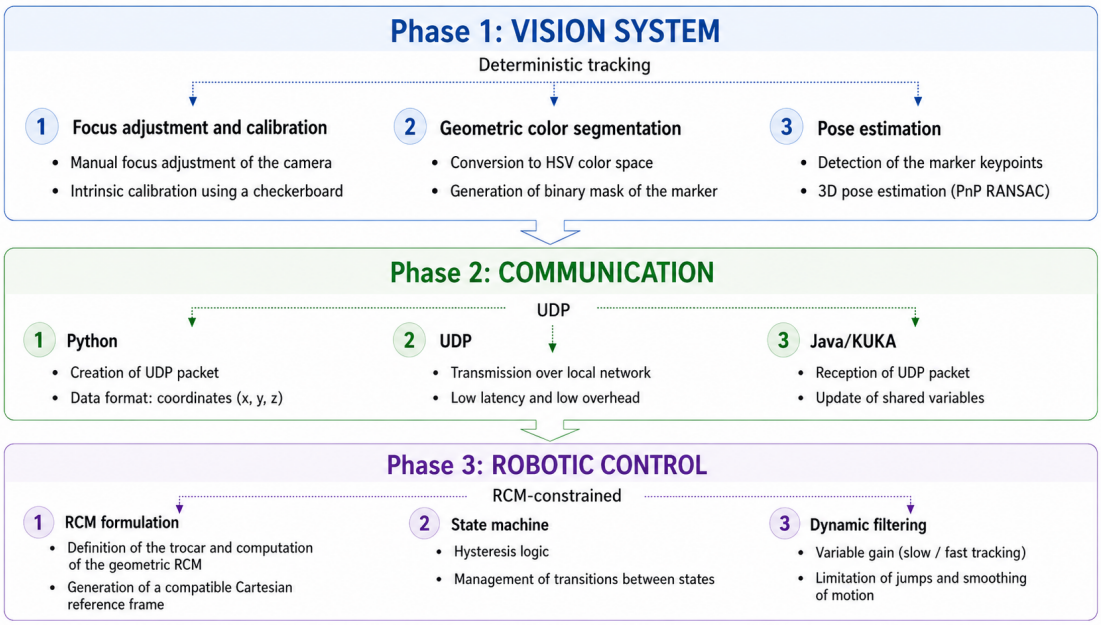


Figure 3.2: Functional diagram of the pipeline developed.

3.2 Deterministic vision and tracking system

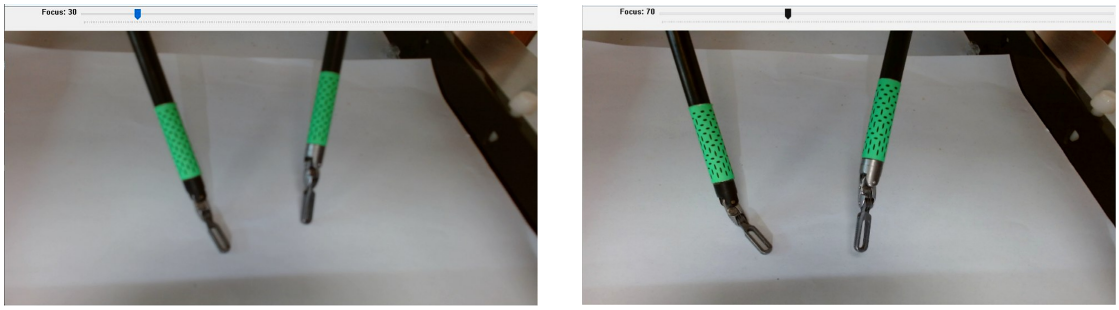
The vision system represents the first functional level of the developed architecture. Its task is to transform the video stream captured by the camera into geometric information that can be used by the robotic control module. In particular, the pipeline must detect the marker attached to the instrument, estimate its three-dimensional position relative to the camera, and make this information available for tracking.

In this work, a deterministic pipeline was adopted, based on colour segmentation and geometric pose estimation. This choice is consistent with the system's objective, which requires a solution that is interpretable, repeatable and compatible with

continuous updating of the visual reference. The implemented method is based on explicit and controllable steps: camera calibration, optical distortion correction, marker segmentation in HSV colour space, and estimation of the three-dimensional pose.

3.2.1 Focus adjustment, camera calibration and distortion model

As a first step, a preliminary adjustment of the camera's focus was carried out. This stage was necessary to establish stable and repeatable acquisition conditions, which were to be maintained throughout the entire process. To this end, the webcam's autofocus was disabled and the focus value was selected manually using a dedicated script. The script captures the video stream at a resolution of 1920x1080 pixels, displays the image in a control window and allows the focus parameter to be adjusted using a slider. The sharpest setting at the intended working distance is then identified visually. Figure 3.3 shows an example of the comparison made between two focus settings.



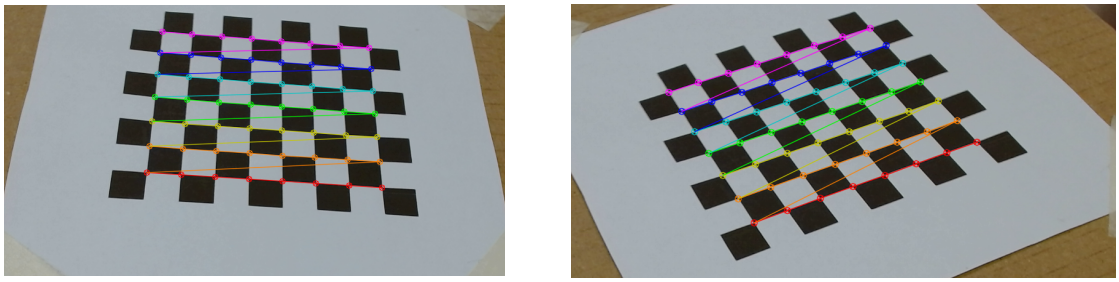
(a) **(b)**

Figure 3.3: Manual focus adjustment for the camera.

The decision to fix the focus does not merely affect the visual quality of the image; it also has a direct impact on the stability of the vision pipeline. The latter consists of colour segmentation and the extraction of geometric points from the marker. If the focus were to shift spontaneously during operation, it could affect the sharpness of the edges, locally alter the colour distribution and make the mask less stable. It must also be borne in mind that the pose estimation depends on the localisation of the marker's feature points. Blurrier images or those captured under variable optical conditions could introduce errors in the position of the detected points. For this reason, the selected focus value was kept constant during calibration, segmentation and the subsequent pose estimation.

Once the camera's optical configuration had been fixed, calibration was performed using a chessboard with known geometry. In this specific case, 8x7 internal intersections were considered, with each square side measuring 10 mm. During the procedure, 25 images of the chessboard were acquired, progressively varying the orientation and distance from the camera. This approach ensured a sufficient variety of pattern orientations. For each valid acquisition, the three-dimensional coordinates of the chessboard points were defined in the reference frame fixed to the pattern, assuming a Z-coordinate of zero, and the corresponding two-dimensional coordinates were determined in the image reference frame.

The acquired images were initially converted to greyscale, as the detection of the chessboard internal corners is based on the variation in intensity between the light and dark regions of the pattern. The internal intersections were identified using the cv2.findChessboardCorners() function and subsequently refined to obtain sub-pixel localisation of the detected points. Some examples of valid images used for calibration are shown in Figure 3.4, where different orientations of the chessboard relative to the camera can be seen and the coloured points represent the automatically detected internal intersections.



(a) **(b)**

Figure 3.4: Examples of images that are useful for camera calibration.

Based on the correspondences between the three-dimensional points on the chessboard and the two-dimensional points detected in the image, the camera parameters were calculated using the cv2.calibrateCamera() function. This function returns the camera's intrinsic matrix, the distortion coefficients and, for each image used, the rotation and translation vectors that describe the pose of the chessboard relative to the camera. The intrinsic matrix K is expressed as:

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

where f_x and f_y represent the focal lengths expressed in pixels, whilst c_x and c_y denote the coordinates of the optical centre.

The parameters obtained were finally saved and subsequently loaded into the main tracking script. During the execution of the pipeline, each captured frame is then corrected using an undistortion procedure, so as to compensate for optical distortion prior to marker segmentation and the estimation of the three-dimensional pose. In this way, the geometric estimation is performed on images that are consistent with the calibrated camera model.

3.2.2 Geometric colour segmentation in HSV colour space

The next stage of the pipeline involves identifying markers within the acquired image. Green cylindrical markers were used for this purpose. This choice was motivated by the geometry of the instruments used in minimally invasive procedures, which often have an elongated, cylindrical shape. The colour green was chosen because it allows for easier segmentation against the background of the experimental scene. Furthermore, the marker wrapped around the instrument's shaft provides a visual reference that is integral to the instrument itself, enabling its tracking without the need for external localisation systems.

In this work, the methodology proposed by Cartucho et al. [39] was adopted, based precisely on a marker with an internal binary pattern. Figure 3.5 shows an example of such a marker. The base code used for this phase is derived from the public CylMarker repository [1]. Compared to the original pipeline, the code has been adapted to the requirements of the experimental setup developed in this thesis, introducing a dedicated calibration of the HSV parameters, a customised segmentation of the marker and separate handling of multiple instruments present simultaneously in the field of view.

Before running the main tracking script, a preliminary calibration of the HSV parameters was carried out. This step was necessary because the colour values detected by the camera depend not only on the marker's nominal colour, but also on lighting conditions, sensor response, working distance and the presence of any reflections. Consequently, inappropriate HSV thresholds could lead to incomplete segmentation of the marker or result in the inclusion of irrelevant portions of the background, potentially causing segmentation failure. For this reason, a dedicated script was used in which a sample image captured in the experimental setup is converted into HSV colour space and displayed in an interactive window. By moving the pointer over the image, the script displays the pixel coordinates and

[1] `https://github.com/Cartucho/cylmarker`

Figure 3.5: A cylindrical marker used for visual tracking. It features a green surface, which is useful for segmentation in HSV colour space, and an internal binary pattern for identifying characteristic points.

the corresponding hue, saturation and value values. In this way, it was possible to identify threshold ranges consistent with the actual colour of the marker observed under the anticipated operating conditions.

The use of HSV colour space is useful because it separates colour information from the luminance component. In BGR colour space, used natively by OpenCV, the three components are directly influenced by both the colour and the brightness of the scene; consequently, a change in lighting can alter all channels simultaneously. In HSV colour space, however, the hue (H) describes the dominant colour, the saturation (S) indicates how vivid the colour is relative to grey, and the value (V) represents the brightness. This representation makes it easier to isolate a marker characterised by a specific colour, as it allows one to select primarily a range of hues and exclude pixels that are too desaturated or too dark.

Within the main script, each frame corrected for distortion is converted from BGR colour space to HSV colour space. A threshold defined by the parameters selected during calibration is then applied to this image: pixels with a hue within the chosen range and with saturation and brightness values above the set minimum thresholds are retained. The result is a binary mask, in which the white regions identify areas where the marker is likely to be present, whilst the black regions represent the rest of the image. In this context, the mask does not yet represent

the internal pattern required for pose estimation, but it delineates the coloured region of the marker and reduces the image area on which subsequent processing is to be carried out. Figure 3.6 shows an example of the result obtained after HSV thresholding. Starting from the acquired image, only the regions associated with the marker's colour are retained, whilst the rest of the scene is excluded from the binary mask.

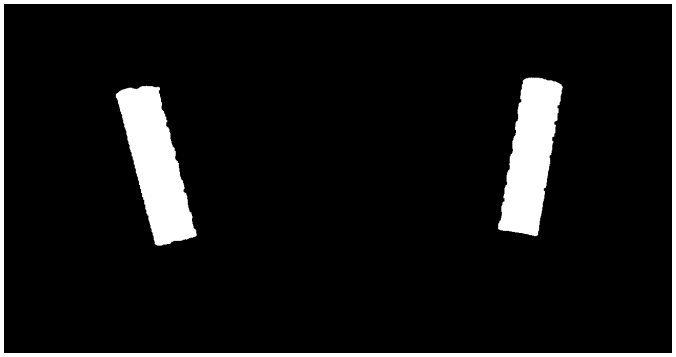


Figure 3.6: An example of a binary mask obtained in HSV colour space. The identified areas are candidate regions for the presence of the marker

Following HSV thresholding, the mask is refined using morphological operations. This step is important because a mask that is too fragmented would make marker identification less reliable, whilst a noisy mask could introduce candidate regions that do not belong to the instrument.

Starting from this candidate region, it is useful to distinguish between two levels of segmentation. The first level identifies the coloured background of the marker, referred to in the code as 'marker background'. In this context, the term 'background' does not refer to the background of the image, but to the coloured surface of the cylindrical marker. The second level, on the other hand, concerns the marker's foreground, i.e. the internal pattern used by CylMarker to identify characteristic points. Figure 3.7 shows an example of foreground segmentation based on the previously identified marker region. After selecting the coloured region using HSV, the code utilises the foreground segmentation function provided by the CylMarker library to extract the pattern solely from within the candidate area. In this way, the HSV segmentation acts as a preliminary spatial filter: it restricts the search for the pattern to areas where the marker is actually likely to be present.

One significant adaptation concerns the simultaneous tracking of multiple instruments. In the method proposed by Cartucho et al. [39], multi-instrument tracking is achieved by using markers with unique binary patterns, so as to distinguish the identity of the different instruments present in the field of view. In the

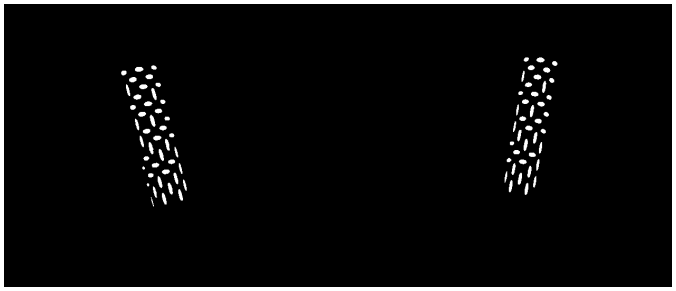


Figure 3.7: An example of a foreground mask derived from the candidate region.

system developed in this thesis, however, the aim was not to recognise each marker independently, but to extract the regions associated with the visible instruments and process them separately in the subsequent stage of the pipeline. For this reason, a simpler solution was adopted, one more consistent with the experimental setup: two instruments can be marked with the same type of marker and separated directly at the image segmentation stage.

Once the overall HSV mask has been obtained, the contours of the segmented regions are then extracted. Each contour delimits a continuous region of the mask, potentially associated with an instrument. The regions are filtered based on their area, discarding components that are too small—likely due to noise—and those that are too large, which are incompatible with the expected size of the marker. A maximum of two candidate regions are retained, consistent with the maximum number of instruments considered in the experimental scenario.

Consequently, there is no need to generate different markers or modify the internal binary pattern. From an implementation perspective, each candidate region is analysed independently thanks to the preliminary separation of the contours.

The pipeline is, in fact, easily adaptable to a larger number of instruments. It is sufficient to adjust the maximum number of candidate regions to be retained according to the experimental scenario, without having to generate different binary patterns.

A separate mask is then constructed for each valid region. This choice is crucial because it prevents patterns belonging to different instruments from being processed as if they belonged to a single marker. In practice, the overall mask obtained from HSV thresholding is divided into local masks, each associated with a single segmented region. For each region, only the foreground corresponding to the marker contained within that specific area is retained.

HSV segmentation therefore plays a preliminary yet essential role within the pipeline. It identifies the regions of the image on which the extraction of the

marker's geometric points is to be performed at a later stage. The quality of this segmentation is of fundamental importance as it directly influences the robustness of the subsequent stage. Incomplete, fragmented or noisy segmentation can compromise the identification of characteristic points. The estimation of the three-dimensional pose based on the regions thus identified is described in the following section.

3.2.3 Three-dimensional pose estimation

Following segmentation, the pipeline has one or two foreground masks, one for each detected marker. The subsequent pose estimation stage uses this local foreground mask, together with the distortion-corrected frame, to identify the marker's feature points and determine its three-dimensional pose relative to the camera. In this context, 'pose' refers to the position and orientation of the reference frame fixed to the marker relative to the camera's reference frame. The intermediate steps required to calculate this pose are described below.

Once the distortion-corrected frame and the foreground mask have been obtained, the function `keypoints.find_keypoints` from the CylMarker library [39] is called; this identifies the features within the marker and reconstructs the binary sequence of the pattern.

The result of this operation is an object containing the geometric information required for pose estimation. If the pattern is not reliably recognised, the function returns `None` and the region is discarded. Otherwise, the method `pttrn.get_data_for_pnp_solver`, also part of the library, returns two sets of points. This step follows the logic described by Cartucho et al. [39], according to which the first set contains the three-dimensional coordinates of the features defined on the basis of the known geometry of the cylindrical pattern. The second set, on the other hand, contains the corresponding two-dimensional coordinates detected in the image, expressed in pixels. The task of the pose estimation algorithm is precisely to estimate the transformation that links the marker's reference frame to the camera's reference frame.

The pose estimation is therefore formulated as a Perspective-n-Point (PnP) problem. In general terms, the PnP problem consists of determining the rotation and translation of a three-dimensional object based on a set of correspondences between known 3D points on the object and their respective 2D projections in the image.

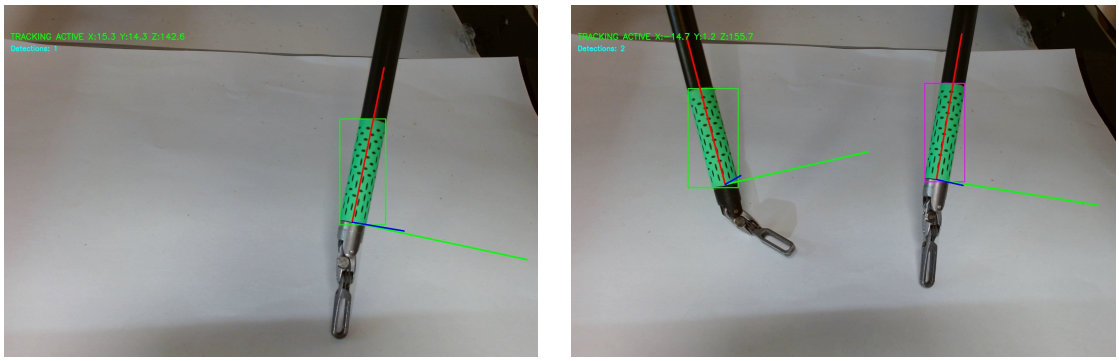
In the code, the PnP problem is solved using the OpenCV function `cv2.solvePnPRansac`. Before calling the solver, a check is carried out to ensure that there are sufficient matches available, as too few matches would not allow for a

robust pose estimate. The function takes as input the 3D points of the marker's features, the 2D points detected in the image and the camera's intrinsic matrix.

The RANSAC variant was introduced to increase the robustness of the estimate. In the presence of reflections, partial occlusions or local errors in feature identification, some 3D-2D correspondences may be inconsistent with the marker's actual pose. RANSAC allows the pose to be estimated by selecting the most consistent solution, thereby reducing the influence of any outliers.

When the estimation is successful, the function returns two main quantities: `rvec` and `tvec`. The vector `rvec` represents the rotation of the marker relative to the camera. The vector `tvec`, on the other hand, represents the translation of the marker's reference frame relative to the camera's reference frame.

If the segmentation identifies only one valid region, the translation vector associated with that region is taken as the visual reference for the current frame, as shown in Figure 3.8 (a). When, however, two valid regions are detected, the pose is estimated separately for each mask and the final reference is obtained by averaging the translational components. An example of this scenario is shown in Figure 3.8 (b), where the two instruments are segmented as distinct regions and processed separately by the pipeline. This approach is consistent with the system's objective, which is not to semantically identify each instrument, but to orient the camera towards a region representative of the surgical scene being observed.



(a) **(b)**

Figure 3.8: Examples of tracking cylindrical markers attached to instruments. (a) Detection of a single marker. (b) Detection of two separate markers, processed independently by the pipeline.

If two markers overlap, are partially occluded or are so close together that they are segmented as a single connected component, the system does not force the separation of the two instruments. The pipeline treats the segmented region as a

single area and produces a single translation vector. An example is shown in Figure 3.9, where the instruments overlap, preventing them from being distinguished as two independent regions. This is an appropriate choice given the present experimental setup. Indeed, the main objective is not to reconstruct the pose of each instrument individually at every instant, but to provide the robot with a stable visual reference for tracking the scene.

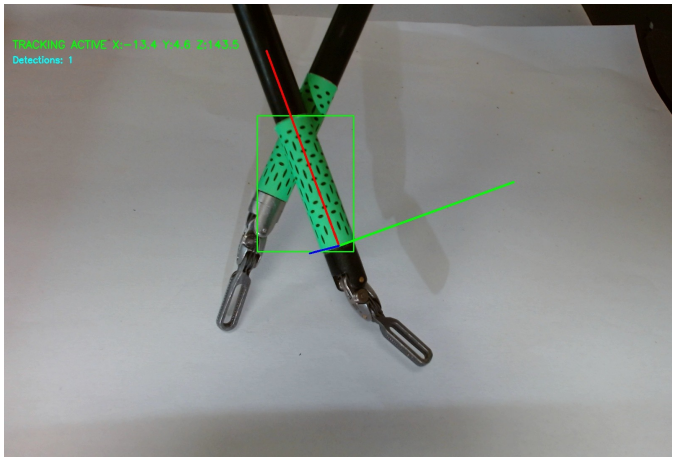


Figure 3.9: An example of two instruments overlapping. In this situation, the detected area is treated as a single region.

Finally, to reduce the effect of frame-to-frame fluctuations, the translational reference is subjected to a time filter. The new estimated vector is compared with the previously accepted one. If the variation exceeds a pre-set maximum threshold of 150 mm, the new value is considered unreliable and the previous reference is retained. This threshold helps to limit the effect of isolated anomalous estimates, which could be caused by incorrect segmentation or incomplete pattern recognition and lead to sudden variations.

When the variation falls within the accepted limits, the vector is filtered using an exponential moving average, equivalent to a first-order low-pass filter. This filter combines the estimated value at the current frame with the filtered value from the previous frame, further attenuating sudden variations caused by segmentation noise or isolated errors in the pose estimation. The implemented formula is as follows:

$$\mathbf{t}_k^{\text{filt}} = \alpha \mathbf{t}_k^{\text{raw}} + (1 - \alpha)\mathbf{t}_{k-1}^{\text{filt}} \tag{3.1}$$

where $\mathbf{t}_k^{raw}$ is the estimated translation vector at the current frame, $\mathbf{t}_{k-1}^{filt}$ is the filtered reference from the previous frame, and $\mathbf{t}_k^{filt}$ is the new filtered reference. The parameter α is the update coefficient and is set to 0.35 in the code; this value

has been chosen to dampen rapid oscillations without making the system's response excessively slow. In fact, the new reference is obtained by combining 65% of the previous value with 35% of the current estimate.

The final result is therefore the marker's pose relative to the camera. However, the relevant information is the position of the target. For this reason, the estimated rotational component is not used during robotic control, whilst the filtered translation vector is selected as the actual output of the vision pipeline. This vector constitutes the information to be transmitted to the robotic module and therefore represents the input data for the subsequent communication phase.

3.3 Communication protocol

The communication phase acts as the link between the vision pipeline and the robotic control module. The target's position must be made available to the robotic controller via continuous updates, so as to enable the robot to adjust its configuration according to the observed position of the instrument.

For this purpose, communication based on the UDP protocol has been implemented. The choice of UDP is motivated by the need to prioritise rapid updating of the reference over the guaranteed delivery of every single packet. In the developed system, each packet contains a new estimate of the target's position; consequently, the occasional loss of a packet is less critical than the introduction of delays. The most relevant information for control is therefore not the complete preservation of the packet sequence, but the availability of the most recent reference.

3.3.1 UDP architecture between the vision module and the robot controller

The first level of communication consists of the Python script dedicated to vision processing. At each processing cycle, the script extracts the filtered translation vector. It is then associated with the cycle time (in ms) of the vision pipeline, calculated as the time required to acquire the frame, correct it, segment the marker, estimate its pose and generate the reference. Together, these two pieces of information form the packet that is sent via UDP to a local address, using the computer as a processing node.

The second level is represented by the intermediate Python script, which acts as a gateway between the vision module and the robot controller. This script receives the packets from the tracking pipeline, extracts the three translational components and the vision cycle time, and then reformats the message to be sent to the robot. Since only the translational component of the visual reference is used in the control

law developed, the rotational components of the packet sent to the robot are set to zero. The forwarded message therefore takes the following form:

```
x, y, z, 0.00, 0.00, 0.00
```

The vision cycle time, on the other hand, is not used directly by the robot controller, but is recorded by the gateway for subsequent performance analysis.

Finally, the third level of communication is implemented in the Java task running on the KUKA controller, specifically SunriseWorkbench. This is a background task that listens on a dedicated UDP port and receives packets from the Python gateway. Once a valid message has been received, the task extracts the first three numerical components and saves them to a variable accessible by the main control program. The UDP task is responsible for updating the available reference, whilst the robotic application uses the latest value received to calculate the new Cartesian destination.

3.3.2 Latency management and data synchronisation

A key aspect of the proposed architecture concerns the timing of data exchange. The vision pipeline and the robotic controller operate at different frequencies. The camera captures images at a nominal rate of 30 fps, whilst the actual frequency at which a new reference is produced depends on the time required for processing operations.

During acquisition via OpenCV, the camera buffer was limited to a single frame, so as to reduce the build-up of unprocessed images and minimise the risk of processing frames that are out of date relative to the current state of the scene.

The Java task dedicated to UDP reception was run cyclically with a period of 10 ms. The aim was to make the robotic side faster than the vision pipeline. In this way, the system checks very frequently for any new packets arriving from the Python module. Consequently, when a new pose estimate is generated and transmitted, the Java task can update it with a minimal additional delay of 10 ms.

As the vision pipeline requires longer processing times and does not necessarily generate new data at every cycle of the robotic task, the UDP socket was configured with a timeout. If no packet is received within the specified time, the task terminates the current iteration and starts another one in the next cycle, preventing network reception from blocking the system's execution.

Similarly, the Python gateway has also been configured so that it does not hang indefinitely whilst waiting for a response from the controller. The script listens for a limited period after sending the packet. If it does not receive an acknowledgement

within the specified time, the lack of response is logged and processing of subsequent packets continues. In this way, any isolated communication losses or delays do not interrupt the overall flow of the system.

The response sent by the Java task to the Python gateway thus allows the receipt of the packet to be confirmed and the Round-Trip Time of the UDP communication to be calculated. The Python gateway records the time immediately prior to sending the packet to the robot and calculates the time elapsed until the confirmation response is received. This is not a one-way latency, but an overall measurement of the round-trip time of the packet between the computer and the robot controller.

The strategy adopted therefore prioritises continuous updating in order to guarantee the most recent reference. Any lost or unacknowledged packets do not cause the pipeline to halt, so that subsequent cycles can provide a new, updated reference. This approach is consistent with the nature of the system developed, in which robotic control does not depend on the receipt of every single packet, but on the progressive updating of the instrument's position.

3.4 Control law and RCM constraint

The control phase is the final stage of the developed pipeline: the three-dimensional reference estimated by the vision module and transmitted via UDP is converted into a Cartesian movement of the camera mounted on the KUKA LBR Med manipulator. The aim of the control is not merely to track the visual target, but to do so whilst preserving the kinematic constraint imposed by the trocar, which in laparoscopic systems acts as the fixed point through which the endoscope enters the patient's body.

As discussed in the state-of-the-art review, the Remote Centre of Motion can be formulated using different approaches, including geometric constraints, dedicated Jacobians, hierarchical strategies in null space and admittance controls. In this work, a geometric and deterministic solution was adopted, consistent with the experimental setup developed. In particular, the trocar is modelled as a static virtual point in the robot's space, whilst the camera is controlled so as to orient its line of sight towards the estimated target, whilst keeping this axis compatible with the imposed pivot point.

To understand how the visual reference is transformed into a robotic command, it is necessary to clarify certain kinematic aspects of the manipulator. In the system developed, the robot is not controlled directly by specifying the position of individual joints, but through Cartesian poses expressed in the robot's reference frame. This approach requires knowledge, at every instant, of the camera's current

pose in the robotic workspace, so as to be able to construct a new destination consistent with the visual target and the RCM constraint.

The ability to determine the camera's pose stems from the forward kinematics of the manipulator. Forward kinematics allows the Cartesian pose of the TCP to be derived from the configuration of the robot's joints; in other words, it makes it possible to determine where the camera is located and how it is oriented within the robotic workspace.

Inverse kinematics, on the other hand, performs the opposite task: starting from a desired Cartesian pose, it determines the joint configuration required to reach it. In this work, this aspect is not explicitly implemented in the code developed. The program calculates the camera's Cartesian destination that is compatible with the RCM constraint, and the transformation of this destination into joint movements is handled internally by the KUKA controller during execution via SmartServo.

3.4.1 Geometric formulation of the RCM constraint

In the implemented setup, the camera has been defined as a tool associated with the robot's flange. The control can therefore be formulated with respect to the TCP, rather than with respect to the flange alone.

Starting from this initial configuration of the camera-robot system, following the initial positioning of the manipulator, the current Cartesian pose of the flange is acquired. A virtual point (the trocar) is then defined, located 300 mm from the flange along the flange's local z-axis. This point is subsequently transformed into the robot's absolute reference frame and kept fixed during the tracking process. Figure 3.10 shows the experimental setup, indicating the main reference frames for the position of the virtual trocar and its geometric relationship with the target. This is a static RCM because the pivot point is not updated during movement but remains as defined in the initial configuration of the manipulator.

The control law utilises the translation vector estimated by the vision pipeline, which describes the position of the target relative to the camera:

$$t = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

where t_x and t_y describe the lateral displacement of the target relative to the camera's z-axis, whilst t_z represents the estimated depth of the marker. The visual reference therefore contains both the information needed to orient the camera towards the target and an indication of its depth relative to the camera.

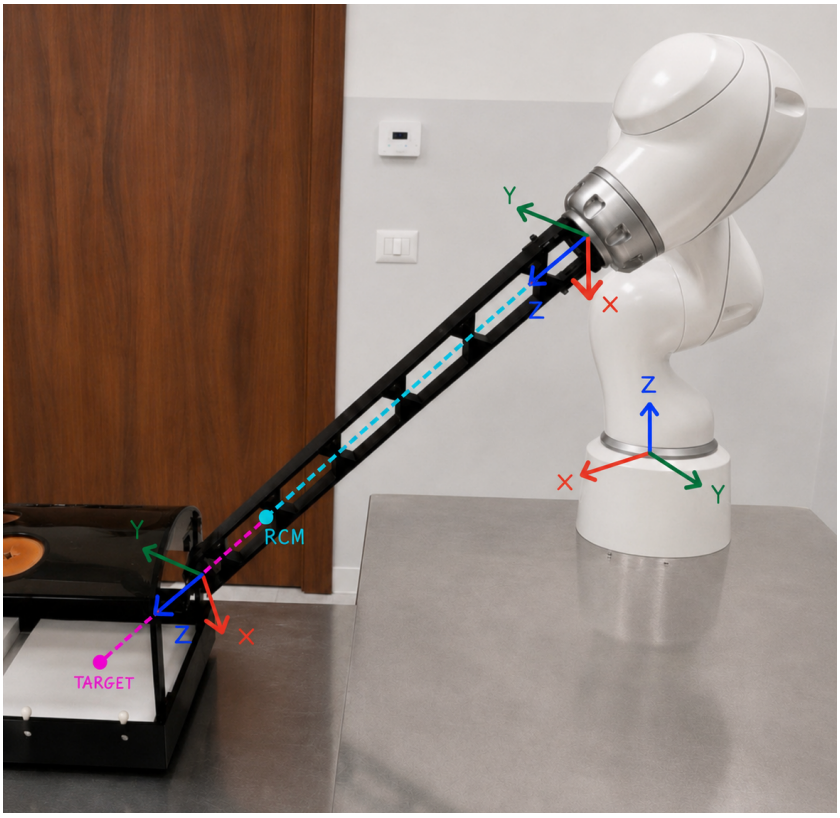


Figure 3.10: Illustration of the initial configuration used to define the RCM constraint.

Based on the visual reference and the current camera pose, the program identifies the point to be observed within the robot's reference frame. This point is not used to move the camera directly towards the target, but to define the direction in which to orient it. The aim is to construct a new pose in which the local z-axis of the TCP, taken as the camera's optical axis, is aligned with the line connecting the virtual trocar to the target to be observed. In this way, the camera's line of sight remains constrained to the pivot point defined by the trocar.

Let $\mathbf{p}_{target}$ denote the position of the target expressed in the robot's reference frame and $\mathbf{p}_T$ the position of the virtual trocar; the desired optical axis is defined as the unit vector connecting the trocar to the target:

$$\hat{\mathbf{z}}_d = \frac{\mathbf{p}_{target} - \mathbf{p}_T}{|\mathbf{p}_{target} - \mathbf{p}_T|} \tag{3.2}$$

This is the core of the implemented RCM constraint: the new camera axis is not chosen freely, but is defined as the direction constrained by the trocar towards

the target. In this way, the camera's movement is formulated as a rotation about a virtual pivot point, rather than as a simple translation towards the visual target.

After defining the camera's new z-axis, corresponding to the desired line of sight, it is necessary to complete the camera's orientation by also defining the other two axes of its reference frame. Indeed, aligning the z-axis towards the target does not uniquely determine the camera's entire orientation, as rotation about the optical axis remains possible. To prevent arbitrary rotations of the image, the code uses the camera's current y-axis as a reference and constructs a new orthogonal triad compatible with the desired z-axis. In this way, the camera is oriented towards the target whilst maintaining continuity with its previous orientation.

After defining the desired direction of the optical axis, the program must also determine the camera's position along that direction. In other words, once the line passing through the virtual trocar and oriented towards the target has been identified, the camera must be positioned along that direction at a distance consistent with the current configuration.

For this reason, an ideal working depth z_0 is introduced, set experimentally to 170 mm based on the geometry of the setup and the observation distance best suited to marker tracking. This value represents the desired distance of the target from the camera along the optical axis. By comparing the estimated depth t_z with z_0, the depth error is calculated:

$$e_z = t_z - z_0 \tag{3.3}$$

The desired position of the camera is then updated to compensate for this deviation. Specifically, once the desired direction $\hat{\mathbf{z}}_d$ has been defined, the program calculates the distance of the camera from the virtual trocar along that direction. Let d denote the distance between the virtual trocar and the new desired position of the camera's TCP, measured along the desired optical axis; this value is constrained within a predefined range:

$$50 \text{ mm} \leq d \leq 350 \text{ mm} \tag{3.4}$$

This limitation prevents depth correction from causing the camera to move forwards or backwards excessively, thereby ensuring patient safety.

Overall, therefore, the visual target defines the direction in which the camera should be oriented, whilst the virtual trocar dictates the pivot point around which this orientation must be achieved. This formulation allows the reference derived from the visual input to be transformed into a pose compatible with the RCM constraint. However, to achieve a smooth and controlled movement, the intensity

of the robot's response must also be adjusted. For this reason, the command is modulated via a state machine, described in the following section.

3.4.2 State machine and non-linear tracking logic

Between two consecutive iterations of the control loop, it may be the case that the UDP module has not yet received a new update from the vision pipeline. For this reason, the received coordinates t_x, t_y, t_z are compared with those from the previous cycle, and the tracking logic is executed only if the data has changed. This prevents the unnecessary recalculation of a new robotic destination when the available visual reference has not been updated.

However, the availability of a new reference does not imply that every variation must be directly converted into a robotic movement. The visual reference from the Python module may, in fact, be subject to noise, minor instabilities in segmentation or apparent variations in the estimation of the marker's position. If such variations were tracked without any modulation, the result could be an overly sensitive behaviour, with continuous micro-corrections and possible vibrations in the camera view.

To mitigate this phenomenon, the control law introduces a state machine with the aim of modulating the robot's response according to the magnitude of the error, associating a specific correction gain with each state. The gain represents the coefficient by which the visual reference is scaled before being transformed into a Cartesian correction by the robot. Low values dampen the movement, whilst higher values make the robot's response more pronounced.

To construct the state machine, a total error e is defined:

$$e = \sqrt{t_x^2 + t_y^2 + e_z^2} \tag{3.5}$$

where t_x and t_y represent the lateral components of the visual reference, whilst e_z is the depth error defined in the previous section. This quantity does not represent the RCM error, but rather the visual target tracking error. It is used as a decision criterion to select the robot's operating state and, consequently, the associated correction gain.

The state machine implemented includes three operating conditions:

- `DEADBAND`: the robot does not generate tracking corrections;
- `SLOW`: the robot performs a slow correction;
- `FAST`: the robot performs a faster correction.

The state is selected on the basis of the total error e defined in Equation 3.5. The deadband zone, in particular, is bounded by two distinct thresholds: an exit threshold of 2.5 mm and a re-entry threshold of 1.5 mm. In this way, the robot only starts to move when the error exceeds the upper threshold, but only comes to a halt when the error falls below the lower threshold.

The transition between slow and fast movement, on the other hand, is governed by a main threshold of 15 mm and a hysteresis margin of 2 mm. Specifically, the transition from `SLOW` to `FAST` occurs when the error exceeds 17 mm, whilst the return from `FAST` to `SLOW` occurs when the error falls below 13 mm.

Figure 3.11 shows a schematic representation of the three operating zones as a function of the total error e. The concentric regions represent the `DEADBAND`, `SLOW` and `FAST` modes, whilst the hatched bands indicate the hysteresis intervals.

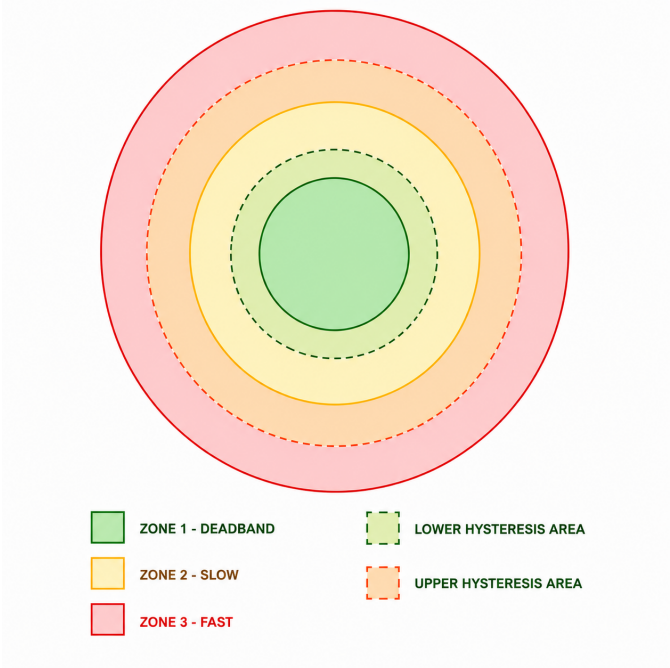


Figure 3.11: Operating modes of the state machine.

This distinction between the activation threshold and the deactivation threshold therefore introduces a hysteresis logic. This choice is important because, in a threshold-based state machine, small variations in the error around a limit value can cause repeated transitions between two neighbouring states. This phenomenon is known as *chattering* and consists of rapid, repeated switching of the control when the error oscillates around a threshold. The problem, therefore, is not merely the

small variations in the estimated target position, but the fact that such oscillation can cause sudden transitions between different states of the robotic control.

In the case of `DEADBAND`, hysteresis reduces this risk by preventing the robot from constantly switching between a stationary state and a tracking state when the error is close to the activation threshold. Similarly, the hysteresis between `SLOW` and `FAST` prevents repeated switching between slow and fast correction when the error is close to the main threshold of 15 mm.

Once the operating state has been selected on the basis of these thresholds, the program must determine the intensity of the correction to be applied. For this reason, a target gain is associated with each state:

$$g^* = \begin{cases} 0, & \text{if } \texttt{DEADBAND} \\ 0.05, & \text{if } \texttt{SLOW} \\ 0.16, & \text{if } \texttt{FAST} \end{cases} \tag{3.6}$$

The zero gain in the `DEADBAND` state prevents the robot from tracking very small changes in the target, as these could be due to noise or instability in the estimate. The reduced but non-zero gain in the `SLOW` state allows for a gradual correction when the error is significant but limited. In the `FAST` state, on the other hand, the target is even further from the desired condition and requires a more pronounced response.

The numerical values of the thresholds and gains were chosen empirically during the experimental tests. The stability of the visual reference and the robot's response were monitored. In particular, the `DEADBAND` was set to prevent small variations in the target—due to segmentation noise or the natural variability of tracking—from generating unnecessary camera movements.

This structure makes the control law non-linear with respect to the error. The command does not simply increase proportionally and continuously, but depends on the operational state of the system. This choice was made to improve the trade-off between stability and responsiveness. A response that was too reactive could have amplified tracking noise and increased the risk of micro-movements; conversely, a response that was too slow would have reduced the system's ability to track significant changes in the instrument's position.

The state machine therefore defines which response mode should be adopted and which target gain should be associated with each operating condition. However, the transition from one state to another could still introduce abrupt changes in the command if the gain were applied instantaneously. For this reason, the following section describes how the target gain is further filtered, so as to make the transition between the different tracking modes more gradual.

3.4.3 Dynamic filtering and vibration mitigation

The generation of robotic motion was implemented using *SmartServo*, a functionality of the KUKA SunriseWorkbench environment that allows controlled movements to be executed in real time, with a destination that can be updated during execution. This choice is central to the architecture developed, because the visual reference is not known in advance and therefore cannot be transformed into a predefined trajectory. Instead, the desired position of the camera must be progressively modified based on new data received from the vision pipeline.

SmartServo specifically allows a continuous robot movement to be initiated and, during execution, the Cartesian destination towards which the camera's TCP must converge to be updated. In this way, the robot does not execute a sequence of independent, pre-calculated trajectories, but maintains a continuous movement whose destination is progressively corrected based on the latest available visual reference.

To improve the fluidity of the movement, SmartServo has been configured with a minimum trajectory execution time of 80 ms. This parameter acts as a temporal smoothing mechanism: it prevents very rapid updates to the destination from resulting in overly abrupt corrections. The relative limits on joint speed and acceleration have also been configured to ensure sufficiently responsive movement, without making the robot's response excessively jerky.

In addition to the choice of SmartServo, the movement was executed under Cartesian impedance control. In this mode, the Cartesian pose calculated by the program remains the desired target, but the robot does not reach it by behaving as a perfectly rigid system. Instead, the controller introduces a more damped and compliant response. Stiffness and damping parameters were therefore configured in the code for both the translational and rotational components, together with parameters specific to the manipulator's null space.

This choice serves a dual purpose. On the one hand, it helps to make the execution of the movement smoother during subsequent updates of the Cartesian destination. On the other hand, it makes the robot's behaviour less rigid in the presence of any external disturbances. For example, there may be accidental contact with the manipulator; in this case, impedance control allows the robot to respond in a more damped manner, avoiding rigid resistance to the imposed displacement.

The robot's target position continues to be constructed in a manner compatible with the RCM constraint, but the manipulator's behaviour is less rigid and better suited to safe interaction within the experimental setup.

The compliance introduced by impedance control makes the execution of the commanded position more damped, but does not directly affect the way in which

that position is generated. This, in fact, depends on the gain selected by the state machine. However, if the gain associated with each state were applied instantaneously, the transition from one state to another could result in an abrupt change in the magnitude of the command.

For this reason, the gain actually applied does not immediately coincide with the target gain g^*, but is filtered through a discrete low-pass filter:

$$g_k = g_{k-1} + \alpha_g \left(g_k^* - g_{k-1}\right) \tag{3.7}$$

where g_k is the gain applied to the current cycle, g_{k-1} is the gain applied to the previous cycle and g_k^* is the target gain set by the state machine. The coefficient α_g is calculated as follows:

$$\alpha_g = 1 - e^{-\Delta t/\tau_g} \tag{3.8}$$

where τ_g is the filter's time constant and determines how slowly the gain transitions from the current value to the new target gain g^*. The term Δt, on the other hand, represents the time interval between two consecutive filter updates, that is, between two successive calculations of the applied gain. In the code, τ_g has been set to 0.15 s, whilst Δt has been approximated to 200 ms, assuming a value of the same order of magnitude as the expected cycle time of the vision pipeline.

The speed of the transition does not depend on the two parameters separately, but on the ratio $\Delta t/\tau_g$, which determines the value of the coefficient α_g. If the ratio is small, the change in gain will be slower and more gradual; conversely, for a higher ratio, the change will be faster. Furthermore, the filter must modulate the gain according to the actual rate at which the command is updated to avoid responses that are either excessively slow or instantaneous. For this reason, Δt has been chosen to be consistent with the frequency at which the system receives new visual references.

The decision to calculate α_g from Δt and τ_g allows the filter to be given an explicit temporal meaning, which would not have been possible had it been assigned a fixed numerical value. In this way, the speed at which the applied gain approaches the target gain does not depend on an arbitrary coefficient, but on the relationship between the update time under consideration and the desired time constant for the transition.

The filter takes on a more interpretable and easily adaptable meaning. If the update time of the visual reference were to change, it would be possible to modify Δt whilst maintaining the same filtering criterion. Conversely, setting a fixed value for α_g directly would have made the choice less transparent and more dependent on

purely empirical calibration. The filtered gain is then used to scale the components of the visual reference:

$$e_x = g_k t_x, \qquad e_y = g_k t_y, \qquad e_z = g_k(t_z - z_0) \tag{3.9}$$

This formulation therefore makes it possible to mitigate the effect of sudden fluctuations in the target and reduce the likelihood that small measurement errors will be translated into jerky robotic movements.

Overall, the quality of robotic motion does not depend on a single parameter, but on the combination of several levels of control. The *deadband* prevents unnecessary corrections, whilst hysteresis reduces repeated switching between operating states. Differentiated gains allow the robot's response to be adapted to the magnitude of the error, distinguishing between no movement, slow correction and faster correction. The gain filter ensures a gradual transition between states, avoiding abrupt changes in the amplitude of the correction. Finally, SmartServo allows the Cartesian target to be updated during execution, whilst impedance control introduces a more damped behaviour compared with purely rigid tracking.

This architecture addresses the gap identified in the state of the art. A reliable visual pose must be integrated into a stable, smooth robotic control system that is compatible with the RCM constraint. The developed system offers a compact and deterministic solution, suitable for the experimental setup under consideration.

3.5 Setup and experimental validation

The vision pipeline, the communication protocol and the robotic control law have been described. It is now necessary to define the experimental setup used to validate the integrated operation of the system. The aim of the validation is to verify, in a controlled environment, the ability of the developed platform to acquire the visual target, transmit the reference to the robot and generate a movement compatible with the RCM constraint.

3.5.1 Simulation of the surgical environment using an medical phantom

The experimental tests were carried out using the KUKA LBR Med manipulator fitted with a Logitech C920 camera.

In the system developed, the camera was mounted at the end of a rigid support fixed to the robot's flange. This support made it possible to simulate the spatial

arrangement of an endoscopic camera, whilst maintaining a known distance between the manipulator's flange and the image acquisition point. The support is 40 cm long and configured to replicate the length of a standard endoscope.

Image acquisition and processing were carried out on a Lenovo IdeaPad 5 15ITL05 laptop with an Intel i7 processor, which connects to the camera via a USB port. The computer acts as the processing node for the vision pipeline and the UDP gateway; consequently, its hardware specifications are relevant for interpreting the cycle times measured during the tests.

A medical phantom, shown in Figure 3.12, was used to simulate the laparoscopic setting. The device consists of a structure with multiple access ports and deformable inserts that allow surgical instruments to be inserted. The phantom is not intended to faithfully reproduce the anatomical or biomechanical complexity of tissues, but it does allow the instruments to be positioned in a geometry compatible with minimally invasive access.

Figure 3.12: Medical phantom used for simulation.

The instruments used during the trials are surgical instruments compatible with the da Vinci robotic system, employed in this setup as independent physical

instruments not connected to the original robotic platform, as shown in Figure 3.13.

Figure 3.13: Surgical instruments used during the experimental trials.

The overall configuration of the experimental setup is shown in Figure 3.14. Once inserted into the phantom, the instruments are moved manually within the camera's field of view, causing lateral and depth variations in the target's position. In this way, it is possible to verify whether the system is capable of responding to the target's movements by updating the camera's orientation whilst maintaining the geometric constraint imposed by the virtual trocar. The tests are not intended for the clinical evaluation of the system, but for the functional validation of the experimental chain: vision, communication and control.

During execution, the data required to quantitatively describe the behaviour of the integrated pipeline is saved. Specifically, the Python module records the vision cycle time, the UDP gateway measures the round-trip time of communication with the robot controller, whilst the Java program calculates and saves the RCM error during movement. These three parameters enable the main levels of the system to be evaluated separately: visual processing, reference transmission and maintenance of the kinematic constraint. The following subsection describes the significance of each metric and the role it plays in experimental validation.

Figure 3.14: Overall configuration of the experimental setup.

3.5.2 Performance evaluation metrics

The experimental evaluation of the system was based on metrics derived directly from the data recorded during execution. The choice of these metrics reflects the structure of the architecture developed.

The first metric considered is the vision pipeline cycle time T_{vision}. This represents the time taken by the Python module to acquire the image, correct its distortion, segment the marker, estimate its pose, filter the reference and prepare the output packet.

The vision cycle time is expressed in milliseconds and makes it possible to assess the effective rate at which the system is able to produce new references. Although the camera is configured with a nominal acquisition rate of 30 fps, the actual refresh rate of the visual reference depends on the time required for subsequent processing. It is therefore possible to derive an approximate effective rate from the average cycle time:

$$f_{vision} \simeq \frac{1000}{\overline{T}_{vision}} \tag{3.10}$$

where $\overline{T}_{vision}$ is the average cycle time. This derived metric makes it easier to

interpret the overall speed of the vision pipeline.

The second metric is the Round-Trip Time of the UDP communication, denoted as T_{RTT}. This is calculated by the Python gateway as the time elapsed between sending a packet to the KUKA controller and receiving the acknowledgement response from the Java task.

The Round-Trip Time is useful for assessing the contribution of communication to the overall pipeline time. In the system developed, communication must be sufficiently fast so as not to introduce a delay that dominates the visual processing time.

The third metric is the RCM error, denoted as e_{RCM}. This quantity measures the minimum distance between the virtual trocar and the camera's actual optical axis during the execution of the movement. Let $\mathbf{p}_T$ denote the position of the virtual trocar, $\mathbf{p}_C$ the current position of the camera's TCP, and $\hat{\mathbf{z}}_C$ the unit vector of the camera's actual optical axis; the RCM error is calculated as:

$$e_{RCM} = |(\mathbf{p}_T - \mathbf{p}_C) \times \hat{\mathbf{z}}_C| \tag{3.11}$$

Figure 3.15 shows the geometric interpretation of this quantity.

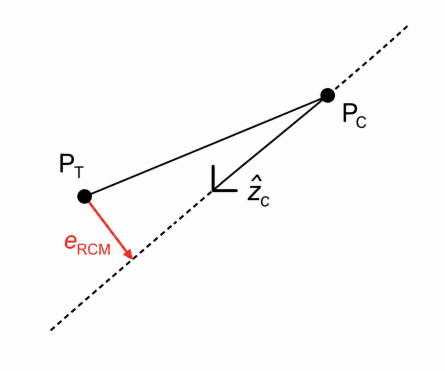


Figure 3.15: Geometric interpretation of the RCM error.

Since $\hat{\mathbf{z}}_C$ is a unit vector, the magnitude of the cross-product directly gives the distance between the trocar point and the line passing through the camera along its optical axis. This metric therefore allows us to assess the extent to which the camera's actual movement remains compatible with the geometrically imposed RCM constraint.

For each metric, the mean value and standard deviation were calculated. In the case of the vision cycle time and the UDP round-trip time, these quantities allow us to assess, respectively, the regularity of visual processing and of communication

with the robotic controller. In the case of the RCM error, however, the mean and standard deviation describe the extent to which the camera's movement has remained close to the geometric constraint imposed by the virtual trocar.

The joint analysis of these quantities makes it possible to verify, overall, whether the system is compatible with continuous and sufficiently stable operation in the experimental setup under consideration.

Chapter 4

Results

This chapter presents the experimental results obtained during the final validation test of the autonomous endoscopic camera tracking system. The analysis was carried out on data recorded during the integrated operation of the vision pipeline, the UDP communication layer and the robotic control program.

The results are organised according to the three performance metrics defined in the experimental protocol: the vision cycle time, the UDP Round-Trip time and the RCM error. The vision cycle time describes the time taken by the Python pipeline to generate a new visual reference. The UDP Round-Trip time describes the time elapsed between sending a packet to the KUKA controller and receiving the acknowledgement message. The RCM error, on the other hand, describes the minimum distance between the virtual trocar and the camera's actual optical axis during the tracking movement.

The three parameters analysed were recorded by different modules of the system. The vision cycle time and the UDP Round-Trip time were saved by the Python gateway for each processed packet, whilst the RCM error was recorded by the Java program during the actual robotic tracking phases. For this reason, the number of valid samples does not necessarily coincide across the different metrics. In each statistical analysis, the number of samples actually available for the parameter in question was therefore reported. Unless otherwise stated, the quantitative results are reported as mean ± standard deviation.

4.1 Vision pipeline cycle time

The vision cycle time was calculated for each frame processed by the Python tracking script. This metric includes image acquisition, distortion correction, marker segmentation in HSV colour space, foreground extraction, pose estimation, temporal filtering of the translational reference, and preparation of the UDP packet.

During the test, 347 samples were recorded over a time interval of approximately 95.9 s. The average vision cycle time was found to be:

$$T_{\text{vision}} = 189.70 \pm 24.98 \text{ ms}$$

The average refresh rate of the visual reference was calculated based on the average cycle time:

$$f_{\text{vision}} \simeq \frac{1000}{T_{\text{vision}}}$$

Substituting the measured average value gives:

$$f_{\text{vision}} \simeq \frac{1000}{189.70} = 5.27 \text{ Hz}$$

The temporal trend of the vision cycle time is shown in Figure 4.1, whilst the distribution of the measured cycle times is shown in Figure 4.2.

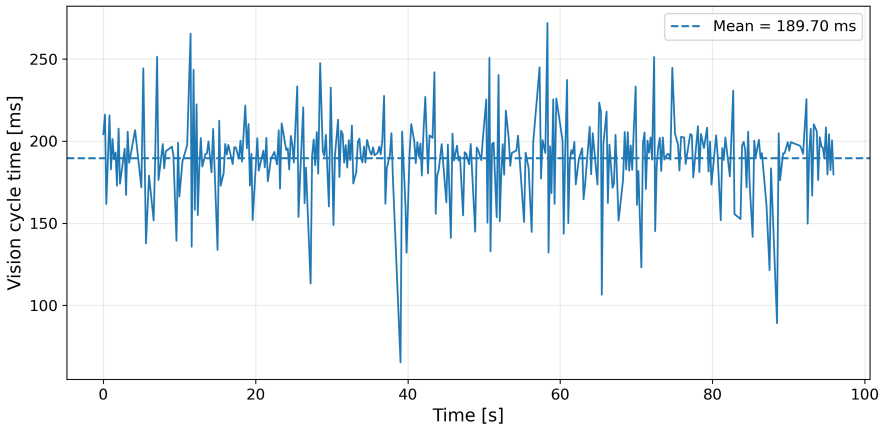


Figure 4.1: Temporal trend of the vision cycle time.

4.2 Round-Trip time of the UDP communication

The Round-Trip time of the UDP communication was measured by the Python gateway as the time between sending the packet to the robot controller and receiving the acknowledgement message sent by the Java task.

During the test 347 samples were recorded over the same time interval of 95.9 s. The average Round-Trip time was found to be:

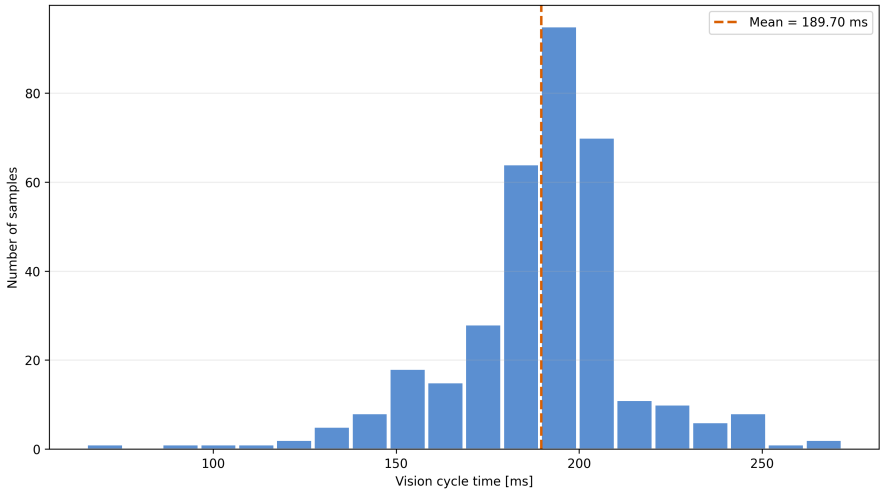


Figure 4.2: Distribution of vision cycle times.

$$T_{\mathrm{RTT}} = 2.48 \pm 1.42 \text{ ms}$$

The temporal trend is shown in Figure 4.3.

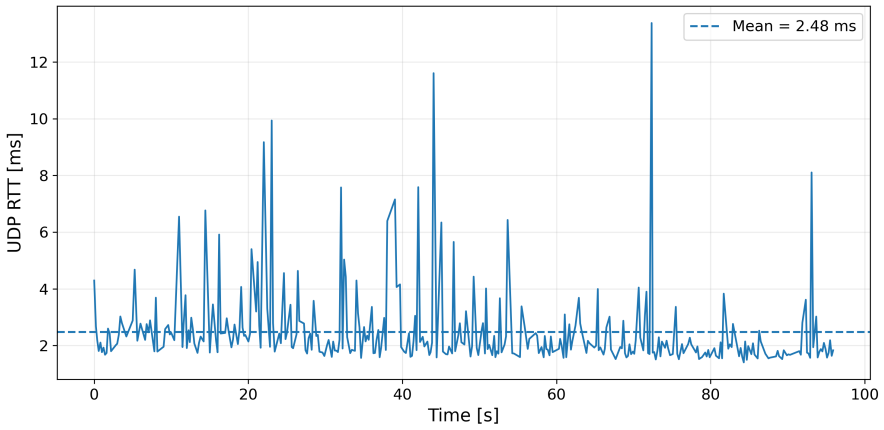


Figure 4.3: Time trend of the UDP Round-Trip time measured between the Python gateway and the KUKA controller.

During the test, no samples were recorded with a sentinel value of -1, which is used in the script to indicate that the confirmation message was not received within the configured timeout.

Taking into account the combined contribution of the average vision cycle time

and the average UDP Round-Trip time, the average time taken to generate and transmit the reference is:

$$T_{\text{vision}} + T_{\text{RTT}} = 189.70 + 2.48 = 192.18 \text{ ms}$$

The corresponding equivalent frequency was calculated as follows:

$$f \simeq \frac{1000}{192.18} = 5.20 \text{ Hz}$$

4.3 RCM error

The RCM error was calculated during the robotic tracking movement as the minimum distance between the virtual trocar and the camera's actual optical axis. Only valid numerical samples were included in the statistical analysis. A single outlier of 11.69 mm, which was not representative of the main signal distribution, was excluded from the statistical analysis. The statistics reported in this section therefore refer to the remaining 327 samples acquired during approximately 94.4 s of tracking.

The average RCM error was found to be:

$$e_{\text{RCM}} = 3.10 \pm 1.41 \text{ mm}$$

Figure 4.4 shows the time-dependent behaviour of the RCM error during the final experimental test. The distribution of the measured values is summarised in a boxplot in Figure 4.5.

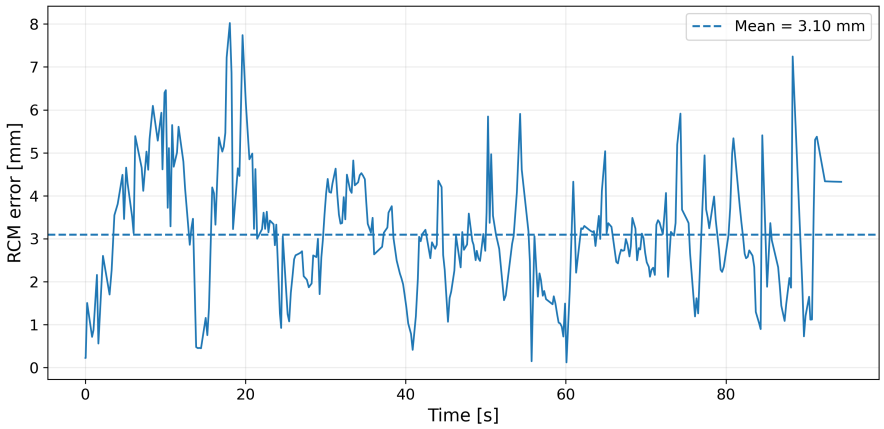


Figure 4.4: Time-dependent trend of the RCM error.

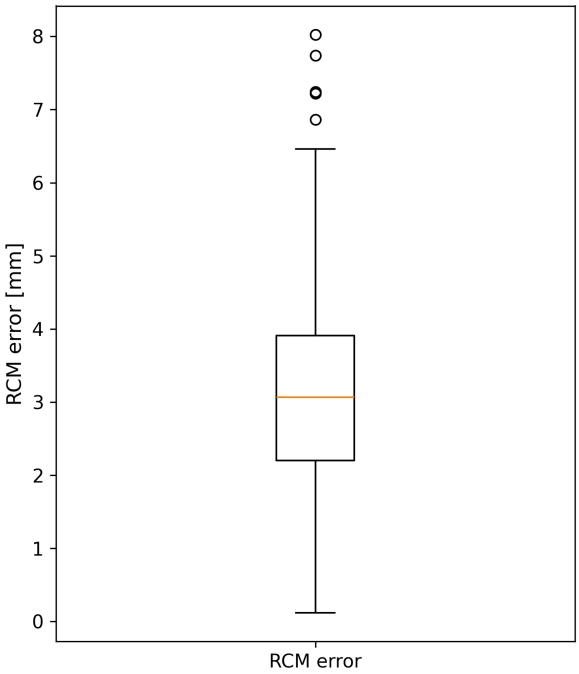


Figure 4.5: Box plot of the RCM error calculated from the 327 valid samples, after excluding the outlier sample of 11.69 mm.

4.4 Summary of quantitative results

Table 4.1 summarises the main quantitative results obtained during the final experimental test.

Table 4.1: Summary of experimental results.

Metric	**N**	**Mean ± SD**	**Min–Max**
Vision cycle time [ms]	347	189.70 ± 24.98	65.22–271.96
UDP RTT [ms]	347	2.48 ± 1.42	1.41–13.37
RCM error [mm]	327	3.10 ± 1.41	0.12–8.02

The maximum recorded vision cycle time was 271.96 ms. The maximum recorded UDP Round-Trip time was 13.37 ms. The maximum recorded RCM error, after excluding the outlier, was 8.02 mm.

Chapter 5

Discussion

This chapter interprets the results obtained during the final validation test of the system. The aim is not merely to comment separately on the measured parameters, but to evaluate them in relation to the overall architecture developed. In particular, the discussion focuses on determining whether the system was able to transform the visual reference estimated by the camera into a continuous, stable robotic movement that was compatible with the geometric constraint imposed by the virtual trocar.

Overall, the results show that the developed pipeline was able to integrate the different levels of the system in a functional manner. The marker attached to the instruments was detected by the camera, its three-dimensional position was estimated, and this was then transmitted to the KUKA controller. Finally, the robot generated a tracking motion whilst maintaining a low average RCM error. The validity of the proposed approach was therefore confirmed. However, the results also highlight certain limitations, which are mainly linked to vision processing time and the absence of clinical validation or validation carried out using a real endoscope.

5.1 Overall assessment of the integrated system

The first significant observation concerns the distribution of times within the pipeline. Taking into account both the vision cycle time and the round-trip time of the UDP communication, the system took on average approximately 192 ms to generate and transmit a new reference to the robotic controller, corresponding to an effective update rate of approximately 5.20 Hz. This value summarises the overall temporal behaviour of the vision-communication chain and makes it possible to identify the stage that most severely limits the system's responsiveness.

This value shows that the main bottleneck of the system is not communication with the robotic controller, but the vision pipeline. In fact, the contribution of

the UDP Round-Trip time is much lower than the time required to acquire and process the frame, segment the marker, estimate its pose and filter the reference. Communication can therefore be considered sufficiently fast relative to the overall system timing, whilst vision constitutes the level that limits the effective command update rate.

From the perspective of robotic control, the average RCM error was found to be 3.10 ± 1.41 mm, with a maximum value of 8.02 mm. This result indicates that the robot succeeded in keeping the camera's optical axis close to the virtual trocar during the tracking movement. The error is not zero, despite the geometric formulation of the constraint, because the measured value depends on the actual behaviour of the entire system.

The result obtained should therefore be interpreted as a functional validation of the control concept, rather than as a definitive measure of clinical accuracy. The system has demonstrated that it can maintain the RCM constraint within a few millimetres in a controlled experimental setup.

5.2 Vision pipeline performance

The vision pipeline was the first component of the architecture. The decision to use a deterministic strategy based on HSV segmentation and geometric pose estimation resulted in an interpretable and controllable system.

However, the average cycle time of approximately 190 ms shows that the pipeline, in its current configuration, does not reach the camera's nominal frame rate. The camera was configured to 30 fps, but the actual rate at which a new reference useful for control is produced is approximately five times lower. This discrepancy is due to the fact that the cycle time includes not only image acquisition but also all subsequent processing steps. In particular, the high frame resolution of 1920×1080 pixels increases the computational cost of undistortion, segmentation and pose estimation operations.

The histogram shows that cycle time values are concentrated around the mean, with limited variation relative to the overall order of magnitude. This demonstrates that the pipeline did not exhibit variability large enough to substantially alter the refresh rate of the visual reference.

Values lower or higher than the main region of the distribution can be attributed to video capture management, marker segmentation and pose estimation. These steps may take slightly different amounts of time depending on the quality of the frame and the visibility of the marker. The cycle time should not be interpreted as

a perfectly constant sampling period, but as a quantity subject to normal variability in the vision system.

There are, however, reasons why high resolution was chosen. This choice allows for a better representation of the marker, as well as more accurate identification of its edges and features. It should also be borne in mind that maintaining high visual quality is consistent with the need to observe the surgical scene clearly. The main trade-off is therefore between image quality and processing speed. In the system developed, greater visual and geometric quality was prioritised, even at the cost of a lower refresh rate.

A frequency of approximately 5 Hz is sufficient to demonstrate the system's operation when moving the instruments within the phantom relatively slowly. However, this could prove to be a limitation in the case of faster or more abrupt movements. Under such conditions, the visual reference might reach the robot with a delay sufficient to result in less smooth tracking or a correction that is not perfectly synchronised with the instrument's current position. For an application closer to the clinical setting, an increase in the effective frequency of the pipeline would nevertheless represent a desirable improvement, achievable, for example, by reducing the resolution, introducing regions of interest, optimising segmentation, or separating acquisition and processing into distinct threads. However, the application scenario must also be taken into account: in a real anatomical context, the movements of the camera and instruments are generally more gradual and limited than those reproduced in the phantom setup. It can therefore be stated that the achieved frame rate does not compromise the validity of the proposed functional demonstration, despite the limitations on responsiveness.

5.3 The role of UDP communication

The results relating to communication show that the UDP protocol was a suitable choice for the type of architecture developed. The average Round-Trip time of 2.48 ms is much shorter than the vision system's cycle time and therefore does not constitute the system's limiting factor. Furthermore, the analysed log did not record any sentinel values of -1, which are used to indicate that an acknowledgement was not received within the specified timeout. Communication between the Python gateway and the KUKA controller therefore remained stable.

The decision to use UDP is coherent with the architecture because each packet contains a new estimate of the target's position. The system does not, in fact, necessarily require the delivery of all packets, but rather the availability of the most recent reference. In this sense, UDP helps to avoid additional delays associated with retransmission management, prioritising the continuity of the update.

It is important to emphasise that the measured value is a Round-Trip Time, not one-way latency. It includes the time taken to send the packet to the controller, its reception by the Java task, the generation of the response, and the return of the message to the Python gateway. Therefore, the value should not be interpreted solely as the time taken for the reference signal to reach the robot. However, as its order of magnitude is much lower than the vision cycle time, even a more precise unidirectional estimate would not alter the main conclusion: communication does not represent the bottleneck in the pipeline.

5.4 Discussion of the RCM error

The average RCM error of 3.10 mm indicates that the robotic motion remained, on the whole, close to the geometric constraint imposed by the virtual trocar. This result is significant because the RCM was not imposed via a physical mechanical constraint, but was calculated and maintained using a software-based geometric formulation.

The presence of a residual RCM error is to be expected because the constraint is imposed at the Cartesian destination position calculated by the control law, but the robot is not necessarily, at every instant, exactly in the desired pose. The geometric formulation defines a pose compatible with the virtual trocar; however, during execution, this pose is reached progressively via SmartServo and impedance control. Gain filtering, introduced to make the movement smoother, and the discrete updating of the visual reference, mean that the control is deliberately not instantaneous. What may therefore happen during tracking is that the camera's actual optical axis may temporarily deviate from the line passing through the virtual trocar. This results in a non-zero RCM error, which should be interpreted as the result of the overall dynamic response of the robot-camera system, rather than as a violation of the geometric formulation of the constraint.

Another factor to consider is that the camera's TCP has been modelled to coincide with the optical centre of the webcam, and that the calculated RCM error represents the distance between the virtual trocar and the optical axis defined in the robotic model. Any minor inaccuracies in the position of the TCP, in the actual alignment of the camera relative to its mount, or in the actual direction of the optical axis may therefore influence the measured value. This does not necessarily mean that the physical constraint has been maintained better or worse than reported, but indicates that the metric obtained describes the consistency of the motion relative to the implemented geometric model. Validation using an external measurement system would be necessary to distinguish more rigorously between the error due to control and that due to the modelling of the setup.

The temporal trend of the RCM error also shows oscillations around the mean value, which may be due to noise in the target estimate, low-frequency vision updates and transitions between control states. However, the boxplot confirms that most values remain concentrated within a limited range. The outlier of 11.69 mm was excluded from the statistical analysis as it was not representative of the main distribution of the signal. Nevertheless, its presence indicates that the system may be subject to occasional deviations.

5.5 The influence of the experimental mechanical setup

A significant limitation of the setup concerns the mount used to attach the camera to the robot's flange. The mount is approximately 40 cm long and was used to simulate an endoscope, but it does not fully replicate the endoscope's mechanical rigidity. Its elongated geometry makes it sensitive to the robot's movement and to external disturbances.

This aspect is particularly relevant because the camera itself provides the visual reference for the control system. If the mount vibrates, the acquired image is also subject to micro-oscillations. Such visual vibrations can affect marker segmentation, keypoint localisation and pose estimation. Consequently, a small mechanical vibration can propagate throughout the pipeline.

In this sense, some of the oscillations observed in the RCM error may not be attributable solely to the control formulation, but also to the mechanical nature of the mount. A real rigid endoscope, designed to maintain greater structural stability along its axis, could reduce this contribution. At the same time, in a real clinical setting, interaction with the trocar and with tissues would also introduce constraints and perturbations different from those of the experimental bench. Therefore, the mount used represents a suitable solution for preliminary validation, but cannot be considered mechanically equivalent to a real endoscope.

The presence of vibrations also has an impact on the assessment of tracking quality. The vision pipeline may appear less stable because the camera-support system oscillates slightly during acquisition. This makes it more difficult to separate the vision-related error from that due to mechanics. In a future evaluation, it would be useful to compare the same algorithm with a stiffer mount or with a real endoscope, in order to quantify the effect of the mechanical structure on the stability of the visual reference and on the RCM error.

5.6 The role of the state machine and filtering

The hysteresis state machine played a central role in stabilising the movement. Without modulating the control signal, even small variations in the visual reference could have triggered continuous micro-corrections of the camera. Greater problems would have arisen in the presence of segmentation noise, oscillations in the estimated pose or mechanical vibrations of the mount.

The deadband allows very small errors to be ignored, preventing the robot from chasing insignificant variations in the target. The SLOW zone allows for a more restrained response when an error is present but small, whilst the FAST zone makes the response more reactive when the target deviates further from the desired condition. Finally, the hysteresis between the thresholds reduces the risk of chattering, i.e. rapid and repeated switching between neighbouring states when the error oscillates around a limit value.

Dynamic gain filtering also contributes to stability. If the gain were to change instantaneously when switching from one state to another, the robot might receive commands with sudden changes in amplitude. The filter makes the transition more gradual and reduces the likelihood of mechanical jerks, improving the smoothness of the tracking.

These elements explain why the system has managed to maintain a low average RCM error. The control law does not merely transform the visual reference into a movement by the robot, but introduces strategies to prevent noise amplification. The combined action of the state machine and dynamic filtering has proved to be of fundamental importance in accommodating the limited update rate of the vision module. These strategies have prevented the processing delay associated with the geometric estimation from resulting in jerky or impulsive robotic corrections. However, stable and smooth tracking requires a compromise in terms of system responsiveness.

5.7 Limitations of the proposed work

The results obtained must be interpreted in light of certain experimental limitations. The first relates to the fact that the validation was carried out on an medical training phantom rather than in a real surgical setting. The phantom allows for the simulation of a geometry compatible with a minimally invasive approach, but does not replicate the anatomical complexity, tissue variability, presence of fluids, surgical smoke, intense reflections or occlusions typical of an intraoperative setting.

The second limitation concerns the acquisition system. The Logitech C920 was mounted on a support to simulate the footprint of an endoscopic camera. This

choice proved useful for validating the system's operation, but the camera differs significantly from an endoscopic one. The optics, lighting, field of view, mounting geometry and mechanical rigidity are profoundly different between the two. As discussed, the mount may introduce vibrations that affect both visual tracking and the quality of the RCM constraint.

The third limitation concerns the vision update rate. The camera operates at 30 fps, but the pipeline produces a usable reference at a frequency of approximately 5 Hz. This frequency is sufficient for an experimental demonstration involving controlled movements, but may be limiting in more dynamic scenarios. The developed system is therefore better suited to slow, gradual tracking rather than rapid instrument movements.

5.8 Future work

The results obtained suggest possible improvements, including the optimisation of the vision pipeline. One possible strategy for improvement is to reduce the frame resolution or to process only a region of interest around the marker. Another possibility is to separate frame acquisition from frame processing using a multithreaded structure.

Furthermore, it would be useful to replace the mount with a stiffer structure or with a real endoscope. This would reduce visual vibrations and allow the system's behaviour to be assessed in a more realistic configuration that is compatible with the intended application. A direct comparison between the current mount and a real endoscope would also make it possible to quantify the effect of mechanical stiffness on the RCM error and on tracking stability.

With regard to control, a significant future development could involve the introduction of an adaptive RCM. In the system developed in this thesis, the trocar was modelled as a static virtual point, defined at the start of execution and kept unchanged during tracking. This choice is consistent with preliminary validation in a controlled environment, but does not take into account any small displacements of the entry point or mechanical interactions between the instrument, the trocar and the tissues.

In a more advanced configuration, the RCM point could vary according to the forces acting at the trocar or the magnitude of the robot's joint torques. Any lateral forces could be used to detect misalignment between the instrument's axis and the fulcrum, thereby enabling the trocar estimate to be dynamically adjusted to reduce mechanical stresses at the entry point. This approach could address the limitation of the static trocar, making the system more realistic in terms of constraint handling within a surgical environment.

Finally, the system could be tested in more complex scenarios. Conditions of partial occlusion and lighting more similar to that found in real endoscopic procedures could be evaluated. This would allow the robustness of the pipeline to be assessed for a robotic assistance system designed for autonomous camera control.

Chapter 6

Conclusion

The work presented in this thesis concerned the development and preliminary validation of a robotic system for the autonomous control of an endoscopic camera under the Remote Centre of Motion constraints. The main objective was to verify whether a collaborative robotic platform, integrated with a vision pipeline and a dedicated control strategy, could autonomously orient the camera towards a visual target whilst maintaining geometric compatibility with the access point defined by the trocar.

The proposed architecture integrates three main levels: visual perception, communication and robotic control. The vision module identified the marker attached to the surgical instruments and estimated their three-dimensional position relative to the camera. Once processed, this information was transmitted to the KUKA controller via UDP communication. The robotic control program then converted the received reference into a Cartesian movement of the camera, whilst respecting the geometric constraint of the virtual trocar. The main contribution of this thesis therefore lies in the development and verification of a complete architecture capable of managing perception, communication and control within a single functional system.

The validation test was carried out on an medical training phantom and demonstrated the feasibility of the proposed approach. The system was able to detect the target, transmit the reference to the robot and generate an autonomous camera movement whilst maintaining the RCM constraint. The average cycle time of the vision pipeline is 189.70 $\pm$ 24.98 ms, corresponding to approximately 5 Hz. The average Round-Trip time for communication is 2.48 $\pm$ 1.42 ms, indicating that the communication layer is not the limiting factor of the system. The average RCM error is 3.10 $\pm$ 1.41 mm, indicating that the camera's optical axis remained, on the whole, close to the virtual trocar during the tracking movement.

These results support the overall validity of the developed architecture. The

deterministic marker-based vision pipeline provided an interpretable and repeatable target estimate of the target. The geometric formulation of the RCM constraint allowed an explicit relationship to be imposed between the camera's orientation and the position of the virtual trocar. Finally, the state machine with hysteresis logic and dynamic gain filtering helped to stabilise the motion, avoiding micro-corrections and reducing the risk of rapid switching between control states. Thanks to these design choices, the system maintained smooth behaviour even in the presence of discrete updates to the visual reference and small oscillations in the target estimate.

However, the results should be interpreted as a preliminary validation. The experimental setup allowed for certain relevant aspects of minimally invasive surgery to be taken into account, but not the full complexity of a real surgical environment. The phantom does not reprocude tissue deformability, fluids, surgical smoke, intense reflections or occlusions that may occur in an intraoperative setting. The camera is also a compromise: it is a Logitech C920 mounted on an elongated experimental support, designed to simulate the dimensions of a rigid endoscope. This configuration proved adequate for verifying functionality, but the camera differs significantly from a real endoscopic camera in terms of optics, lighting, field of view, mounting geometry and mechanical behaviour.

The main limitation of the current system is the effective frame rate of the vision pipeline. The camera was configured to an acquisition rate of 30 fps, and the complete processing pipeline generated a usable reference at approximately 5 Hz. This frame rate was sufficient for the experimental tests conducted, as the instruments were moved manually and progressively within the phantom. In more dynamic scenarios, however, rapid movements of the target might require greater responsiveness of the system. The decision to use high resolution improved visual quality and the localisation of marker features, but also increased the computational cost of the distortion correction, segmentation and pose estimation stages.

A further limitation concerns the static nature of the RCM constraint. In the system developed, the trocar was defined as a virtual point at the start of execution and was kept fixed during tracking. This choice is consistent with preliminary validation in a controlled environment, but represents a simplification compared with a real surgical procedure, in which the entry point may be influenced by tissue deformations, small patient movements or interaction forces between the endoscope and the trocar. For this reason, a significant future development could involve the introduction of an adaptive RCM, capable of updating the estimate of the access point based on additional information, for example derived from the measurement of forces, joint torques or external tracking systems.

Possible future improvements include increasing the responsiveness of the vision pipeline. The image resolution could be reduced, process only a region of interest

around the marker, optimise segmentation, or separate acquisition and processing using a multithreaded architecture. From a mechanical perspective, the support could be replaced with a real endoscope to more accurately assess the stability of the tracking and the accuracy with which the RCM constraint is maintained. The system should also be tested under more complex visual conditions, including partial occlusions, variable lighting and scenarios more representative of the real endoscopic environment.

This work has provided a preliminary validation of a robotic control architecture for the autonomous orientation of the endoscopic camera towards a visual target, whilst maintaining compatibility with the RCM constraint. The developed system has integrated deterministic visual tracking, UDP communication and constrained robotic control. Although further improvements and validation are required, the work carried out provides a solid basis for future developments in autonomous robotic assistance for camera manipulation in minimally invasive surgery.

Bibliography

[1] Chun-Li Wang, Gang Qu, and Hong-Wei Xu. «The short- and long-term outcomes of laparoscopic versus open surgery for colorectal cancer: a meta-analysis». In: *International Journal of Colorectal Disease* 29.3 (Mar. 2014), pp. 309–320. ISSN: 0179-1958, 1432-1262. DOI: 10.1007/s00384-013-1827-1. URL: http://link.springer.com/10.1007/s00384-013-1827-1 (visited on 05/30/2026) (cit. on p. 2).

[2] Rafał Watrowski, Stoyan Kostov, and Ibrahim Alkatout. «Complications in laparoscopic and robotic-assisted surgery: definitions, classifications, incidence and risk factors - an up-to-date review.» In: *Videosurgery and Other Miniinvasive Techniques* (Aug. 28, 2021), pp. 501–525. ISSN: 2299-0054. DOI: 10.5114/wiitm.2021.108800. URL: https://www.mp.pl/videosurgery/issue/article/16862/ (visited on 05/30/2026) (cit. on p. 2).

[3] Draga-Maria Mandi et al. «Aesthetic Outcomes and Patient Satisfaction in Laparoscopic vs. Open Incisional Hernia Repair: Have We Asked the Patients?» In: *Chirurgia* 119.3 (2024), p. 260. ISSN: 1221-9118. DOI: 10.21614/chirurgia.2024.v.119.i.3.p.260. URL: https://www.revistachirurgia.ro/pdfs/aesthetic-outcomes-and-patient-satisfaction-in-laparoscopic-vs-open-incisional-hernia-repair-have-we-asked-the-patients (visited on 05/31/2026) (cit. on p. 2).

[4] Giulio Dagnino and Dennis Kundrat. «Robot-assistive minimally invasive surgery: trends and future directions». In: *International Journal of Intelligent Robotics and Applications* 8.4 (Dec. 2024), pp. 812–826. ISSN: 2366-5971, 2366-598X. DOI: 10.1007/s41315-024-00341-2. URL: https://link.springer.com/10.1007/s41315-024-00341-2 (visited on 06/01/2026) (cit. on pp. 2, 3).

[5] Xiaoqin Zhou, Haijun Zhang, Mei Feng, Ji Zhao, and Yili Fu. «New remote centre of motion mechanism for robot-assisted minimally invasive surgery». In: *BioMedical Engineering OnLine* 17.1 (Dec. 2018), p. 170. ISSN: 1475-925X. DOI: 10.1186/s12938-018-0601-6. URL: https://biomedical-

engineering-online.biomedcentral.com/articles/10.1186/s12938-018-0601-6 (visited on 06/04/2026) (cit. on pp. 2, 6).

[6] Caspar Gruijthuijsen, Luis C. Garcia-Peraza-Herrera, Gianni Borghesan, Dominiek Reynaerts, Jan Deprest, Sebastien Ourselin, Tom Vercauteren, and Emmanuel Vander Poorten. «Robotic Endoscope Control via Autonomous Instrument Tracking». In: *Frontiers in Robotics and AI* 9 (Apr. 11, 2022), p. 832208. ISSN: 2296-9144. DOI: 10.3389/frobt.2022.832208. arXiv: 2107.02317[cs.RO]. URL: http://arxiv.org/abs/2107.02317 (visited on 05/30/2026) (cit. on pp. 3, 6, 10, 12, 13, 19).

[7] Mohammad S. A. Amin et al. «Evaluation of a remote-controlled laparoscopic camera holder for basic laparoscopic skills acquisition: a randomized controlled trial». In: *Surgical Endoscopy* 35.8 (Aug. 2021), pp. 4183–4191. ISSN: 0930-2794, 1432-2218. DOI: 10.1007/s00464-020-07899-5. URL: https://link.springer.com/10.1007/s00464-020-07899-5 (visited on 05/31/2026) (cit. on pp. 3, 6).

[8] Kristen H. Goodell, Caroline G.L. Cao, and Steven D. Schwaitzberg. «Effects of Cognitive Distraction on Performance of Laparoscopic Surgical Tasks». In: *Journal of Laparoendoscopic & Advanced Surgical Techniques* 16.2 (Apr. 2006), pp. 94–98. ISSN: 1092-6429, 1557-9034. DOI: 10.1089/lap.2006.16.94. URL: https://journals.sagepub.com/doi/full/10.1089/lap.2006.16.94 (visited on 05/31/2026) (cit. on p. 3).

[9] Kim Platte, Chantal C.J. Alleblas, Joanna Inthout, and Theodoor E. Nieboer. «Measuring fatigue and stress in laparoscopic surgery: validity and reliability of the star-track test». In: *Minimally Invasive Therapy & Allied Technologies* 28.1 (Jan. 2, 2019), pp. 57–64. ISSN: 1364-5706, 1365-2931. DOI: 10.1080/13645706.2018.1470984. URL: https://www.tandfonline.com/doi/full/10.1080/13645706.2018.1470984 (visited on 05/31/2026) (cit. on p. 3).

[10] Priscila Rodrigues Armijo, Chun-Kai Huang, Tyson Carlson, Dmitry Oleynikov, and Ka-Chun Siu. «Ergonomics Analysis for Subjective and Objective Fatigue Between Laparoscopic and Robotic Surgical Skills Practice Among Surgeons». In: *Surgical Innovation* 27.1 (Feb. 2020), pp. 81–87. ISSN: 1553-3506, 1553-3514. DOI: 10.1177/1553350619887861. URL: https://journals.sagepub.com/doi/10.1177/1553350619887861 (visited on 05/31/2026) (cit. on p. 3).

[11] L. S. G. L. Wauben, M. A. Van Veelen, D. Gossot, and R. H. M. Goossens. «Application of ergonomic guidelines during minimally invasive surgery: a questionnaire survey of 284 surgeons». In: *Surgical Endoscopy And Other Interventional Techniques* 20.8 (Aug. 2006), pp. 1268–1274. ISSN: 0930-2794,

1432-2218. DOI: 10.1007/s00464-005-0647-y. URL: http://link.springer.com/10.1007/s00464-005-0647-y (visited on 05/31/2026) (cit. on p. 3).

[12] Gyusung Lee, Tommy Lee, David Dexter, Carlos Godinez, Nora Meenaghan, Robert Catania, and Adrian Park. «Ergonomic risk associated with assisting in minimally invasive surgery». In: *Surgical Endoscopy* 23.1 (Jan. 2009), pp. 182–188. ISSN: 0930-2794, 1432-2218. DOI: 10.1007/s00464-008-0141-4. URL: http://link.springer.com/10.1007/s00464-008-0141-4 (visited on 05/31/2026) (cit. on pp. 3, 6).

[13] Martyn C. Stott, Jenifer Barrie, Domnic Sebastien, Christa Hammill, and Daren A. Subar. «Is the Use of a Robotic Camera Holder Economically Viable? A Cost Comparison of Surgical Assistant Versus the Use of a Robotic Camera Holder in Laparoscopic Liver Resections». In: *Surgical Laparoscopy, Endoscopy & Percutaneous Techniques* 27.5 (Oct. 2017), pp. 375–378. ISSN: 1530-4515. DOI: 10.1097/SLE.0000000000000452. URL: https://journals.lww.com/00129689-201710000-00012 (visited on 05/31/2026) (cit. on pp. 3, 6).

[14] Andre Luiz Gioia Morrell, Alexander Charles Morrell-Junior, Allan Gioia Morrell, Jose Mauricio Freitas Mendes, Francisco Tustumi, Luiz Gustavo De-Oliveira-E-Silva, and Alexander Morrell. «The history of robotic surgery and its evolution: when illusion becomes reality». In: *Revista do Colégio Brasileiro de Cirurgiões* 48 (2021), e20202798. ISSN: 1809-4546, 0100-6991. DOI: 10.1590/0100-6991e-20202798. URL: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-69912021000100302&tlng=en (visited on 06/04/2026) (cit. on pp. 4, 6).

[15] Andrew A. Wagner, Ioannis M. Varkarakis, Richard E. Link, Wendy Sullivan, and Li-Ming Su. «Comparison of surgical performance during laparoscopic radical prostatectomy of two robotic camera holders, EndoAssist and AESOP: A pilot study». In: *Urology* 68.1 (July 2006), pp. 70–74. ISSN: 00904295. DOI: 10.1016/j.urology.2006.02.003. URL: https://linkinghub.elsevier.com/retrieve/pii/S0090429506002202 (visited on 06/04/2026) (cit. on pp. 4, 6).

[16] S.E. Butner and M. Ghodoussi. «Transforming a surgical robot for human telesurgery». In: *IEEE Transactions on Robotics and Automation* 19.5 (Oct. 2003), pp. 818–824. ISSN: 1042-296X. DOI: 10.1109/TRA.2003.817214. URL: http://ieeexplore.ieee.org/document/1236755/ (visited on 06/02/2026) (cit. on p. 4).

[17] John W Yaxley et al. «Robot-assisted laparoscopic prostatectomy versus open radical retropubic prostatectomy: early outcomes from a randomised controlled phase 3 study». In: *The Lancet* 388.10049 (Sept. 2016), pp. 1057–

1066. ISSN: 01406736. DOI: 10.1016/S0140-6736(16)30592-X. URL: https://linkinghub.elsevier.com/retrieve/pii/S014067361630592X (visited on 06/02/2026) (cit. on p. 4).

[18] Induni N Weerarathna, David Raymond, and Anurag Luharia. «Human-Robot Collaboration for Healthcare: A Narrative Review». In: *Cureus* (Nov. 21, 2023). ISSN: 2168-8184. DOI: 10.7759/cureus.49210. URL: https://www.cureus.com/articles/206887-human-robot-collaboration-for-healthcare-a-narrative-review (visited on 06/04/2026) (cit. on p. 4).

[19] Guang-Zhong Yang et al. «Medical robotics—Regulatory, ethical, and legal considerations for increasing levels of autonomy». In: *Science Robotics* 2.4 (Mar. 2017), eaam8638. ISSN: 2470-9476. DOI: 10.1126/scirobotics.aam8638. URL: https://www.science.org/doi/10.1126/scirobotics.aam8638 (visited on 06/03/2026) (cit. on p. 5).

[20] Yanwen Sun, Bo Pan, Yili Fu, and Guojun Niu. «Visual-based autonomous field of view control of laparoscope with safety-RCM constraints for semi-autonomous surgery». In: *The International Journal of Medical Robotics and Computer Assisted Surgery* 16.2 (Apr. 2020), e2079. ISSN: 1478-5951, 1478-596X. DOI: 10.1002/rcs.2079. URL: https://onlinelibrary.wiley.com/doi/10.1002/rcs.2079 (visited on 06/11/2026) (cit. on pp. 10, 19).

[21] Tommaso Da Col, Andrea Mariani, Anton Deguet, Arianna Menciassi, Peter Kazanzides, and Elena De Momi. «SCAN: System for Camera Autonomous Navigation in Robotic-Assisted Surgery». In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Las Vegas, NV, USA: IEEE, Oct. 24, 2020, pp. 2996–3002. ISBN: 978-1-7281-6212-6. DOI: 10.1109/IROS45743.2020.9341548. URL: https://ieeexplore.ieee.org/document/9341548/ (visited on 06/11/2026) (cit. on p. 11).

[22] Yisen Huang, Jian Li, Xue Zhang, Ke Xie, Jixiu Li, Yue Liu, Sze Hang Ng, Philip Wai Yan Chiu, and Zheng Li. «A Surgeon Preference-Guided Autonomous Instrument Tracking Method With a Robotic Flexible Endoscope Based on dVRK Platform». In: *IEEE Robotics and Automation Letters* 7.2 (Apr. 2022), pp. 2250–2257. ISSN: 2377-3766, 2377-3774. DOI: 10.1109/LRA.2022.3143305. URL: https://ieeexplore.ieee.org/document/9682508/ (visited on 06/11/2026) (cit. on pp. 11, 12).

[23] Beili Dong, Junhong Chen, Zeyu Wang, Kaizhong Deng, Yiping Li, Benny Lo, and George Mylonas. «An Intelligent Robotic Endoscope Control System Based on Fusing Natural Language Processing and Vision Models». In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. 2024 IEEE International Conference on Robotics and Automation (ICRA).

Yokohama, Japan: IEEE, May 13, 2024, pp. 8180–8186. ISBN: 979-8-3503-8457-4. DOI: 10.1109/ICRA57147.2024.10611534. URL: https://ieeexplore.ieee.org/document/10611534/ (visited on 06/12/2026) (cit. on p. 12).

[24] Khusniddin Fozilov, Jacinto Colan, Ana Davila, Kazunari Misawa, Jie Qiu, Yuichiro Hayashi, Kensaku Mori, and Yasuhisa Hasegawa. «Endoscope Automation Framework with Hierarchical Control and Interactive Perception for Multi-Tool Tracking in Minimally Invasive Surgery». In: *Sensors* 23.24 (Dec. 16, 2023), p. 9865. ISSN: 1424-8220. DOI: 10.3390/s23249865. URL: https://www.mdpi.com/1424-8220/23/24/9865 (visited on 06/12/2026) (cit. on pp. 12, 21).

[25] Austin Reiter, Peter K Allen, and Tao Zhao. «Marker-less Articulated Surgical Tool Detection». In: () (cit. on p. 14).

[26] Juan Sandoval, Med Amine Laribi, Jean-Pierre Faure, Cyril Breque, Jean-Pierre Richer, and Said Zeghloul. «Towards an Autonomous Robot-Assistant for Laparoscopy Using Exteroceptive Sensors: Feasibility Study and Implementation». In: *IEEE Robotics and Automation Letters* 6.4 (Oct. 2021), pp. 6473–6480. ISSN: 2377-3766, 2377-3774. DOI: 10.1109/LRA.2021.3094644. URL: https://ieeexplore.ieee.org/document/9477029/ (visited on 06/12/2026) (cit. on pp. 15, 23).

[27] Xin Ma, Chengzhi Song, Philip Waiyan Chiu, and Zheng Li. «Autonomous Flexible Endoscope for Minimally Invasive Surgery With Enhanced Safety». In: *IEEE Robotics and Automation Letters* 4.3 (July 2019), pp. 2607–2613. ISSN: 2377-3766, 2377-3774. DOI: 10.1109/LRA.2019.2895273. URL: https://ieeexplore.ieee.org/document/8627918/ (visited on 06/12/2026) (cit. on p. 16).

[28] Xin Ma, Chengzhi Song, Philip Waiyan Chiu, and Zheng Li. «Visual Servo of a 6-DOF Robotic Stereo Flexible Endoscope Based on da Vinci Research Kit (dVRK) System». In: *IEEE Robotics and Automation Letters* 5.2 (Apr. 2020), pp. 820–827. ISSN: 2377-3766, 2377-3774. DOI: 10.1109/LRA.2020.2965863. URL: https://ieeexplore.ieee.org/document/8959288/ (visited on 06/12/2026) (cit. on p. 16).

[29] Jacinto Colan, Ana Davila, and Yasuhisa Hasegawa. «Constrained Motion Planning for a Robotic Endoscope Holder Based on Hierarchical Quadratic Programming». In: *2023 8th International Conference on Control and Robotics Engineering (ICCRE)*. 2023 8th International Conference on Control and Robotics Engineering (ICCRE). Niigata, Japan: IEEE, Apr. 21, 2023, pp. 198–203. ISBN: 979-8-3503-4565-0. DOI: 10.1109/ICCRE57112.2023.10155579. URL: https://ieeexplore.ieee.org/document/10155579/ (visited on 06/13/2026) (cit. on pp. 17, 20, 21).

[30] Jinze Shi, Chunlin Zhou, Luming Wang, Wenhan Lin, Song Zhou, Zhehao He, Honghai Ma, Jian Hu, and Dongqin Feng. «Hands-Free Camera Assistant: Autonomous Laparoscope Manipulation in Robot-Assisted Surgery». In: *The International Journal of Medical Robotics and Computer Assisted Surgery* 21.4 (Aug. 2025), e70103. ISSN: 1478-5951, 1478-596X. DOI: 10.1002/rcs.70103. URL: https://onlinelibrary.wiley.com/doi/10.1002/rcs.70103 (visited on 06/13/2026) (cit. on p. 18).

[31] Juan Sandoval, Gerard Poisson, and Pierre Vieyres. «A new kinematic formulation of the RCM constraint for redundant torque-controlled robots». In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Vancouver, BC: IEEE, Sept. 2017, pp. 4576–4581. ISBN: 978-1-5386-2682-5. DOI: 10.1109/IROS.2017.8206326. URL: http://ieeexplore.ieee.org/document/8206326/ (visited on 06/13/2026) (cit. on p. 18).

[32] Martin Huber, John Bason Mitchell, Ross Henry, Sebastien Ourselin, Tom Vercauteren, and Christos Bergeles. «Homography-based Visual Servoing with Remote Center of Motion for Semi-autonomous Robotic Endoscope Manipulation». In: *2021 International Symposium on Medical Robotics (ISMR)*. 2021 International Symposium on Medical Robotics (ISMR). Atlanta, GA, USA: IEEE, Nov. 17, 2021, pp. 1–7. ISBN: 978-1-6654-0622-2. DOI: 10.1109/ISMR48346.2021.9661563. URL: https://ieeexplore.ieee.org/document/9661563/ (visited on 06/13/2026) (cit. on p. 19).

[33] Pierre Berthet-Rayne, Gauthier Gras, Konrad Leibrandt, Piyamate Wisanuvej, Andreas Schmitz, Carlo A. Seneci, and Guang-Zhong Yang. «The i2Snake Robotic Platform for Endoscopic Surgery». In: *Annals of Biomedical Engineering* 46.10 (Oct. 2018), pp. 1663–1675. ISSN: 0090-6964, 1573-9686. DOI: 10.1007/s10439-018-2066-y. URL: http://link.springer.com/10.1007/s10439-018-2066-y (visited on 06/13/2026) (cit. on pp. 19, 21).

[34] Martin Wagner et al. «A learning robot for cognitive camera control in minimally invasive surgery». In: *Surgical Endoscopy* 35.9 (Sept. 2021), pp. 5365–5374. ISSN: 0930-2794, 1432-2218. DOI: 10.1007/s00464-021-08509-8. URL: https://link.springer.com/10.1007/s00464-021-08509-8 (visited on 06/13/2026) (cit. on p. 20).

[35] Hang Su, Shuai Li, Jagadesh Manivannan, Luca Bascetta, Giancarlo Ferrigno, and Elena De Momi. «Manipulability Optimization Control of a Serial Redundant Robot for Robot-assisted Minimally Invasive Surgery». In: *2019 International Conference on Robotics and Automation (ICRA)*. 2019 International Conference on Robotics and Automation (ICRA). Montreal, QC,

Canada: IEEE, May 2019, pp. 1323–1328. ISBN: 978-1-5386-6027-0. DOI: 10.1109/ICRA.2019.8793676. URL: https://ieeexplore.ieee.org/document/8793676/ (visited on 06/13/2026) (cit. on p. 21).

[36] Ehsan Nasiri and Long Wang. «Admittance Control for Adaptive Remote Center of Motion in Robotic Laparoscopic Surgery». In: *2024 21st International Conference on Ubiquitous Robots (UR)*. 2024 21st International Conference on Ubiquitous Robots (UR). New York, NY, USA: IEEE, June 24, 2024, pp. 51–57. ISBN: 979-8-3503-6107-0. DOI: 10.1109/UR61395.2024.10597442. URL: https://ieeexplore.ieee.org/document/10597442/ (visited on 06/13/2026) (cit. on pp. 22, 23).

[37] Theodora Kastritsi and Zoe Doulgeri. «A passive admittance controller to enforce Remote Center of Motion and Tool Spatial constraints with application in hands-on surgical procedures». In: *Robotics and Autonomous Systems* 152 (June 2022), p. 104073. ISSN: 09218890. DOI: 10.1016/j.robot.2022.104073. URL: https://linkinghub.elsevier.com/retrieve/pii/S0921889022000367 (visited on 06/16/2026) (cit. on p. 23).

[38] «KUKA Sunrise.OS Med 2.7, KUKA Sunrise.Workbench Med 2.7». In: () (cit. on p. 26).

[39] João Cartucho, Chiyu Wang, Baoru Huang, Dan S. Elson, Ara Darzi, and Stamatia Giannarou. «An enhanced marker pattern that achieves improved accuracy in surgical tool tracking». In: *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization* 10.4 (July 4, 2022), pp. 400–408. ISSN: 2168-1163, 2168-1171. DOI: 10.1080/21681163.2021.1997647. URL: https://www.tandfonline.com/doi/full/10.1080/21681163.2021.1997647 (visited on 06/23/2026) (cit. on pp. 31, 33, 35).