POLITECNICO DI MILANO

School of Industrial and Information Engineering

Master of Science in Automation and Control Engineering

Real-Time Solution for Long-Term Tracking of Soft Tissue Deformations in Surgical Robots

Supervisor: Prof. Marcello Farina

Co-supervisor: Prof. Giancarlo Ferrigno

Master Thesis Dissertation by:
Martin Vasilkovski
Student ID 882555

Academic Year 2018/2019

ABSTRACT

Robot-assisted surgery (RAS) is a type of surgery performed by a robotic system in collaboration with a surgeon-operator. Despite the numerous benefits introduced by robotic systems in Minimally Invasive Surgery (MIS), complications (such as intra-operative bleeding) are still plausible and likely to affect the outcome of the procedure. Blood vessels can be accidentally damaged by unintentional contact with surgical tools, or by motion in sub-surface areas which are not visible to the operator. Assistive guidance tools represent a possibility to correct surgical gestures and are a big step forward towards safer procedures in the Operating Room (OR). The only component of a surgical robot that can provide insight of the interaction between surgical tools and the protected vessels is an endoscope equipped with a stereo camera. Creating an image analysis framework that can provide stable, robust and noise invariant solution for real-time implementation in a surgery, is yet to be overcome. However, once done, an assistive tool can be provided to surgeons that will correct the tool movement and notify them if a critical tissue is in danger of being injured. The introduction of Active Constraints (AC) is the first step towards safer surgeries, and this thesis is providing a tracking tool that will transform the AC accordingly in real-time during surgical procedures. The aim of this thesis is to develop a computer vision algorithm to robustly track areas of soft tissue, defined intra-operatively by a surgeon-operator based only on a real-time endoscopic video stream. The proposed framework combines feature tracking and adaptive recognition algorithms to track, localize and redefine the considered soft tissue after a tracking failure has occurred due to occlusion or severe deformation. The performance is assessed on two datasets, representing a controlled environment and a real-world in-vivo pancreatectomy. The results demonstrate that the proposed method successfully tracks and rediscovers the region of interest with good performance while maintaining real-time computing.

Key words: robot-assisted surgery, soft-tissue tracking, endoscopic image analysis, anatomy-based constraint, online adaptive recognition, real-time

SOMMARIO

Un intervento Robot-Assistito è un tipo di intervento effettuato da un chirurgo in collaborazione con un sistema robotico. Nonostante i numerosi benefici che un intervento Robot-Assistito ha introdotto nell'ambito degli interventi poco invasivi, alcune complicazioni (come le perdite di sangue durante l'intervento stesso) sono ancora possibili ed essi possono condizionare il risultato finale dell'operazione. I vasi sanguigni possono essere danneggiati da un contatto accidentale con gli strumenti chirurgici, o attraverso un movimento che avviene sotto la superficie di un'area non visibile dal chirurgo. Una possibile soluzione per la correzione dei movimenti chirurgici sono gli strumenti di guida che rappresentano un notevole passo avanti verso procedure sempre più sicure all'interno delle Sale Operatorie. L'unico componente di un robot chirurgico che può fornire informazioni riguardo l'interazione tra strumenti chirurgici e vasi sanguigni protetti, è un endoscopio equipaggiato con una stereo-camera. Creare un framework di analisi delle immagini che può fornire una soluzione stabile, robusta e non affetta da rumore della dinamica che circonda questo tipo di intervento, è un obbiettivo ambizioso. In ogni modo, una volta raggiunto, strumenti di assistenza basati su questo framework possono affiancarsi al chirurgo che correggerà i movimenti errati e li notificherà nel caso in cui un tessuto critico corre il pericolo di essere danneggiato. L'introduzione di vincoli attivi è il primo passo verso interventi sempre più sicuri.

Questa tesi illustra lo sviluppo di un sistema di tracciamento di vincoli attivi in grado di lavorare in tempo reale durante l'intervento stesso. Lo scopo di questa Tesi è infatti quello di sviluppare un algoritmo di visione artificiale per tracciare in modo robusto i tessuti morbidi, che il chirurgo, basandosi su una visualizzazione su schermo in tempo reale, definisce durante l'intervento. Il metodo proposto combina algoritmi di tracciamento e di riconoscimento additivi per tracciare, localizzare e ridefinire i tessuti morbidi considerati dopo che è stato rilevato un errore dovuto a una mancanza di visuale o a una deformazione. Il risultato è stato valutato su sue insiemi di dati sperimentali, rappresentanti un ambiente controllato e una pancreatectomia in-vivo. I risultati hanno dimostrato che i metodi proposti tracciano e riscoprono con alti valori di successo le regioni di interesse su tutte le metriche delle prestazioni calcolate, mantenendo un calcolo in tempo reale.

Parole chiave: intervento Robot-assistito, tracciamento di tessuti morbidi, analisi di immagine endoscopica, vincoli anatomici, riconoscimento additiva online, tempo reale.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
Introduction

The introduction of Robots in Minimally Invasive Surgery (RMIS) allows to overcome many obstacles introduced by traditional laparoscopic techniques, by improving surgeon dexterity and the ergonomics during the surgical procedure and restoring the surgeon hand-eye coordination.

Since their introduction in cardiosurgery, robots have entered all surgical subspecialties. Hundreds of robotic systems are commercially available, and the most widely known are the da Vinci System (Intuitive Surgical, Sunnyvale, CA, USA), Zeus and Aesop (Computer Motion, Goleta, CA, USA), RoboDoc (Integrated Surgical Systems, Sacramento, CA, USA), and Naviot (Hitachi Ltd., Tokyo, Japan). Advanced robots now assist surgeons in procedures, which were unthinkable just a few years ago, ranging from minimally invasive surgery in laparoscopy to complex reconstruction surgery.

Despite the increased adoption of robot-assisted surgery (RAS), the execution of surgical tasks on soft tissue remains entirely manual under a human-controlled paradigm. Functional outcomes, including complication rates, have remained highly variable owing to human factors, such as a surgeon’s hand-eye coordination and experience. With more than 44.5 million soft tissue surgeries in the United States each year, autonomous soft tissue surgery promises substantial benefits through improved safety from reduction of human errors, increased efficiency due to procedure time reduction, and potential access to optimal surgical techniques and consistent outcomes independent of surgeon training, condition, or experience.

In current years, research on RMIS has been very popular and evolving at an exponential rate. Research institutions and clinics around the world have been adopting RMIS, and it is one of the most popular topics worldwide by constantly attracting funding for new ideas to improve the overall structure of medical robots.

Compared to conventional surgery, surgical robots have many advantages, but also many disadvantages. While they are more dexterous and precise while replicating the operator's motion, they are still prone to mistakes from the surgeon. The introduction of active constraints as a concept in surgical robots provides the opportunity to decrease errors whether the surgeon is a junior surgeon or an expert one.

Active constraints allow the surgeon to decide what region of the patient's organs should not be touched during surgery. This would provide minimal bleeding and even less surgical injuries. A complex algorithmic solution is needed to succeed to follow and track all of the movements of the organs inside the patient's body in order to transform the active constraint while the patient is breathing or unconsciously moving. Stochastic environments, such as surgeries, require incredible knowledge, powerful computational machines and complex mathematical solutions. In addition, a general solution is needed because each patient is different than the others. To date, no direct solution has been created to allow precise, stable and real-time constrained control of surgical robots. Such control would be able to track and estimate the deformations of the selected organ and force the robot out of a precisely defined area (i.e. a vein) to protect that area, achieving a safer environment and removing many injury prone situations.

The work done in this thesis is meant to be an integral part of the SMARTsurg project [1], funded by the European Commission's Directorate-General for Research and Innovation (DG RTD), under its Horizon 2020 Research and innovation programme (H2020).

The present thesis work consists of seven chapters. In Chapter 2, the latest developments of tissue deformation tracking are presented, as well as state of the art research work conducted on real-time active constraints implementations. In Chapter 3 and Chapter 4 the proposed method is presented along with brief and functional explanation of pre-existing methods. In Chapter 5, experimental results and validation of the proposed method are presented, along with comparisons with existing solution

CHAPTER 2
State of the art in medical robots

2.1. Medical robots – General overview

Advances in engineering have paved the way in the past 20 years to the development of highly flexible and functional robots, which assist surgeons during surgery. These robots are called medical (or surgical) robots and the surgeries where they are used are called robot-assisted surgeries. Many innovative surgical techniques can be now implemented and performed in spaces too difficult to reach and/or visualize without the help of this technology. The first surgical robots ever used in a procedure is the PUMA 560 (Unimation, Danbury, CT, USA) and NeuroMate (Integrated Surgical Systems, Davis, CA, USA). They are adaptations of the technology available in the late 80s which were predominantly based on technology developed for the industrial sector. The latter are the first robots who have bridged the gap between industrial robotics and neurosurgery and stereotactic biopsy. A major positive step in the field of surgery was made when a robot was first used in the theater of surgery about 25 years ago. The robot was a PUMA 200 (Westinghouse Electric, Pittsburgh, PA) which was used for needle placement in a CT-guided brain biopsy [2]. Since then it has been exciting to see that the field of robotic surgery grows in leaps and bounds.

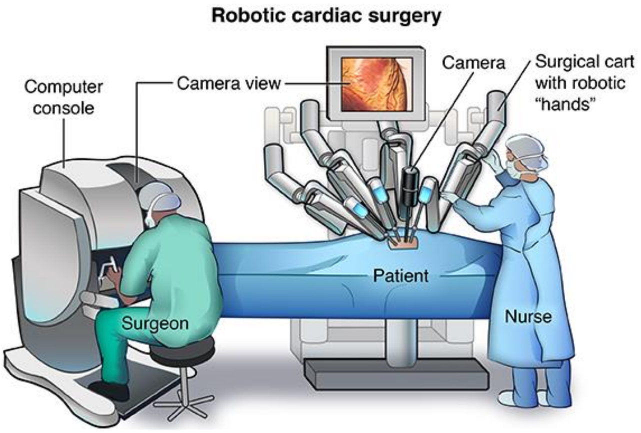


Figure 2.1 Robot-assisted surgery setup

The field of robotics has expanded and has been delivering many options for surgical procedures and different solution for ease of use. Latest advanced robots can assist surgeons in such procedures that have been unthinkable years ago, ranging from minimally invasive laparoscopy to complex reconstruction surgery and neurosurgery.

Robotic surgery, or robot-assisted surgery (RAS), allows doctors to perform many types of complex procedures with more precision, flexibility and control than is possible with conventional techniques. Robotic surgery is usually associated with minimally invasive surgery (MIS) — procedures performed through tiny incisions. It is also sometimes used in certain traditional open surgical procedures. The bond between RAS and MIS has become very strong, leading to certain MIS being only done with RAS. The benefits of minimally invasive surgery include:

- Fewer complications, such as surgical site infection
- Less pain and blood loss
- Quicker recovery
- Smaller, less noticeable scars

However, robot-assisted surgery involves risk, some of which may be similar to those of conventional open surgery, such as a small risk of infection and other complications.

2.2. Computer – assisted surgery

Computer-assisted surgery (CAS) is a surgical concept and set of methods, that use computer technology for surgical planning, and for guiding or performing surgical interventions. CAS is a leading factor in the development of robotic surgery.

General principles of CAS include:

- **Creating a virtual image of the patient**: this is conducted through a number of medical imaging technologies including Computer Tomography (CT), Magnetic Resonance Imaging (MRI), x-rays, ultrasounds and many more. For this generated 3D model, the anatomical region to be operated needs to be scanned and uploaded into the computer system.
- **Image analysis and processing**: it involves the manipulation of the patient's model to extract relevant information from the data using different image processing techniques.
- **Diagnostic, preoperative planning, surgical simulation**: using specialized software. The gathered dataset, rendered as a virtual 3D model of the patient, can be easily manipulated by a surgeon to provide views from any angle and at any depth within the volume, thus gaining the ability to better assess the case and establish more accurate diagnosis.
- **Surgical navigation**: in CAS, the actual intervention is defined as surgical navigation. Using the surgical navigation system, the surgeon uses special instruments which are tracked by the navigation system. The position of a tracked instrument in relation to the patient's anatomy is shown on images of the patients, as the surgeon moves the instrument.

2.3. Surgical navigation system

A surgical navigation system is similar to the common automotive navigation system, in the sense that both of them attempt to localize or determine a position in space in the context of its surroundings. However, the actual technology used differs by a great margin as surgical navigation is not able to use triangulation such as a global positioning system (GPS). Modern surgical navigation systems use a stereoscopic camera emitting infrared light which can determine a 3D position of particular structures, like reflective marker spheres. This allows for real-time tracking of the marker spheres. As presented in the paper of U. Mezger et. al. [3], this basic setup includes a stereoscopic camera, computer platform with screen and proprietary navigation software. Camera movements are intraoperatively possible because only the tracked instruments of the tracked patient reference is relevant.

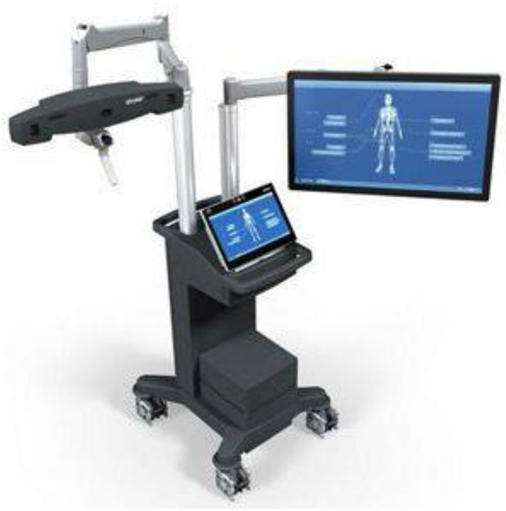


Figure 2.2 Surgical navigation system

A big improvement to this approach is using preoperative knowledge, usually done by acquiring imaging data from preoperative CT or MRI images which will improve the navigation in the Operating Room (OR). This type of approach is called “image-based”. The patient’s 3D model is created from the CT or MRI scans, and using image registration it is matched to the current patient position. This is to establish a relation to the “real” coordinate system.

A state-of-the-art image-guided surgical navigation allows surgeons to actively look inside of the body and see lesions inside cavities, narrow passages and hidden tumors positioned deeply inside of the tissue. In classical surgeries, an endoscope can also be used for helping the surgeon to analyze the tissue but for this, first the surgery must be paused so that the surgeon can view the endoscope output on a monitor. In robot-assisted surgeries, there is no need for such delay. The surgeon is always actively focused on the tissue that needs to be removed or operated on.

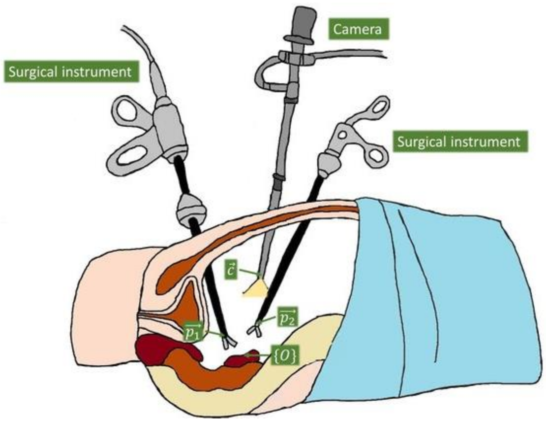


Figure 2.3 Intraoperative surgical navigation setup

2.4. da Vinci Robotic system

Ever since 1995 the USA based company "Intuitive Surgical" has been developing robotic-assisted systems to empower doctors and hospitals to make surgery less invasive than an open approach. This has resulted with one of the most advanced systems called "da Vinci". It is responsible for more than 6 million minimally invasive surgeries worldwide by 2018.

The da Vinci Surgical System (dVSS) is a telerobotic surgical system assembled using the da Vinci Research Kit (dVRK), a collection of robotic components from the first-generation da Vinci Surgical System provided by Intuitive Surgical. It includes controllers developed at Johns Hopkins University and Worcester Polytechnic Institute, and software developed at John Hopkins University. The dVSS is also the only US Food and Drug Administration (FDA) – approved robotic system for surgery since 2000.

Complex activities such as surgeries require a complex system with many components working synchronously. The components of a surgical robot, in this case dVSS, are the following:

Surgeon Console:

- Using the da Vinci Surgical System, the operator is seated comfortably in front of a console from where he/she can operate while viewing a high definition, 3D image inside of the patient’s body.
- There are two da Vinci Master Tool Manipulators (MTMs) with force feedback which are at arm’s length, and the operator uses his/her fingers to grab the master controls attached to these components. They are placed slightly below the display to allow the operator a natural and comfortable movement and positioning of hands and wrists relative to the eyes.
- The system replicates and translates the operator’s hand movements into precise, real-time motion of all surgical elements.
- A foot pedal tray, as part of dVSS, may be used by the operator for multiple purposes such as moving the camera, reconfiguring the sitting position, disabling movement etc.

Patient-side cart:

- This refers to he place where the patient is positioned during surgery.
- Includes three to four da Vinci Patient Side Manipulators (PSMs), or robotic arms, that carry out the operator’s commands.
- PSMs move around fixed pivoting points located at the “core” (remote center of motion) where the instruments are inserted.
- It is required that every maneuver is under the direct control of the surgeon, making the robot an assistant to the surgeon instead of an automatic device. Fail-safe systems and repeated safety checks prevent any independent movement of the instruments or robotic arms.
- Each PSM has seven degrees of freedom, which is even greater than the four degrees of freedom that the human wrist has.

Vision system:

- The vision system is equipped with a high-definition 3D endoscope. It is a flexible tube with stereo camera and a light at the tip. Alongside this there is a high-level image processing equipment that provides true-to-life images of the patient’s anatomy.
- The operating field is viewable to the entire Operation Room team on a large viewing monitor, placed on the vision cart. This widescreen view provides the

surgical assistants at the patient's side with a broad and detailed perspective of the procedure, albeit in 2D.

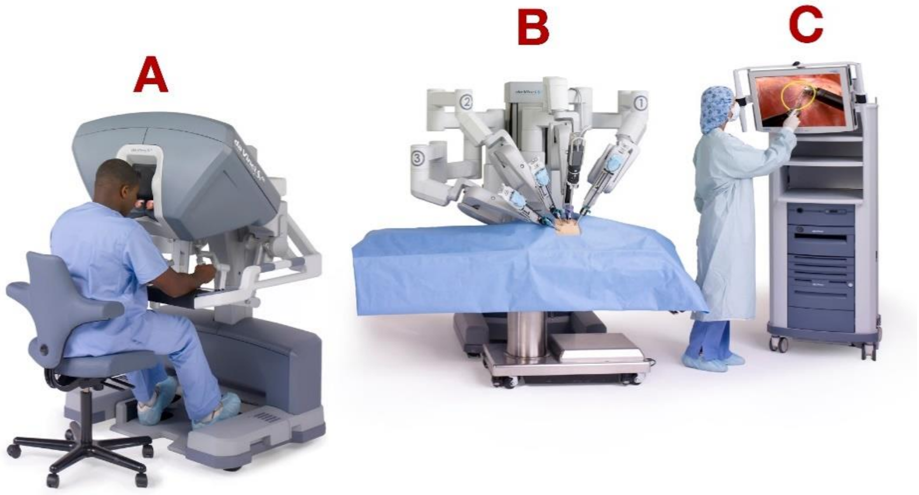


Figure 2.4 da Vinci Surgical System setup. A) Surgeon cart; B) Patient cart; C) Vision system

Endowrist Instruments:

- The line of dVSS-specific surgical tools is called Endowrist Instruments
- A full range of instruments is available to the surgeon while operating.
- Each instrument is designed for the specific surgical application such as clamping, cutting, cauterizing, suturing and tissue manipulation.
- Quick-release levers are used for fast instrumental change during surgery.

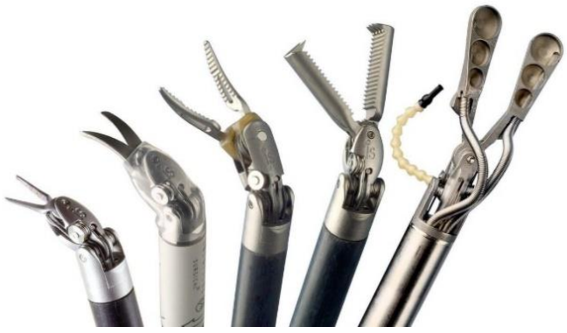


Figure 2.5 Endowrist instruments

The system enables a minimally invasive approach to traditional laparoscopic surgeries by allowing entire operations to be performed through relatively small incisions. In traditional laparoscopic surgeries, a larger incision is made in the patient’s body and the surgeon performs the operation while standing using long-shafted instruments and viewing their movements on a nearby video screen. These surgeries typically last many hours and can be quite exhausting for the surgeon. In contrast, the da Vinci Surgical System allows the surgeon to perform the entire operation while seated at an ergonomic console. The surgeon rests their head on a soft pad in a downward-facing position, as it would be if they were performing the surgery directly, and views their movements through a high-resolution, stereoscopic display of the video feed from the endoscopic camera inserted in the patient. The camera can be controlled with the Endoscopic Camera Manipulator (ECM). The MTMs are placed where the surgeon’s hands would be if they were performing the surgery directly. The motion of the MTMs is translated to scaled motion of the PSMs with seven degrees of freedom, and the PSMs also feature tremor cancellation to reduce potential shakiness of the surgeon’s hands.

2.5. The da Vinci Research kit

The da Vinci Research Kit, however, differs from the full DVSS. The Research Kit is a collection of first-generation da Vinci components that can be used to assemble a research platform for exploring telerobotics in medicine. The Kit contains the following components:

- Two da Vinci Master Tool Manipulators (MTMs)
- Two da Vinci Patient Side Manipulators (PSMs)
- A stereo viewer
- A foot pedal tray
- Manipulator Interface Boards (dMIBs)
- Basic accessory kit

The dVRK began as an attempt to create an open-source telerobotics research platform from an existing complete telesurgical system [4]. Because the da Vinci

Surgical System is a proprietary product and was not meant to be an open source product, entirely new controller hardware had to be designed and produced to allow complete access to all control points. The hardware is based on an approach known as centralized computation and distributed I/O, by which a real-time communication network allows all control computations to be implemented on a high-performance computer while keeping the I/O distributed, thereby preserving the advantages of reduced cabling.

Though the newest model of the da Vinci Surgical System includes two MTMs, three PSMs, and one ECM, the actual dVRK includes only two MTMs and two PSMs. Nevertheless, it is a simple straigthforawd procedure to combine dVRK electronics and software with a full da Vinci Surgical System. To control a single manipulator, two FPGAs controller boards are needed and each of these boards required a unique ID to be properly addressed by the communication protocol.

2.6. da Vinci Research kit vision system

The dVRK is equipped with a stereo camera enclosed in an endoscope. Unlike dVSS, which has an ECM, dVRK is using a rigid endoscope. A rod-lens endoscope, design proposed and patented by Harold Hopkins, was proposed in order to improve on previous rigid endoscopes which had low light transmittance and poor image quality. Tomkinson et al. in their paper have provided an extensive comparative study on rigid endoscopic relay systems [5] and their experimental use in different applications.

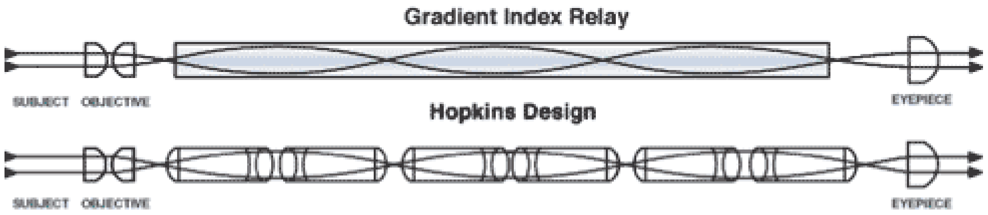


Figure 2.6 Gradient Index Relay vs Hopkins Rod-lens design

The essential components of an endoscope are:

- **A rigid or flexible tube –** depending on whether the Endoscope is Rigid Endoscope or Flexible Endoscope

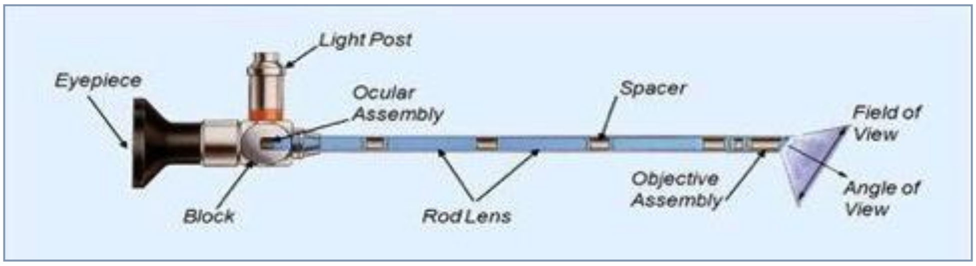


Figure 2.7 Rigid tube endoscope

- **Light delivery system** to illuminate the organ or object in focus. The light source is normally outside the body and the light is typically directed through optical fibers.
- **Objective lens** turns the light into an image, projecting it onto the following component. It comprises of between two and nine lenses as well as a prism if different viewing directions are required.
- **Relay System** transmits the image from the objective lens to the viewer, typically a Rod- relay lens system in the case of rigid endoscopes or a bundle of fiber optics in the case of a fiberscope.
- **Eyepiece lens -** The eyepiece lens magnifies the image transmitted by the rod lenses, providing the viewer with a large image circle. Depending on the use, this can be implemented for a camera system.

The rod lens system was developed by Hopkins and therefore referred to as the Hopkins System.

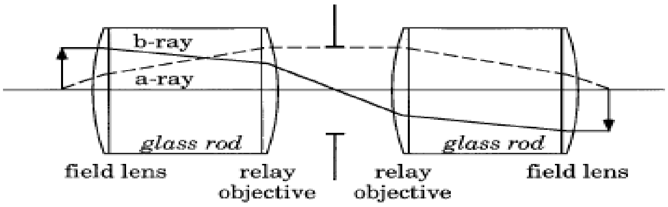


Figure 2.8 One stage of Hopkins Rod Lens

The reversal system consists of a series of rod lenses (relay lenses). They serve to transmit the image within the endoscope. Rod lenses made of glass provide a clearly higher light transmission efficiency compared to conventional lenses where the area filled with air between the lenses is relatively large.

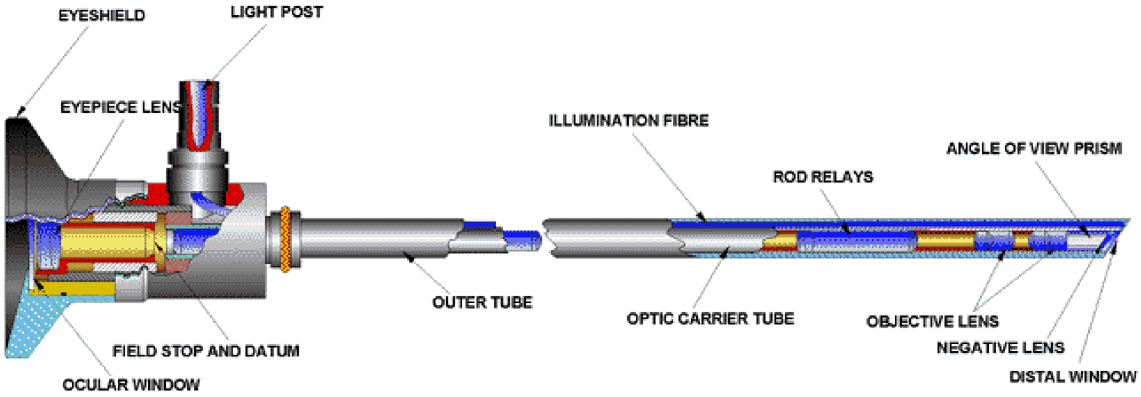


Figure 2.9 Section drawing of a typical rod lens endoscope

The focus control unit is controllable through a foot pedal and is capable of automatic and manual focus adjustment. The two Camera Control Units (CCU) are bolted to the vision cart, which are directly connected to the optical cables, which lead to the video sensors on the tip of the endoscope.

The endoscope is entirely motorized and controllable by the surgeon. One of the robotic arms is dedicated to the endoscope, and by pressing one of the foot pedals the operator can gain instant access to the endoscope and position it properly during surgery.

2.7. Image registration and 3D reconstruction

In robotically assisted Minimally Invasive Surgery (MIS), recovering of 3D structure of the operating field is crucial for registering pre-operative data to the surgical field-of-view for providing dynamic active constraints (to be introduced later in this chapter) and motion control near the operated tissue. Tomographic imaging can potentially provide anatomical information about the 3D shape and morphology of the soft tissues, but their implementation in operating theatres is a great challenge [2]. Currently, the most practical method of recovering the 3D structure of the operating site is through optical techniques using a stereo laparoscope. This information can be used to align multimodal information within a global reference 3D coordinate system and enhance robotic instrument control. However, the recovery of 3D geometry from stereo imaging during robotic procedures is difficult due to tissue deformation, partial occlusion due to instrument movement, and specular inter-reflections.

The recovery of 3D information from stereo images is one of the greatest challenges in the field of computer vision. Given a calibrated stereo vision rig, the task is to identify the unique correspondence across a stereo image pair. Recent review articles [6], [7] provide a good summary of progress in the field of 3D reconstruction.
In the study of Stoyanov et.al. [6], a technique is proposed for building a semi-dense reconstruction of the operating field in MIS that can operate in real-time. The method starts with sparse 3D reconstruction based on feature matching across the stereo pair and subsequently propagates structure into neighboring image regions.

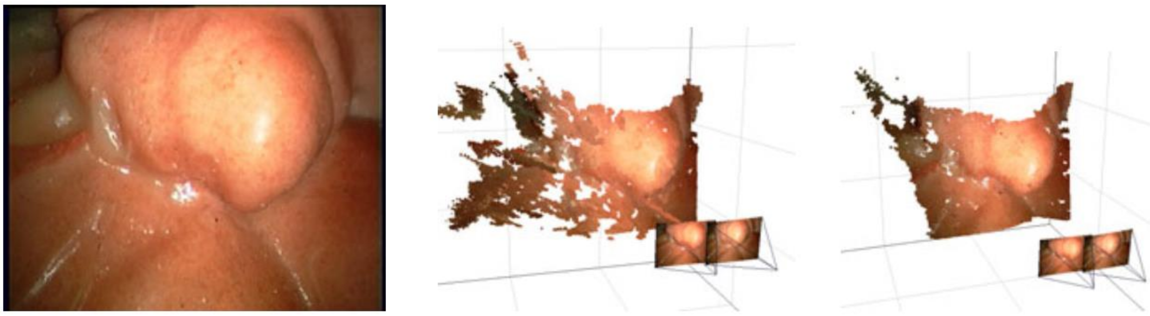


Figure 2.10 Example images from stereo-laparoscope and the corresponding stereo reconstructions [5]

The proposed method which was tested through two 3D reconstruction experiments applied on two different datasets with ground truth obtained by CT scan data, and experimentally gave disparity error 0.89 [∓1.13] and 1.22 [∓1.71] pixels, with respect to both experiments. The mean disparity error with value of 1/10 with respect to the other compared methods BP, RT, CUDA with the only setback of having a semi real-time processing by approaching a maximum processing speed of 15 Hz for compressed images to resolution of 360 x 288 pixels.

Kowalczuk et al [8], propose a method which does not depend on preoperative CT or MRI scans, but rather acquiring knowledge by applying image-based techniques that perform stereo matching only on the images obtained from a stereoscopic camera (Figure 2.11).

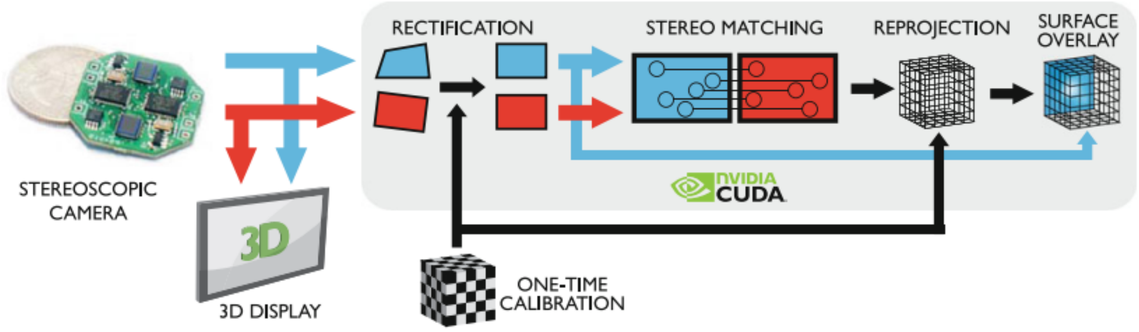


Figure 2.11 Flow diagram of the proposed digital stereoscopic method [8]

Experimental results from a real-time generation of a 3D model of porcine surgery show that the measurements extracted from the 3D model differ from those obtained manually by <1.5 mm, resulting in a mean absolute error of 0.637 mm.

One of the latest advances in real-time 3D reconstruction of soft tissue is presented in the work of H. Zhou and J. Jagadeesan [9]. ORB-SLAM framework is implemented as the basis for this work which can be seen in Figure 2.12, but in order to overcome difficulties with feature matching in surfaces with repeating textures, a novel histogram voting scheme is introduced along with a novel 1-point RANSAC based algorithm. According to the paper, a semi real-time computation is achieved at 13.1 Hz with no compression of the video resolution. The achieved precision averages root mean square errors in the range from 1.3 to 2 pixels acquired from experiments on datasets with 960 x 540 pixels.

A setback for most of these methods is that they are assuming a close-to-static environments with minimal deformations between frames and are suitable on for static or semi-static minimally invasive surgeries.

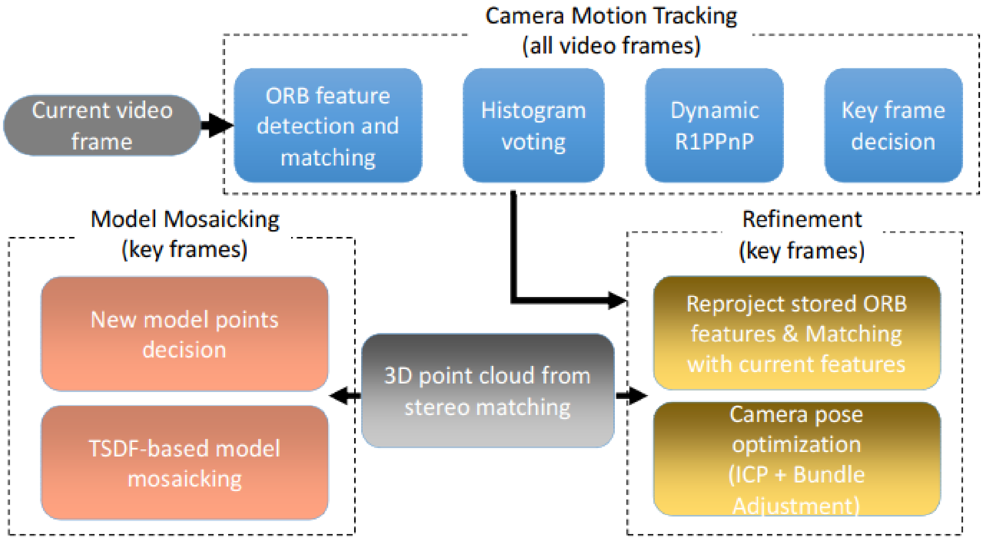

Camera Motion Tracking
(all video frames)
Current video frame
ORB feature detection and matching
Histogram voting
Dynamic R1PPnP
Key frame decision
Model Mosaicking
(key frames)
New model points decision
TSDF-based model mosaicking
3D point cloud from stereo matching
Refinement
(key frames)
Reproject stored ORB features & Matching with current features
Camera pose optimization
(ICP + Bundle Adjustment)

Figure 2.12 Flow chart of proposed method [9]

2.8. Advances in haptic feedback

Human-machine communication is predominantly based on only two senses: sight and hearing. Haptic feedback is a mode of communication rather than a specific technology or application. The tactile sensations named "touch" are part of what is known as the somatosensory system. This sensory system also includes a huge variety of sensations, including touch, vibration and pressure. These three sensations are some of the principles of how haptic feedback is implemented. An everyday use is the vibrations of a telephone which mimic the sensation of pressing a button.

One of the setbacks of robot-assisted surgery is the loss of haptic feedback. Though robotic surgery has many benefits over conventional surgery such as motion scaling for finer motion control, stereoscopic vision, increased dexterity and additional degrees of freedom, loss of haptic feedback is one setback that can make a difference. The ability of robotic surgical systems to apply strong compressive and shear forces has led to the increased risk of tissue damage, reduced performance and increased number of mistakes [10]. As RAS become more popular, implementation of haptic feedback systems (HFS) become more present in every commercially available solution.

Having an advanced system which will provide the surgeon with multiple sensory excitation that can convey tactile information of pressure applied and tool-tissue interference is of great importance and a way to improve human-machine communication in robot-assisted surgery.

A. Abiri et al [11] provide an advanced multi-modal solution to haptic feedback devices for surgical robots, claiming that Kinesthetic force feedback (KFF) is the most broadly researched area of haptics [12] because its relative ease of integration with the master controls of surgical robots. Their proposed solution is having a multi-modal haptic feedback system with the goal to convey more than one aspect of touch by targeting multiple classes of mechanoreceptors in the skin and muscles. The proposed sensory excitations are pressure and vibration. The experimental results show a comparison between: free hand grip, da Vinci robot grip with no HFS, and tri-modal HFS applied to the da Vinci, where he average forces applied are 0.88 N, 2.78 N and 1.27 N respectively.

These numbers provide a conclusion that a combined tactile sensation to the surgeon improves safety and is able to provide a more natural communication between the surgeon and the surgical tool. However, haptic feedback does not have to only be used for tactile sense of the pressure applied to the tissue. One other possible implementation is for guidance of the tool through a path or to protect going into a possible region of the organ which must be protected. Such concept as the latter is called active constraints.

2.9. Active constraints

Active constraints (AC) can be defined as collaborative control strategies, which can be used in human manipulation tasks to improve or assist by iso or anisotropically regulating motion. Motion regulation is achieved by attaching tools to a robotic arm, which is primarily controlled by a human user, under teleoperation control. Throughout operation the robot controller monitors tool motion and analyzes it with respect to known restricted regions. The AC controller then attenuates or nullifies any user command, which will cause the manipulator to digress from a plan. Referring to survey paper of Bowyer et al [13] regarding their implementation, they can be divided in several groups, such as:

- **Impedance device control**, where the impedance constraint which, when active, applies force to the user which will nullify the motion that violates the constraint;
- **Admittance device control**, where the admittance constraint allows the component of motion which does not violate the constraint;
- **Attractive**, which encourage movement to the permitted region once the restricted region has been breached. Usually used with guidance AC, to encourage movement along the path (see Figure 2.13 - a) ;
- **Repulsive**, which is active when the tool is near the restricted space. Usually used with regional AC, to filter out components of motion trying to breach the region (see Figure 2.13 - b);
- **Regional**, which prevents the tool from entering the defined region (see Figure 2.13 - c);
- **Guidance**, which encourages the tool to move in a specific path and conversely, try to nullify when breaching the boundaries of the guidance path (see Figure 2.13 - d);
- **Unilateral**, which acts only on one side. It will prevent tool motion into the restricted space. Conventionally, regional constraint would be constructed from unilateral surfaces (see Figure 2.13 - e);
- **Bilateral**, which acts on both sides. It will enforce tool motion along the guidance path (see Figure 2.13 - f). Conventionally, guidance constraint would be constructed from bilat

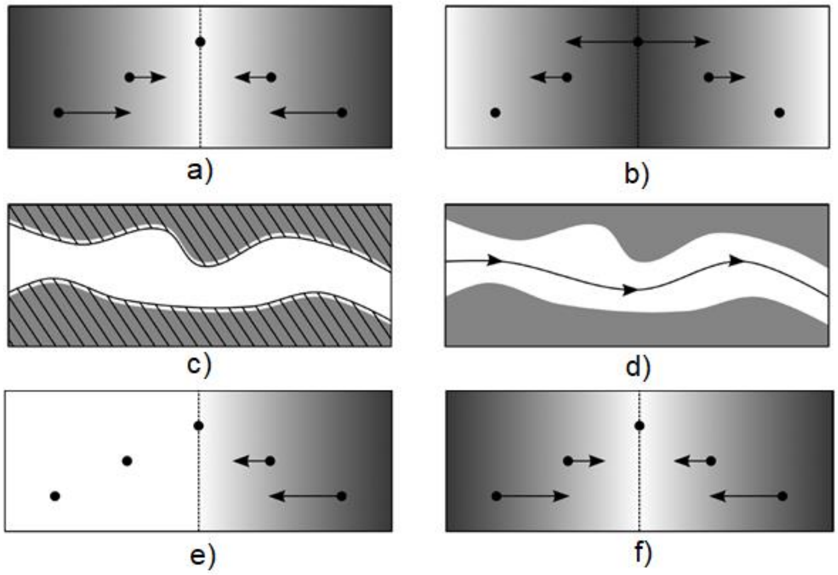


Figure 2.13 Examples of constraints. a) Attractive; b) Repulsive; c) Regional; d) Guidance; e) Unilateral; f) Bilateral. Permitted regions are shown in lighter colors and restricted regions are shown in darker colors. [13]

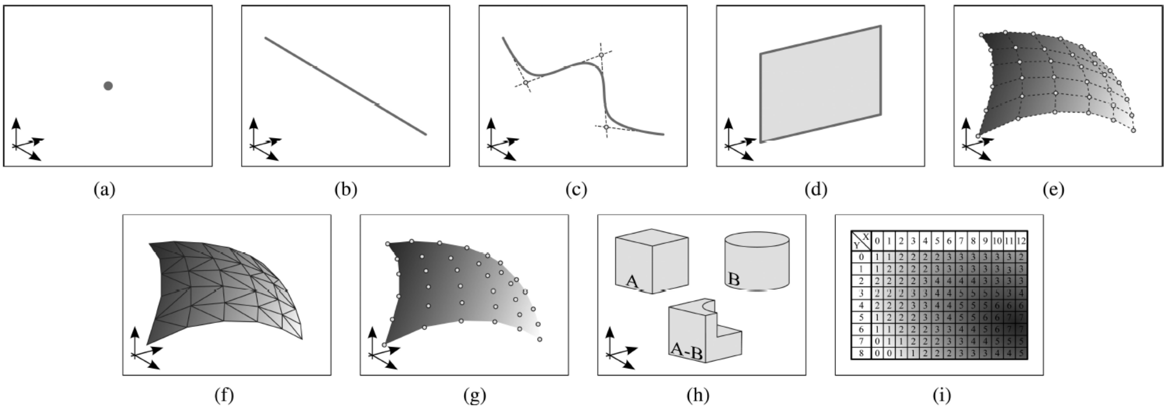
The final and conceptually most important classification of active constraints is:

- **Static AC**, where the environment is considered as stable enough that no change of the AC is needed in time.
- **Dynamic AC**, where the AC is a moving and deforming geometric figure as result of the environmental changes.

The implementation and use of the aforementioned constraints depends on how they are defined. Their definition needs to be computationally feasible, but they need to also be implemented correctly in order to describe geometrically regions that may be mathematically complex to define. Based on the type of definition, they can be:

- **Point constraints**, are the most simple type with one simple 3-D coordinate, used for tool positioning (see Figure 2.14 (a)) .
- **Linear constraints**, are formed from vectors within the task space. Used for straightforward guidance constraints, for direct approach from point to point (see Figure 2.14 (b)).
- **Parametric curve constraint**, have a complexity range from sinusoidal functions to splines. They are geometrically flexible and can be used to describe a wide range of complex tool paths (see Figure 2.14 (c)).

- **Hyperplanar constraints**, are simple to implement and can be used to separate task space into subspaces, as well as regional and guidance constraints (see Figure 2.14 (d)).
- **Parametric surface constraints**, are complex to implement due to need for nonuniform rational B-splines (NURBS), but research [14] done has shown success (see Figure 2.14 (e)).
- **Polygonal mesh constraints**, are very complex to implement, construct, evaluate and store. If the surface is extracted from "real-world" surfaces, they are very useful (see Figure 2.14 (f)).
- **Point cloud constraints**, represent a sample set of Cartesian points on the surface of a geometry. Common in practice, and usually produced from 3-D scanners, range cameras and fiducial tracking systems. Very important is that they are very simple to implement in real time, and in literature has found use in static and dynamic active constraints (see Figure 2.14 (g))


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

(f) (g) (h) (i)

Y \ X	0	1	2	3	4	5	6	7	8	9	10	11	12
0	1	1	2	2	2	2	3	3	3	3	3	3	2
1	1	2	2	2	2	3	3	3	3	3	3	3	3
2	2	2	2	2	3	4	4	4	4	3	3	3	3
3	2	2	2	3	3	4	4	5	5	5	5	3	4
4	2	2	2	3	3	4	4	5	5	5	6	6	6
5	1	2	2	2	3	3	4	5	5	6	6	7	7
6	1	1	2	2	2	3	3	4	4	5	6	7	7
7	0	1	1	2	2	2	3	3	4	4	5	5	5
8	0	0	1	1	2	2	2	3	3	3	4	4	5

Figure 2.14 Example illustrations of the constraint representations described within the literature. (a) Point; (b) Linear; (c) Parametric curve; (d) Planar; (e) Parametric surface; (f) Polygonal mesh; (g) Point cloud; (h) Volumetric primitive; (i) Explicitly described

Implementation of AC in a real control system is directly connected to the control unit. Their definition affects the control of multiple motion parts of a robot. A block diagram of how they are implemented in a real system is shown in Figure 2.15.

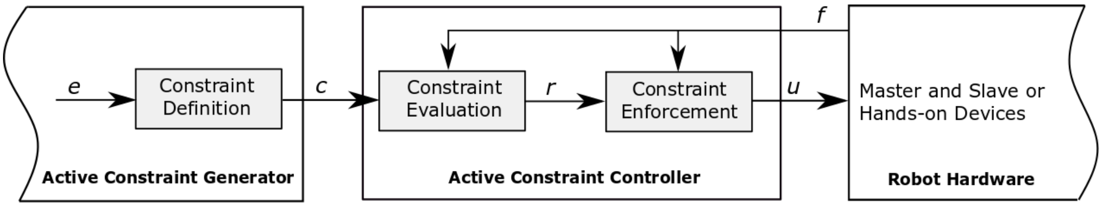

e
Constraint Definition
c
Active Constraint Generator
Constraint Evaluation
r
Constraint Enforcement
u
f
Active Constraint Controller
Master and Slave or Hands-on Devices
Robot Hardware

Figure 2.15 Generalized active constraint implementation framework

Having a static constraint requires that the operated region is not affected by the environment and will have little to no deformation or movement during surgery. However, such assumption very big and unlikely to be fulfilled, because it is difficult to provide such scenario in a real surgical procedure. In the literature, there has been extensive research on static AC but very little on dynamic AC. The reason for this is that dynamic AC are difficult to realize and computationally very expensive.

2.10. Dynamic active constraints – Overview of literature

Dynamic active constraints are a type of AC where the constraint geometry moves continuously, as a result of changes in the physical environment or the particular task being undertaken. An investigation of dynamic active constraints was carried out using simple proximity based constraints by Gibo *et al.* [15]. They constructed an experiment where a linear actuator was used to move a soft tissue phantom in 1 DOF, while a human user attempts to affect the model tissue (phantom) by a fixed amount using a teleoperated robot. They constructed a regional dynamic active constraint, which provided guidance through the tissue phantom at a predefined tissue depth, where the user is assisted while the tissue was moved periodically and randomly. Gibo *et al.* used two methods for computing the necessary position of the dynamic constraint; one based on the current tissue position and one based on its predicted position. They found that the two methods gave similar results and both were significant improvements on static constraints or unconstrainted operation.

Big portion of dynamic AC research has focused on beating heart surgery, Navkar et al. [16] considered the heart's left ventricle and generated multidimensional dynamic AC based on a proximity function. A dynamic guidance curve was generated in real time, applied to the end effector, between the inner walls of a beating heart. A haptic master device was implemented to render constraint force to the user. The results show that off-path error was reduced compared to cases with no guidance or with only visual guidance.

In the paper of Shademan et al. [17] an entire concept is proposed on autonomous robotic soft tissue surgery. They propose that a supervised autonomous soft tissue surgery in an open surgical setting is the next step for robot-assisted surgeries. In this method, the key elements to achieving this is using plenoptic three-dimensional and near-infrared fluorescent (NIRF) imaging system and autonomous algorithm for the surgical procedure of suturing. One of the biggest advantages a surgeon can have to a robot is the awareness of environmental changes and following them. The proposed system integrates NIRF and 3D plenoptic vision, force sensing, nano-scale positioning and actuated surgical tools. The tissue is marked at reference points with NIRF markers. Markers can be easily distinguished from the rest of the frame due to the specific light they emit.

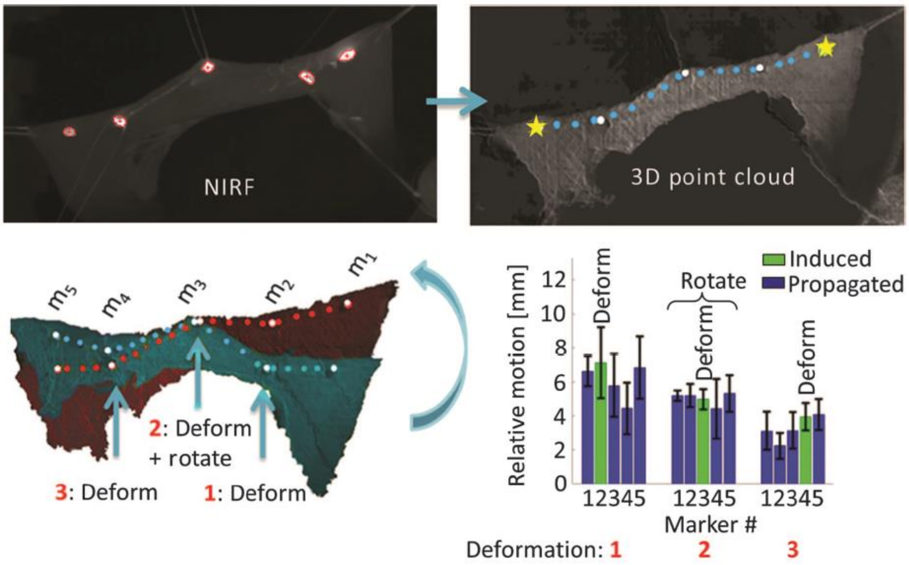


Figure 2.16 Top left - Marking of tissue with NIRf markers; Top right - Point clouds of initial (blue) and deformed (red) tissue; Bottom left - 3D point cloud before (blue) and after (red) deformations; Bottom right - Representative average marker deformation [17]

Preoperative CT model is used to generate a model of the tissue, and an offline registration of the 2D NIR image and 3D point cloud data. To this point cloud, initial NIRF markers are added with blue color. In online surgery, the current locations of these points are applied with the color red. The approach is motivated by Finite

Element Method, by separating the distances between two markers as finite rigid elements.

This method does not provide complete knowledge of the tissue deformation, but only around reference points which are of interest for a particular application such as suturing. This approach is trying to remove the surgeon from the actual procedure, and only have one in preoperative planning and as supervisor. However, the results show improvements with respect to conventional robot-assisted surgery (RAS) at certain aspects such as reduced mistakes and suture spacing performance while it falls back behind the conventional open surgery, laparoscopy and RAS in most of the other aspects.

2.11. Soft tissue deformation tracking

Dynamic AC are directly connected to the physical geometry of the organ. Thus, if the same organ is translating, rotating or deforming, the dynamic AC must always be in line with these changes. Estimating these changes in such environment is a very difficult task and is yet to be overcome. Environmental effects can be fast changing and easily cause malfunctions of tracking algorithms.

In the work of P. Mountney et al. [18] a framework is proposed which incorporates an online learning algorithm with feature tracking method that is suitable for *in vivo* applications. The problem of feature tracking is formalized as a classification problem where the classifier is trained with unlabeled data and adaptive updates during the tracking process. This approach does not assume about the type of image transformations or visual characteristics, which makes it suitable for dealing with nonlinear tissue deformations. The claimed strength of the algorithm when dealing with drift and occlusions, as well as tissue deformation is demonstrated in the experimental results done on simulated, porcine and in vivo data (*Figure 2.17*). Compared to three different techniques (SIFT, LK, mean-shift) it achieves bigger and very stable sensitivity to deformations during long term datasets.

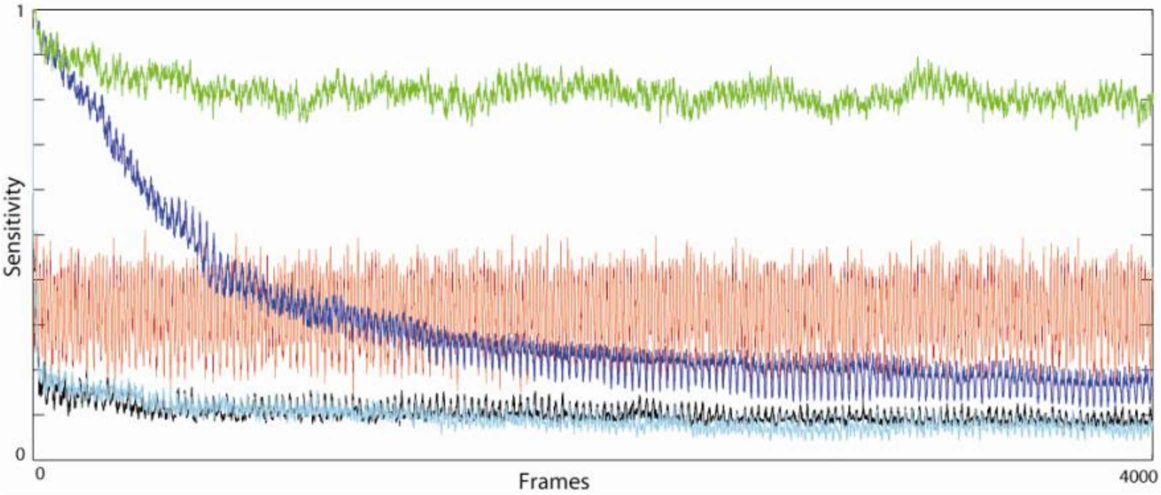


Figure 2.17 Relative performance values for the five different tracking techniques compared; green – the proposed tracker, red – SIFT, dark blue – Lucas Kanade, black – mean-shift 1 and light blue – mean-shift 2 [18]

In the work of D. Stoyanov and G. Yang [19],a framework is presented which uses a 2D video stream, instead of the usual 3D. Based on geometric surface representation, surface deformation is inferred from a reliable set of tracked salient feature points, which may be obtained using any reliable feature-based approach. The calculation time, however, is largely affected by the number of nodes in the mesh which is tracked. This mesh acts as an active constraint to be tracked and transformed.

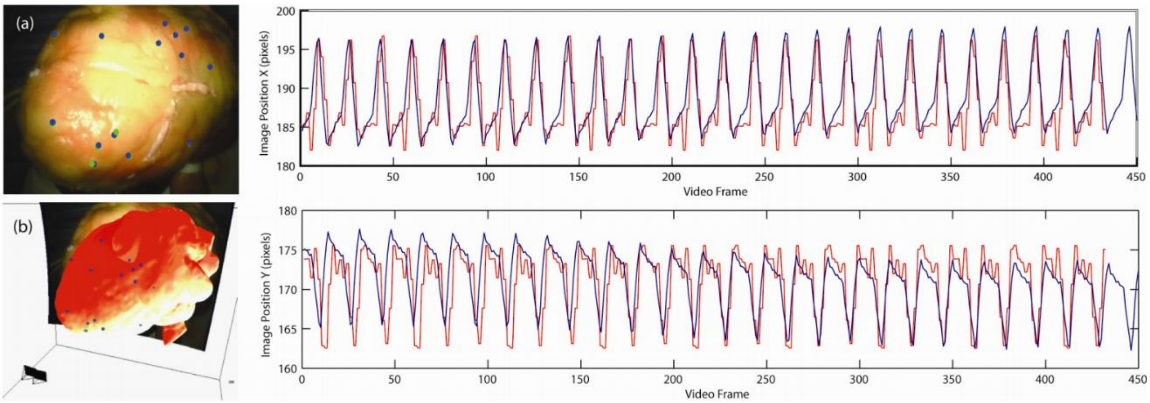


Figure 2.18 (a) Laparoscopic image of the phantom heart model with CT fiducials rendered onto the image to align with the observed points; (b) 3D rendition of the phantom model and fiducials within the camera’s coordinate system; (c-d) trajectory motion of a fiducial recovered using the proposed surface tracking approach shown in blue and compared to the data obtained from the CT ground truth shown in red. [19]

This proposed method achieves good repeatability and robustness in environments not largely affected by scene changes and tool interference.

A probabilistic framework to track affine-invariant anisotropic regions has been developed by Giannarou et al [20]. where a recovery strategy from potential tracking failure has been approached using spatial context and region similarity information to update an Extended Kalman Filter tracking framework.

Puerto-Souza and Mariottini in their work [21] introduced Hierarchical Multi-Affine (HMA) algorithm to map features between two endoscopic images allowing to recover features that were lost after a complete occlusion or sudden camera motions. It is a method which improves over existing feature-matching methods because of the larger number of image correspondences, increased speed and higher accuracy and robustness. In the provided test results, HMA outperforms the existing methods in terms of speed, accuracy and robustness.

A fast and adaptive algorithm for tracking non-rigid objects is proposed in the paper by Duffner et al. [22], where the unconstrainted problem of tracking a non-rigid, moving and deforming object is addressed. Their proposed method generalizes the unseen and new appearances of soft tissues within a video sequence and avoids drift, by applying an adaptive approach which is a combination of a detector using pixel-based descriptors and a probabilistic segmentation framework. The pixel-based detector is developed by using a Hough voting scheme. This method shows great computational speed which allows for real-time application in surgical robots.

An approach which directly addresses the problem of 3D deformation tracking in Minimally invasive surgery (MIS) is provided in the work of V. Penza et al. [23] which is a solution for safety volume tracking. This framework provides an approach to minimize the risk of intraoperative bleeding during abdominal MIS. Following on the published work of Penza et al. [24], long-term tissue tracking and dense tissue 3D reconstruction method, the 3D information obtained from this reconstruction is used to identify a Safety Volume (SV) fitted around the area it aims to protect. Any time an instrument approaches the SV, the surgeon is warned through graphical representation of the distance between the instruments and the reconstructed surface. This proposed method was realized and implemented in a dVRK system and was tested and validated under realistic Robotic MIS (RMIS) conditions.

In addition to the aforementioned works of Penza, another published work by Penza et al. [25] is done concerning long-term Safety Area (SA) tracking. This framework combines optical flow algorithm with a tracking-by-detection approach in order to be robust against failure. A Model Update Strategy (MUpS) is additionally implemented to improve the SA re-detection after failures, taking into account changes of appearance of the SA model through time. The presented method was tested and verified in order to assess it's capability of maintaining high tracking performance for extended periods of time (length of 5 min, containing different types of occlusions). Results show high precision and recall values, 0.85 and 0.6 respectively. These results show great promise, however the computational time at each cycle is at least 1.6 s up to 8 s. This does not allow real time implementation due to possible latency and blockage of motion commands. The results concerning method effectiveness are shown in the following *Table 1*.

Table 1 F-measure values (without/with MUpS) for three different overall threshold (low = 0.2, medium = 0.5, high =0.8) and Recovery Time [# frames] (without/with MUpS) [25]

	In-vivo			*ex-vivo*
	EV1	**EV2**	**EV3**	**IV1**
Low	0.93/0.96	0.44/0.96	0.97/0.97	0.34/0.60
Medium	0.93/0.95	0.44/0.96	0.90/0.93	0.34/0.60
High	0.80/0.71	0.30/0.45	0.20/0.38	0.22/0.32
Rec. time	0.84/0.50	37.50/0.88	7.00/2.00	16.00/8.04

2.12. SMARTsurg project

The SMARTsurg project, funded by the European Commission's Directorate-General for Research and Innovation (DG RTD) under its Horizon 2020 Research and innovation programme (H2020), is a collaboration between several European research institutions. The main vision of the SMARTsurg project is to enable complex minimally invasive surgical operations by developing a novel robotic platform for assisting the surgeon in such tasks. Advanced features will be developed and integrated into the proposed platform including:

- Wearable surgical system to provide natural usability and high dexterity to allow the undertaking of more complex surgical procedures and to reduce the surgeon's cognitive load.
- Anthropomorphic multi-fingered surgical instrument controlled by the anthropomorphic wearable system, enabling user-centered design and modifications by means of additive manufacturing.
- Software embedded visual and force augmentation for increased safety and dependability.
- Functionalities enhancing the system's cognition abilities and dependability, *such as dynamic active constraints construction and enforcement*, as well as user intention detection

This thesis, as a part of the SMARTsurg project, was done under the guidance of prof. Giancarlo Ferrigno in **NE**uroengineering and medica**AI** Robotics **Lab** (NEARLab)

2.13. Aims of the work

In practice, having a well posed system for tissue deformation tracking and dynamical active constraint transformation has not been successfully implemented. There are barriers for adoption for real-time implementation of the aforementioned system, such as providing reliable mathematical and programming basis required to capture the fast changes in a very mixed and dynamic environment that is a robot-assisted surgery. Furthermore, providing a system that is able to correct itself and learn at each step in order to recover from the changes in tissue structure, lighting, projection, and tool interference in the scope of view. It is also needed to reduce the complexity of the solution in order to provide calculations fast enough not to lose any information that is provided at each frame received. For this reason, the main purpose of this thesis is development of an estimation technique which will account for all of these occurrences, particularly the deformation of the tissue that occurs from tissue manipulation by the surgeon and due to natural movement of organs during surgery due to breathing, heartbeat etc.

Tracking of certain characteristic features of the tissue of interest is a key process of the behavior that is of interest to analyze. By monitoring the changes of these features, the amount of geometrical change of the tissue can be assessed. If there is an excess of noise, parameters can be changed in order to stabilize the process.

A deep understanding of the correlation between the motion of characteristic features and the geometrical change is fundamental to know how to estimate the geometrical transformation of the organ in a mathematical manner, and to know how to control and correct the parameters so that the entire process is stable and reliable. While the method that will be proposed in this work is completely working on 2-D input images, the results will be projected into 3-D space point cloud so the data can be effectively implemented in the GUI designed by the collaborators from the SMARTsurg project (Chapter 2.12)

Experimental campaigns have been carried out on datasets provided from real life surgeries done by surgeons using a medical robot. Each provided dataset was specifically picked out to encapsulate all possible disturbances that may happen in the operating room in a real surgery. Additional tests were done using a kidney phantom in order to cover particular behaviors in a controlled test environment.

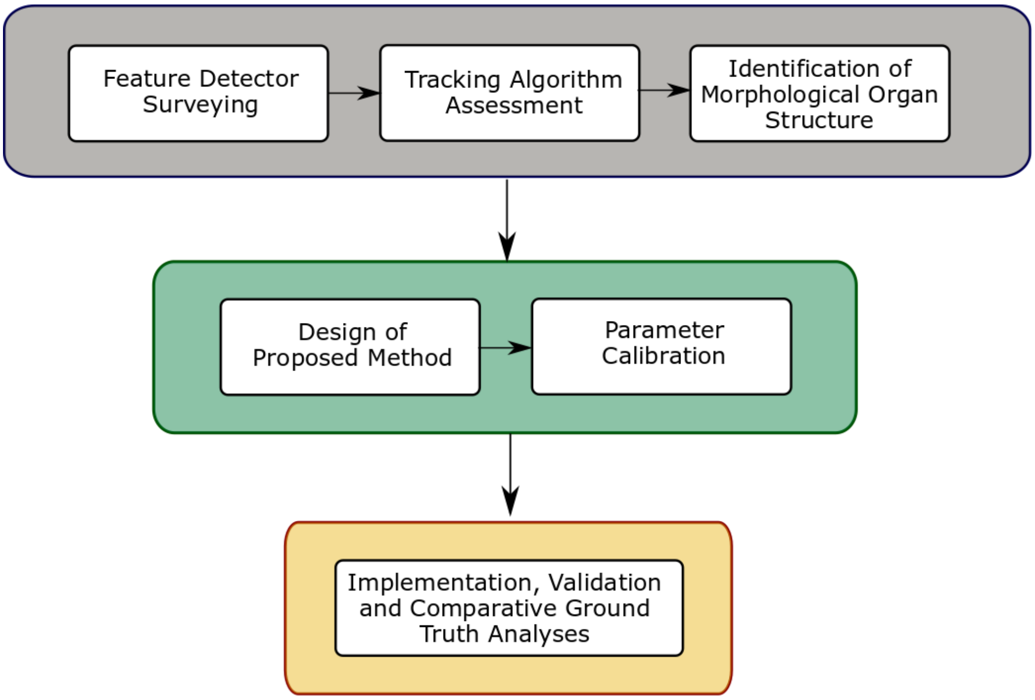

Feature Detector Surveying
Tracking Algorithm Assessment
Identification of Morphological Organ Structure
Design of Proposed Method
Parameter Calibration
Implementation, Validation and Comparative Ground Truth Analyses

Figure 2.19 Thesis work logical scheme

CHAPTER 3
Methodological background

In this chapter, a brief yet detailed analysis can be found of the pre-existing processing algorithms used in this thesis work is discussed. Because of the large number and diversity of these algorithms, they have been separated into groups depending on their function and in which stage of the full process they are used.

3.1. Color Spaces

3.1.1. Red – Green – Blue (RGB) and Grayscale

In the world of graphics there are additive and subtractive colors. The primary additive colors are red, green and blue, hence RGB. By combining them, the entire spectrum of visible colors can be recreated.

The color of each pixels is presented by three values, each corresponding to the intensity of each primary color. The value of each pixel ranges from 0 to 255.

The triples corresponding to black (i.e. absence of light) and white (i.e. absolute presence of light)

$$RGB_{black} = (0, 0, 0) \qquad RGB_{white} = (255, 255, 255)$$

respectively. Also, the triples corresponding to the primary colors are:

$$R = (255, 0, 0) \qquad G = (0, 255, 0) \qquad B = (0, 0, 255)$$

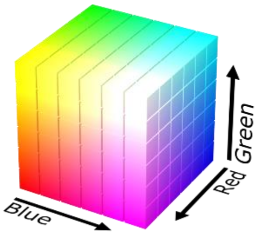


Figure 3.1 RGB cube of color distribution

An RGB image is a m x n x 3 matrix, where m and n are the width and height of the image respectively, and the third dimension refers to having three matrices with size m x n for each R, G and B channel.

In order to transform an RGB image from three dimensions to two, a calculation needs to be done in order to keep as much as information possible. The most used approached is ‘”grayscale”. To convert a color in grayscale, we compute:

$$GS_{x,y} = \frac{R_{x,y} + G_{x,y} + B_{x,y}}{3}$$

Eq. 3.1

This is a very basic and simple step and can be done for any other color space but due to different definitions of values in each channel, it only makes sense to be used for RGB.

3.1.2. Hue – Saturation – Value (HSV)

One of the alternative representations to the RGB color model is HSV which stands for Hue-Saturation-Value. It was designed in order to align colors in a way which is close to the way human vision perceives color-making attributes.

For this reason, each component has a different role in this color space:

- Hue is the color portion of the model, and it can be expressed as a number from 0° to 360°.
- Saturation describes the amount of gray in a particular color, and ranges from 0 % to 100 %. Reducing this component produces a faded effect.

- Value (or brightness) works in conjunction with saturation and describes the brightness or intensity of the color, that it ranges from 0 % to 100 %. Here, 0 % stands for absolute black and 100 % is maximum brightness.

These values are mathematically correlated to RGB because most visual multimedia is encoded in RGB. This connection is important, and it will be further discussed.

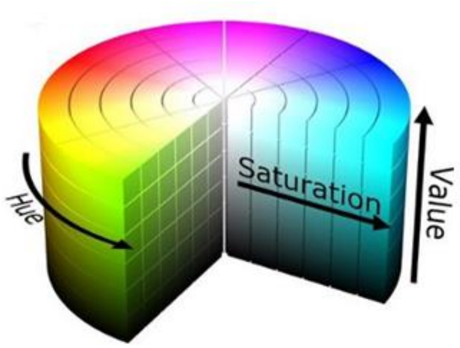


Figure 3.2 HSV cylinder of color distribution

This color space gives the opportunity to easily process colors based on shades of color, which would show very useful throughout this thesis, especially the Saturation channel.

3.1.3. Luma – Channel Blue – Channel Red (YCbCr)

Another color space focused on how the human eye percepts colors is YCbCr which stands for:

- Y for Luma component;
- Cb for blue component;
- Cr for red component.

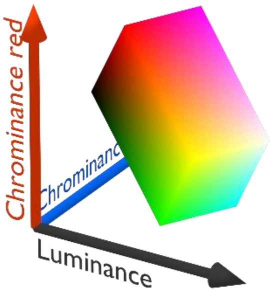


Figure 3.3 YCbCr cube of color distribution

The human eye is most sensitive to the Y component and during the conversion or transmission this channel is most accurate. Cb and Cr are less important because the human eye does not react as sensitively as to Y.

However, this gives a good range of opportunities especially when working with Histogram thresholding of organs and tissues. The Cb channel has a well posed distinction between red-rose shades and the rest of the spectrum.

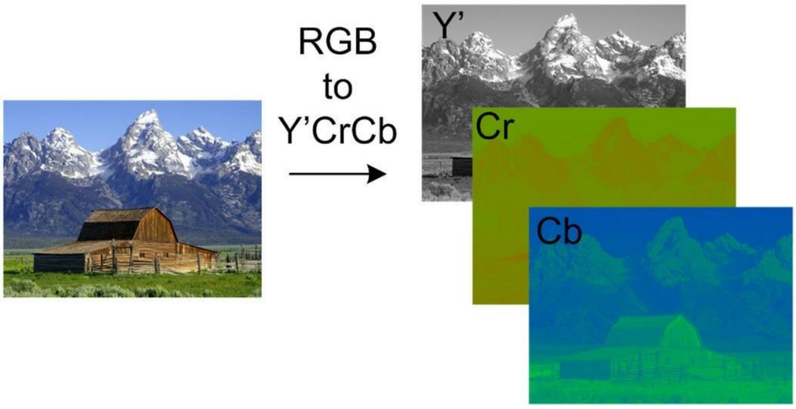


Figure 3.4 YCbCr channel decomposition

3.1.4. CIELab

Lab, or CIELab, is the most complex and robust color space for quantitative comparisons. Lab makes assumptions about the colors in the environment based on the specific lighting conditions. In this color space, each pixel has three different values in order to represent a color. The description of each of the three values is what Lab stands for:

- L for Luminance (black to white) ranging between 0 % and 100 %,
- a for channel green-to-red ranging from -100 to 100,
- b for channel blue-to-yellow ranging from -100 to 100.

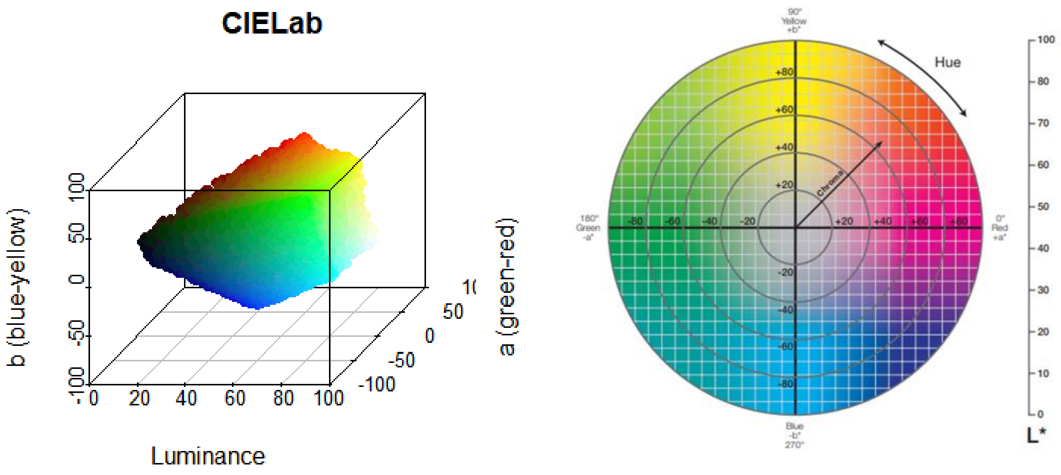


Figure 3.5 CIELab color distribution

Lab color space is defined as a perceptually uniform color space, meaning that the sets of colors separated by the same distance in Lab space will seem about equally different in sense of color hue or shades.

3.2. Image Preprocessing

In image processing procedures, the most important step is preprocessing, as it prepares the image for further analysis. Once the image is correctly preprocessed to improve contrast, decrease noise effect, equalize intensity etc. for a particular application, it will give much more information than an image that has not been prepared for analysis in the same experimental setup.

Therefore, the principles presented in this part are the ones used in preprocessing the video frames before analyzing them.

3.2.1. Gamma correction

Gamma correction is a nonlinear operation that was primarily developed in order to correct the luminance in video and images [26]. It is applied directly to an RGB image to each pixel individually. When applied, the intensity of the entire image will be modified depending on the choice of the γ coefficient. In its simplest form it is defined by the expression:

$$V_{out} = AV_{in}^{\gamma}$$

Eq. 3.2

Where V_{out} is the non-negative real output , V_{in} is the non-negative real input value which is raised to the power of γ and multiplied by the constant A. Since Gamma Correction is applied to each pixel individually, V_{in} is the pixel color value of the original image while V_{out} is the pixel color value of the image after gamma correction is applied. A is commonly 1, and the inputs and outputs are normalized to fit a range of values between 0 and 1.

There are three cases:

- γ < 1 means that the correction is doing gamma compression, which results with a darkened image
- γ = 1 will results with the exact same image on the output as the input
- γ > 1 means that the correction is doing gamma expansion, which results with a more luminated image.

This method was primarily used to match the colors of the image or video taken from a camera with the intensity which the human eye is supposed to see it. There is a nonlinear relationship between these two occurrences.

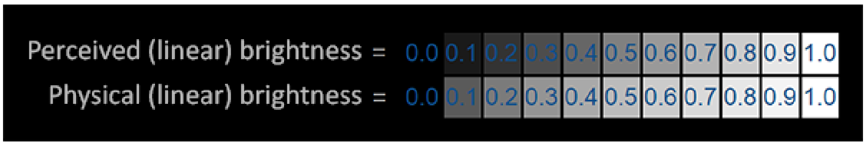

Perceived (linear) brightness = 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
Physical (linear) brightness = 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

Figure 3.6 Perceived vs Physical brightness

The power-law takes care of this nonlinear relationship with the aforementioned equation which takes the form of the curve specified in the following graph (*Figure 3.7*).

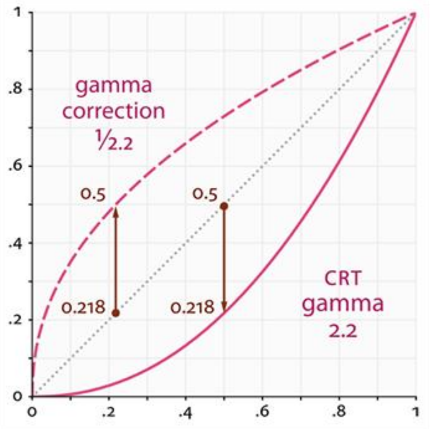


Figure 3.7 Gamma correction non-linear dependencies

This operation is performed pixel-wise and is not an approach which introduces correlation between pixels, unlike other procedures which involve moving windows.

Even though it was primarily developed for the use in monitors, it has many applications in other fields. One of them, which is used in this thesis, is the capability to compress visual characteristics of certain areas of an image. By using gamma correction, regions with no explicit boundaries have a more distinguishable visual outlook than the rest of the image. In a way, it is a step before approaching a background-foreground segmentation. Thanks to it, it is possible to acquire an image that resembles a surrounding which encapsulates a salient object.

3.2.2. Gaussian blur

One of the most common and efficient ways to blur an image is Gaussian blur or Gaussian smoothing. The Gaussian smoothing operator is a 2-D convolution operator that is used to 'blur' images and remove noise. In this sense it is similar to the mean filter, but it uses a different kernel that represents the shape of a Gaussian curve. The kernel takes advantage of Gaussian distribution to create a 2-D moving window of weights which is then applied through the image. It can be applied only on

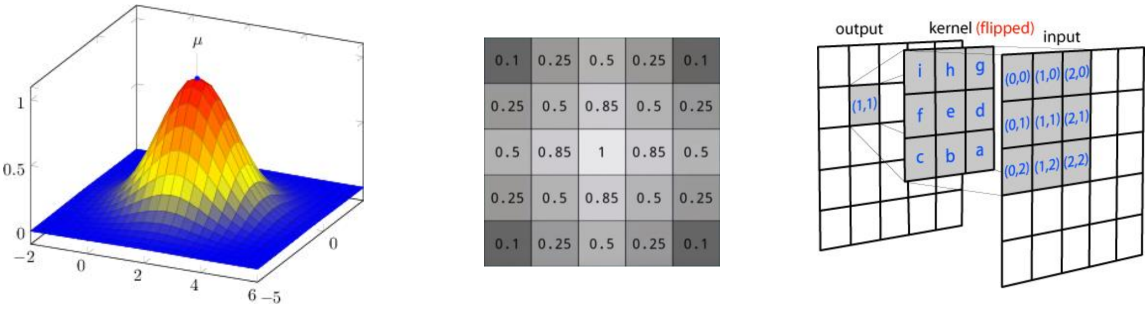
2-D images, such as grayscale, or on 3-D images by applying it separately to each channel (i.e. separately to R, G and B).

The mathematical formula of a Gaussian function in two dimensions is:

$$G(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Eq. 3.3

where x is distance from the horizontal axis, y from the vertical axis and σ is the standard deviation of a Gaussian distribution. A kernel is then created based on this distribution, in two dimensions. Because Gaussian distribution is asymptotic, a kernel of infinite size will be needed to cover the entire distribution. A finite size needs to be chosen which is of size n by n, where n is an odd number. An odd number is needed in order to make sure there is a central element, at position $(\frac{n-1}{2},\frac{n-1}{2})$. The element at this position is the core element and will be in line with the processed pixel. The kernel is moved through the image in the same way a moving window is done. In *Figure 3.8*, the first image shows a 2D gaussian distribution, where it can be seen that the highest value is in the central point. The kernel is defined by this distribution but in a discrete manner, as shown in third image in *Figure 3.8*. The value of 1 is positioned in the center, and by following the Gaussian distribution lower values are set to the other pixels in waves of concentric circles. This approach gives the maximum priority to the central element. By the principle of convolution, the kernel is taken through the image. When multiplied with a set of n by n pixel sub-matrix from the original image, only the central pixel will change the value. Each element pixel sub-matrix from the original image will be multiplied with the corresponding element of the kernel. After this, the sum of all multiplication results is calculated and then divided by the sum of all weights in the kernel. This value will be the new value of the central pixel, as shown in the last image of *Figure 3.8*.



0.1	0.25	0.5	0.25	0.1
0.25	0.5	0.85	0.5	0.25
0.5	0.85	1	0.85	0.5
0.25	0.5	0.85	0.5	0.25
0.1	0.25	0.5	0.25	0.1

Figure 3.8 Gaussian blur kernel definition

A good survey can be found in the work of Gedraite et al. [27] on the effects of Gaussian Blur in image filtering and image segmentation. One of the setbacks of this procedure, especially for the use in this thesis, is that blurring an image reduces the clear edges throughout the image

3.2.3. Contrast – Limited Adaptive Histogram Equalization (CLAHE)

A common image processing technique to improve contrast is Adaptive histogram equalization (AHE), and it differs from ordinary histogram equalization in the sense that it will compute several histograms, each corresponding to a separate distinct section of the image, used to redistribute the lightness values of the image. However, this principle tends to overamplify noise in relatively homogenous regions of an image. This is easily noticeable in a situation where a small histogram window is used, and that particular window is dominated by noise effect. The result will be a histogram based on noise.

A variant of AHE called contrast limited adaptive histogram equalization (CLAHE) prevents by limiting the amplification. Ordinary AHE tends to overamplify the contrast in near-constant regions of the image, since the histogram in such regions is highly concentrated.

CLAHE uses a windows in which the histogram is equalized, and is very important for this window to be larger than the features to be preserved. For the calculation of CLAHE, color space LAB is to be used. From the Lab image, a histogram with N bins is calculated only on the L (lightness) channel. that should not be more than 256. It limits the maximum contrast in its intensity transfer function, by implementing a clip limit.

Anything that exceeds this clip limit will be cut off from the top and redistributed at the bottom equally among all histogram bins (Figure 3.9). The value at which the histogram is clipped, the so-called clip limit,and depends on the normalization of the histogram and thereby on the size of the neighborhood region. Common values limit the resulting amplification to between 3 and 4 [28]

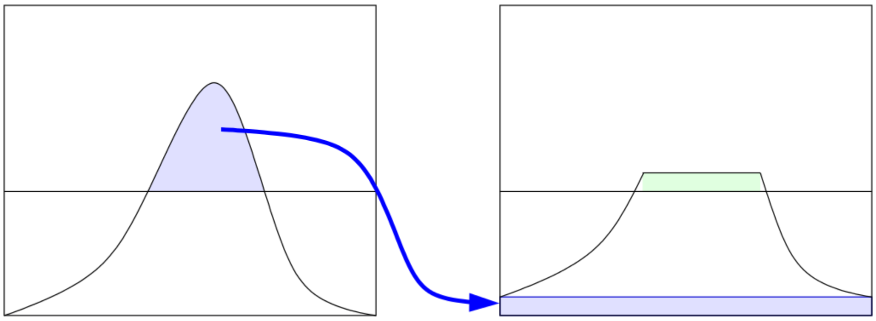


Figure 3.9 Redistribution of the part of a histogram above the clip limit to the lower bound

3.3. Features, Detectors and Descriptors

In image processing, a feature is a piece of information that includes relevant data, necessary for resolving computational tasks for image processing applications. In a sense, it is similar to pattern recognition but by having a very sophisticated collection of features. They are specific structures in an image such as points, edges or objects depending on the feature detection algorithm.

An interest point (key point or salient point) detector is an algorithm that chooses points from an image based on some criterion. Typically, an interest point is a local maximum of some function, such as a "cornerness" metric. The detector uses a sophisticated set of rules in order to decide whether a certain point qualifies to be a point of interest. Together, an interest point and its descriptor is usually called a local feature. Local features are used for many computer vision tasks, such as image registration, 3D reconstruction, object detection, and object recognition. There are many types of features, and each one of them requires a certain algorithm to be used as a detector. Generally, they are divided in the following groups:

- Corner detection:
  - FAST (Features from Accelerated Segmentation Test)
  - Harris Corner Detector
  - Shi and Tomasi
  - SIFT (Scale-Invariant Feature Transform)
- Blob detection:
  - SURF (Speeded Up Robust Features)
  - KAZE

- MSER (Maximally Stable Extremal Regions)

- General feature detection:
  - BRIEF (Binary Robust Independent Elementary Features)
  - ORB (A hybrid of Oriented FAST and Rotated BRIEF)

For the purpose of choosing the most appropriate feature detector and descriptor in this thesis, a brief survey, motivated by the work of Tareen et al. [29] was done on a video sequence of partial nephrectomy in order to experimentally see which feature detection algorithm will be most usable for this thesis. The final choice was made according to:

- Average number of features detected in the video sequence
- Average detection time per frame
- Dispersion of features

The results are shown on the *Table 2*.

Table 2 Comparative study of feature detectors

Name	**Average Time**	**Average Features Detected**
FAST	0.0073 s	224.8246
Min-Eigen	0.1283 s	693.6864
Harris	0.1037 s	178.4189
BRISK	0.2098 s	390.3268
SURF	0.0439 s	178.2127
KAZE	0.1070 s	1274.1
MSER	0.1422 s	258.0570
ORB	0.0861 s	2072

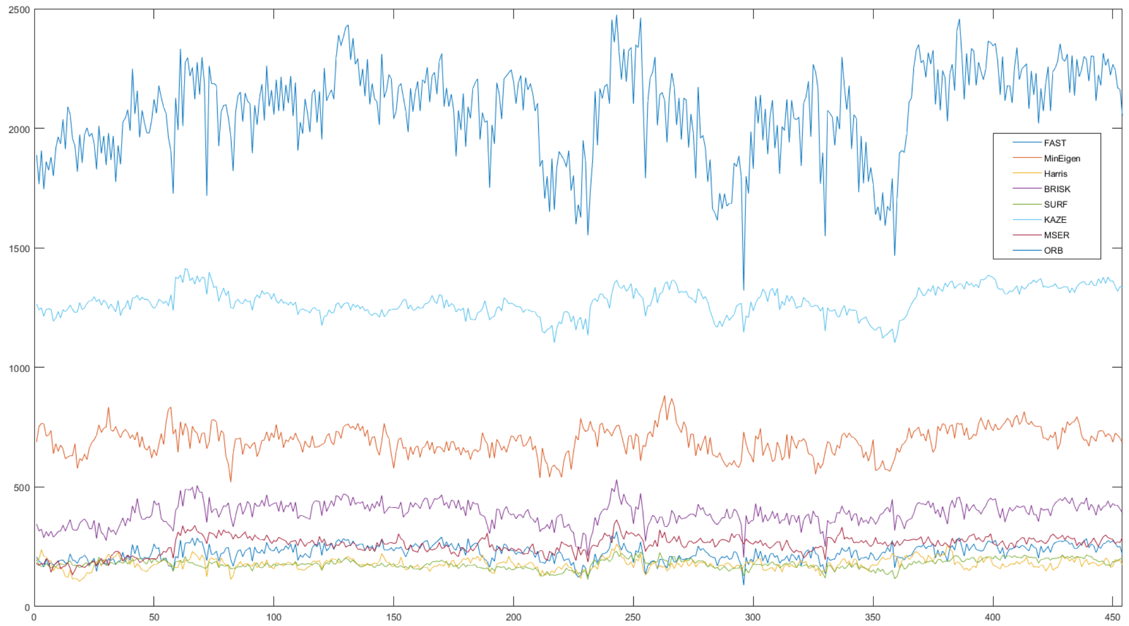


Figure 3.10 Comparison of different feature detectors applied to the same video sequence with length of 405 frames

The conclusions are:

1. The fastest algorithm is FAST but gives only third smallest number of features.

2. The algorithm with most average features per frame is ORB and has third fastest execution time. The only problem is that without filtering the frame, it does not cover the entire frame.

3. KAZE gives a fairly good amount of features, and its execution time is in the middle. However, it is crucial to emphasize covers the entire frame with trackable features and has a very low standard deviation compared to the average number of detected features (see *Figure 3.10 Comparison of different feature detectors applied to the same video sequence with length of 405 frames*).

4. SURF is the most cost-efficient detector of blob-like structures.

5. ORB is the most cost-efficient detector of points of interest (based on adaptive FAST).

6. What can not be seen from the table is the dispersion of trackers. While SURF and ORB seem to be the most cost-efficient, their performance in registering features more sparsely through the image was below average. This lead to choosing KAZE as the used feature detector because it was able to detect features throughout the image at a very stable and predictable performance. Since the region of interest may be at any segment of the image, choosing KAZE was the obvious choice. From here on each mention of feature related processes will imply that KAZE were used.

In the core of most feature detectors and descriptors lies the analysis on **image gradient**. Image gradient is the directional change of intensity or color in an image. Mathematically, it is a function with two variables at each point of an image, as a 2D vector with components given by the **derivatives** in the **horizontal** and **vertical** directions. At each point, this vector points in the direction of the largest intensity, and the length corresponds to the rate of change in that direction.

The novelty in KAZE features [30] is the computation of nonlinear scale space in 2D, to detect features of interest that exhibit a maxima of scale-normalized determinant of the Hessian response through the nonlinear scale space.

For detecting **keypoints** of interest, the **response** of a normalized determinant of the Hessian is computed at multiple scale levels. In the case of multiscale detection, differential operators are normalized with respect to the scale, because the amplitude of spatial derivatives decreases with scale:

$$L_{Hessian} = \sigma^2 \left(L_{xx}L_{yy} - L_{xy}^2\right)$$

Eq. 3.4

Where $(L_{xx},\ L_{yy})$ are the second order derivatives in horizontal and vertical directions respectively, and L_{xy} is the second order cross derivative, and σ is the scale level. The derivatives are approximated by using 3 x 3 Scharr filters of different derivative step sizes. Aside from the importance of the response $L_{Hessian}$ in order to determine if some point is a keypoint feature, it can also be used as a **good metric** for how good and how distinctively has that point been described by the descriptor.

To obtain rotation invariant descriptors, estimation of the dominant orientation in a local neighborhood centered at the keypoint location is needed. In a circular area with a radius of 6σ with a sampling step of size σ for each sample, first order derivatives are weighted with a Gaussian kernel centered at the interest point. The derivative responses are represented in vector space and the dominant orientation is found by summing the responses within a sliding circle segment.

The descriptor is built upon the M-SURF descriptor, but adopted to a nonlinear scale space framework.

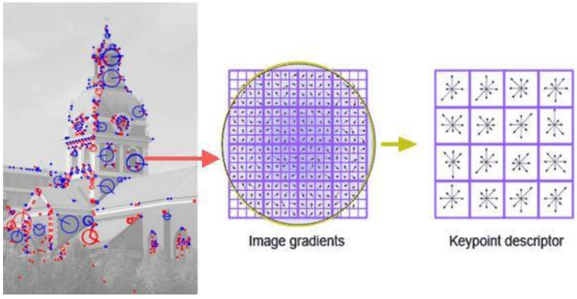


Figure 3.11 Structure of KAZE feature descriptor

From experimental results, it can be seen that KAZE is a step in-front of the other descriptors in its class in detector repeatability, as well as precision in nearest neighbor matching strategy by using any of the three possible diffusivity protocols (G1, G2 or G3)

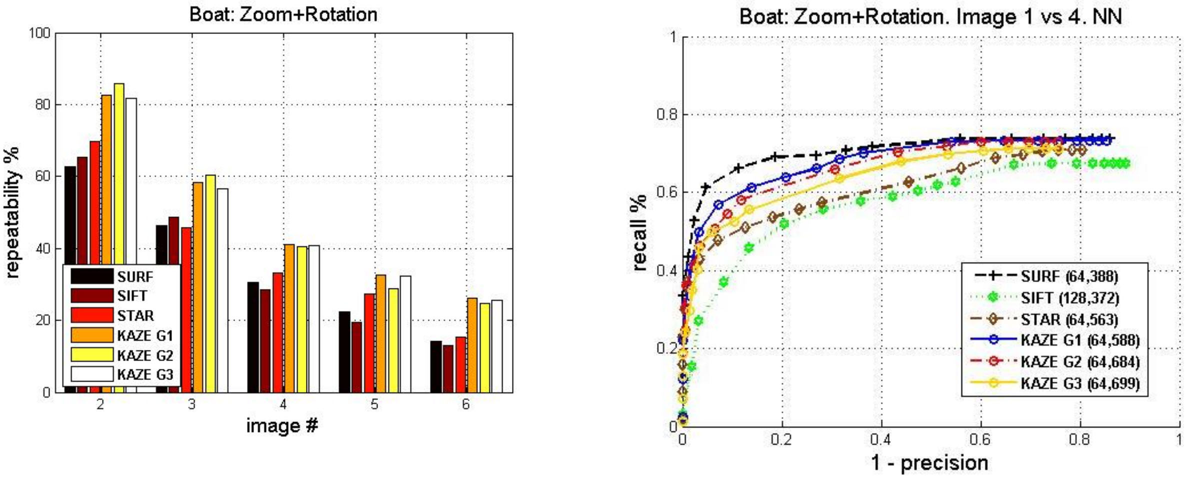


Figure 3.12 On the left, repeatability graph while zooming and rotation is featured in the testing set. On the right, precision and recall scores for the same test set.

The most important results regarding this thesis is KAZE's capability of great precision in matching deformable surfaces. Detection time is larger than most of the detectors but having a good descriptor to track is better than redetecting at every few frames.

3.4. Feature tracking

In the realm of computer vision, the Lucas-Kanade method [31] is perhaps the most widely used differential method for optical flow estimation. It works under the assumption that the flow is constant in a local neighborhood of the pixel in consideration and computes the basic optical flow its neighborhood by the least squares criterion. Another assumption is that the displacement is less than 1 pixel between two frames. It is specifically a local method and does not provide flow information of uniform regions of the image. Since motion tracking can be sparse or dense, this method falls under Sparse Optical Flow tracking.

A more popular name for this algorithm is KLT (Kanade-Lucas-Tomasi) but in that implementation there is a corner detection implemented by the Tomasi-Kanade feature extraction framework called GoodFeaturesToTrack. For this thesis, only Optical Flow Estimation by Lucas-Kanade will be used because the GoodFeaturesToTrack output did not provide any useful features to track.

Problem statement:

Two images in grayscale are provided to the algorithm, called I and J.

- I(x,y) is the grayscale value of a pixel from the image I at (x,y).
- Let $u = [u_x \; u_y]^T$ be a point of interest on the first image I.
- $d = [d_x \; d_y]^T$ is the image velocity at u, or the optical flow at u.
- The goal is to find v in J, where I(u) and J(v) are similar enough.

$$v = u + d = \begin{bmatrix} u_x + d_x \\ u_y + d_y \end{bmatrix}$$

Eq. 3.5

The residual function is two-dimensional least squares method covering the neighborhood of the pixel tracked. The leading cost function that should converge and lead to a solution is this residual function which is being minimized with respect to the two components of the optical flow $[\mathrm{d_x}\ \mathrm{d_y}]^\mathrm{T}$.

A big issue appears when object moves by more than one pixel. Since this method only can be applied to small movements, it should not be able to track that. However, Pyramidal Implementation takes care of this part.

The concept behind this is that the pixel is tracked at multiple levels. The pyramid representation, as presented in the work of Bouguet [32], is built recursively in which Level 0 is the original image, Level 1 is scaled down to one quarter of the original image (both width and height are scaled to half), Level 2 is one eighth and so on.

Let L be the pyramidal level, from here it is follows that

$$u^L = \frac{u}{2^L}$$

Eq. 3.6

And for the displacement (optical flow)

$$d = \sum_{L=0}^{L_m} 2^L d^L$$

Eq. 3.7

A simplified overall pyramid tracking algorithm is depicted on *Figure 3.13*.

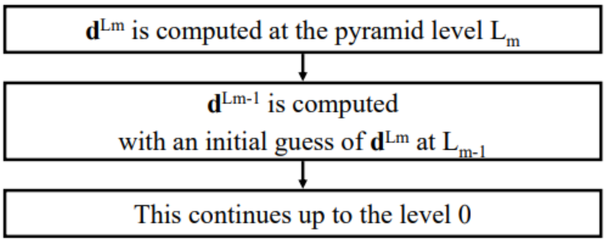

$\mathbf{d}^{\mathrm{Lm}}$ is computed at the pyramid level $\mathrm{L_m}$
$\mathbf{d}^{\mathrm{Lm-1}}$ is computed
with an initial guess of $\mathbf{d}^{\mathrm{Lm}}$ at $\mathrm{L_{m-1}}$
This continues up to the level 0

Figure 3.13 Simple overall pyramid tracking algorithm

With respect to the level L, it must be defined as:

$\bar{v} = [v_x \; v_y]^T = \; d^L$

$p = [p_x \; p]^T = \; u^L$

Eq. 3.8

The previously mentioned residual function being minimized with respect to d implemented for pyramidal representation will be:

$$\varepsilon(\bar{v}) = \; \varepsilon\left(v_x, v_y\right) = \sum_{x=p_x-w_x}^{p_x+w_x} \sum_{y=p_y-w_y}^{p_y+w_y} (A(x,y) - B\left(x + \; v_x, y + \; v_y\right))^2$$

Eq. 3.9

Where ε is the residual, $\left(d_x, d_y\right)$ is the displacement vector, $(w_x \,, \; w_y \,)$ are the dimensions of the integration window, $(p_x \,, \; p_y)$ is the point vector, (x,y) are the coordinates in the source image, A is the source image, and B is the destination image.

To find the optimum of $\left(d_x, d_y\right)$ the following differential equation must be solved:

$$\left.\frac{\partial\varepsilon(\bar{v})}{\partial\bar{v}}\right|_{\bar{v}= \; \bar{v}_{opt}} = [0 \;\; 0]$$

Eq. 3.10

This is solved iteratively by the first order Taylor expansion about the point $\bar{v} = [0 \; 0]$.

For a clearer representation of the previously discussed subject, the following flow chart gives a functional representation of how one tracking sample works.

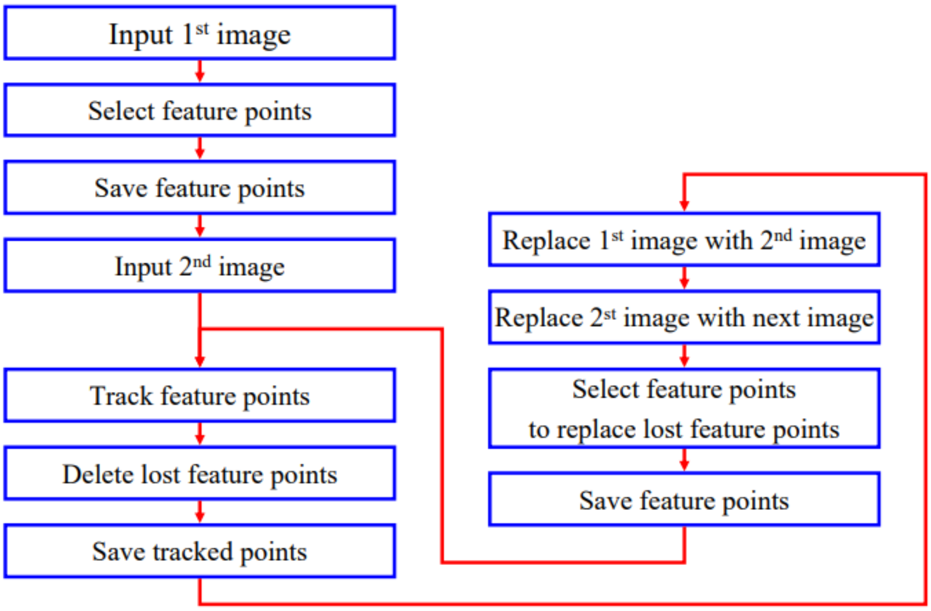


Figure 3.14 Flow chart of an example implementation of LK tracking

Optical flow estimation is a widely used concept with many different implementations which improve performance in their respective applications. It is completely implemented in OpenCV with a very stable performance. Many of the mentioned equation have parameters which must be user defined in order to achieve useable results since using default parameters only works good for stock images and videos.

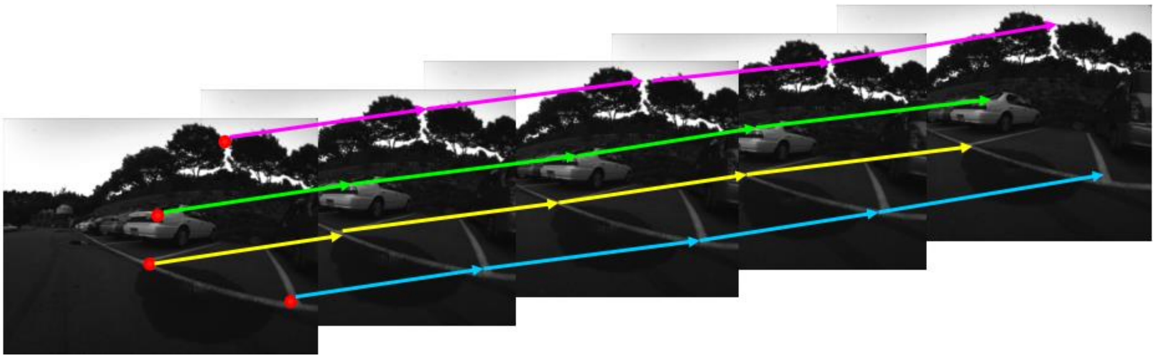


Figure 3.15 Graphical representation of tracked feature points with LK

3.5. Feature matching

In general there are two common approaches for matching features in Image Processing. They are Brute Force matching and FLANN-based matching. Brute force tries to find the best match between all the features of both images using a particular method such as Euclidean distance (L2 – norm), while Flann (Fast Library for Approximate Nearest Neighbors) looks for an approximate nearest neighbor. Flann can be much faster but only finds an approximation which the cost paid for gaining on speed. Flann is more commonly used for large sets of features (above 1000) which is not the case in this application.

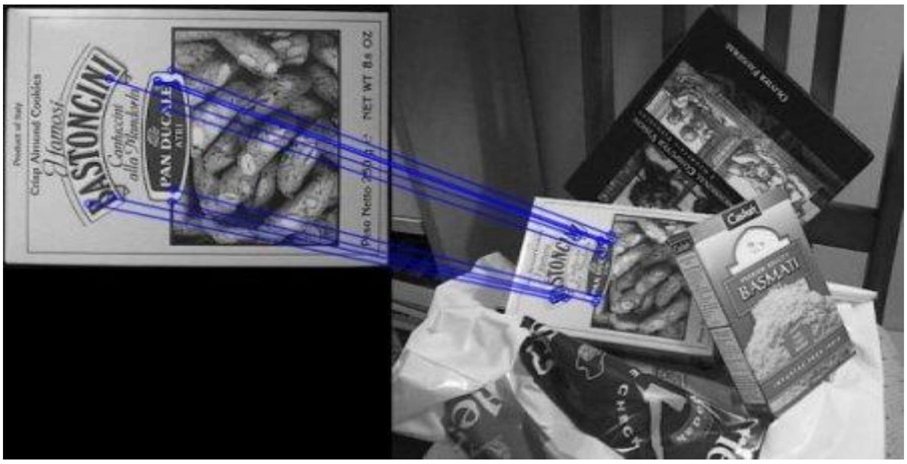


Figure 3.16 Feature matching by using Brute Force approach

Because feature matching in this thesis is a difficult task on its own, the method used will be Brute Force in order to find the best possible match. There are a couple of steps prior to matching which make sure a small number of very strong features is matched to reduce workload, which will be explained in the next sub-topic.

3.6. Types of transformations and transformation matrix estimation

3.6.1. Types of transformations

A linear transformation is a function which maps a vector space into another, in a form of a matrix. A mapping is a linear transformation if it preserves vector addition and scalar multiplication. To apply a linear transformation to a vector (x,y coordinates of one pixel point), it is necessary to multiply this vector by a matrix which represents the linear transform. The output will be a new vector with the transformed coordinates.

There are wo classes of linear transformations - projective and affine. Affine transformations are a special case of the projective transformations. Both of the transformations can be represented with the following matrix:

$$\begin{pmatrix} a_1 & a_2 & b_1 \\ a_3 & a_4 & b_2 \\ c_1 & c_2 & 1 \end{pmatrix}$$

Where:

- $\begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix}$ is a rotation matrix.
- $\begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$ is a translation vector.
- $(c_1 \quad c_2)$ is a projection vector. For affine transformations all elements are always equal to 0.

If *x* and *y* are the coordinates of a point, the transformation can be done by the simple multiplication:

$$\begin{pmatrix} a_1 & a_2 & b_1 \\ a_3 & a_4 & b_2 \\ c_1 & c_2 & 1 \end{pmatrix} \times \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix}$$

Eq. 3.11

Here, *x'* and *y'* are the coordinates of the transformed point.

The only difference between these two transformations is in the last row of the transformation matrix. For affine transformations, the first two elements of this row are zeros. This leads to different properties of the two operations:

- The projective transformation does not preserve parallelism, length, and angle. But it still preserves collinearity and incidence.
- Since the affine transformation is a special case of the projective transformation, it has the same properties. However, unlike projective transformation, it preserves parallelism.

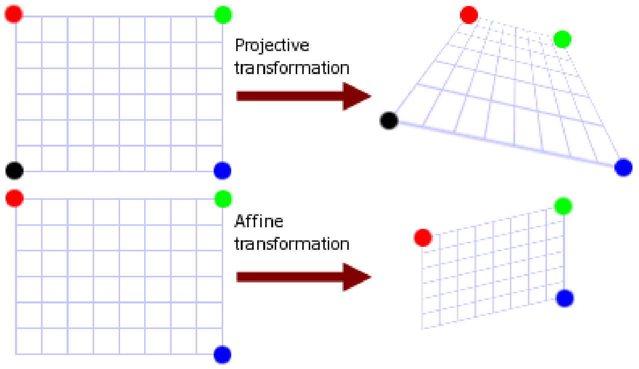


Figure 3.17 Graphical representation of Projective and Affine transformation

In literature, projective transform can be found also as perspective transform.

There are also more complex and non-linear types of transformations, but the transformation between two frames is assumed to be small enough to be estimated by perspective transformation, as it is the linear transformation that can cover most sub-transformations (parallelism, length and angle).

3.6.2. Transformation matrix estimation

Going back to feature matching (Chapter 3.5) , once features have been matched, many of them have been discarded because not a suitable match has been found. In applications where matching is not done on a salient object, but rather on a region from a very non-homogenous image where background and foreground are not easily segmentabl, many features will be discarded becausee they have not been matched.

In addition, there will be many wrong matches since the background is spread throughout the image with similar repeating patterns. In order to make an estimation on how the features have transformed from one image to another, it is necessary to estimate a transform matrix while also discarding many possible outliers which may cause difficulties when estimating this matrix.

If not all of the point pairs (Source Points, Destination Points) fit the rigid perspective transformation (meaning, outliers are accounted for), this initial estimation will be poor. In this case, one of the two robust methods should be used, RANdom Sample Consensus (RANSAC) or Least Median of Square regression (LMeDS). Both of them try many different random subsets of the corresponding point pairs (of four pairs each), estimate the homography matrix using this subset and a simple least-square algorithm, and then compute the quality/goodness of the computed homography (which is the number of inliers for RANSAC or the median re-projection error for LMeDs). The best subset is then used to produce the initial estimate of the homography matrix and the mask of inliers/outliers.

Regardless of the method, robust or not, the computed homography matrix is refined further (using inliers only in case of a robust method) with the Levenberg-Marquardt method [33] to reduce the re-projection error even more.

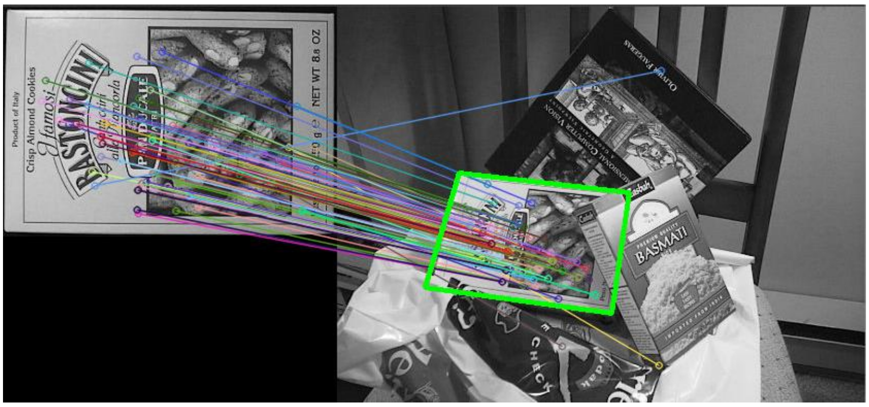


Figure 3.18 Matching of a template with applied bounding box by estimated perspective transform

The RANSAC method is able to any ratio of outliers but requires a threshold to distinguish inliers from outliers. The method LMeDS does not need any threshold but it works correctly only when there are more than 50% of inliers.

The function can be used to find initial intrinsic and extrinsic matrices. Homography matrix M is determined up to a scale. Thus, it is normalized.

A RANSAC reprojection threshold is added, such as the maximum allowed reprojection error in order to treat a point pair as an inlier. That is, if:

$$\left|\left|dstPoints_i - convertPointsHomnogenous(M * srcPoints_i)\right|\right| > ransacReprojThreshold$$

Eq. 3.12

Holds, then point i is considered an outlier. If source and destination points are measured in pixels, it is a rule of thumb for this threshold to be between 1 and 10.

Because of the aforementioned reasons and capabilities of each method, and from experimental results which show 50+ % inliers cannot be always achieved, RANSAC is the best suitable choice. Creating an adaptive threshold is something that has been tried and implemented in many applications, but no foolproof design has been adapted for applications similar to the one it will have in this thesis.

3.7. Statistical methods

3.7.1. Density - based spatial clustering of applications with noise (DBSCAN)

DBSCAN is a well-known data clustering algorithm that is commonly used in data mining and machine learning [34]. It is a density-based clustering non-parametric algorithm. From a given data set, it will group points into groups that are closely packed together (points with high number of neighbors).

DBSCAN requires three inputs:

- Data set of points described by coordinates of location,
- ε is a parameter which specifies the radius of a neighborhood with respect to a point,
- minPts is minimum number of connected points in order to consider a group as a cluster.

The working principle of DBSCAN can be described through core points:

- A core point must have at least minPts points within radial distance ε of it
- There can be many core points in a cluster
- A data point is directly reachable from a core point if the same point is within distance ε from the core point. A point can be directly reachable only from core points
- All points which are not reachable from any other point are outliers or noise points

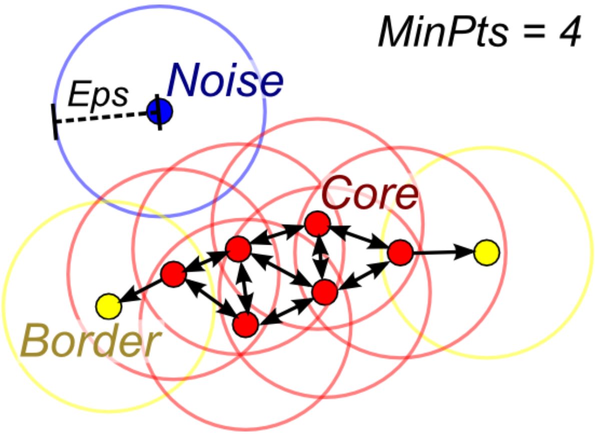


Figure 3.19 Working principle of DBSCAN

In *Figure 3.19*, the working principle of DBSCAN is depictured. The minimum points parameter minPts is set to 4, while the radius in an arbitrary *ε.* All red points are core points, because the area surrounding them in *ε* radius contains at least 4 points (including itself). Because they are all reachable from one another, they form a single cluster. Points B and C are not core points but are reachable from A (by other core points) and therefore belong to the cluster as well. Point N is a noise point that is neither a core point nor directly reachable.

For a cluster to be defined, two properties must be satisfied:

- All points within the cluster are mutually density-connected.
- If a point is density-reachable from any point of the cluster, it is part of the cluster as well.

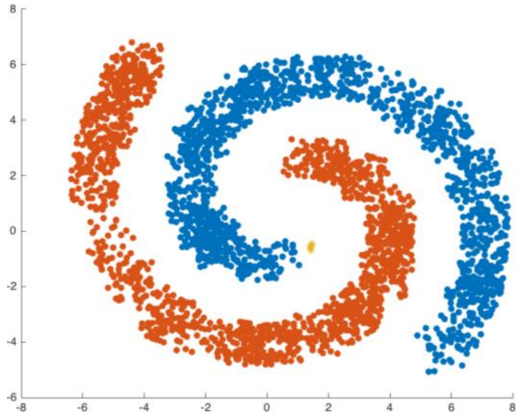


Figure 3.20 DBSCAN clustering example

3.7.2. Jaccard similarity score

Jaccard similarity score or score measures the similarity between two sets. It is also called Intersection over Union because of its working principle and is a very simple mathematical calculation which provides estimate of how similar two sets are. The mathematical formula is

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

Eq. 3.13

Where A and B are to distinct sets, and J is the Jaccard similarity score. A predefinition must be made, claiming in the case of both sets A and B being empty, J(A,B) = 1.

The result is ranged between 0 and 1, where 0 is no equality possible and 1 means complete equality. The Jaccard distance, which measures dissimilarity between sets has a very similar mathematical formula to the previously mentioned equation *Eq. 3.13*.

$$d_J(A,B) = 1 - J(A,B)$$

Eq. 3.14

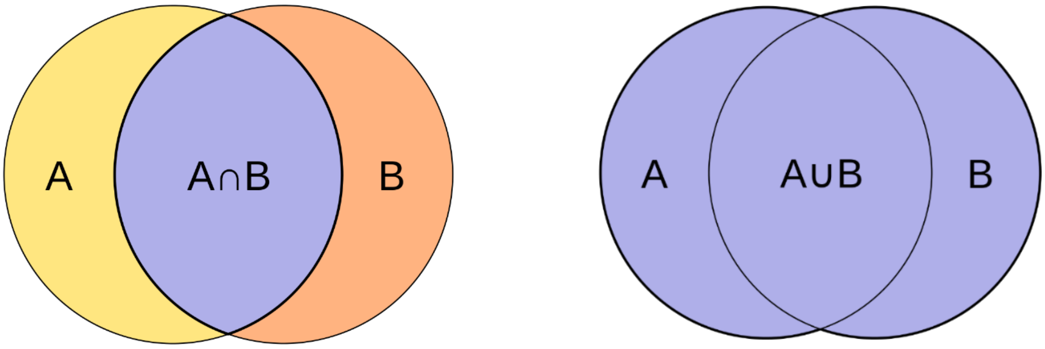


Figure 3.21 On the left, intersection of two sets A and B. On the right, union of two sets A and B

In image processing, in can be used on bounding box estimation of a recognized patter compared to the ground truth. Alongside its native functional purpose, it has found extensive use in medical imaging as described in the paper of Yeghiazaryan et al. [35].

In this thesis it will be used to calculate a reliability index of an estimation, as well tracking quality metric compared to predefined ground truth.

3.8. Edge detection – Canny

Canny edge detector, as proposed in the groundbreaking paper of J. Canny [36], is a sophisticated edge detection operator which uses a multi stage algorithm to detect a wide range of edges in a grayscale image.

Even though in the basis of the algorithm there is preprocessing being done, it is a good practice to do it a priori.

This algorithm is of great importance in this project because it provides information upon which stable long-term tracking is possible at very low computational cost.

In order to understand it, the five steps of Canny need to be mentioned and explained. The input picture must be 2-D, which leads to using a grayscale image or single channel from any color space representation. In the respective order of execution, they are:

1. Gaussian filter is applied with adaptive parameters to reduce noise.

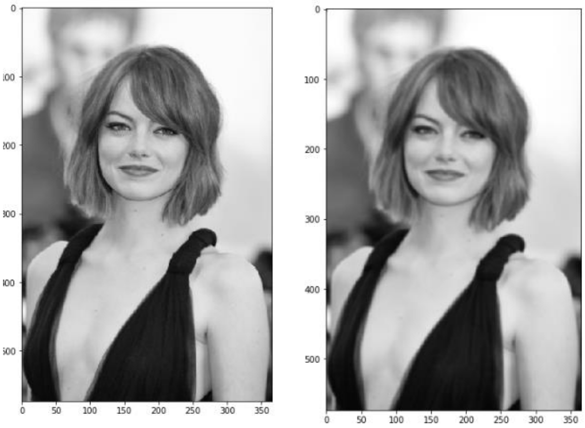


Figure 3.22 Gaussian filter effect on stock image

2. Finding intensity gradients of image is done in four directions (horizontal, vertical and in two diagonals). The Sobel operator is most widely adapted method for this and returns the first derivative in horizontal direction (G_x) and vertical direction (G_y). From this information, the edge gradient and direction can be determined with

$$G = \sqrt{G_x^2 + G_y^2}$$

Eq. 3.15

$$\theta = arctan2(G_x, G_y)$$

Eq. 3.16

where G is the edge gradient amplitude, and θ is the edge orientation.

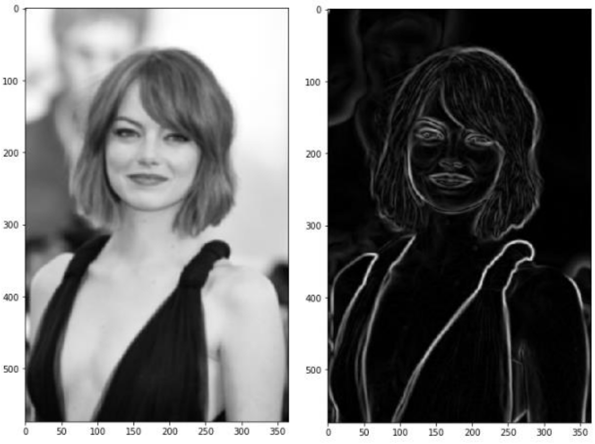


Figure 3.23 Output of Sobel operator

3. Non-maximum suppression is used as technique to acquire thin edges. Canny gives binary results with logical ‘one’s where it has detected an edge and logical ‘zero’s in the rest of the image, which means it returns an edge mask. The purpose of the algorithm is to check if the pixels on the same direction are more or less intense than the ones being processed. If there are no pixels in the edge direction having more intense values, then the value of the current pixel is kept.

   Each pixel has two main information: edge direction and pixel intensity. Based on these inputs the non-maximum suppression steps are:

   - Create a matrix of zeros with same size as the original intensity matrix;
   - Identify the edge directions from the angle matrix;
   - Check if any pixel in this direction has a higher intensity than the pixel in consideration;
   - Return the image processed with the non-max suppression algorithm

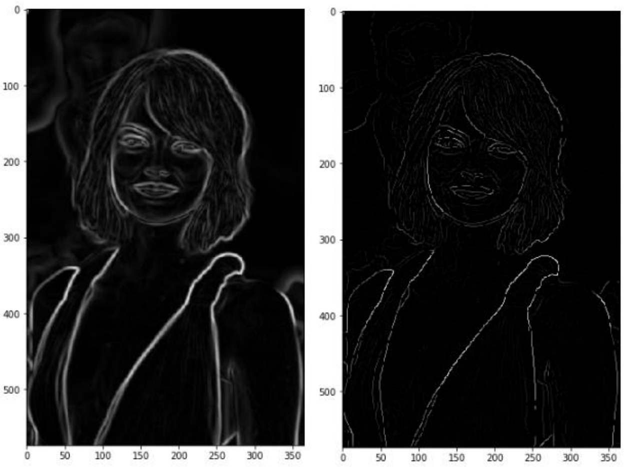


Figure 3.24 Non-maximum suppression

4. The double threshold step identifies 3 kinds of pixels: strong, weak, and irrelevant:

   - Strong pixels, pixels with intensity so high that it is sure they contribute to the final edge.

- Weak pixels, pixels that have an intensity value that is not enough to be considered as strong, but not small enough to be considered as irrelevant for the edge detection.

- Other pixels are considered as irrelevant for the edge

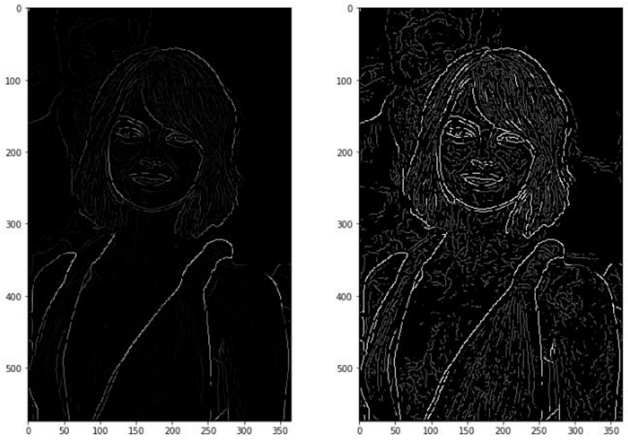


Figure 3.25 Double threshold

5. Edge tracking by hysteresis uses the threshold results, and transforms weak pixels into strong ones, if and only if at least one of the pixels around the one being processed is a strong one and belongs to the edge.

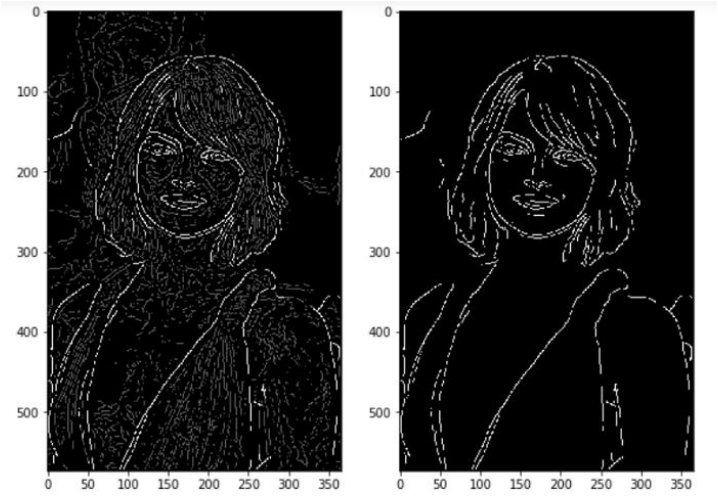


Figure 3.26 Edge hysteresis

3.9. Post-processing and refining

Until now, a couple of methods have been described which lead to localization and recognition of the Region of Interest (ROI) defined at the very initialization of the algorithm. The following subtopic concentrates on methods used in re-describing the region in order to feed the algorithm with a new boundary of the region after deformation, occlusion or partial occlusion.

Because there is no time window to specify a training data set so more complex recognition networks can be used, a set of sequential procedures are applied in order to morphologically find the best description of the region the robot needs to track.

3.9.1. Convex Hull

The convex hull of a set of points is defined as the smallest convex polygon, that encloses all of the points in the set. Convex means, that the polygon has no corner that is bent inwards. The points are defined by Euclidean 2D space coordinates.

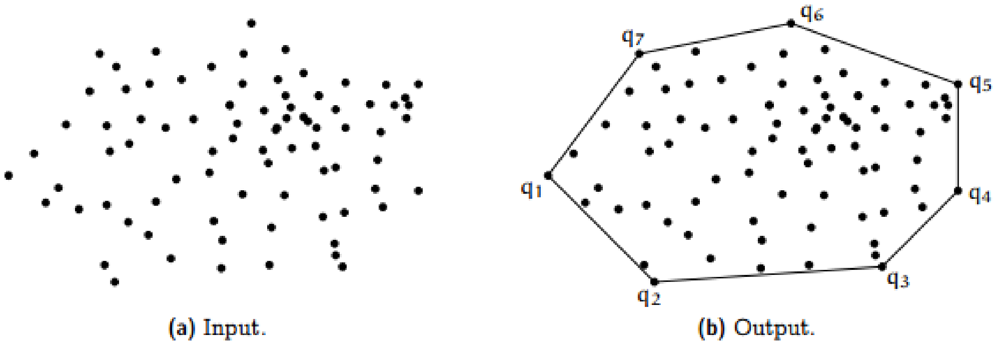


Figure 3.27 Convex Hull

This approach will be applied to the output of previously described DBSCAN.

3.9.2. Active contours

Segmentation is a part of Image Processing best described as a process of partitioning a digital image into multiple segments as sets of pixels. It has variety of applications ranging from segmenting written text to segmenting tumors from healthy brain tissue in an MRI image.

Active contours, also called snakes, is a subset of techniques used for iteratively finding the outline of an object with hard defined edges but as well as outlines of softly defined edges of an object.

The snakes model is popular in computer vision, and is widely used in applications like object tracking, shape recognition, segmentation, edge detection and stereo matching.

A snake is an energy minimizing, deformable spline directed by constraint and image forces that pull it towards object contours and internal forces that resist deformation, controlled by two evolution parameters. They are a special case of the general technique of matching a deformable model to an image by energy minimization. Snakes do not solve the entire problem of finding contours in images, since the method requires knowledge of the desired contour shape beforehand. Rather, they need an initial mask upon which they evolve.

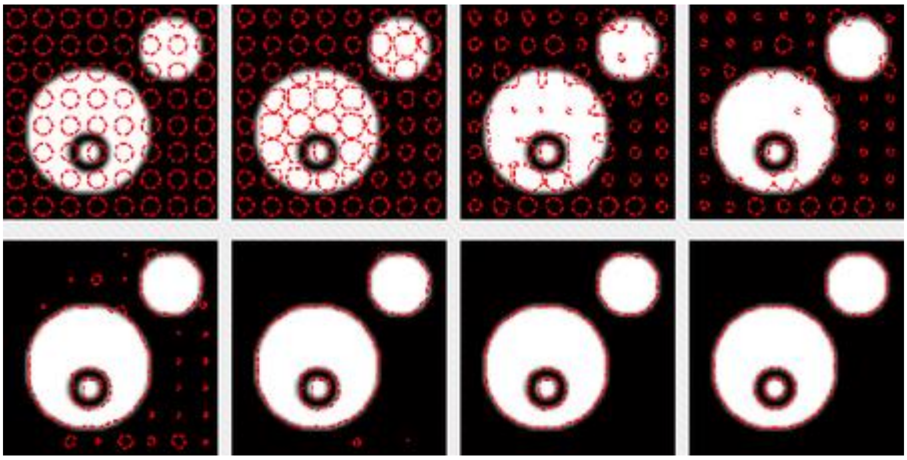


Figure 3.28 Evolution of Chan-Vese snake

The Chan-Vese approach is a segmentation algorithm designed to segment objects without clearly defined boundaries. In this thesis Morphological Chan-Vese (MCV) is used, an approach based on level sets that are evolved iteratively to minimize an energy function, which is defined by weighted values corresponding to the sum of intensity differences from the average value outside the segmented region, and a term which is dependent on the length of the boundary of the segmented region. It requires a 2-D input image with predefined initial mask. Once initialized, MCV will expand the initialization mask through the energy equation in all directions, inwards and outwards, depending on the parameter definitions. The energy equation is being directed by:

- lambda1 : weight parameter for the outer region
- lambda2 : weight parameter for the inner region
- smoothing parameter for nonlinear interpolation of edges.

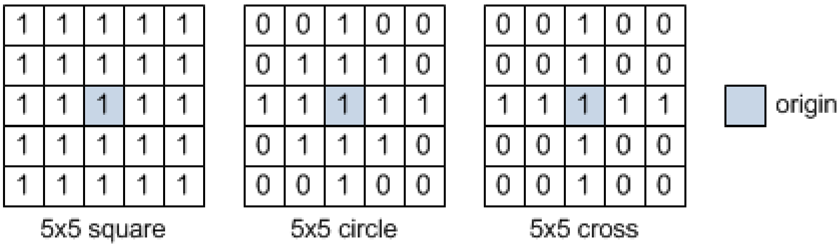
If lambda1 is larger than lambda2, the snake will force towards expanding and vise versa.

Snakes might not always return only one region, but rather a couple of contours. In order to separate them in different objects, a procedure for finding contours is applied.

But before this is used, the regions must be processed in order to discard of small specks of grouped pixels and different spikes and peaks on the outskirts of a region which are most likely noise. This procedure is called Morphological Opening.

3.9.3. Binary filters

Morphological Opening is the process of applying dilation after erosion using a structuring element. A structuring element is a 2-D matrix element which can be of different types, see figure below.



1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

5x5 square

0	0	1	0	0
0	1	1	1	0
1	1	1	1	1
0	1	1	1	0
0	0	1	0	0

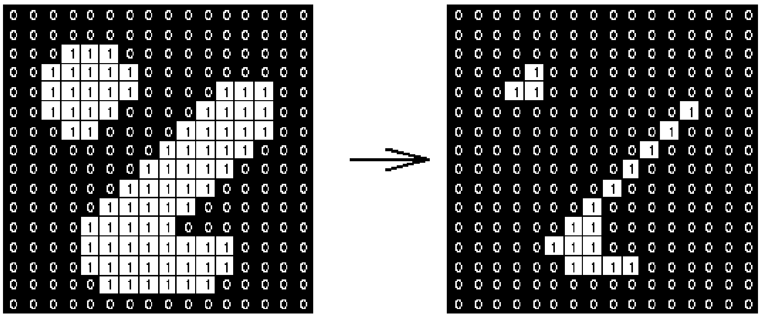
5x5 circle

0	0	1	0	0
0	0	1	0	0
1	1	1	1	1
0	0	1	0	0
0	0	1	0	0

5x5 cross

Figure 3.29 Types of kernels (structuring elements)

This element is applied to each pixel just outside of the boundary of a binary contour. In the case of erosion, if the pixels in the contour correspond at that iteration with some of the logical 'one's from the element they are switched to a logical 'zero'. This way the contour decreases inwards depending on the size and type of the element. Having a larger structuring element will result with a more extreme erosion effect.



0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0	0	1	1	1	0	0
0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
0	0	0	1	1	0	0	0	0	1	1	1	1	1	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3.30 Effects of erosion on binary image

In the case of dilation, the opposite happens. When the logical ‘one’s from the element reach a logical ‘zero’ from the contour, they will switch it to ‘one’ resulting with an expanded contour.

Morphological opening will first use erosion to remove boundary parts of the contour which are most likely to be caused by noise, and afterwards gives a well-posed contour by expanding the firm boundaries and filling up concave cavities.

CHAPTER 4
The proposed method

In this chapter, the proposed method will be described. It will be seen how the previously described algorithm were implemented and the specific purpose they have. Also, the choice of parameters will be provided and explained why that value has been used. The additional metrics, created for the specifically for this thesis, will be explained along with some functions proven to give great results regarding precision and computational effectiveness.

4.1. Control Problem Statement and Solution Concept

The elements of a conventional control system are:

- Input - reference signal,
- Controller,
- Controlled plant,
- Feedback.

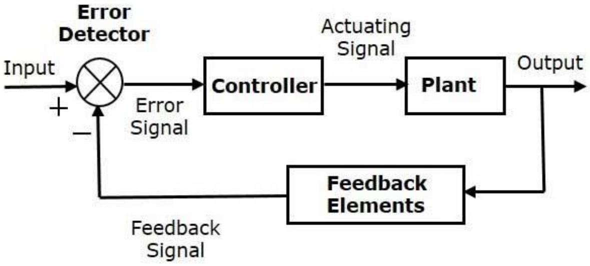


Figure 4.1 Sample control block diagram

A medical robot is a complex system which requires synchronous control of each moving element, (joints of the manipulator arms). The control of each of these joints, installed as rotational or translational elements, is directly commanded by a master control unit. The position of the end effectors, in the case of medical robots it is either the endoscope or a surgical tool, can be directly computed from the initial calibration as presented by Roh et al. [37]. By knowing:

- initial points of end effectors,
- amount of movement for all moving elements (actuators) of each arm,
- distances between each joint of an arm (fixed translations),
- static positions of the ‘root’ of each robotic arm,
- transformation matrices (D-H parameters) between each joint of an arm
- transformation matrices (D-H parameters) between each arm,

it is safe to assume that at each point of time the 3D position and orientation of the end effector and every element of the arm is known (Figure 4.2).

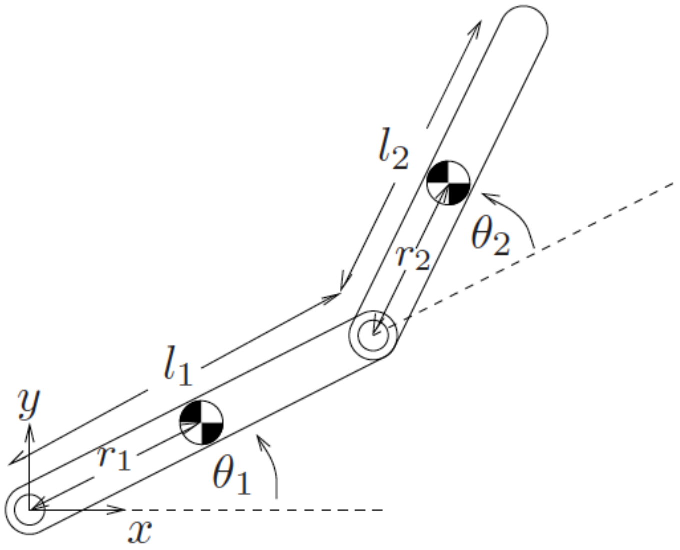


Figure 4.2 Robotic arm motion variables

This allows for the master control unit to plan the path of each element of each robotic arm with full control of the motion. Having such freedom, the application of active constraints (AC) becomes plausible.

The reference motion and position signals are provided to the controller, which accounts for the constraint generator and the dictated constraints from it. With this influence on the output motion of the robot, the actual effective motion is:

$$ActualMotion = ReferenceMotion - ConstraintPenalty$$

The controller uses the active constraint generator effectively to modify the output in order to satisfy the constraint. Such constrained control system is the one presented in *Figure 4.1*, where the controller takes the form of *Figure 2.15*. The type of constraint is not of interest to this thesis, as it only affects the motion and position of the tools tips and does not affect the visual aspect of the surgery. However, it is confident to say that a mixture of positioning and motion constraints is used. Their effectiveness is directly related to the performance of the AC tracking method, as the positional vicinity and penalty magnitude is dictated by the current definition of the AC

The initial concept of the solution presented in this chapter was an adaptation of Model Predictive Control, which is an advanced method for that is used to control a process while satisfying a set of constraints through optimization at each step. After some time, many aspects evolved or diverged from the idea based on MPC, but some details remain. A flow chart capturing the general outlook of the proposed method is shown in *Figure 4.3*.

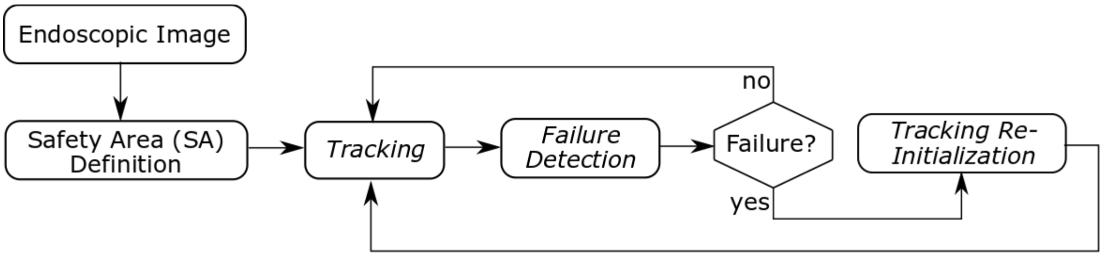


Figure 4.3 General flow chart of the proposed framework

The proposed method in this thesis is completely implemented as a 2D concept. However, it is required to provide this information as 3D information in order to be effectively used in a surgery. It is needed to provide them in 3D in order to cover the area of an organ, which is in 3D. Not only cartesian position is needed, but also depth. For this reason, a method is used for projecting 2D points to a 3D point cloud. In this way, the active constraint will be able to protect the tissue at all times regardless of the absolute distance between the tool tip and the tissue being operated.

In addition, a brief yet informative set of data is being displayed at the top left corner to provide the surgeon at all times with data on how reliable the AC tracking at every moment of the surgical procedure is. The proposed method is entirely implemented for a 2D video stream, most often by using the left camera from a stereo camera pair, but it will be shown in the end how these 2D points can be projected onto the 3D surface reconstructed from image pairs from the stereo camera.

4.2. Implementation overview

The main program is written in Python, and the most used library is OpenCV. OpenCV is natively written in C++. In OpenCV, all algorithms are implemented in C++. However, these algorithms can be used in different languages like Python, Java etc. This is made possible by the bindings generators. These generators create a bridge between C++ and Python which enables users to call C++ functions from Python. It allows for the performance of running image processing functions in C++, while having the simplicity of Python for the rest of the project. The second most used library is Numpy, which is a dedicated Python library for numerical calculation. The third most important is Scikit-learn, which is a library vastly used for scientific purposes and it includes many published algorithms which are useful in many situations including numerical analysis, image analysis and statistics. In this chapter, the proposed method will be explained.

For visual representations of the processed frames of each step from the proposed method, a surgical video of pancreatectomy is used with resolution of 1280 x 720 and frame rate of 25 Hz.

4.3. Initialization and Pre-processing

The initialization phase is responsible for creating instances of all methods to be used in this method. From *Figure 4.3*, it encapsulates the two blocks outside of the loop. The definition of the AC to be tracked will happen in this phase and everything that will happen further on is to keep *this* AC definition alive and as close to the original possible. A brief flow chart of the workflow of this phase is depicted in *Figure 4.4*.

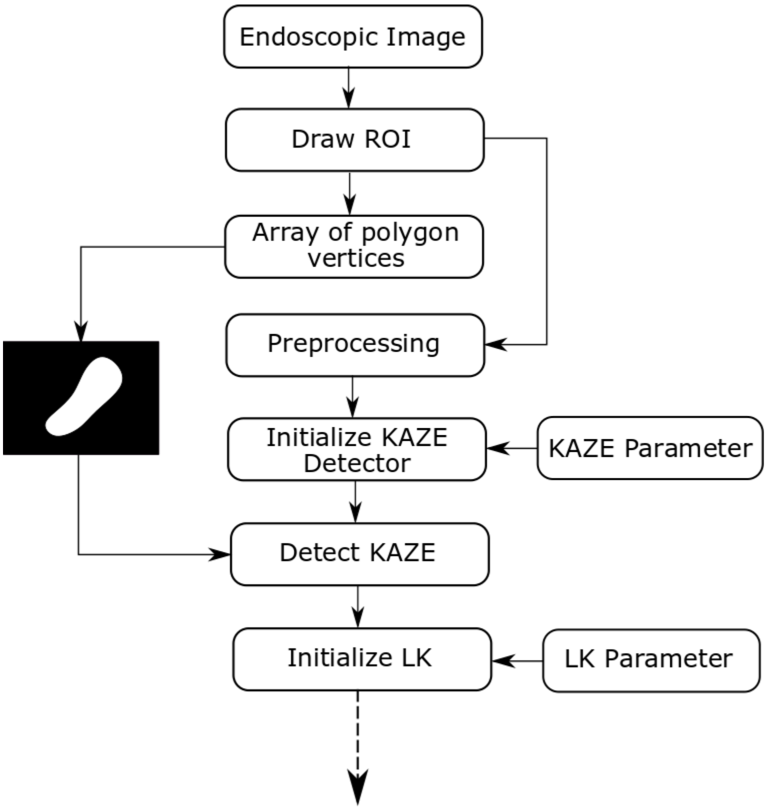


Figure 4.4 Initialization phase flow chart

The initialization phase starts at the moment the algorithm is activated or called. At that instance, a frame from the video stream is received in RGB color space. This image (*Figure 4.5*) is used and displayed in front of the operator. Then, he/she is able to use any kind of input device, such as touch, stylus, or mouse, to define in a free-hand manner region of the tissue, which needs to be kept reserved and protected (active constraints from Chapter 2.9).

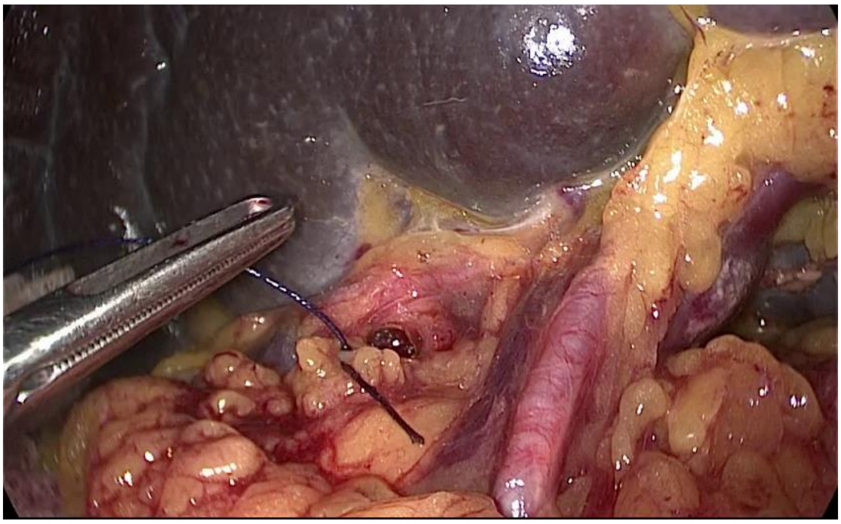


Figure 4.5 Initial frame from video stream

The vertices of the polygon are shown on *Figure 4.6* as red points. The goal of this thesis is to be implemented in partial nephrectomy, which is a procedure where the key tissues (renal vein and renal artery) require protection from possible damage. This means that active constraints will be most likely applied on these vessels.

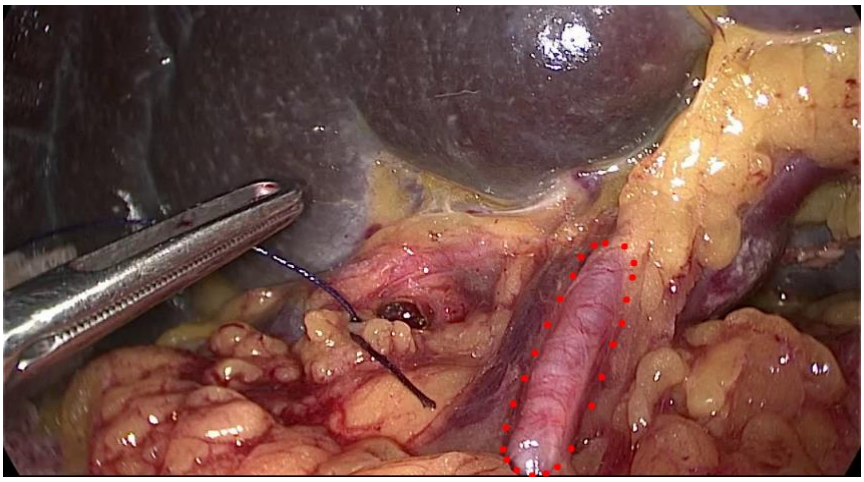


Figure 4.6 Selected points of Active constraint

Once this region is selected, the program will translate the drawn points to an array of vertices with horizontal and vertical coordinates of a 2D Euclidean space:

$$SA = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix}$$

Eq. 4.1

This is to be stored and saved in the program’s memory, because it will be used in a couple of different situations.

After defining the Region of Interest (ROI), which is the region described by the Active Constraint (AC), the initialization phase can continue. Firstly, the image needs to go through a sequence of image processing steps.

The very first applied method is resizing. Both axis, x and y, are scaled by particular ratio coefficients, ratio_x and ratio_y. In this thesis the ratio 0.5 was used, meaning that:

ratio_x = ratio_y = ratio = 0.5 .

This will scale the image to one quarter of the original size, because both of the axis are halved. This will size the frame down from 1280x720 to 640x360.

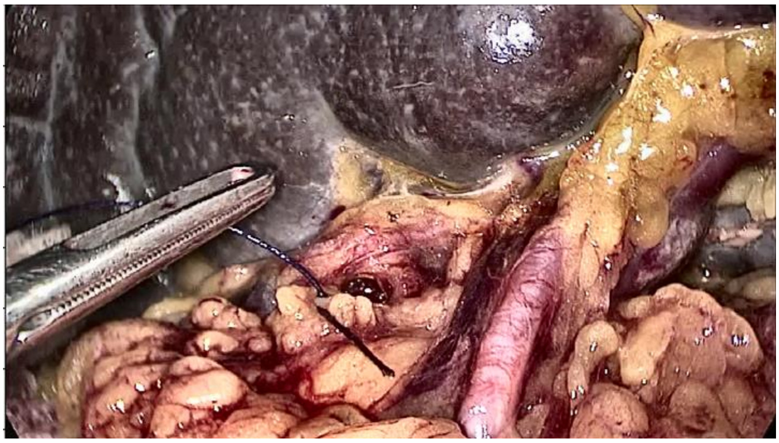


Figure 4.7 Contrast-Limited Adaptive Histogram Equalization

After resizing, Contrast-Limited Adaptive Histogram Equalization (CLAHE) is applied with a ClipLimit of 3. The theory and meaning behind CLAHE is covered in Chapter 3.2.3. Contrast – Limited Adaptive Histogram Equalization (CLAHE)This will adjust the local contrast of the inner blocks with size of 8 by 8 pixels in order to equalize the redistribution of lighting throughout the image and locally enhance contrast to increase visibility of edges. CLAHE is computer on the Lightness channel of Lab color space representation of the processed image. The pictured figures *Figure 4.7* and *Figure 4.8*, have been converted back to RGB for ease of display.

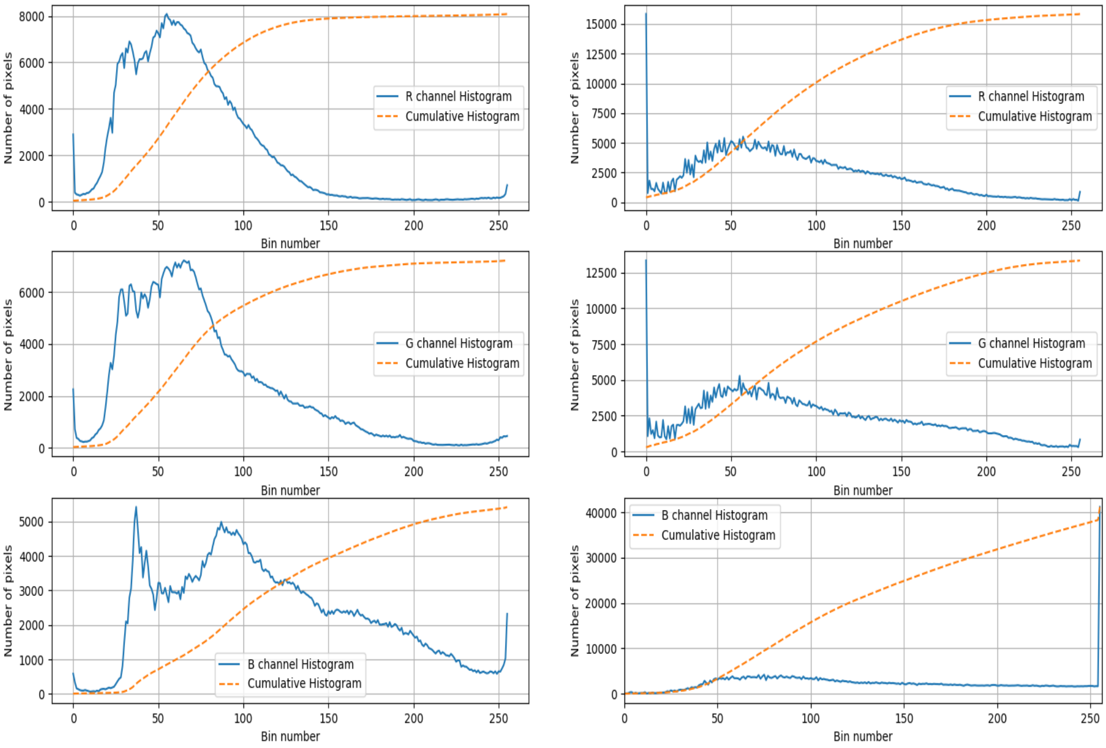


Figure 4.8 The first row shows the histograms of the frame before CLAHE is applied, with each sub-figure corresponding to each R, G and B channel. The second row shows the histograms of the frame after CLAHE is applied. The dashed line is the cumulative representation of each histogram

The visual effect of CLAHE can be seen in *Figure 4.7*, while the output histogram effects compared to the input of the CLAHE algorithm are shown in *Figure 4.8*.

The following step is Gamma Correction (Chapter 3.2.1. Gamma correction) This will adjust the saturation of pixels with the goal to again, enhance edge revealing characteristics even more. The gamma coefficient is set to be γ = 0.5, and applied to the equation:

$$V_{out} = AV_{in}^{\gamma}$$

Eq. 4.2

where A is commonly set to 1. This low value will introduce darkening of the image, but for the computer it will mean more details regarding edges and characteristic small regions that will provide much more information for edge detection and feature extraction. The visual effect can be seen by comparing *Figure 4.7* to *Figure 4.9*, where the latter one is after applying Gamma adjustment. Gamma is intensity based, and is applied to each pixel of each RGB channel individually.

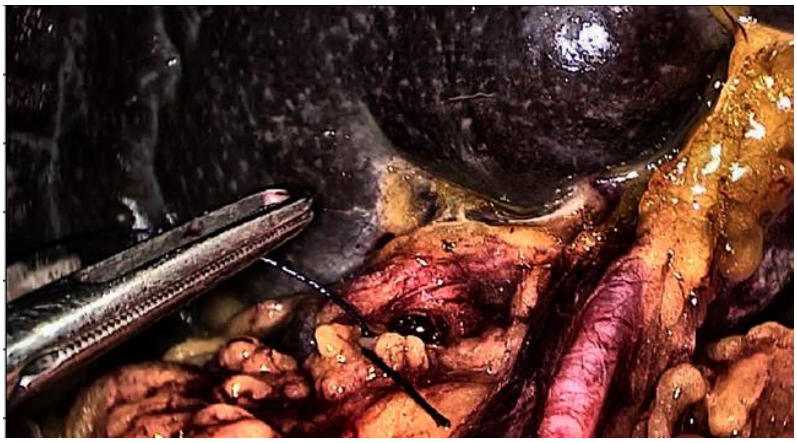


Figure 4.9 Gamma correctiont

This will result with a change of distribution of the histogram bins, due to the nature of Gamma correction which compresses some parts of the image and stretches others. This is best seen with the "spikes" which appear in the second row of *Figure 4.10*.

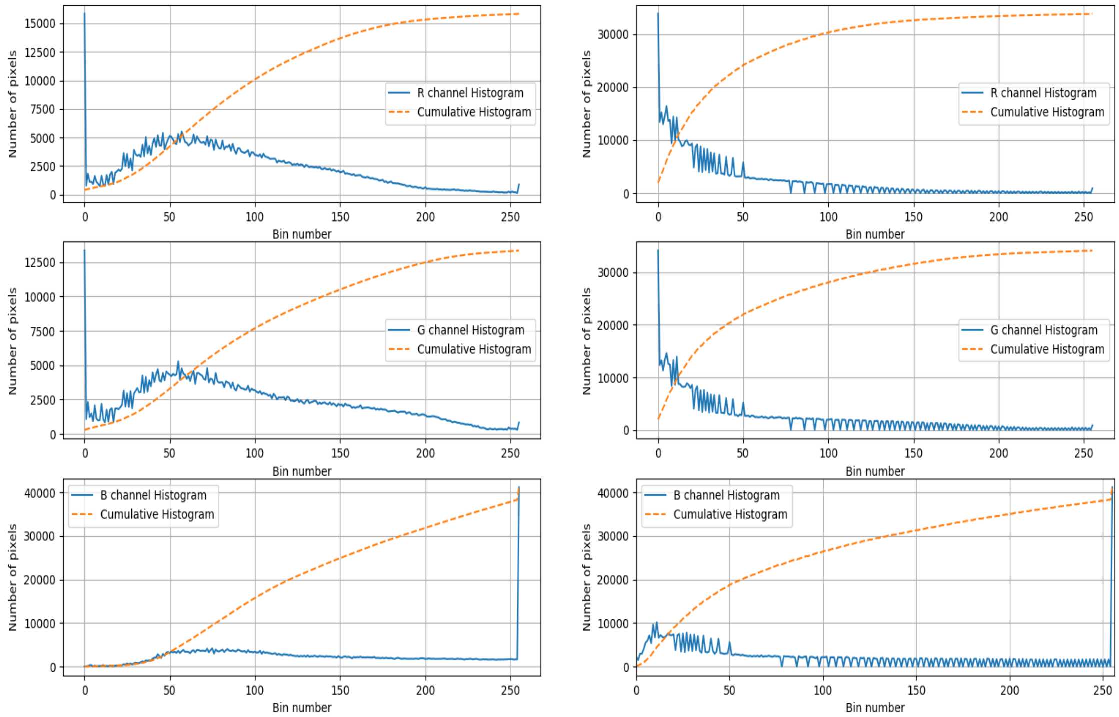


Figure 4.10 The first row shows the histograms of the frame before Gamma Correction is applied, with each sub-figure corresponding to each R, G and B channel. The second row shows the histograms of the frame after Gamma Correction is applied. The dashed line is the cumulative representation of each histogram

The next filter to be applied is Gaussian Blur or Gaussian smoothing. This is used to smooth out objects affected by noise. It is usually done in the beginning of image preprocessing phase, but here it is done at the end in order to decrease the noise effect that might have been amplified by edge preserving filters done prior to this. The reason for this is the assumption that much of the noise is due to non-regular lighting in the image. It is proposed here to first enhance the effect of lighting in order to gain stronger edge definitions. Gaussian blur is defined by a kernel, which acts as a moving window in 3-D matrix.

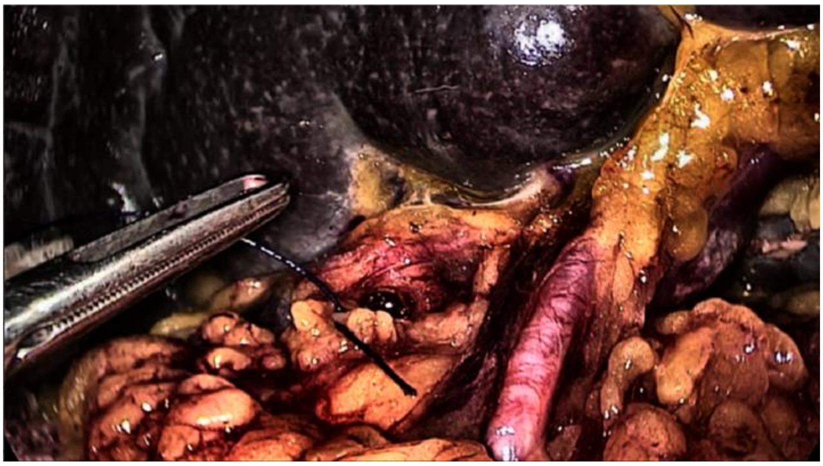


Figure 4.11 Gaussian smoothing

The size of this kernel in this thesis is 5 by 5. The reason for this size is that according to the size of the scaled image, everything above 7 by 7 introduces too much blur effect, which becomes rather more visible instead of just removing small noise content and smoothing out edges. Anything less than 5 would have little to no effect. The results are not visible to the naked eye but can be monitored through the outputs from the algorithm and improvement of stability and repeatability.

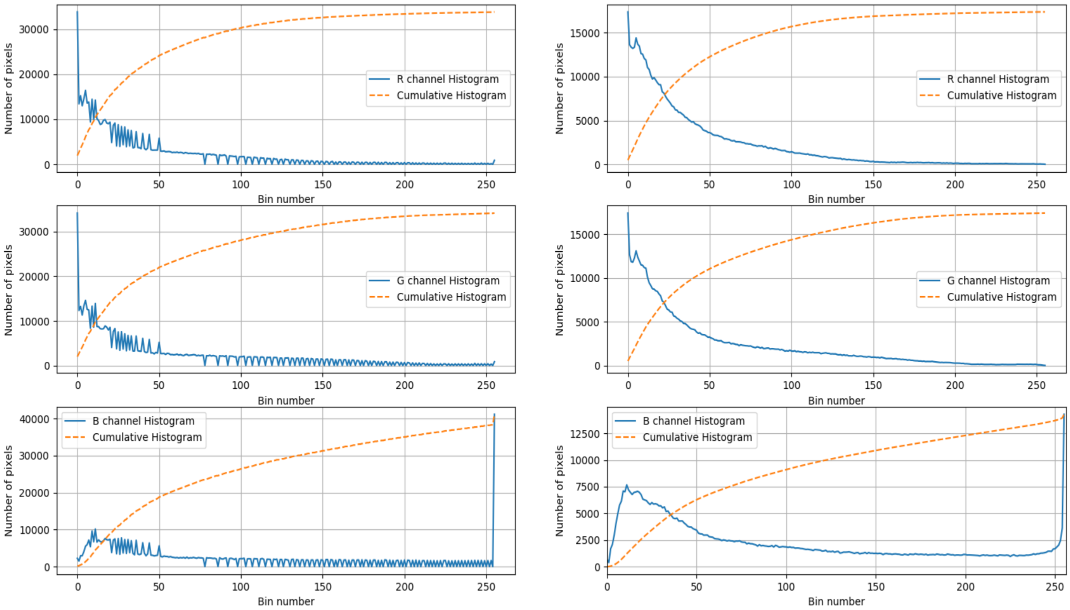


Figure 4.12 The first row shows the histograms of the frame before Gaussian smoothing is applied, with each sub-figure corresponding to each R, G and B channel. The second row shows the histograms of the frame after Gaussian smoothing is applied. The dashed line is the cumulative representation of each histogram

At this point, all procedures applied on the three channels RGB and Lab are done. The following methods work with grayscale images.

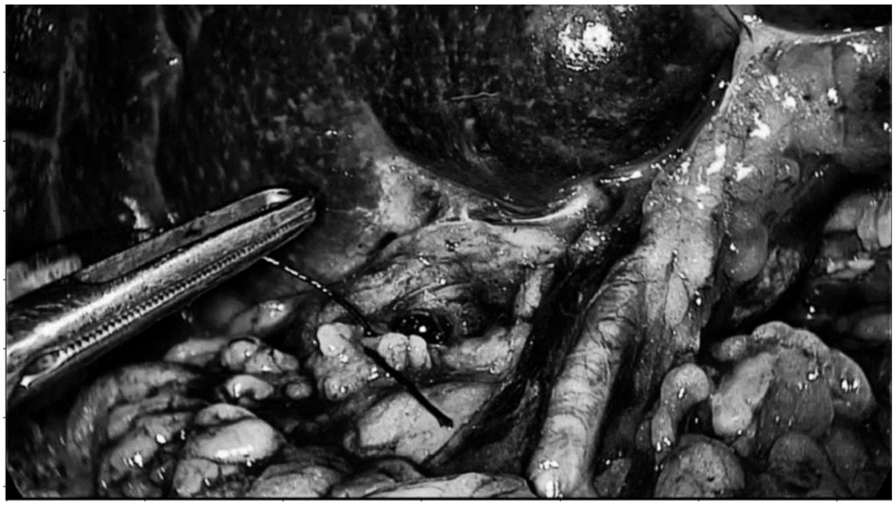


Figure 4.13 Grayscale frame

The first one is Histogram Equalization. It will provide a normalized image with respect to the intensity of each pixel in a grayscale representation. Through empirical results, it was chosen to use a grayscale image for this step instead of doing it separately for each channel because there were no substantial improvements, especially not ones which will gain more visible and detailed image, while, when using histogram equalization on grayscale, an image is gained which has much more visibility and much more details.

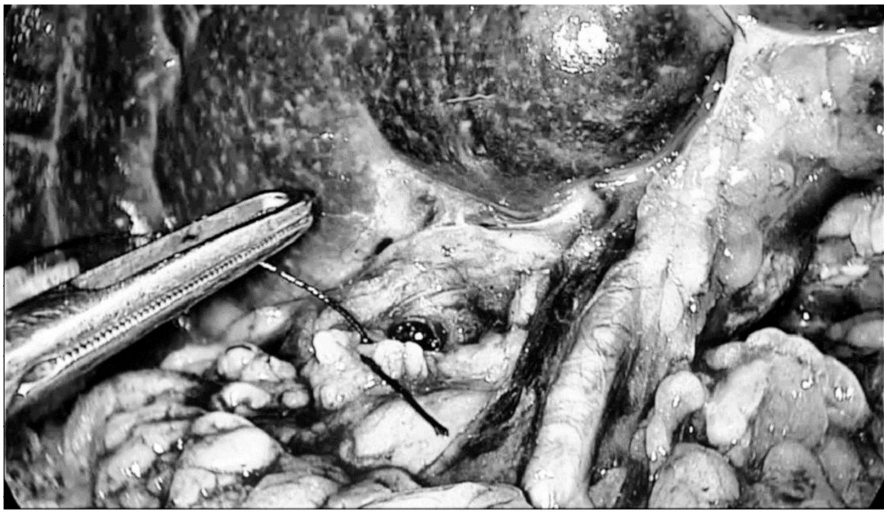


Figure 4.14 Histogram equalization

It is noticeable from *Figure 4.15* that the effect of Histogram equalization (HE) has a similar effect to CLAHE (*Figure 4.8*) in the sense of stretching and compressing the image. However, from the cumulative slope it can be seen how HE affects the distribution of pixel values in order to equalize it through each color value. Thus, gaining a cumulative histogram with the form of linear equation. This makes up a more robust image with an even distribution and average pixel value of 128.

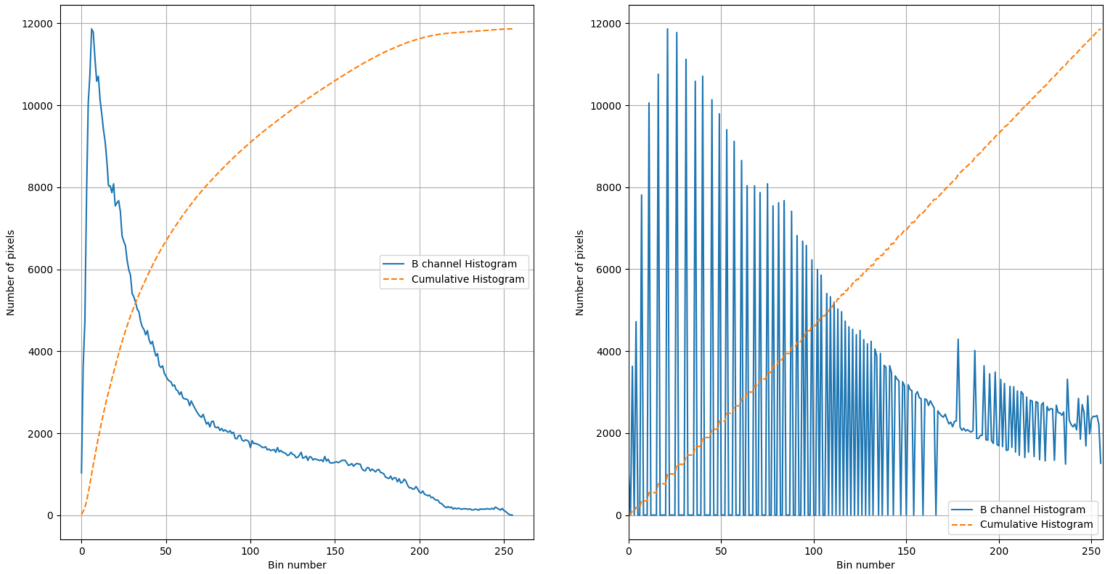


Figure 4.15 The first plot shows the histograms of the frame before HE is applied; the second row shows the histograms of the frame after HE is applied. The dashed line is the cumulative representation of each histogram

Histogram equalization provides a more even distribution of light intensity through the image, and improves performance in distinguishing tools from tissue.

With this, the preprocessing is finished. A grayscale image is gained with well-defined, enhanced and preserved edge definitions, distinctive details and with noise reduced as low as possible.

Now it is required to detect and track certain morphological and color features of the video stream input. A surgery is an environment with many objects overlapping, and no salient objects can be defined for a well-posed foreground-background segmentation. Thus, it is proposed to take advantage of the enhanced edge definitions. Previously, it was shown what type of preprocessing each frame is put through, with the motivation of robustness and edge preservation.

This following step is one of great importance in this thesis, and an idea that is rarely used in practice. Here the Canny Edge (Chapter 3.8) detector is introduced as the input for the tracking phase. To understand how and why it is used, it will be divided in two parts.

Firstly, the image received from preprocessing is fed into the Canny detector. This image has been prepared in order to provide such basis that it will be possible to extract as much well-defined edges as possible. Here arises the problem of choosing the correct lower and higher threshold parameters in order to have a stable output, robust towards noise content.

For this purpose, Otsu's threshold is used [38]. It is an algorithm used for adaptive conversion from grayscale to binary images. It was applied to randomly chosen frames from five segments of three videos from different surgical procedures, and the mean value was used on each video. An interesting finding was that each segment had a mean value of Otsu's segment ranging from 0.078 to 0.0795. After averaging all data gathered, the number 0.0783 was chosen. In practice Otsu's threshold is considered as the upper threshold for Canny Edge Detection, while the lower threshold is 0.0783/2.5 = 0.0313. This proportion of 2.5 is used as rule of thumb (taken from MATLAB's implementation of Canny Edge detector) when it is expected for the objects not to have distinct edges, which is the case here where there is no salient object present. The output from Canny with these coefficients is shown in the *Figure 4.16*.

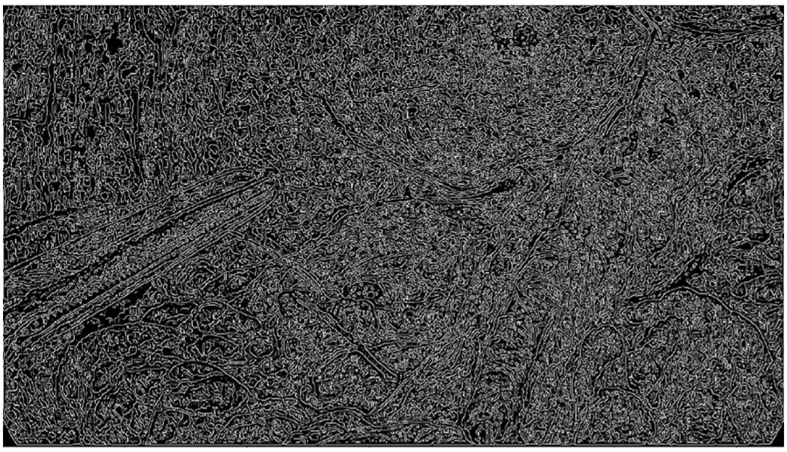


Figure 4.16 Canny edge detection output

However, this binary image looks like a messy mosaic with randomly dispersed white pixels on top of a black background. Feature detectors and tracking algorithms expect grayscale images rather than binary ones, so it is possible to infuse more details to the binary output of Canny's detector in order to make the edges even more characteristic and distinguishable.

The next step, which gives detail to these edges, is element-wise multiplication of the binary image (mask) with edges and the grayscale image used as input to Canny (*Figure 4.14*). This will give color and morphological value to the edges. In a sense, not only the shape of an edge is of great use to the procedure but also the values of each pixel of the edge. The final look of a frame is shown in *Figure 4.17*.

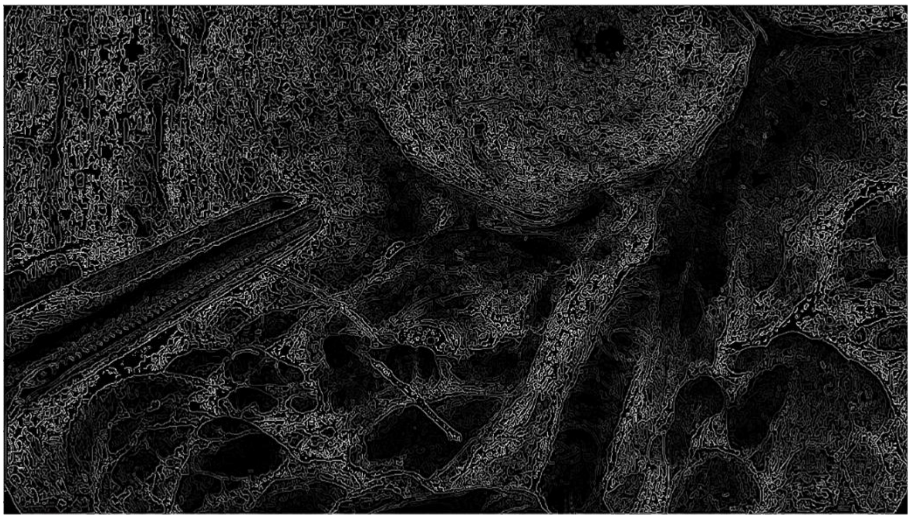


Figure 4.17 Enhanced Canny output with morphological structure

It is noticeable that some sections are dark, even though it is expected to have information there. This is not a setback, and the reason is that they are most likely not to produce any valuable information for the procedures that will be applied later, or that they are regions that lack texture.

Processing an image filled with non-binary edges will produce much more features than needed. Working with all of them is neither needed, nor effective. For the purpose of tracking only the selected region, that segment will be extracted from the image by selecting only that part. This selection is done by applying a mask, which is created upon the polygon defined by the surgeon in initialization. This mask will allow only the pixels inside of it to keep their value, while everything outside of it will be kept to ‘0’. Having a black background will not produce any feature detections. The output from this masking is shown on the image *Figure 4.18*.

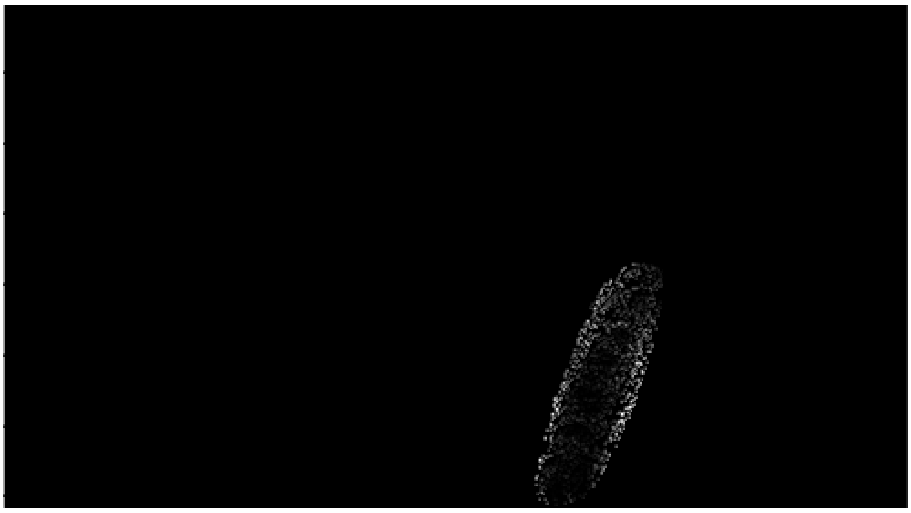


Figure 4.18 Enhanced Canny edge inside of AC

During the initialization phase, one more thing needs to be introduced. This is the ‘Buffer’ concept, which includes a an accumulator of AC models which will be used for re-initialization of the AC definition. Together with the models, several different specific information for each model are stored. These data that come together with each model are used either to decide if the model needs to be updated with a newer one, or some information that will be used for re-initialization of the AC definition. A very specific voting scheme is used in order to fill each place in the Buffer, and for this some equally specific definitions need to be made.

The Buffer has length of 13 elements (Temp refers to template, or a model), of which:

- $Temp_1$: The first model is always the one defined by the surgeon.

- $Temp_{21,22,23}$: The following three (2-4) models are the ones with highest number of detected features.

- $Temp_{31,32,33}$: The following three (5-7) models are the ones with strongest features, features with highest response.

- $Temp_{41,42,43}$: The following three (8-10) models are the ones with highest number of detected feature during re-initialization and acceptable Jaccard score (part of re-initialization sub-chapter).

- $Temp_{51,52,53}$The last three models are the last known ROI transformations during tracking with an acceptable number of trackers alive (>80%)
A pseudo-code representation of the Buffer array is

$$\mathrm{Buffer} \\ = \{(\mathrm{Temp}_1, \mathrm{No.KAZE}, \mathrm{Av.Resp}, \mathrm{No.KAZE}_{\mathrm{reinit}})_1, .., (\mathrm{Temp}_{53}, \mathrm{No.KAZE}, \mathrm{Av.Resp}, \mathrm{No.KAZE}_{\mathrm{reinit}})_{13}\ \}$$

Each element of the array, has a model and three additional different elements:

- $No.KAZE$: Number of detected KAZE features
- $Av.Resp$: Average response of the strongest 40 features (Chapter 3.3, equation $Eq.\ 3.4$)
- $No.KAZE_{reinit}$: Number of detected KAZE features at re-initialization phase

Each of the tissue models in the buffer will provide insight and data on different aspects of feature tracking and detection.

For now, it is enough to be said that at initialization every element of Buffer will be filled with the same model and that model is the one from the very first frame on which the operator draws the ROI . As for the other data in the Buffer, each element of any kind is set to 0, the virtual minimum. The reason for this is that the first element is never changed, and the rest need to be updated as quickly as possible. By setting the virtual minimum, all elements will be updated in the first 13 computational cycles. The model is acquired by applying the mask which allows data to be preserved only by the pixels inside of the AC. This masked image is than cropped with respect to the bounding box around the AC definition, as shown in *Figure 4.19*. By excluding everything from the image that is not in the region of interest, a more robust feature matching method will be achieved.

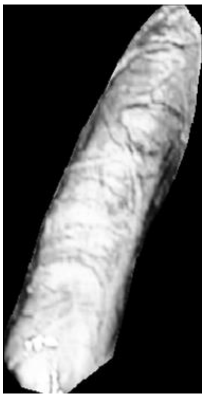


Figure 4.19 Model template

Going back to the masked figure with non-binary edges (*Figure 4.18*), the next step is to apply feature detection. As mentioned in the previous chapter, in this thesis KAZE detector and descriptors are used. Using the stock default parameters, the results are not very useful. Therefore, a set of parameter used for KAZE detection are:

- Threshold = 0.0001. Responsible for detection of local extrema. Higher values mean only more significant are taken into consideration.

- Number of Octaves = 3. The highest order of scaling allowed. Similar to Pyramidal Method from LK (Chapter 3.4), it will scale down to detect over a compressed image for larger objects.

- Number of Octave Layers = 3. Amount of levels between scales, used to achieve smoother transition between layers.

Characteristic keypoints, called features, will be returned from KAZE's detector. These points will be considered as reference points for tracking until re-initialization is required. Because the only area of interest is what is inside the AC definition, KAZE detection is applied only to that area. Instead of using a normal grayscale image, the method in this thesis uses the enhanced Canny mask, depicted in *Figure 4.18*. This grayscale edge defined image will produce much more features with edge specific feature descriptors. In *Figure 4.20*, a graphical representation of all descriptors detected is given, with red circles for each keypoint.

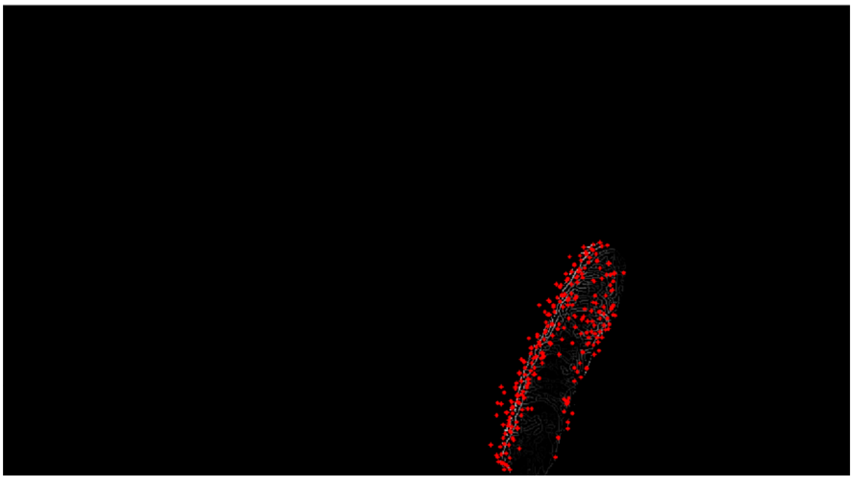


Figure 4.20 KAZE detection in AC area

4.4. Tracking

Upon acquiring these points, it is now possible to initialize the LK (Lucas-Kanade) tracker, see Chapter 3.4. Tracking is a phase which should be as stable as possible in order to achieve performance that will allow stable estimation of the tissue deformation without re-initializing when not completely needed. A simple flow chart of the tracking phase workflow is depicted in *Figure 4.21*.

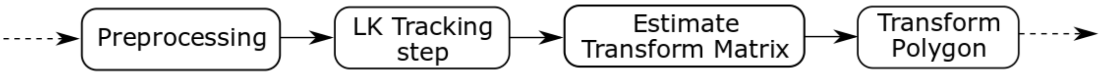


Figure 4.21 Tracking phase workflow

For this, two parameters need to be set. One is the size of window which goes through the image to look for matching features to be tracked. It needs to be in compliance with the size of the image, not too small nor too big. By window size, it is meant the integration window in each dimension. As for the other parameter called MaxLevel, it means the maximum allowed scaling depth of Pyramidal method.

These are given to the system only at the first initialization and the same parameters are used throughout the entire runtime. Using these parameters an LK tracker object can be initialized, and once it is initialized at each new frame it will be called. For this LK requires to be fed with keypoints to be tracked, and this is why KAZE detection was done on the initial frame. The window size use is 31 by 31, and MaxLevel is 4.

At each frame, LK must be provided with:

- Origin frame (last frame)

- Points to be tracked

- Destination frame (current frame).

In the very first cycle of LK, the points to be tracked are directly the ones detected by KAZE. They are the initialization points for tracking, and at each cycle they can only be found or lost unless it has been re-initialized with a new set of points to be tracked. In Figure 4.22, a finite set of 40 points are tracked and the ones which are successfully tracked are connected with a colored line in both images.

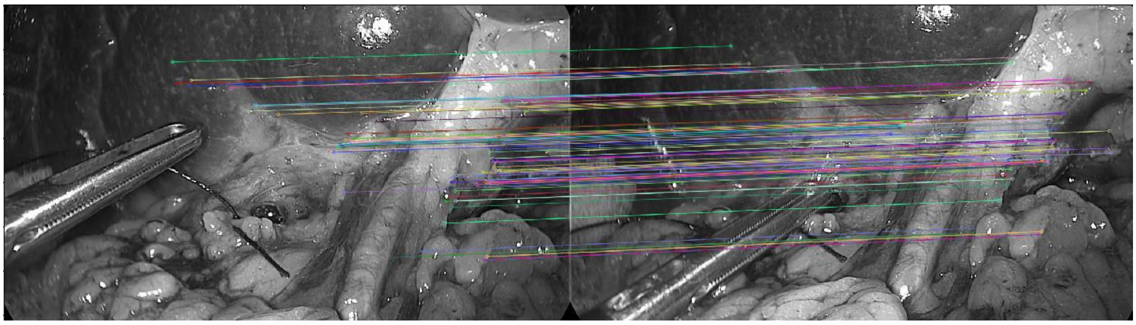


Figure 4.22 LK tracked points

Once Pyramidal Optical Flow estimation is done by LK tracker, it will provide an array with the found points and their new estimated locations. Along this matrix, there is a 'status' array. This array of type mask has the size of the input array and has:

- 'TRUE' at the place corresponding to a point which was found and tracked in the destination frame;

- 'FALSE' at the place corresponding to a point which was not found nor tracked in the destination frame.

Now the goal is to find a mathematical representation on how the points have transformed from one frame to the other. These points are the tracked features, and the following assumptions are made:

- The transformation between two frames is at most of type Perspective (rigid).
- There are more than 5 live tracked points, because 4 is the required minimum for estimating perspective transformation (Chapter 3.6.2. Transformation matrix estimation).
- LK requirements have been correctly held and satisfied, and estimation procedure will converge.

The tracked features, which were estimated in the destination frame can be considered good estimations as they must meet the minimal requirements defined in LK. But still, some of them may be not correct, or in other words outliers.

Therefore, an estimation technique for the transformation matrix must be used which includes discarding of outliers, and one such is used. It is a stock OpenCV function called FindHomography (Chapter 3.6.2. Transformation matrix estimation)., which is a technique based on RANdom SAmple Consensus (RANSAC).

The method is iterative, and randomly picks 4 pairs of each image. The way RANSAC is included here is that the rule for RANSAC to consider some random sampling pair as outlier, the re-projection error must be below some predefined threshold. If the re-projection is less than this value, it will be an inlier.

For the tracking phase, this RANSAC reprojection threshold is defined to be 25. With respect to the size of the image, and the characteristic of LK to only track subpixel movements, this may be considered a very large value. However, it is very useful to have it as a large value for the following reasons:

- Setting a small value will make the algorithm very strict, and any small movement will be discarded. This is not useful because a tracker will be lost, even though in the end estimation this small movement will not affect the transformation matrix.

- Because of the Pyramidal method, there will be occurrences of wrongly tracked points. This value is fail-safe to exclude large movements even though they are highly unlikely to happen.

At each iteration, an estimation of the transformation matrix is gained. Also, while excluding outliers at each iteration, a new transformation matrix is acquired from the new subset of pixel pairs with respect to the RANSAC, meaning that at each iteration the transformation is improving and refining.

The end results of this function is a transformation matrix M and a matrix of type 'mask' which has 'TRUE' if a point was considered as an inlier.

By multiplying the array of the operator-defined Safety Area polygon with this transformation matrix, each point will be transformed. This way it can be considered that now the new and deformed Safety Area is acquired. Based on the estimated rigid transformation of the tissue, a transformation matrix is calculated that only accounts for the transformation of the pre-selected section of the frame.

$$SA_n = MxSA_{n-1}, \qquad M = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix}$$

Eq. 4.3

where, SA_n is the transformed SA, $SA_{n-1}$is the SA from the last frame, and M is the transformation matrix used to transform SA_{n-1} to SA_n. Bear in mind that SA is an array of vertices of the SA polygon.

The current frame and the Safety Area that has been accordingly transformed with matrix M and is considered the new estimated Safety Area after a minor deformation between two frames, will now be changed from Destination Frame and New Safety Area Points to Origin Frame and Old Safety Area Points. At the next iteration, when a new frame is acquired, it will be the state to which it is desired to estimate the transformation and deformation.

Ideally, the algorithm would stop here. However, this is not the case, because the environment in a surgical procedure is very dynamic and sudden changes. Trackers can and will be lost at almost each new frame. Although using Canny filter allows tracking to be more robust, there are many occurrences that make trackers disappear, such as:

- Partial or full occlusion
- Abrupt lighting change or change of size of projection
- Tool interference.

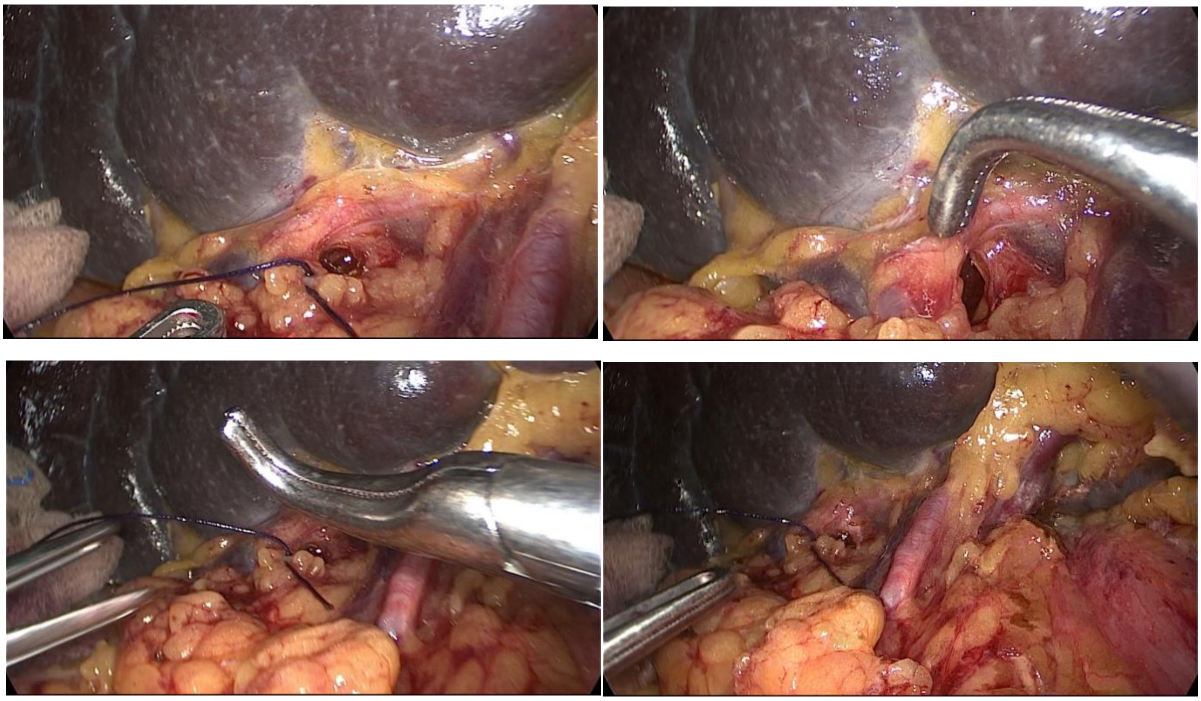


Figure 4.23 1) Partial occlusion; 2) Full occlusion; 3) Tool interference; 4) Change of focus

This implies that a good metric and voting rules for tracking failure need to be made. The first intuitive one is to save the amount of trackers initialized, and check if the number of live ones has dropped below 70%. However, this simple type of failure check has faults. One of them is when the number of lost trackers is small, but the ones which are still alive are densely situated in one part of the SA region. The subroutine responsible for deciding if a tracking failure has happened will check the requirement of having 70% trackers alive, and it will decide that no failure has happened. However, the tracking dynamics will be wrong because the densely populated part will dominate, shown in *Figure 4.24*. Estimating a transformation matrix in this situation will result with a very precise estimation but only for that densely populated part. This is a very bad situation because the other end of the ROI will diverge instead of having a stable transformation, and it will also be erroneously considered as trustworthy.

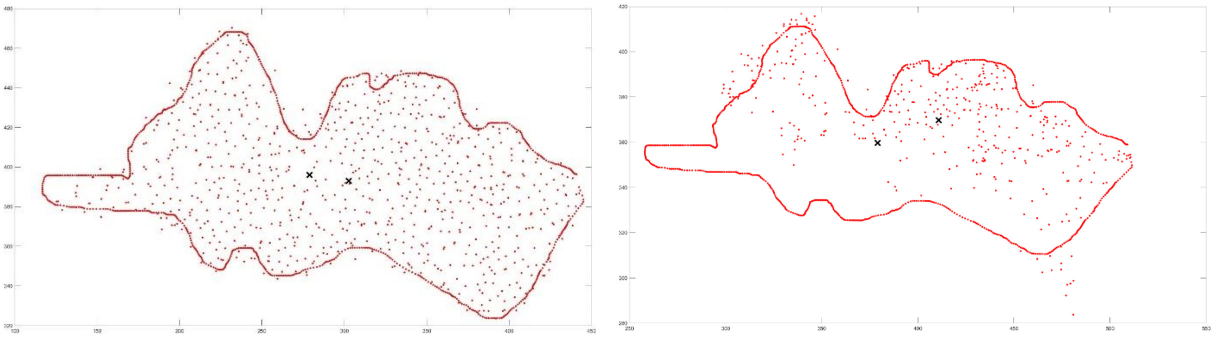


Figure 4.24 On the left, a good uniform distribution of descriptors. On the right, an upper right corner piling up of descriptors

For this purpose a new metric is needed, one for uniformness of distribution of points in an irregular 2-D space.

Evaluation of uniform distribution of points inside a non-regular form is a very difficult task. However, an exact solution for this is not needed. For this particular application, in this thesis a dispersion of trackers inside the AC is considered uniform when the Centroid of the AC polygon and the Centroid of the set of trackers are below a certain Δ value. This Δ is dependent on the size of the frame and also the area covered by the AC. The centroid is calculated as the mean value of all vertices (for AC) or all points in the set of trackers, for x and y respectively. Or, mathematically:

$$C_{AC} = (mean_{AC,x}, mean_{AC,y})$$

Eq. 4.4

$$C_{set} = (mean_{set,x}, mean_{set,y})$$

Eq. 4.5

$$\delta = \sqrt{\left(mean_{AC,x} - mean_{set,x}\right)^2 + \left(mean_{AC,y} - mean_{set,y}\right)^2}$$

Eq. 4.6

$$\Delta = f(size(frame), area(AC))$$

Eq. 4.7

$$\begin{cases} \delta < \Delta, & uniform \\ \delta \geq \Delta, & not\ uniform \end{cases}$$

Eq. 4.8

If these two tracking failure metrics do not give a signal that tracking re-initialization is needed, the algorithm continues to work in tracking mode. Before the new cycle starts, the confirmation that no failure has happened requires that the current AC definition needs to be checked if it fits to replace any model of the Buffer. The buffer was defined in Chapter 4.3, but the updating protocol will be now explained in detail.

At each successful tracking cycle, KAZE detector is applied to the area inside of the current AC definition. Once done, for this array of KAZE features it is calculated to see how many detected features there are; the average value of the strongest 40 responses.

$$KAZE_{responses} = sortbyresponse(KAZEfeatures)$$

Eq. 4.9

$$Avg.Resp = \frac{1}{40} * \sum\nolimits_{0}^{39} KAZE_{responses}$$

Eq. 4.10

Newer AC models are always better than old, and at each step the new AC definition received from tracking a model candidate for updating the Buffer (see *Figure 4.25*). If the number of detected features in the model candidate is more than 85% of the model in place $Temp_{21}$, than it will take its place. Or, if the number of detected features is more than 75% of the model in place $Temp_{22}$, than it will take its place. And lastly, if the number of detected features is more than 65% of the model in place $Temp_{23}$, than it will take its place. The first model of the Buffer (at position Buffer[0]) is the initial AC definition and therefore never changed during runtime.

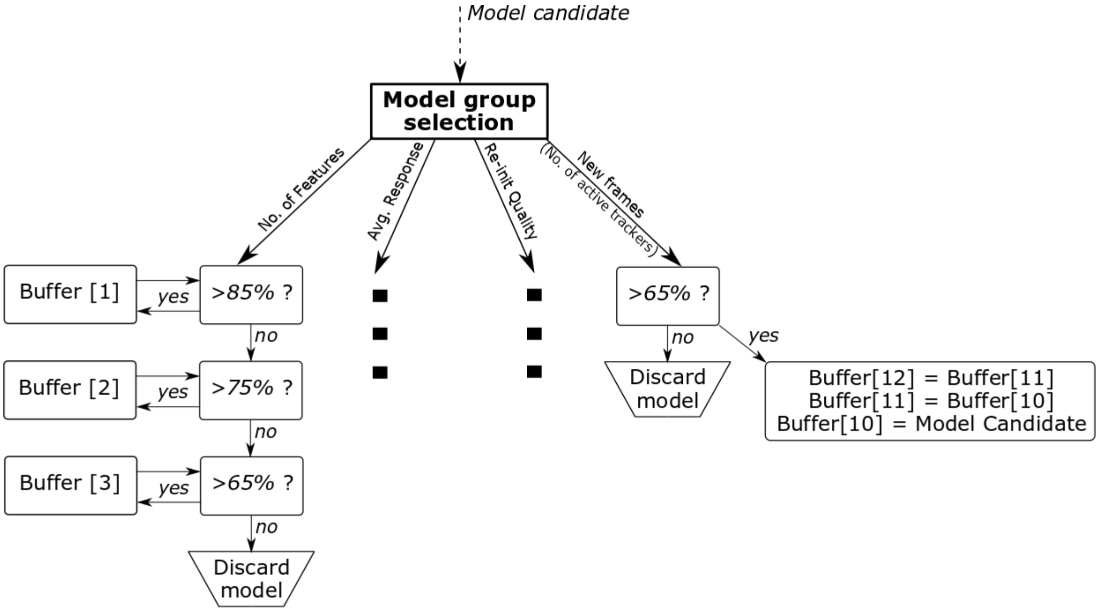


Figure 4.25 Buffer update scheme

Same principle goes for the rest of the Buffer groups, with exact same percentages. Only difference is that, $Temp_{31,32,33}$ is updated regarding response of features, $Temp_{41,42,43}$ is updated regarding quality of re-initialization (this is not updated when tracking). $Temp_{51,52,53}$ is updated regarding newest frames with more than 65% active trackers, where the oldest of the three is updated to the newest model candidate.

If one of the two tracking failure metrics gives a value that is not acceptable for trustworthy tracking, it means that the region needs to be redefined if possible. By redefining the region, it is meant that the SA which was initially given by the operator needs to be redrawn and all the features that are tracked need to be re-initialized. This phase is called ‘Re-initialization’.

4.5. Re-initialization

The procedure for re-initialization is a complex and intricate procedure, which will be separated in three general sub-chapters: Localization, and Redefinition and refining. A flow chart of the workflow of this phase is presented in . The current frame received by the video stream is considered in this part as the input, to assess the location and description of the AC. Each matching applied is between a model in the Buffer and the current frame.

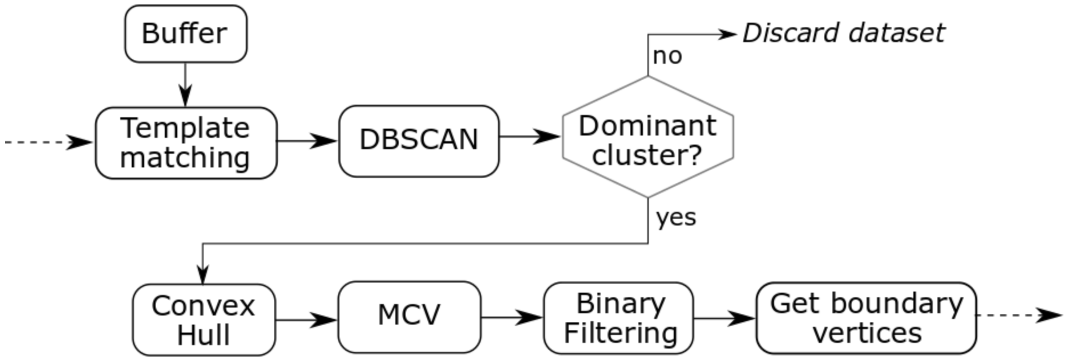


Figure 4.26 Flow chart of Re-initialization phase

4.5.1. Localization

The previously defined model buffer, filled with models of the tissue of interest, is the primary source of information in this part. Each of the models is being matched to the current frame, which is the one that has been detected to have given a tracking failure. Each model will give a different set of matched features with this current frame.

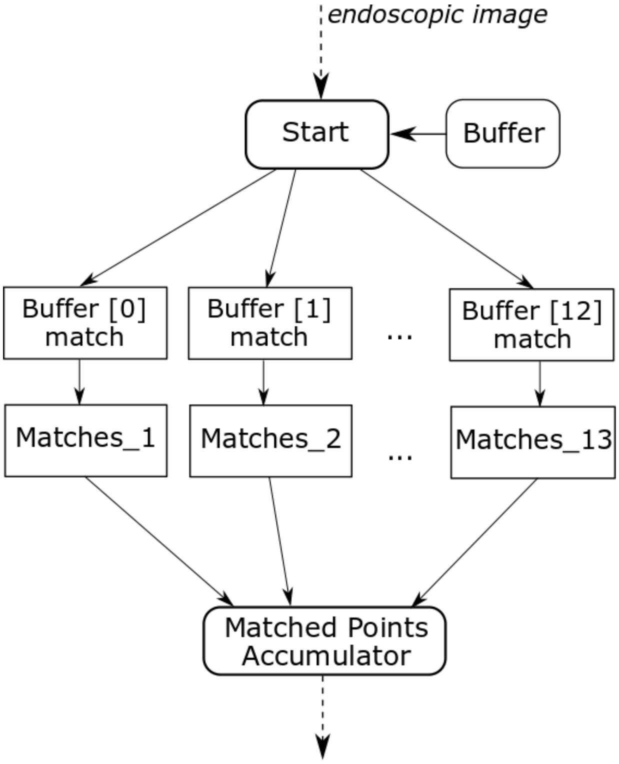


Figure 4.27 Flow chart of template matching

The total number of models to be matched is 13, as defined for the model buffer previously. This is done in parallel processes, and each of these 13 procedures gives a set S with different amounts of matched features (keypoints). Once all of them are gathered, they are stacked in one big array called Accumulator (*Figure 4.27 Flow chart of template matching*).

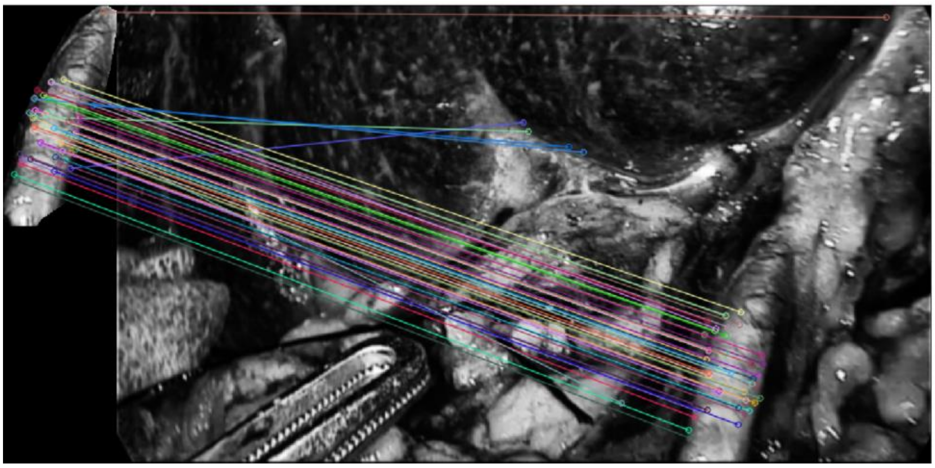


Figure 4.28 Single template matching

Feature matching between a template and the current frame is done using the Brute Force approach (Chapter 3.5), as can be seen in *Figure 4.28* where matched points are connected by lines, matched points are good enough for localization of the area but will not describe the region fully, which is the motivation for using multiple templates for matching with different characteristics. All matching is done on grayscale images, but for ease of understanding the data will be displayed on color images.

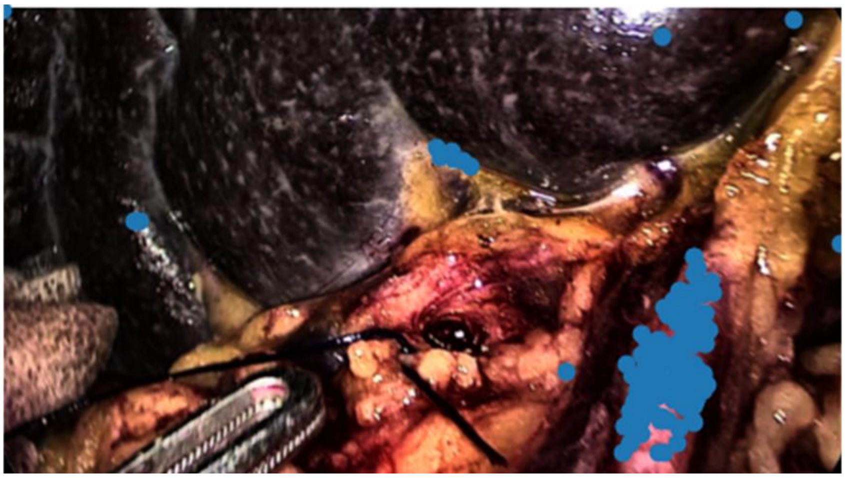


Figure 4.29 Features matched from all available templates

In *Figure 4.29*, all matched points have been graphically presented by a blue filled circle. This is a graphical representation of the points found in the Accumulator. The vast majority of them are directly positioned on the tissue, while there are others dispersed through the image by mismatching. In order to discard these mismatches, the use of a density-based algorithm is proposed, in particular Density-based spatial clustering of applications with noise (DBSCAN), as explained in Chapter 3.7.1. Density - based spatial clustering of applications with noise (DBSCAN)

It is most likely that a big portion of the matched destination points from the current frame will appear multiple times and it is of great importance to keep all of them. They may appear multiple times at the matching between the model and the current frame ($[x_i,y_i] \in Si$), and also the same points may appear even in different model matchings ($[x_i,y_i] \in S1 \cap S2... \cap S13$). Although having the same point multiple times may seem as non-beneficiary to the general knowledge at this stage, it will present itself as very important in the next step which is the use of DBSCAN. The minPoints and radius parameters were identified by series of experiments, and the results were correlated to the size of the image and the size of the tracked tissue. They are adaptive and have been calculated as a function of the input video stream's width and height, and also the area covered by the AC definition. The most common values for MinPts and radius were 35 and 25, respectively.

The large set of data in the Accumulator of all matched points is fed into the DBSCAN. It requires two parameters in order to function, one is the search radius and the second one is the required minimum (of neighboring points), as discussed in Chapter 3.7.1. Density - based spatial clustering of applications with noise (DBSCAN) The aforementioned copies of same points will come in handy here. The reason for this is that if a point is matched multiple times, it will be considered with great trust that this point is an available feature that is found in the tissue of the current frame. In addition, the distance between these copies is 0 and according to DBSCAN it will be considered as a core point with great trust. This will happen because the Euclidean distance between two points with same exact coordinates is 0 and is always less than the search radius defined for DBSCAN. According to DBSCAN clustering theory, if there is number of points larger than the required minimum provided through a parameter within a radius also given through a parameter, than those points can be considered as core points. This acts like a small voting scheme for including or discarding certain points.

Having these copies is also very likely to happen in the region matched from the model. This helps DBSCAN to get rid of the mismatched outliers which are dispersed through the image. The algorithm for matching used is of Brute Force type, but even this thorough approach will have wrong conclusions.

Once DBSCAN has returned the clusters found, the largest one is chosen. The reason why it is always acceptable to consider the biggest one, is again because of the matching copies, which provide the possibility for this cluster to have amount of points which is multiple times larger than the other clusters found, thus making it a dominant cluster. This way of thought was motivated by the Hough accumulator, which acts in a similar manner by voting certain points or lines as winners if during the analysis they appear more times than others in continuity. This implementation of the Hough accumulator is using Hough transform which is an algorithm used for line detection in images [39]. The resulting dominant cluster provided by DBSCAN is graphically presented in *Figure 4.30*, where each point of the cluster is shown as a filled blue circle.

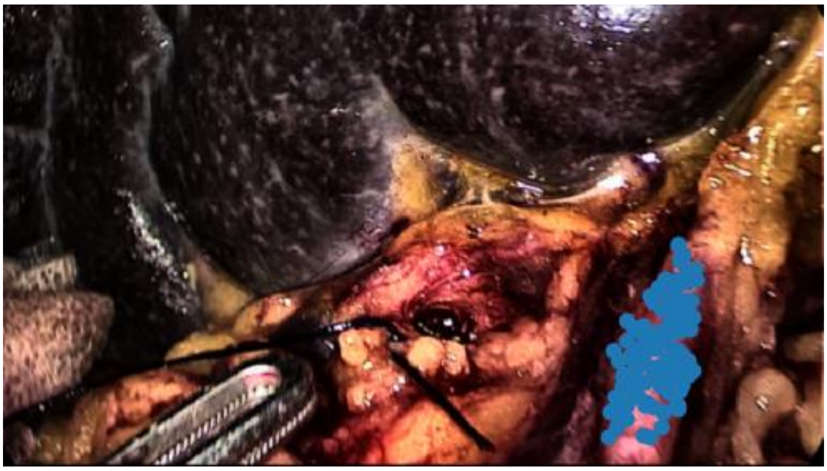


Figure 4.30 DBSCAN output cluster

One way of creating a robust binary mask out of this data set is by finding the outer boundary and then filling it with white pixels. Most common approach for this is by using Convex Hull (Chapter 3.9.1. Convex Hull). However, Convex Hull has a big problem with finding boundaries if the distance between points which are candidates for boundary vertices have 0 distance between them. For this, a set of only unique values is required. This is achieved by checking for each point if there is another point with the same coordinates, and removing each occurrence of it after the first one has been found. Hence, achieving a set of unique values.

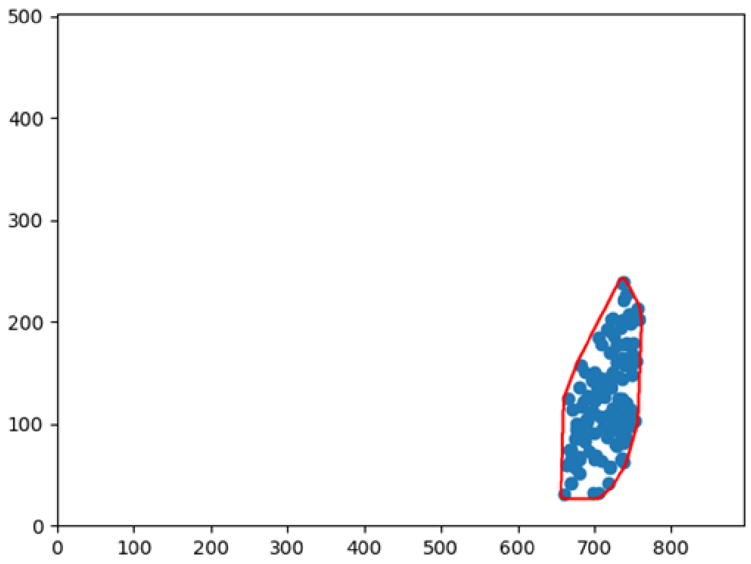


Figure 4.31 Convex Hull of DBSCAN cluster

4.5.2. Re-definition and refining

Using the Morphological implementation of Chan-Vese (MCV) active contours (Chapter 3.9.2. Active contours), which belongs to the group of Active Contours Without Edges (ACWE), it is possible to expand the region that has been defined by Convex Hull. For this reason, the initial mask for active contours will be exactly the mask acquired from the boundaries defined through Convex Hull. The energy function, which MCV needs to minimize has two coefficients, with lambda1 working in favor of expanding while lambda2 in favor of contracting. It has been shown experimentally that the cluster which localizes the tissue is usually inside the tissue, so a mask based on this will need to expand. MCV will be initialized with parameters lambda1 = 8 and lambda2 = 0.4, with the difference of exactly 20 times in order to prefer expansion. MCV is applied on the first channel of HSV color space representation of the current frame. It is used because the Hue channel (first channel of HSV) provides a good distinction between different shades of a color, and especially between different colors.

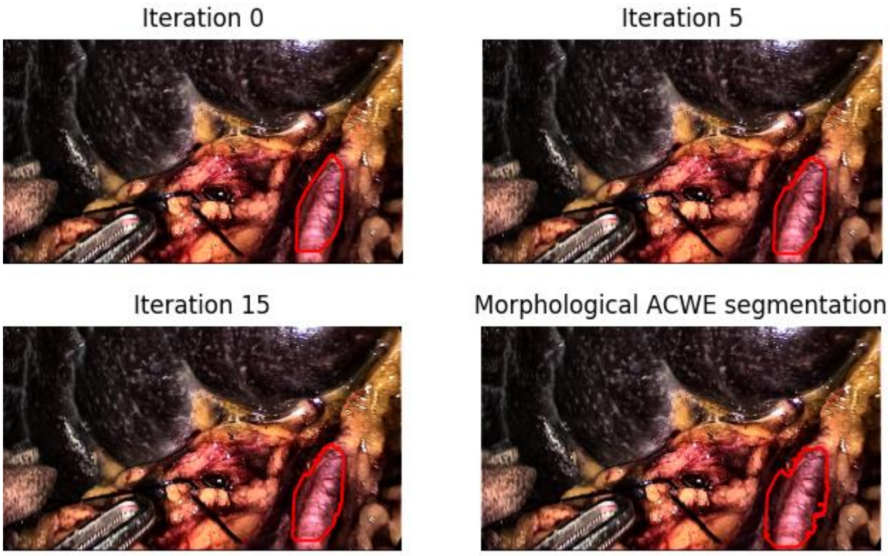


Figure 4.32 Morphological Chan-Vese evolution at different iterations

Since it is largely affected by color, noise and morphological structures due to the minimization of the residual equation, two outcomes are possible:

- Well posed contour with a dense core and very small sparse 'spikes' around it.
- Badly posed contour with a small core and big sparse 'spikes' around it.

If the region MCV wants to describe is from the same object it will expand in a stable manner and will not grow sparsely in multiple directions

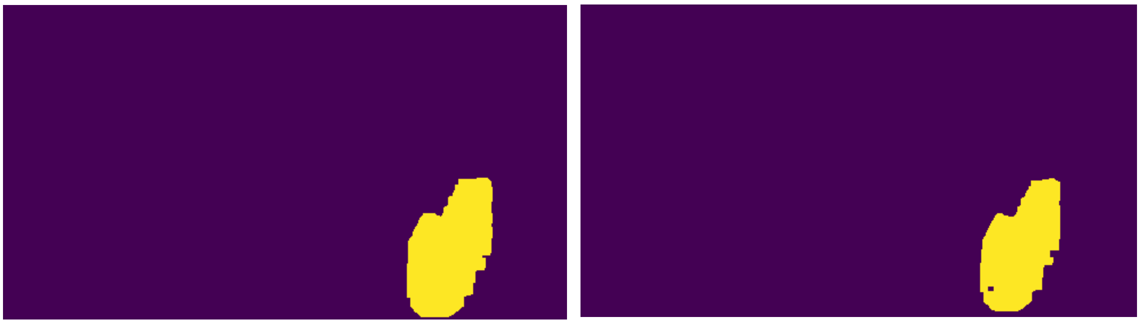


Figure 4.33 On the left, the mask created upon MCV's method is implemented. On the right, the mask acquired after binary filtering operations.

Using Morphological Opening and erosion, these spikes will be removed, and small gaps inside will be filled. Aside from refining the contour, it will be useful to have a reliability index as measure for this new contour. Assuming a badly posed contour will have these spikes which are vastly lost after erosion, Jaccard score can be used. Again. Jaccard similarity score will calculated over two sets. An intersection between two sets will be a coordinate where both have a logical ‘one’. The three conclusions to be drawn from Jaccard index are:

- If the sets before and after the binary filters are very similar, it will mean it is well posed (*Figure 4.34* on the left) and the Jaccard score will be very high (0.85-1).
- If it is badly posed, erosion will remove a large portion of these spikes and Jaccard score will be low (<0.6)This will be considered as unacceptable for redefinition of AC (*Figure 4.34* on the right)
- Everything between 0.6 and 0.85 will be shown in a circle, while changing color shades from red for <0.6, through yellow to green for >0.85.

This classification, with the color shades described, is used to provide the surgeon with online information on the tracking quality and status.

At each point, the operator can see this metric and decide if the re-initialization is good enough. If it is not, a manual switch can be used to force a re-initialization.

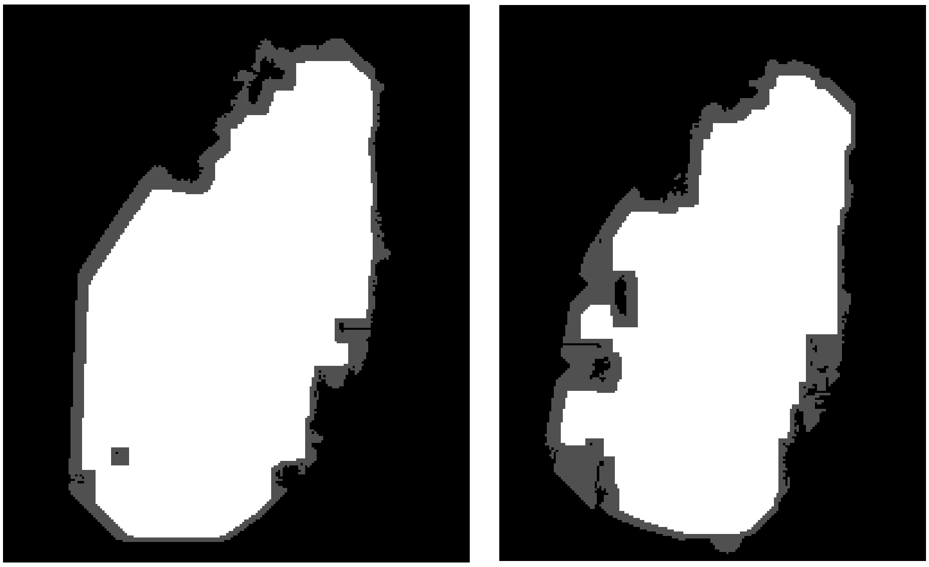


Figure 4.34 The gray zone is the mask before erosion, and in white after erosion. Jaccard score on the left is 0.83 (good estimation), and on the right Jaccard score is 0.68 (bad estimation)

In the case of a reliable description of the AC, the boundary vertices of the contour are calculated, and they are returned to the tracking algorithm, as seen in *Figure 4.35*. The tracking algorithm instead of using the last known SA, uses the new form of the SA retrieved by this redefinition phase and re-initializes all parameters as in the very first cycle, as it was described in the first sub-topic from this chapter. Once re-initialization is finished, the new AC is checked if it should be used to update the model buffer, as discussed at the very end of the previous sub-chapter (Chapter 4.4)

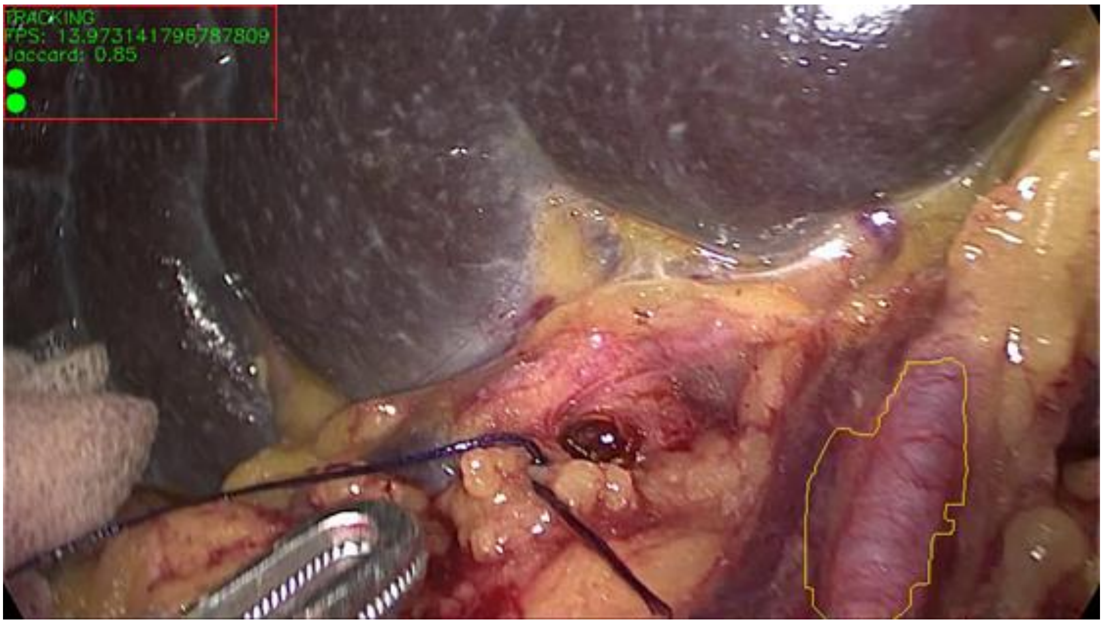


Figure 4.35 Estimated new SA definition is used in tracking

In *Figure 4.35* there can also be seen five lines of information, selected with a red bounding box. The meaning of each line provides a different type of information provided to the surgeon and their meaning is respectively:

- Phase, and it can be TRACKING or RE-INITIALIZATION
- Computational speed, in HZ
- Jaccard score of current re-initialization
- Colored circle for conveying a visual AC Reliability index (red for not reliable, through yellow for semi-reliable, to green for reliable)
- Colored circle for tracking reliability index as measure of active trackers available (red for not reliable, through yellow for semi-reliable, to green for reliable)

4.6. 2D to 3D projection

As discussed in the beginning, this method is completely implemented in 2D. However, when implemented in a surgical robot it needs to provide 3D information on how the area of the AC has transformed. For this, an approach to transform 2D points to a 3D point cloud is used.

The developed GUI, SmartSURG, is used for definition of the point clouds from which the AC is modelled. The used technique (*Figure 4.36*) deploys a virtual plane placed between the observer (viewpoint) and the point cloud (3D scene). On this plane, using a mouse or a master robotic arm, the surgeon can draw an area to delimit the AC (*Figure 4.37*).

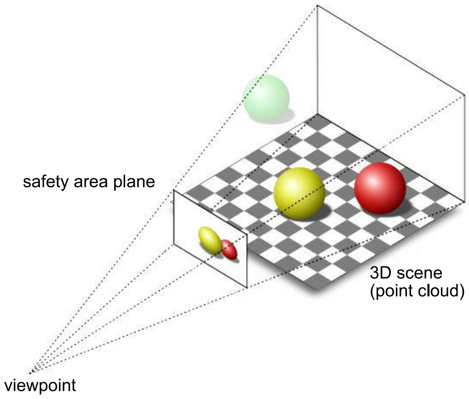


Figure 4.36 Structure of the projection of 3D scene on the plane where safety areas are drawn.

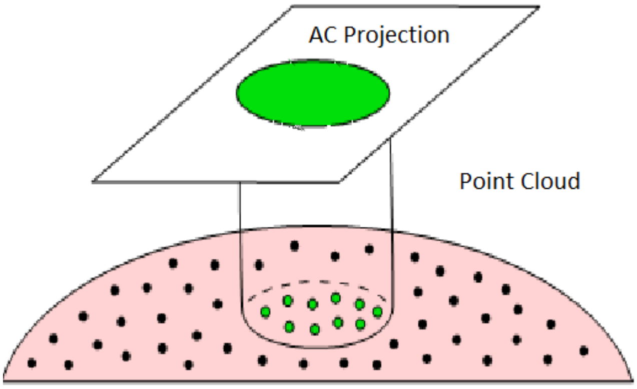


Figure 4.37 Projecting Point cloud on a 2D plane where the AC contour is then drawn (polygon bounding the green area) and projecting the points inscribed in green area back to the 3D scene, identifies the points belonging to the AC as a point cloud (green spots).

The projection of the points included in the safety area projected on the plane back to the 3D scene identifies the points belonging to the AC as a point cloud (i.e. defines the AC to be rendered visually and haptically).

This step is of fundamental importance since the selected points will be used to reconstruct the 3D surface of the anatomical structure that will form the AC volume, from which the surgical tool will be steered away. The problem is defined in the 3D space, so the primary challenge consists in identifying the 3D points confined within a 2D polygon seen from the perspective point of the user who drew the area.
To solve the problem, it was decided to transform the 3D formulation into a 2D one. Instead of projecting the AC on a parallel plane beyond the scene to create a polyhedron, it turned out to be more convenient to project the point cloud on the same plane of the AC (*Figure 4.36*). Following this step, by applying a 2D ray-casting algorithm, it is possible to identify the points inside the polygon and then project them back to their original position to obtain the points of the safety volume's surface.

This solution for projecting the 2D polygon of AC to 3D is being used continuously to provide a 2D-to-3D projection of the 2D defined AC polygon from the method proposed in this thesis to the reconstructed 3D image from the endoscope.

CHAPTER 5
Experimental results

In this chapter, the experimental results used to validate the proposed method will be presented. Several metrics and statistical methods validation for of the method will be briefly discussed, as well as description of each experiment and the purpose it holds in the experimental campaign. The videos used in all experiments are pre-recorded and simulate a different aspect of a real surgery and potential situations that may arise. The videos are recorded using endoscopes, and once processed the data acquired while processing will be used to assess the performance of the algorithm.

.

5.1. Experimental setup

5.1.1. Computing setup

The conducted experiments were done using a workstation provided by NearLab laboratory. The algorithm developed to execute the proposed method was built, compiled and tested on the same machine. This workstation is equipped with:

- Intel i9 9900k @ 3.60 GHz with 8 cores / 16 threads and max turbo speed of 5.00 GHz
- 16 GB DDR4 RAM at 2133 MHz
- Quadro M5000 GPU with 8 GB of GDDR5 memory

The machine was running on Ubuntu distribution version 16.04.6 LTS. The complete programming was done in Python 3.6.8 with Ubuntu 16.04.6 compatible versions of the OpenCV, numpy and scikit-learn libraries.

During this experimental campaign, the proposed method was not implemented on the dVRK.

5.1.2. Monitoring setup

This method is purely visual and does not depend on any other input from the robot except for what is provided by the camera, which in this application is the endoscope.

Richard-Wolf stereo ENDOCAM Epic 3D HD endoscope

This endoscopic system includes two high definition (HD) cameras on the tip of the endoscope shaft to provide a fully HD image with three-dimensional depth (*Figure 5.1*). Human beings use their two eyes for orienting in a space, and it is why ENDOCAM Epic 3DHD also uses two cameras to generate an image of a space.

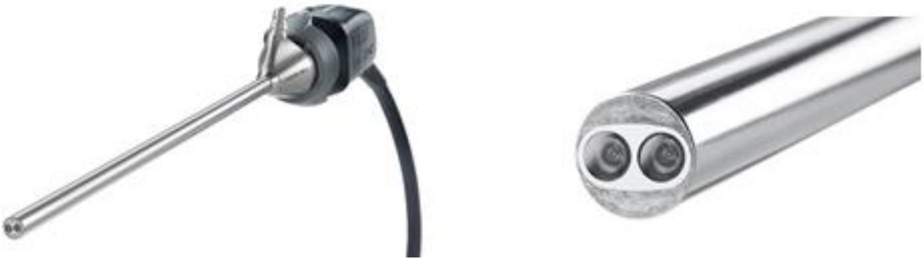


Figure 5.1 Richard-Wolf ENDOCAM

All experiments were done using prerecorded video sequences as input to of the method proposed in this thesis, in order to test and assess specific aspects of its behavior and performance. The used video datasets are classified in two groups:

- Controlled environment: 4 datasets,
- Real world surgery: 1 dataset.

The dataset from an actual surgery used was provided by the TrackVes repository [40] and it is not disclosed what type of endoscope it is used. However, it is safe to assume that it is a da Vinci RAS surgery and the endoscope is a proprietary device. In *Table 3 Endoscopes used in experiments*, each experiment to be done is identified by the endoscope used in it.

Table 3 Endoscopes used in experiments

Experiment	Endoscope
Controller environment 1	Richard-Wolf stereo ENDOCAM
Controller environment 2	Richard-Wolf stereo ENDOCAM
Controller environment 3	Richard-Wolf stereo ENDOCAM
Controller environment 4	Richard-Wolf stereo ENDOCAM
Real surgery	Not disclosed

5.2. Methodological overview of statistical metrics

To assess the algorithm behavior and performance, and to be able to compare to other works, several approaches were used. In this sub-chapter these metrics will be discusses, as well as the theory behind them.

5.2.1. Visibility of AC area

Regarding visibility of AC, as proposed by Penza et al. [25] where Safety Area is used to annotate the area inside the AC, the video may have four different (see *Figure 5.2*) visibility situations:

- Safety Are Visible (SAV)
- Partial Occlusion (PO)
- Total Occlusion (TO)
- Out of Field of View (OFV)

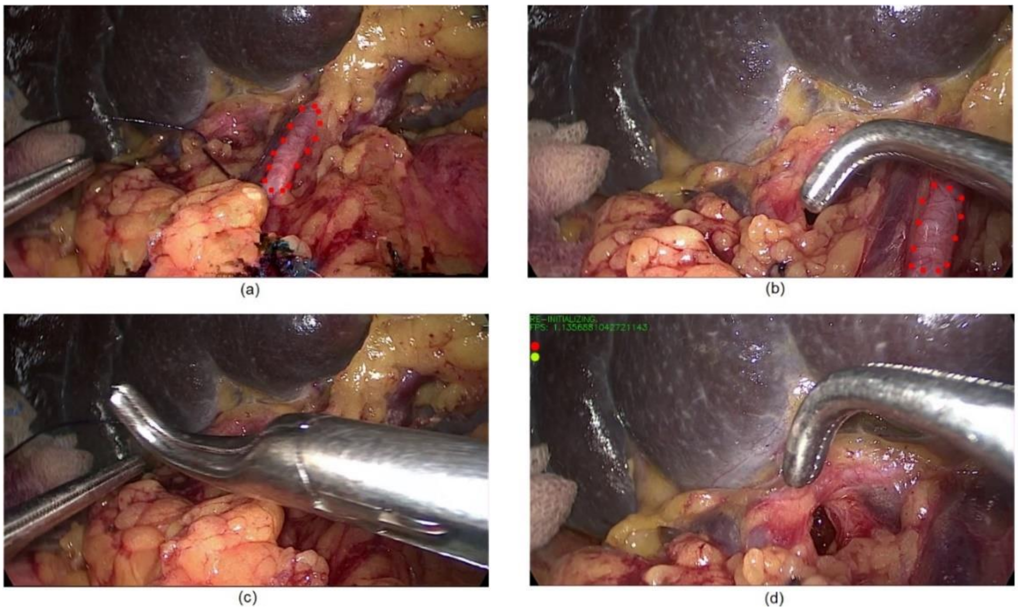


Figure 5.2 Occlusion types.(a)SAV; (b) PO; (c) TO; (d) OFV

5.2.2. Ground truth labeling

Ground truth (GT) is a term used in statistics that means checking the results of the statistical estimations for accuracy against the real world. It is a term borrowed from meteorology for independent confirmation at a site, for information obtained by remote sensing.

These tests allow researchers to refine their algorithms for better accuracy.

For instance, in this thesis the GT that is used for determining the accuracy of the method is a polygon drawn at an interframe step of k frames. This means that the GT will be defined at each kth frame. The region that will be drawn by the user is relative truth and does not always represent the absolutely correct AC definition to be expected, therefore it may not always be the best description of the region to be assessed for accuracy.

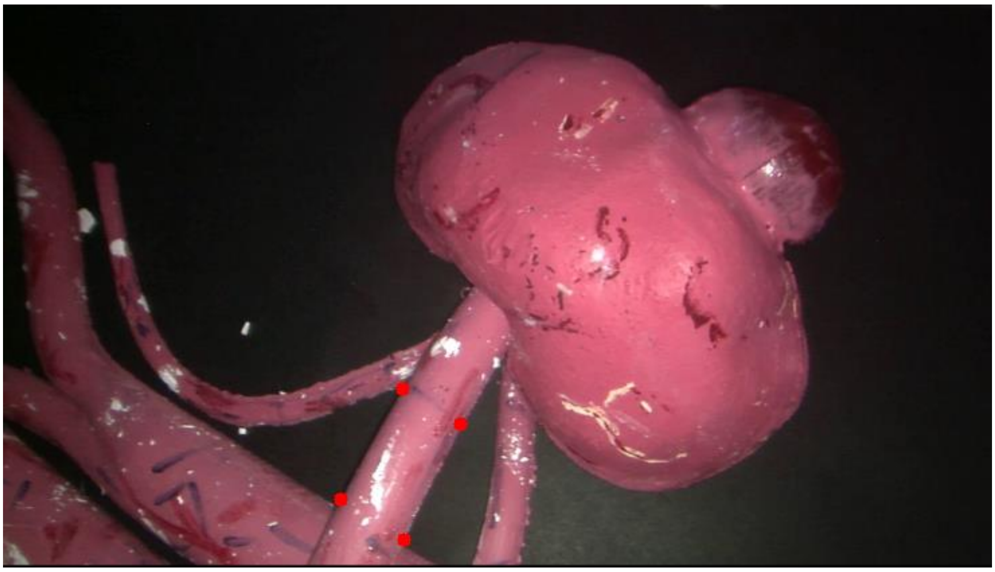


Figure 5.3 GT labeling

This is why similarity coefficient of 100% is never to be expected, but rather everything above 80% similarity will be considered as “equal”.

5.2.3. Jaccard similarity score

Jaccard similarity score (JSS) is a metric used to assess how similar are two sets. It will be used in the experimental results of this thesis for assessing the performance in two different aspects:

- Re-initialization reliability index,
- Quality of tracking.

Re-initialization reliability index

This index (see Chapter 3.7.2. Jaccard similarity score) shows how good is the region proposed by Morphological Chan Vese (MCV) in the phase of re-definition while re-initializing. JSS is calculated as the similarity score between S_1 and S_2, where S_1 is a binary mask over the region proposed by MCV to be the new AC, and S_2 is a binary mask over the region that is received after Morphological Opening and erosion has been applied to MCV. A low Jaccard similarity index will mean that there were many sparse elements on the boundary of the region, meaning it was affected by noise (*Figure 5.4*, right image).

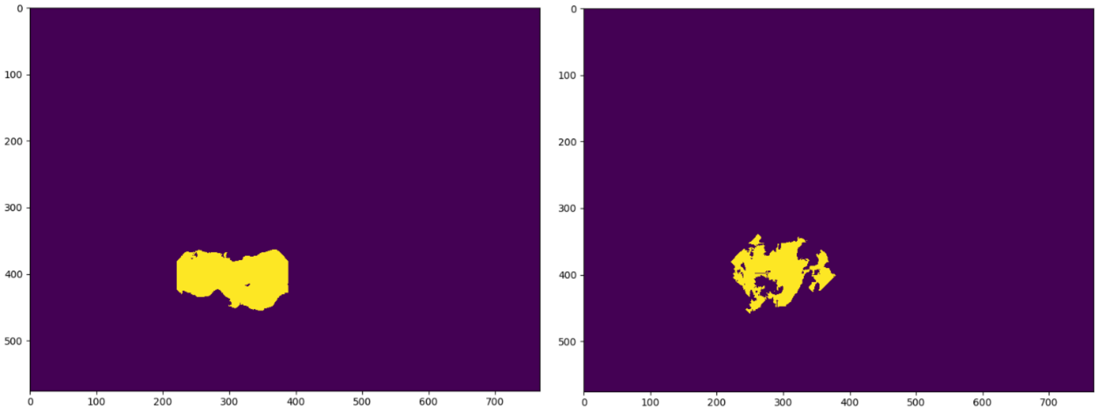


Figure 5.4 MCV output. On the left, a well-posed contour. On the right, a contour largely affected by morphological noise

Quality of tracking

In quality of tracking, Jaccard is the set similarity score between the AC acquired by the proposed method and the AC definition provided by GT. Three outcomes are possible:

- 0, if there is no common element. Which leads to the conclusion that False Positive, False Negative or mismatching has occurred.

- Low value (<0.4) means that the AC redefinition is ill-posed.

- High value (>0.4) means that the AC redefinition is well-posed.

Jaccard similarity score is a *dimensionless quantity*, ranging from 0 for no intersection at all, to 1 for complete overlap of sets.

5.2.4 Precision, Recall and Accuracy

Building a statistical or mathematical model is an integral part in the fields of computer science, statistics and patter recognition. Once built, this model needs to be put through tests to assess how good it is, and if it is possible to improve it in some way. The evaluation of the model built is the most important task of a data science project to delineate how good the predictions are. Model based estimations can be of several types, and it is best described by using the so-called ‘confusion matrix’ (see *Table 4*)

Table 4 Confusion matrix

		Predicted Class	
		Class = Yes	Class = No
Actual Class	Class = Yes	True positive	False negative
	Class = No	False positive	True negative

True positive and true negatives are the observations that are correctly predicted and therefore shown in green. The goal is to minimize false positives and false negatives, so they are shown in red color.

In order to explain the mathematical background behind these concepts, first some abbreviation must be defined.

TP – Number of True Positive occurrences, is how many times the algorithm has detected and defined some AC correctly.

FP – Number of False Positive occurrences, is how many times the algorithm has detected the AC even though it is not actually visible at that moment.

TN – Number of True Negative occurrences, is how many times the algorithm has not detected the AC because it is not actually visible.

FN – Number of False Negative occurrences, is how many times the algorithm has not detected the AC even though it is visible.

However, AC estimations are not binary (True or False) but rather arrays of vertices describing an area, and this means that there needs to be a definition to what is True Positive.

In order to consider something as True Positive Jaccard similarity index will be used again. In this case, if the algorithm’s description of the AC and the GT differ by less than a threshold it will be considered as TP. Or, mathematically:

$$\partial = \frac{|AC_{PM} \cap AC_{GT}|}{|AC_{PM} \cup AC_{GT}|}$$

Eq. 5.1

where AC_{PM} is the AC described by the proposed method (PM) and AC_{GT} is the ground truth AC.

If ∂ is over a predefined threshold ε, the occurrence of TP will be accounted for.

$$O_{TP} = \begin{cases} 1, & \partial \geq \varepsilon \\ 0, & \partial < \varepsilon \end{cases}$$

Eq. 5.2

where O_{TP} is the occurrence of True Positive point. In the experiments conducted in this chapter, the ε values which will be used are : 0.3, 0.5, 0.7 and 0.8.

Accuracy

Accuracy is the most intuitive performance measure and is simply the ratio between correctly predicted observations and the total observations. Accuracy is a great measure but only when the number of false positives and false negatives is almost the same. If not, other measures should be considered. Mathematically, Accuracy is calculated as:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Eq. 5.3

Precision

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. Mathematically, precision is calculated as:

$$Precision = \frac{TP}{TP + FP}$$

Eq. 5.4

Recall

Recall is the ratio of correctly predicted positive observations to the all observations in actual class – yes from *Table 4*.

While, recall is calculated as:

$$Recall = \frac{TP}{TP + FN}$$

Eq. 5.5

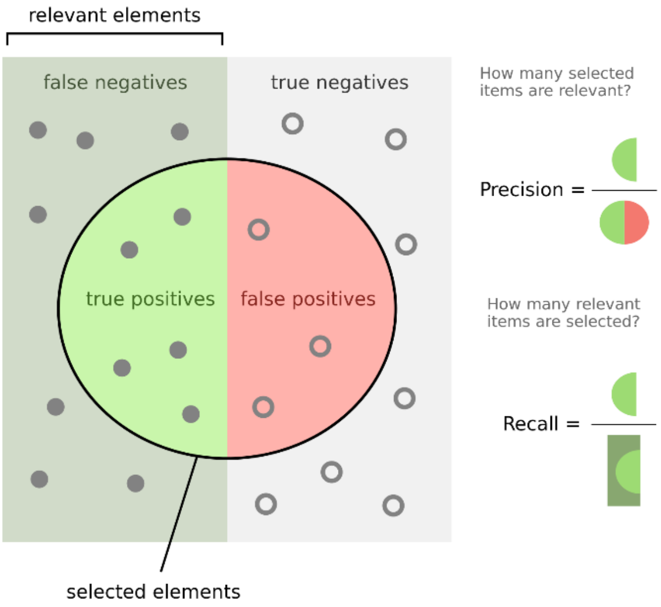


Figure 5.5 Precision and Recall definitions

Accuracy, Precision and Recall are dimensionless quantities, ranging from 0 to 1. In both cases, 0 is considered bad performance, while 1 is considered perfect performance.

5.2.5. F1 Score

In most problems, priority can be given to maximizing precision, or recall, depending on the type of problem. But in general, there is a metric that can take into account both precision and recall, and it can be used to maximize this number to make the method more accurate. The generic form is called Fβ score, where β is a parameter used to prioritize either precision or recall. The mathematical formula of Fβ is :

$$F\beta = (1 + \beta^2) * \frac{Precision * Recall}{\beta^2 * Precision + Recall}$$

Eq. 5.6

A special case of Fβ is F1 score, which is simply the harmonic mean of precision and recall. Setting parameter β to 1 gives equal importance to both precision and recall.

$$F1Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Eq. 5.7

F1 Score can also be considered as the weighted average of Precision and Recall and is considered the measure which encapsulates most outcome. It is often more useful than accuracy especially when the class distribution is uneven.

Same as Precision and Recall, F1 Score is a dimensionless quantity, ranging from 0 to 1. A F1 score of 0 is considered bad performance, while 1 is considered perfect performance.

As proposed by Wu et al. [40], it is widely used as tracking benchmarking measure.

5.3. Validation of the method in controlled environment

The first four experiments are done using molded objects based on 3D computer models of kidney, tumor, renal vein, renal artery, inferior vena cava and descending aorta (see Figure 5.6, on the left). The 3D printing consists of printing of the molds using a PLA 3D printer (see Figure 5.6, on the right). PLA is preferred due to its non-toxicity and extrusion ease. This material is manufactured using 40% in volume of Prochima Prolastix silicon base, 40% of cross-linking agent and 20% silicon oil to obtain correct anatomical density. They are molded with a silicon-based material which is soft to the touch and can mimic deforming behavior similar to the real situation.

The complexity in this type of experiments is in the lack of texture and the object having an almost homogenous surface of same color hue. Unlike real organs, this object lacks details such as bumps, vessels and other irregularities that are unique and can be particularly distinguished from other parts of the object and environment.

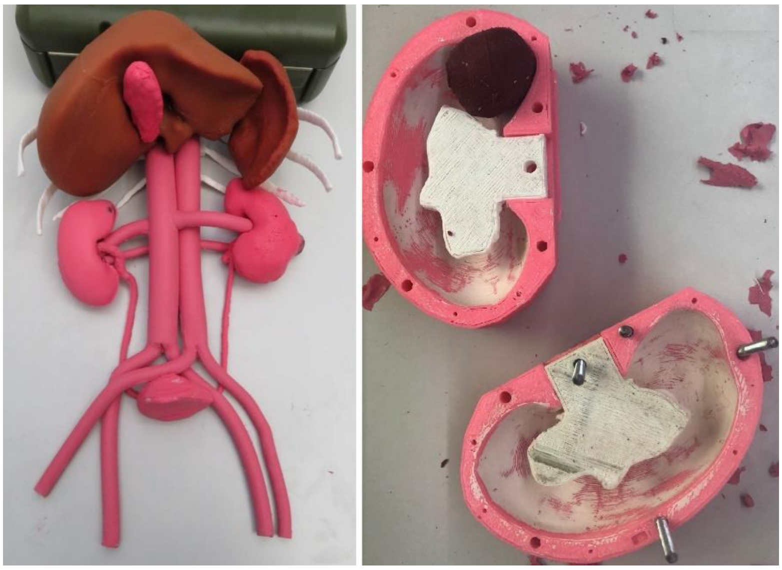


Figure 5.6 On the left, Phantom kidney setup. On the right, kidney molds

Some blood vessel-like details were drawn on the kidney phantom manually with a color marker.

The video sequences used in this group of experiments are recorded using the Richard-Wolf stereo ENDOCAM (Chapter 5.1.2. Monitoring setup).

5.3.1 Result analysis of experiment 1

The first experiment is done using the proposed method with video input from a dataset which has basic movement such as *rotation* and *translation* but does not have any occlusions nor deformations. Here only the kidney phantom with a tumor will be used, without any of the aforementioned additions. This test is done in an enclosed and controlled environment, using the kidney phantom over a dark background. Having a dark background provides a test which is not affected by specular noise and can show the level of reliability of the method in such stable environment. This, however, is not applicable to the real surgery environment which is highly dynamic.

In this experiment, the tracked AC is always visible. The table concerning AC area visibility is shown in *Table 5*.

Table 5 AC area visibility coverage in Experiment 1

SAV	PO	TO	OFV
100%	0%	0%	0%

Figure 5.7 Frame from Experiment 1

This dataset has a resolution of 1840 x 1044 pixels and is 19 seconds long at 25 Hz, which amounts to a total of 475 frames. The selected AC encircles a region from the tumor artificially applied to the phantom kidney, depicted in *Figure 5.8*.

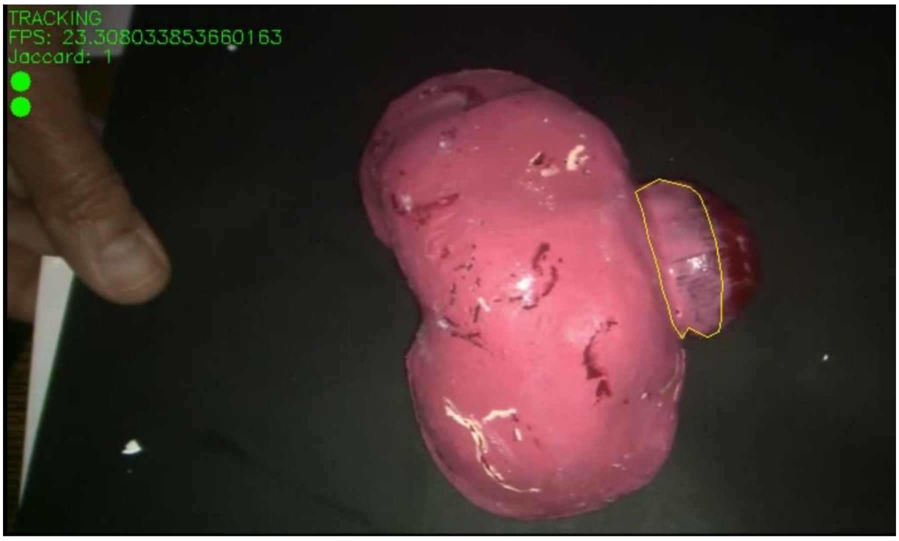


Figure 5.8 Tracked AC on phantom kidney

Precision, Recall, Accuracy and F1 Score

Precision, recall, accuracy and F1 Score will be calculated and compared in order to have some sense of precision, repeatability and false detection, These values, as previously stated (Chapter 5.2.4 Precision, Recall and Accuracy), were calculated by using several thresholds for considering something as True Positive, with respect to the current ground truth. Ground truth is defined at each 50 frames, which for this video results with 9 labeled frames from which the following calculations are acquired.

Table 6 Statistical results for Experiment 1

Threshold	*Precision*	*Recall*	*Accuracy*	*F1 Score*	*False Positive*	*False Negative*	*True Positive*
0.3	1	1	1	1	0	0	9
0.5	1	1	1	1	0	0	9
0.7	1	1	1	1	0	0	8
0.8	1	1	1	1	0	0	4

The results presented in *Table 6* *Statistical results for Experiment 1* are 'ideal', which is not in any way surprising because the dynamics in this experiment are very low. Even with low texture, the edge-based approach in this method for tracking proves itself as reliable. Depending on the threshold set, only the number of TPs changes.

These results only consider if a detection has happened and if it is correct, in a very 'binary' manner. However, it gives no information on how the tracked AC compares to the GT.

Tracked AC quality performance

In *Table 7*, it can be seen how the tracked AC differs from the GT. For this purpose and because AC and GT can be considered as sets, Jaccard similarity score (JSS) is used again to determine how much do the sets differ.

Table 7 AC tracking quality in Experiment 1

	Mean	Standard deviation	Minimum	Maximum
AC quality JSS	0.779	0.041	0.688	0.819

From this Jaccard similarity score graph (*Figure 5.9*), it can be observed that the maximum value is the starting one with 0.82 and the lowest is 0.69. This can be considered a narrow range of 0.13 score value, considering a mean value of 0.78 and standard deviation of 0.041.

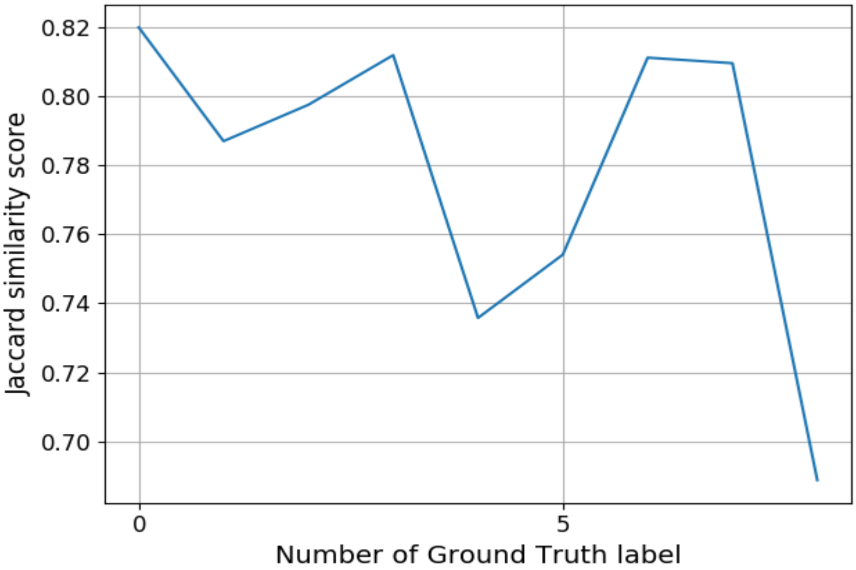


Figure 5.9 AC tracking quality in Experiment 1

There are no clearly visible defects of the AC throughout the tracking process, except for minor transformation difficulties because of the oval geometry of the object, which makes tracking of points difficult when they move out of view due to rotation (projection)*.* From this data it can be concluded that during this experiment, the method has successfully tracked the AC and when compared to the provided GT the Jaccard similarity score has a mean value of 0.78.

Number of active trackers

Even in such stable environment, movement of types translation and/or rotation will cause change in: reflection on the object with respect to the fixed background lightning; moving points out of view and projection differences. Considering this, it is important to track how many trackers are active at any moments. In *Figure 5.10*, the number of active trackers for LK method are shown at all times.

Table 8. Number of active trackers in Experiment 1

	Mean	Standard deviation	Minimum	Maximum
No. of active trackers	209.825	3.168	207	219

In this scenario, the number of trackers falls only to 95% (*Table 8*)of the initialized tracker. Drops of active trackers can be easily correlated to sudden movements which cause an abrupt reflection or projection changes. Since until the end of the dataset, no more than 5% of trackers are lost the re-initialization voting scheme never votes for a re-initialization. The initial drawing of AC is always considered as the correct AC definition, so Jaccard score to that AC definition is set always set as 1 to initialization AC polygon. According to what was previously stated, re-initialization in this scenario never happens and Jaccard score for the AC definition remains 1 during the entire video

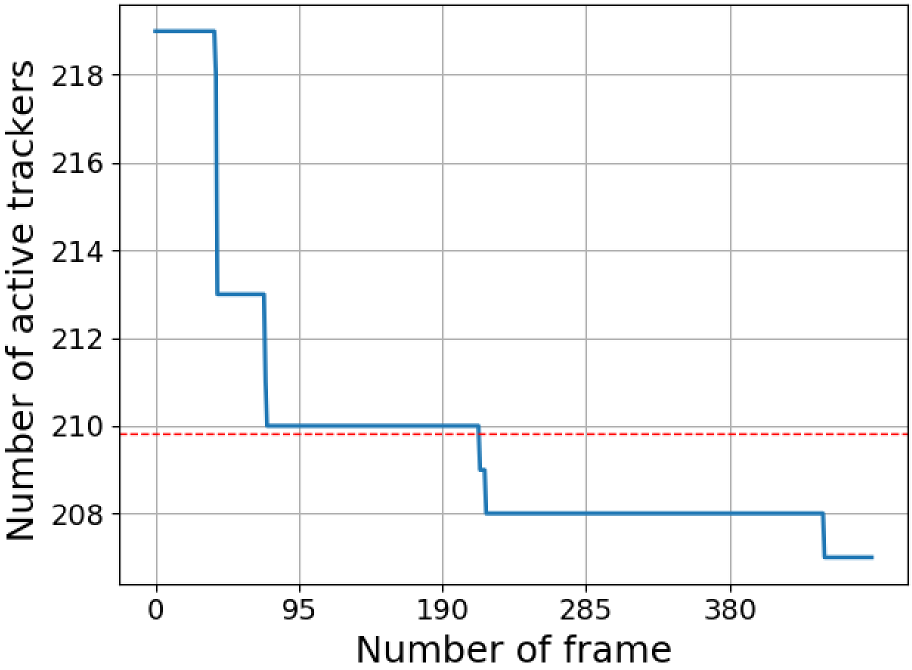


Figure 5.10 Number of active trackers in Experiment 1

Program execution speed

In this experiment, as shown in *Figure 5.11*, the computational speed had an average value 73.505 Hz, with standard deviation of 5.93 Hz, with a minimum of 20.78 Hz and maximum of 87.229 Hz.

Table 9. Program execution speed in Experiment 1

	Mean	Standard deviation	Minimum	Maximum
Program execution speed [Hz]	73.505	5.933	20.780	87.229

These numbers lead to the conclusion that real-time application in this scenario is plausible. The average computational speed on the desktop computer for this experiment of 73.505 Hz is several times more than the minimum required for real-time image processing which is 25 Hz.

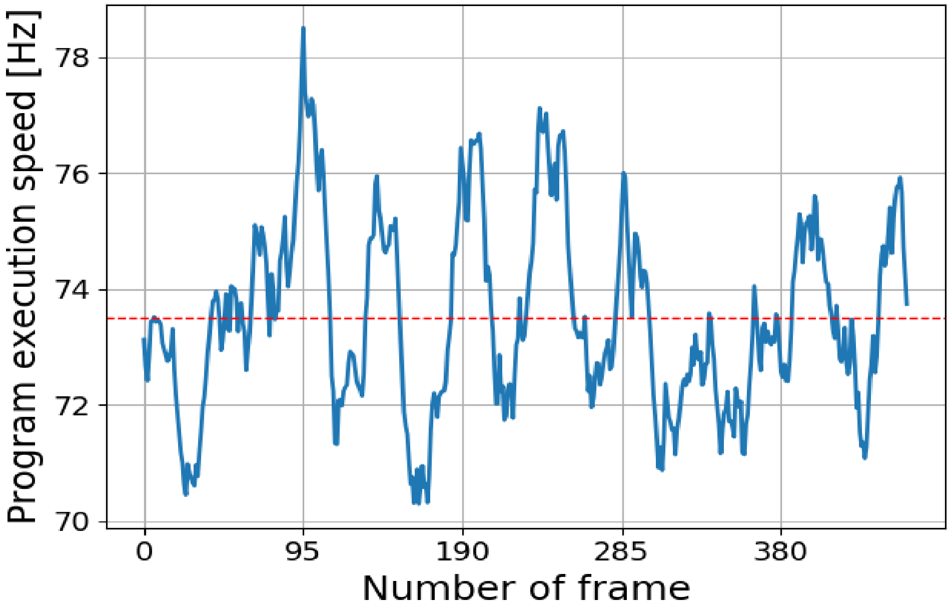


Figure 5.11 Program execution speed in Experiment 1

Since there has been no re-initialization occurrence in this experiment, there are no metrics regarding re-initialization.

5.3.2 Result analysis of experiment 2

The second experiment is done using a more complex organ setup. Urethra, renal vein, renal artery, inferior vena cava and descending aorta are added to the phantom kidney (*Figure 5.12*). The test is done in an enclosed and controlled environment, using the phantom setup over a dark background. Having a dark background provides a test which is not affected by specular noise and can show the level of reliability of the method in such stable environment.

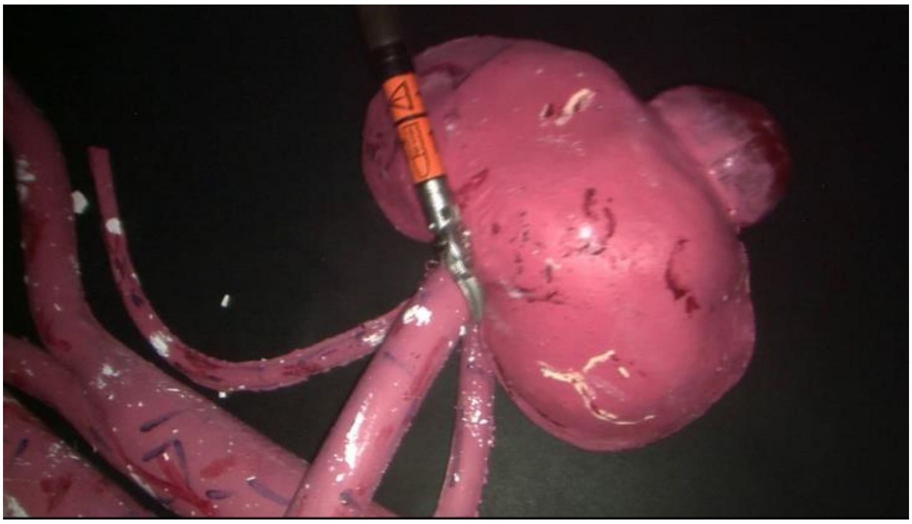
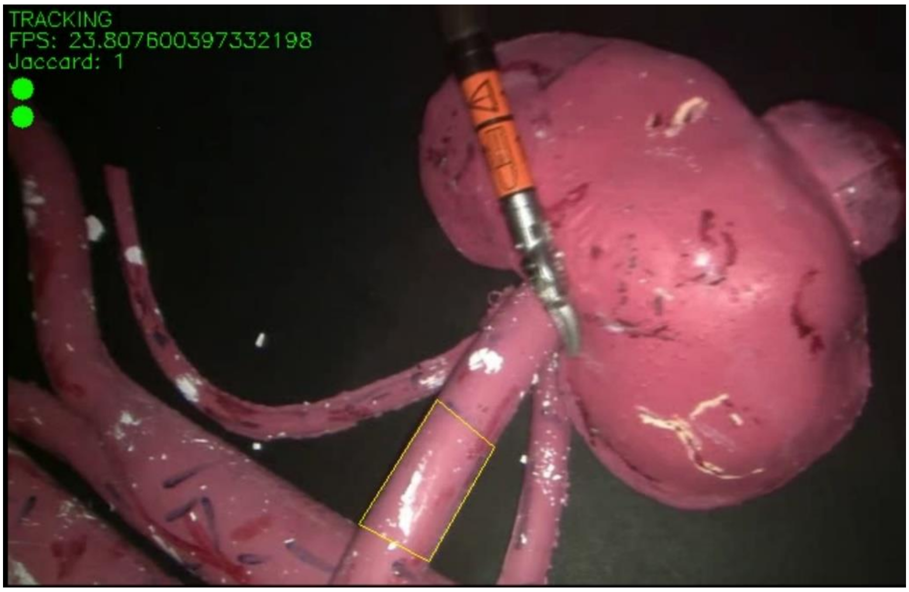


Figure 5.12 Sample frame from experiment 2

This dataset has a resolution of 1840 x 1044 pixels and is 62 seconds long at 25 Hz, which amounts to a total of 1550 frames. The selected AC encircles a region from the urethra (middle and biggest tubal structure exiting the kidney) artificially applied to the phantom kidney, depicted in *Figure 5.13*. In order to create a longer video sequence to test the behavior when it is supposed to track for a longer period of time, this video is created by looping a 15.5 seconds long video for four times. The reason for this is to accumulate the error which is done by the tool manipulation of the object and assess this repeatable behavior). Unlike Experiment 1, here the setup is static but the main testing goal is achieved when a surgical tool is used to manipulate with each part of the setup. For brief periods of time, it will also interfere with the AC.



```
TRACKING
FPS: 23.807600397332198
Jaccard: 1
```

Figure 5.13 AC definition on kidney phantom

In this experiment, the tracked AC is always visible so the table concerning AC visibility is shown in *Table 10*.

Table 10 AC visibility coverage in Experiment 2

SAV	PO	TO	OFV
98.6%	1.4%	0%	0%

Precision, Recall, Accuracy and F1 Score

Precision, recall, accuracy and F1 Score will be calculated and compared in order to have some sense of precision, repeatability and false detection, These values, as previously stated, were calculated by using several thresholds for considering something as True Positive, with respect to the current ground truth (see Chapter 5.2.4 Precision, Recall and Accuracy).Ground truth is defined at each 50 frames, which for this video results with 31 labeled frames from which the following calculations are acquired.

Table 11 Statistical results for Experiment 2

Threshold	*Precision*	*Recall*	*Accuracy*	*F1 Score*	*False Positive*	*False Negative*	*True Positive*
0.3	1	1	1	1	0	0	31
0.5	1	1	1	1	0	0	31
0.7	1	1	1	1	0	0	19
0.8	1	1	1	1	0	0	9

The results presented in *Table 11* are again ‘perfect’, which is not in any way surprising because the dynamics in this experiment are very low and the AC area to be tracked is almost always visible (*Table 10 AC visibility coverage in Experiment 2*) . Even with low texture, the edge-based approach in this method for tracking proves itself as reliable.

These results only take into account if a detection has happened and if it is correct, in a very ‘binary’ manner. However, it gives no information on how the tracked AC compares to the GT.

Tracked AC quality performance

In *Table 12* it can be seen how the tracked AC differs from the GT. For this purpose and because AC and GT can be considered as sets, Jaccard similarity score (JSS) is used again to determine how much do the sets differ.

Table 12 AC tracking quality in experiment 2

	Mean	Standard deviation	Minimum	Maximum
AC quality JSS	0.737	0.098	0.574	0.903

From these Jaccard similarity score results presented *Table 12*, it can be observed that the maximum value is the starting one with 0.903 and the lowest is 0.574. From, it can also be seen that the mean value to be expected is 0.737 and standard deviation of 0.098.

The consistent descending of the value depicted in *Figure 5.14*, infers that no re-initialization has been voted by the voting scheme.

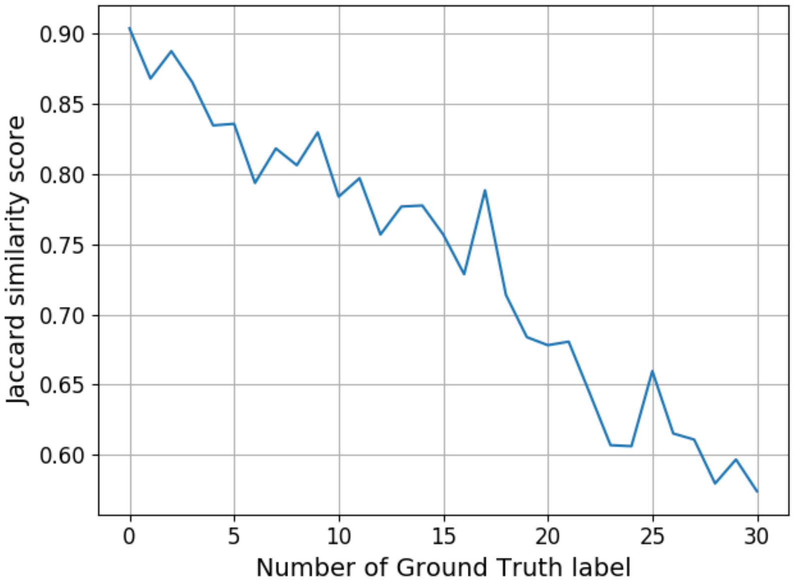


Figure 5.14 AC tracking quality in experiment 2

There are no clearly visible defects of the AC throughout the tracking process, except for minor transformation difficulties because of the oval geometry of the object in the form of drift that (see *Figure 5.15*) have occurred from the minor interferences between surgical tool and AC and the lighting changes cause by the tool. From this data it can be concluded that during this experiment, the method proposed has successfully tracked the AC and when compared to the provided GT the Jaccard similarity score has a mean value of 0.737.

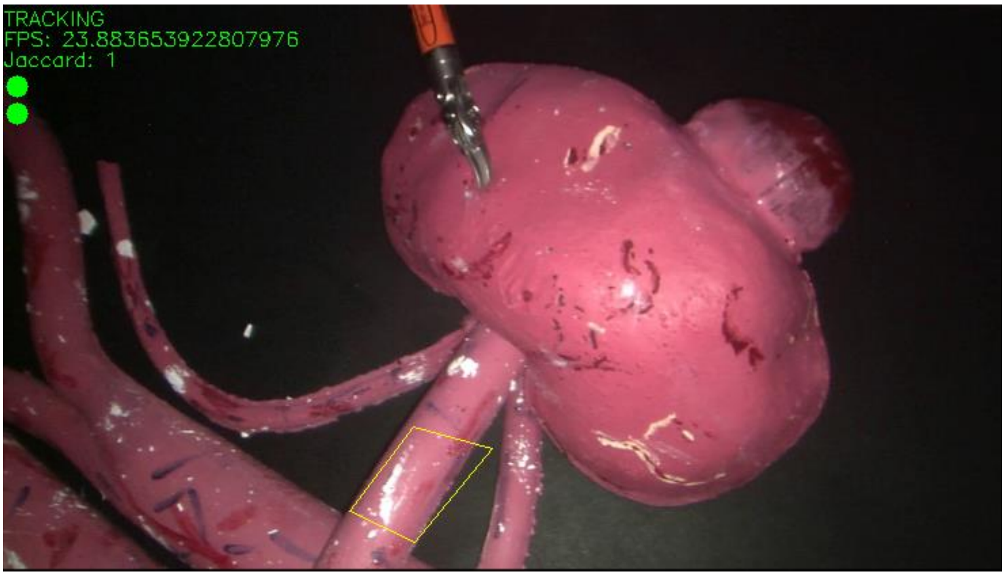


Figure 5.15 Occurrence of drift in experiment 2

Number of active trackers

Even in such stable environment where the phantom kidney is almost completely static, even minor tissue manipulations with the surgical tool can be critical for accurate tracking if they are not accounted for. Tissue manipulation can cause loss of trackers, but also the interference of the tool tip inside of the AC area will directly cover sub-areas filled with trackers causing them to be lost. In Figure 5.16, the number of active trackers for LK method are shown at all times.

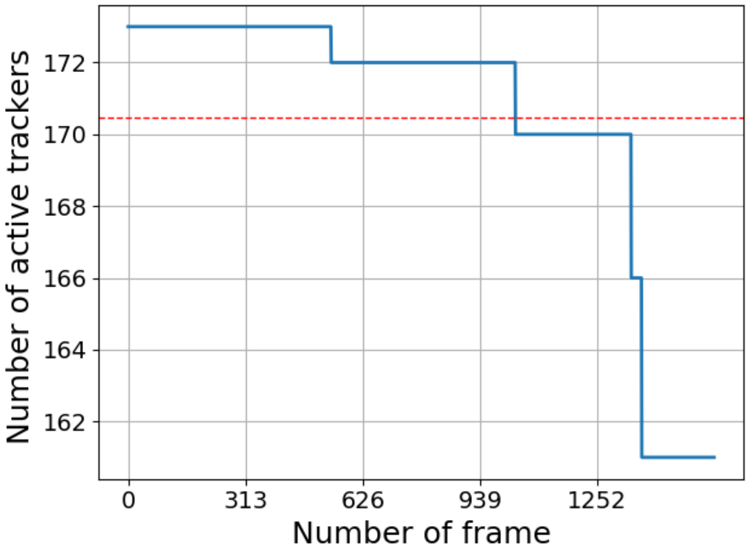


Figure 5.16 Number of active trackers in experiment 2

In this scenario, the number of trackers falls only to 93% of the initialized tracker. Drops of active trackers can be easily correlated to sudden movements which cause an abrupt reflection or projection changes. Since until the end of the dataset, no more than 7% of trackers are lost the re-initialization voting scheme never votes for a re-initialization.

Table 13 Number of active trackers in experiment 2

	Mean	Standard deviation	Minimum	Maximum
No. of active trackers	170.480	3.804	161	173

The initial drawing of AC is always considered as the correct AC definition, so Jaccard score to that AC definition is set always set as 1 to initialization AC polygon. According to what was previously stated, re-initialization in this scenario never happens and Jaccard score for the AC definition remains 1 during the entire video.

Program execution speed

In this experiment, as shown in *Table 14* ,the computational speed had an average value 75.078 Hz, with standard deviation of 6.801 Hz, with a minimum of 7.662 Hz and maximum of 93.504 Hz.

Table 14 Program execution speed in experiment 2

	Mean	Standard deviation	Minimum	Maximum
Program execution speed [Hz]	75.078	6.801	7.662	93.504

These numbers lead to the conclusion that real-time application in this scenario is plausible. The average computational speed on the desktop computer for this experiment of 75.078 Hz is several times more than the minimum required for real-time image processing which is 25 Hz.

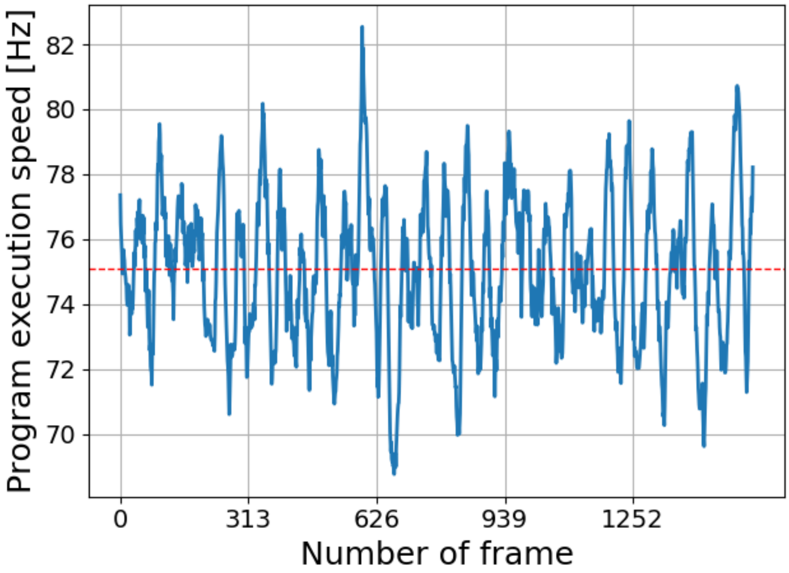


Figure 5.17 Program execution speed in experiment 2

Since there has been no re-initialization occurrence in this experiment, there are no metrics regarding re-initialization.

5.3.3 Result analysis of experiment 3

The previous two experiments covered motion and minor partial occlusion. In this experiment total occlusion will be introduced such that re-initialization is forced to assess the behavior of the method when the tissue is not present in the field of view and how well does it localize and describe it after reappearing.

The kidney setup is the same as in experiment 2, where to the kidney phantom with tumor, urethra, renal vein and renal artery are attached, while the renal vein is attached to the inferior vena cava. In this experiment, the inferior vena cava will be the organ to be protected by AC (biggest tubal structure located in the lower left corner). Same as in Experiment 2, this kidney phantom setup is static. A big object (human hand) will interfere with the camera's scope of view to induce total occlusion. Because of the fast movement, automatic contrast adjustment and autofocus from the camera, additional darkening of the camera output will appear. In this video sequence there is no physical contact and manipulation of any part of the kidney phantom setup.

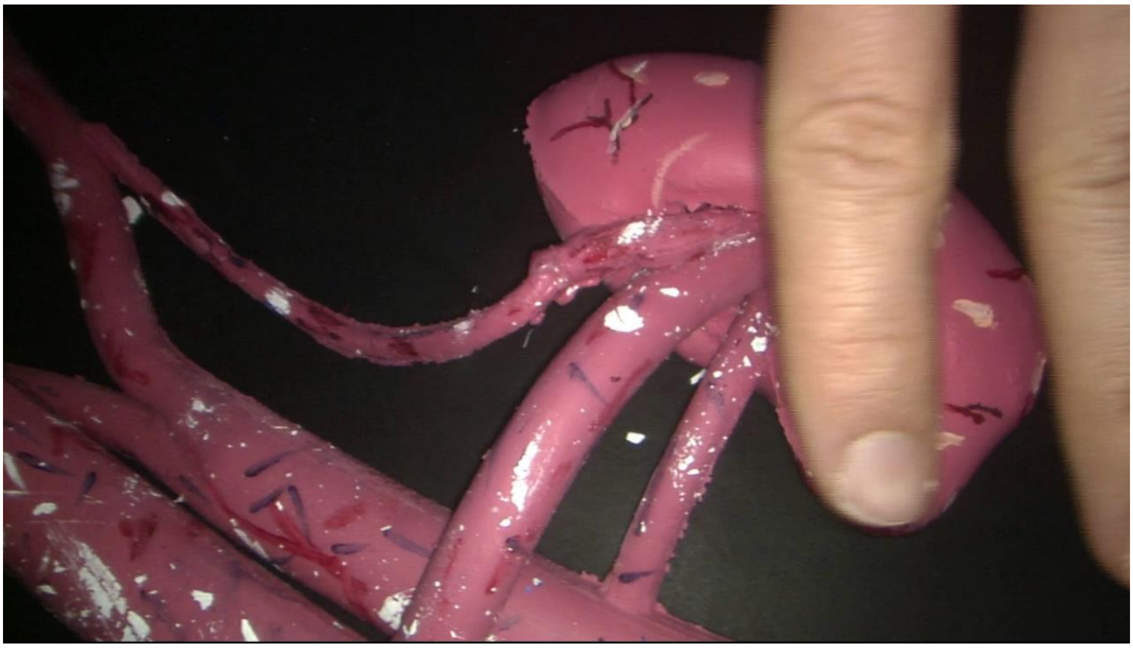


Figure 5.18 Sample frame from experiment 3

This dataset has a resolution of 1840 x 1044 pixels and is 46 seconds long at 25 Hz, which amounts to a total of 1150 frames. The selected AC encircles a region from the artificial inferior vena cava. In order to create a longer video sequence to test the behavior when it is supposed to track for a longer period of time, at randomly chosen times the hand will either partially or totally cover the camera's scope of view. The reason for this is to assess the ability of the algorithm to learn online and rediscover the AC region being tracked after being entirely out of the scope of view.

In this experiment, the tracked AC is not always visible so the table concerning AC visibility is shown in *Table 15*.

Table 15 AC visibility coverage in Experiment 3

SAV	**PO**	**TO**	**OFV**
79.2%	3.3%	17.5%	0%

The environment that is provided in this experiment covers a very usual occurrence in real world surgeries which is having surgical tools interfering in the field of view of the camera, thus removing visibility from the algorithm. In such interference, even being close to the tracked AC may cause problems due to the addition of shade and directly changing the overall lighting.

Precision, Recall, Accuracy and F1 Score

Precision, recall, accuracy and F1 Score will be calculated and compared in order to have some sense of precision, repeatability and false detection. These values, as previously stated, were calculated by using several thresholds for considering something as True Positive, with respect to the current ground truth (Chapter). Ground truth is defined at each 50 frames, which for this video results with 23 labeled frames from which the following calculations are acquired.

Table 16 Statistical results for Experiment 3

Threshold	*Precision*	*Recall*	*Accuracy*	*F1 Score*	*False Positive*	*False Negative*	*True Positive*
0.3	1	1	1	1	0	0	21
0.5	1	1	1	1	0	0	20
0.7	1	1	1	1	0	0	17
0.8	1	1	1	1	0	0	10

The results presented in *Table 16* are yet again ‘perfect’. This results is not true, and is a consequence of the 50 frames interframe step for labeling ground truth. The occurrence of false positive was not documented by the statistical methods and therefore will not be considered in these calculations. However, it is necessary to mention that it has happened even though it was not registered. Having a false positive means that the AC was detected by the method, even though it was not visible and the ground truth at that moment was None (empty). These scores are very high but must be taken with caution because there is always a possibility for something to have happened between the labeling timestamps distanced 50 frames between two consecutive test points. Moreover, 50 frames add up to 2 seconds which is not a big timespan in a real surgery that may last in terms of hours.

These results only take into account if a detection has happened and if it is correct, in a very ‘binary’ manner. However, it gives no information on how the tracked AC compares to the GT.

Tracked AC quality performance

In *Table 17* it can be seen how the tracked AC differs from the GT. For this purpose, Jaccard similarity score (JSS) is used again to determine how much do the sets differ. It can also be observed that the maximum value is 1.0 and the lowest is 0.0. In addition, the mean value has value of 0.711 and standard deviation of 0.225.

Table 17 AC tracking quality in experiment 3

	Mean	Standard deviation	Minimum	Maximum
AC quality JSS	0.711	0.225	0	1

A Jaccard score of 0 means either an empty set (no AC or GT) or AC and GT have no common element (completely wrong estimation).

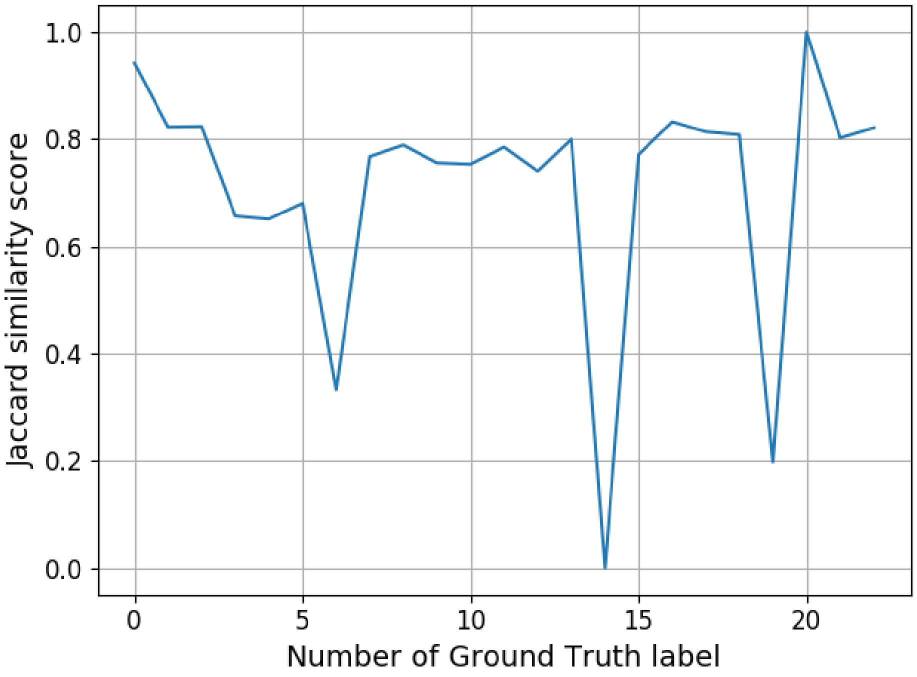


Figure 5.19 AC tracking quality in experiment 3

But these numbers are not to be taken directly as truth, because one data point is due to either wrong detection, false positive or false negative and gives no significant information on the quality of AC tracking. Aside from this 0 score value, there are also two additional low points. One of them has recovered fast to a high score value of 0.74 (from data point at step 6 to step 7), while the second one (data point at step 19) is followed by a 'perfect' recovery which leads to the conclusion that there has been a failure to detect and is most likely to have been a long lasting occlusion.

If these three points are neglected by the assumption that at these times failure to track has occurred or is anticipated to occur, the mean value rises to 0.753 with standard deviation of 0.11.

Another very important aspect of this method is self-learning and optimizing. In other words, once a wrong re-definition has happened it should be able to correct itself and to reposition the AC correctly.

By adding additional ground truth instances a posterior in order to specifically catch these instances, this method provides a reaction time of at most 7 frames and at least

1 frame according to this experiment. This means that if a wrong redefinition of the AC has occurred, the algorithm is most likely correct itself and move the AC definition back to the proper within 7 sequential frames, assuming the tracked region is visible.

Number of active trackers

The environment that is provided in this experiment covers a very usual occurrence in real world surgeries which is having surgical tools interfering in the field of view of the camera, thus removing visibility from the scope of view. In such interference, even being close to the tracked AC may cause problems due to the addition of shade and directly changing the overall lighting. These are the main reasons for great loss of trackers from LK tracking algorithm. In *Figure 5.20*, the number of active trackers for LK method are shown at all times.

Table 18 Number of active trackers in experiment 3

	Mean	Standard deviation	Minimum	Maximum
No. of active trackers	789.285	261.465	20.0	1158.0

Unlike the previous experiments, here the number of active trackers is not always decreasing. Re-initialization happens when they drop below the threshold defined as 80% of the number of initialized LK trackers, which reactivates a new set of trackers. Sudden drops of large magnitude can be identified as total occlusions of the AC area being tracked. They are always followed with sudden jumps to a higher value of active trackers which is caused by re-initialization.

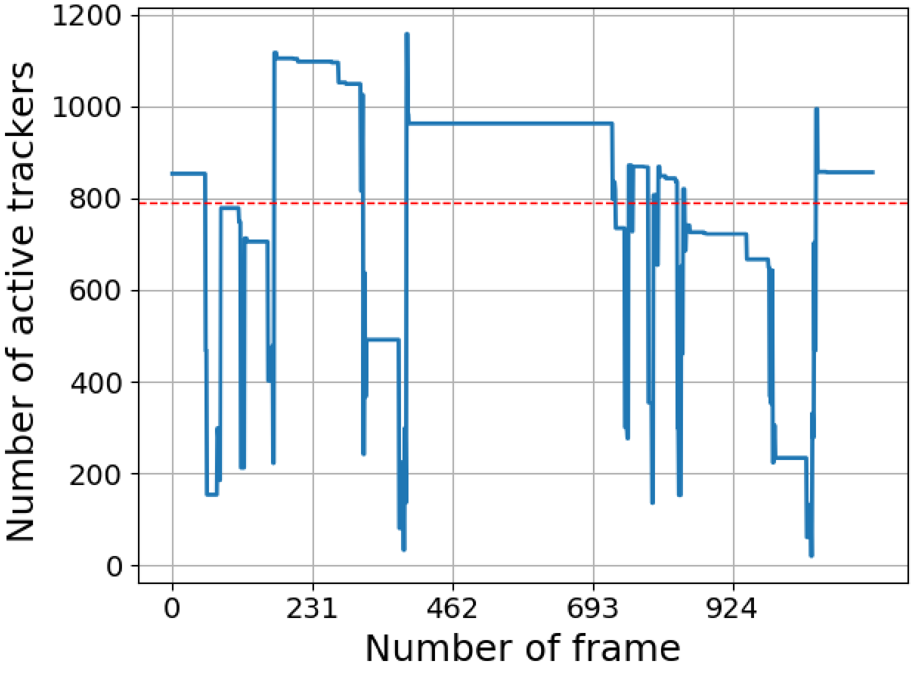


Figure 5.20 Number of active trackers in experiment 3

Program execution speed

In this experiment, as shown in *Table 19 Program execution speed in experiment 3*,the computational speed had an average value 27.830 Hz, with standard deviation of 3.374 Hz, with a minimum of 0.594 Hz and maximum of 44.649 Hz. These numbers lead to the conclusion that real-time application in this scenario is plausible. The average computational speed on the desktop computer for this experiment of 27.830 Hz is more than the minimum required for real-time image processing which is 25 Hz.

Table 19 Program execution speed in experiment 3

	Mean	Standard deviation	Minimum	Maximum
Program execution speed [Hz]	27.830	3.374	0.594	44.649

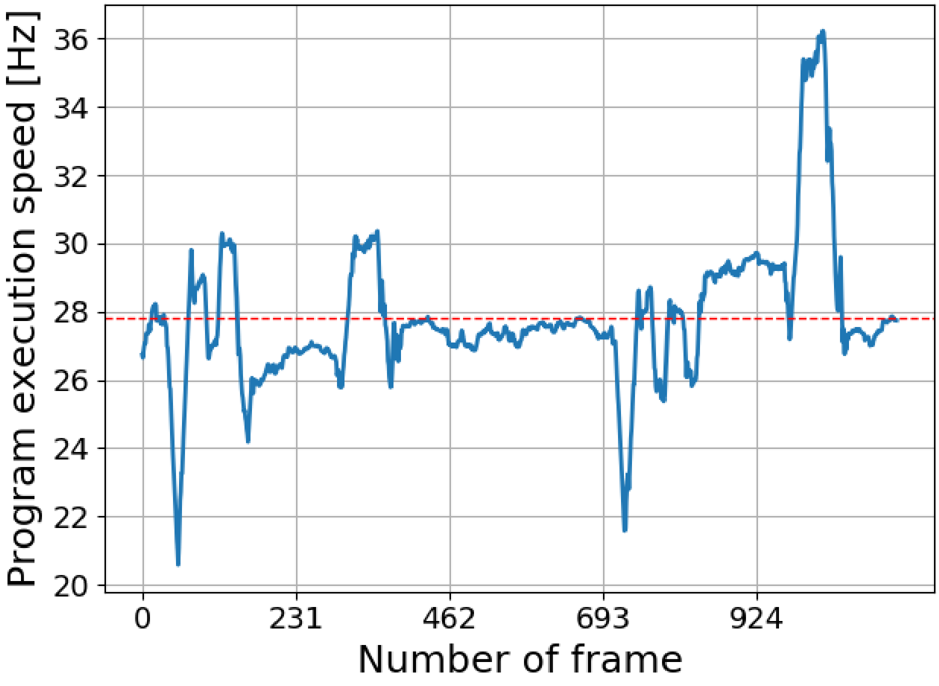


Figure 5.21 Program execution speed in experiment 3

Unlike the previous two experiments, in this experiment there are total occlusions registered during runtime. Because of this two more aspects need to be inspected, Reliability index and length of tracking between re-initializations.

Reliability index

When re-initialization happens, localization and AC redefinition techniques are applied as described in Chapter 4.5. In order to determine if they are reliable or not, the assumption made in Chapter 4.5 is yet again considered and it states that a well-posed redefinition of AC will be only mildly affected by binary erosion once it is applied. The metric used for this is yet again Jaccard similarity score.

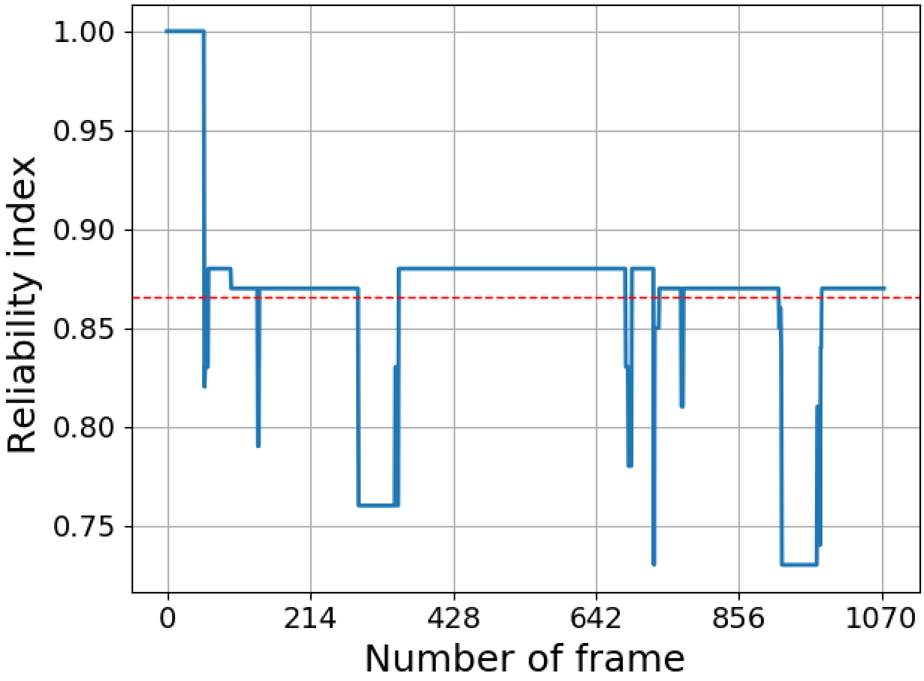


Figure 5.22 Reliability index in experiment 3

The changing value in this figure, which was previously always 1, shows how ‘good’ are the re-definition and refining. The mean value is 0.866, as presented in *Table 20 Reliability index in experiment 3*, and standard deviation is 0.051. Having a mean of 0.865 score value shown that the proposed method in this thesis tackles the problem of redefinition successfully by having an average similarity score of 0.865 from the possible maximum 1.

Table 20 Reliability index in experiment 3

	Mean	Standard deviation	Minimum	Maximum
Reliability index	0.866	0.051	0.73	1.0

Length of tracking phases

Another difference between this experiment and the previous two is that tracking occurs continuously without failure in those experiments. Here, it is also important to assess how well does it accompany the concept of long-term tracking.

In *Figure 5.23* all tracking phases are shown with the length of each of them. There are two maximums of 342 and 338 frames length of tracking, which adds up to 13.68 s of tracking without re-initialization per tracking phase.

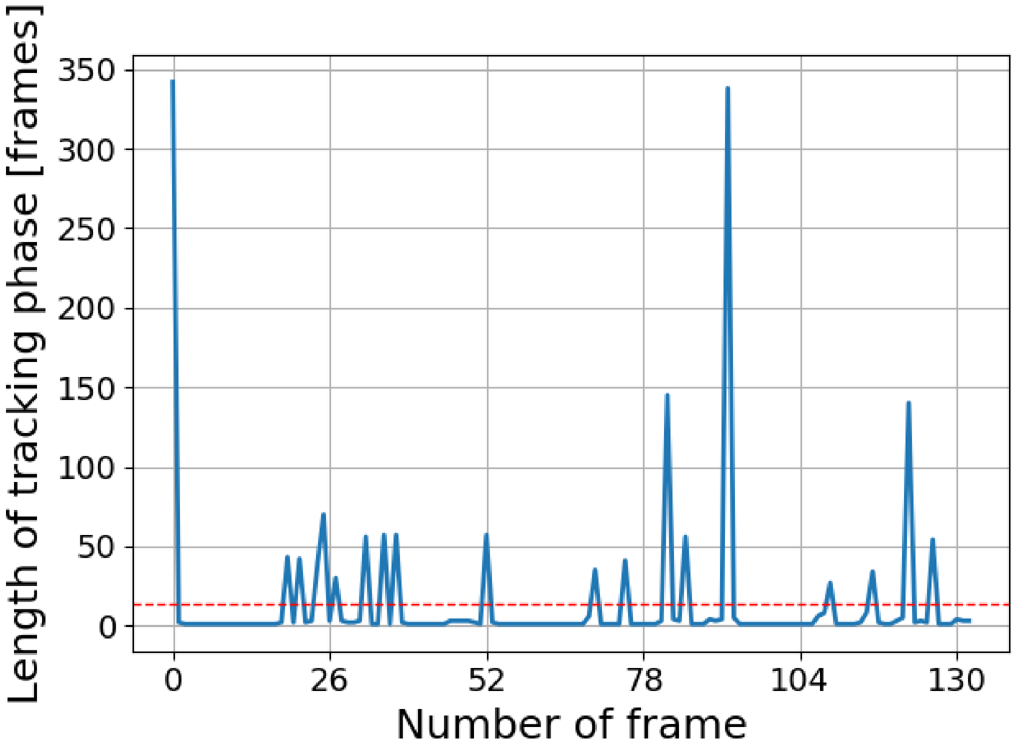


Figure 5.23 Length of tracking phases in experiment 3

The mean value of all tracking phases is 14.015 frames (see *Table 21*), while the lowest value is 1, when there is no tracking but rather only attempts to re-initialize. Accordingly, the sequential values of 1 are associated to total occlusion.

Table 21 Length of tracking phases in experiment 3

	Mean	Standard deviation	Minimum	Maximum
Tracking lengths [frames]	14.015	45.960	1.0	342.0
Corrected tracking lengths [frames]	40.045	3.296	3.0	342.0

If all instances where the value of tracking length phase is below three are discarded, a graph will be acquired representing all phases where there has been effective tracking of at least 3 sequential frames, as shown in *Figure 5.24*. This data where only effective tracking of more than three consecutive frames is considered will be call ‘corrected’. The mean value will rise to 40.045 frames sequential frames of tracking with a lowered standard deviation to 3.296 frames (see *Table 21*)

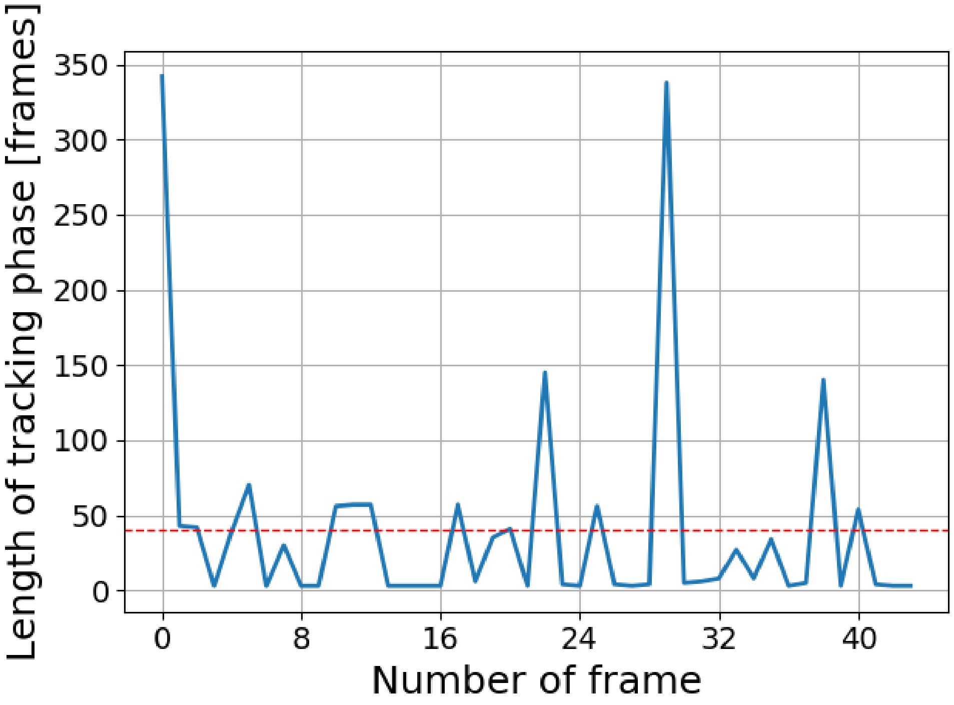


Figure 5.24 'Corrected' length of tracking phases in experiment 3

Visual defects

During this experiment, several visual occurrences were noticed which cannot be inferred from the statistical data, such as:

- Overestimation at re-initialization (*Figure 5.25 (d))*
- Wrong re-initialization (*Figure 5.25 (c))*
- Bad re-initialization (*Figure 5.25 (b))*
- Correction after bad - reinitialization

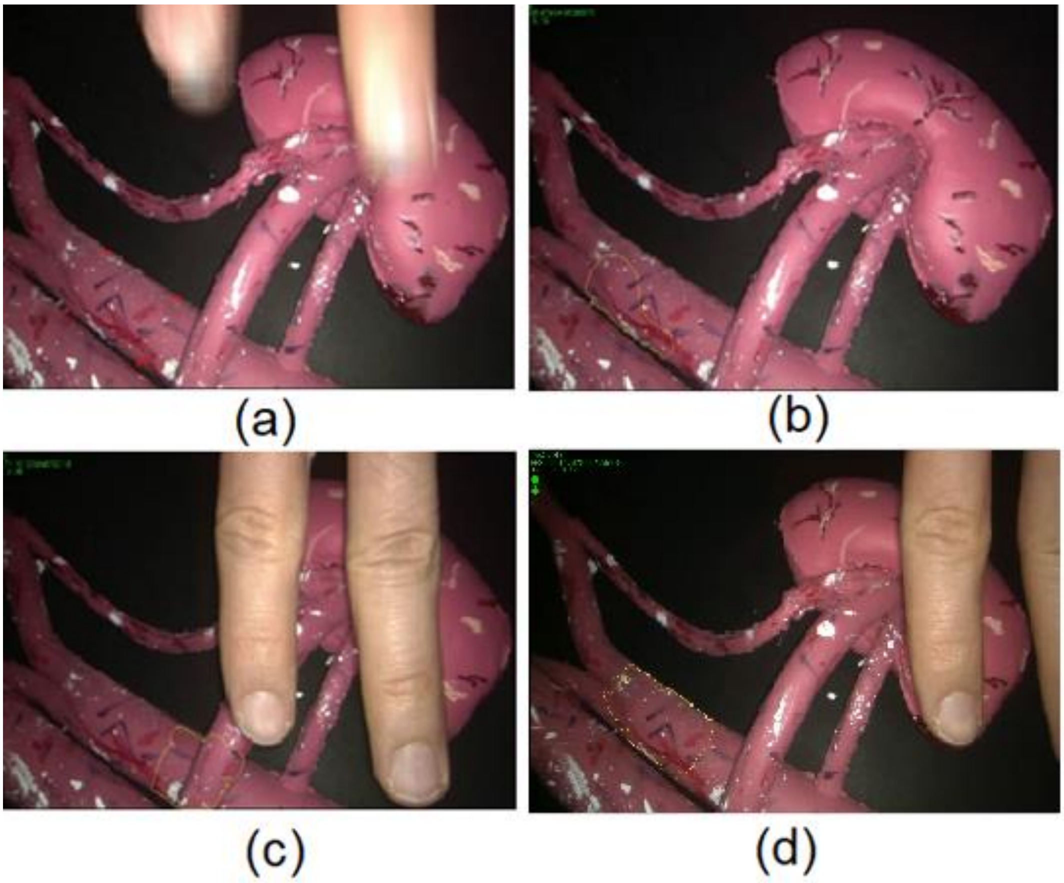


Figure 5.25 (a)GT definition; (b) bad re-initialization; (c) wrong re-initialization; (d) overestimation

5.3.4 Result analysis of experiment 4

This fourth and last experiment from this group of experiments on the kidney phantom focuses on deformation. The video sequence starts with the object deformed (*Figure 5.26*, upper picture), and after a while the pressure is released resulting in the object going back to its normal form phantom focuses on deformation. The video sequence starts with the object deformed (*Figure 5.26*, lower picture). In addition to this, there is rotation occurring. The kidney phantom is positioned at a relatively close distance from the camera. The deformation is severe, and the object regains its primary form with a very fast and abrupt movement. This will cause loss of majority of active trackers. This is to be expected, and the goal of this experiment is to assess how reliable is this method in redefining the AC when it is under a great deformation.

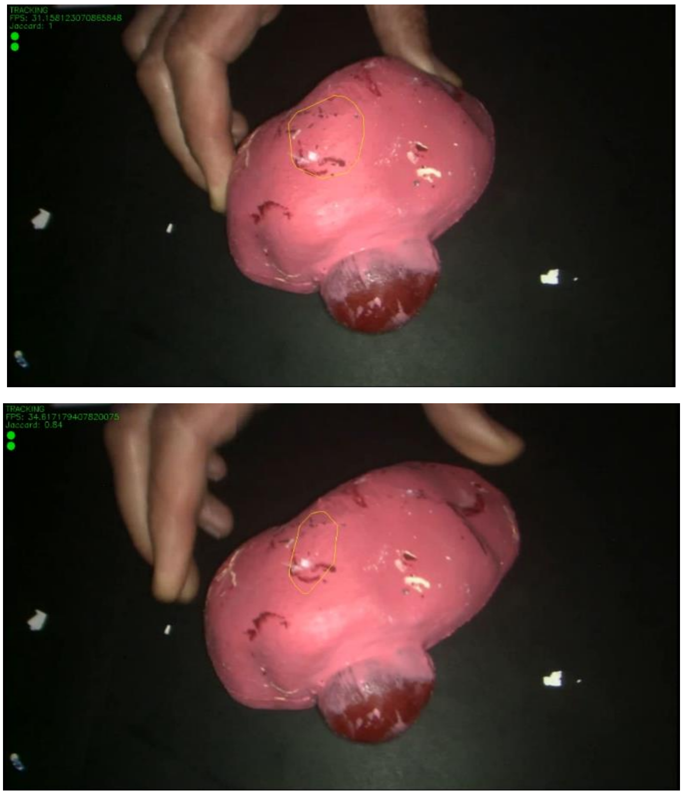


Figure 5.26 Upper: deformed model; Lower: recovered model

The previous experiments covered motion, minor partial occlusion and total occlusion. In this experiment deformation will be introduced, such that tracking and re-initialization are forced to try and track change, movement and loss of feature points in order to assess the behavior of the method when the tissue loses its primary form and how well does it localize and describe it after deforming. The kidney setup is the same as in experiment 1, where to the kidney phantom with tumor is the only component of the experimental test setup. In this experiment, a randomly chosen region from the kidney surface will be the region to be protected by AC . Same as in Experiment 2, this kidney phantom setup is almost completely static. A manipulating device (human hand) will apply force to the sides of the kidney phantom to induce deformation and partial occlusion. Because of the fast and abrupt movement, automatic contrast adjustment and autofocus from the camera will create additional darkening of the output.

This dataset has a resolution of 1840 x 1044 pixels and is 20 seconds long at 25 Hz, which amounts to a total of 500 frames. The selected AC encircles a region from the artificial kidney model. In order to create a better video sequence to test the behavior when it is supposed to track for a longer period of time but with big magnitudes of deformation, at randomly chosen times the hand will partially change the overall shape of the kidney phantom. The reason for this is to assess the ability of the algorithm to learn online, track in a dynamic environment and rediscover the AC region being tracked after being occluded and deformed.

In this experiment, the tracked AC is not always completely visible so the table concerning AC visibility is shown in *Table 22*.

Table 22 AC visibility coverage in Experiment 4

SAV	PO	TO	OFV
93%	7%	0%	0%

The environment that is provided in this experiment covers a very usual occurrence in real world surgeries which is having surgical tools manipulating the tissue and changing its shape regardless of rotation and translation, thus removing visibility from the algorithm.

Precision, Recall, Accuracy and F1 Score

Precision, recall, accuracy and F1 Score will be calculated and compared in order to have some sense of precision, repeatability and false detection, These values, as previously stated, were calculated by using several thresholds for considering something as True Positive, with respect to the current ground truth (Chapter 5.2.4 Precision, Recall and Accuracy). Ground truth is defined at each 50 frames, which for this video results with 10 labeled frames from which the following calculations are acquired.

Table 23 Statistical results for experiment 4

Threshold	*Precision*	*Recall*	*Accuracy*	*F1 Score*	*False Positive*	*False Negative*	*True Positive*
0.3	1	1	1	1	0	0	10
0.5	1	1	1	1	0	0	10
0.7	1	1	1	1	0	0	3
0.8	1	1	1	1	0	0	1

The acquired values of the results are all equal to 1, as in the first two experiments, which leads to the conclusion that the detection algorithm well. There is no surprise about these results because the AC is always visible and easily localized, even when deformed. There are short spans of sequences in this video which have very high dynamic, but the results regarding Precision and Recall are not affected by this. As seen in the *Table 23 Statistical results for experiment 4*, no false positive estimation has occurred even with the strictest threshold of 0.8.

These results only take into account if a detection has happened and if it is correct, in a very 'binary' manner. However, it gives no information on how the tracked AC compares to the GT.

Tracked AC quality performance

In *Table 24* it can be observed how the tracked AC differs from the GT. For this purpose, Jaccard similarity score (JSS) is used again to determine how much do the sets differ. It can also be observed that the maximum value is the starting one with 0.887 and the lowest is 0.603 with a mean value of 0.683 and standard deviation of 0.078.

Table 24 AC tracking quality in experiment 4

	Mean	Standard deviation	Minimum	Maximum
AC quality JSS	0.683	0.078	0.603	0.887

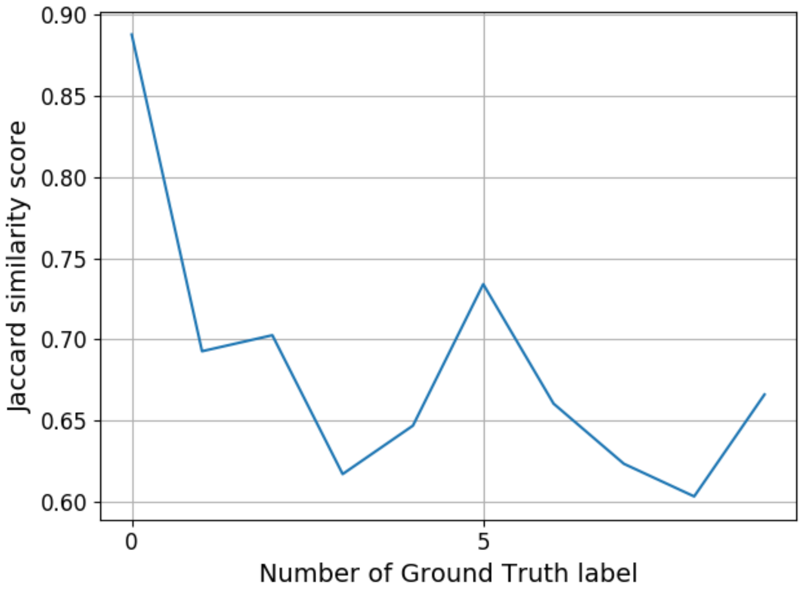


Figure 5.27 AC tracking quality in experiment 4

From Jaccard similarity score graph pictured in *Figure 5.27*,it can be observed that the maximum value is the starting one with 0.887 and the lowest is 0.603. This can be considered a fairly wide range of 0.284 score value, considering a mean value of 0.683 and standard deviation of 0.078. Between GT label steps 3 and 5 a great recovery can be seen which is the adaptive algorithm finding new similarities to the models from the buffer. This is one of the most critical measures for this experiment as it is visually difficult to identify several parts of the AC area. These clearly visible defects of the AC throughout the tracking and re-initialization process, and they are directly affected by two reasons:

- Lack of texture in the kidney phantom
- Big and fast deformation is applied

Once the deformation has been applied, or most trackers are lost, re-initialization procedure starts. It always locates it AC well but does not always expand properly to describe it better and refine this redefinition. Lack of texture makes a problem for Morphological Chan Vese algorithm which requires color and morphological differences in order to minimize the energy equation. Nevertheless, a mean value of 0.683 is still good result having in mind that it means 68.3% of the sets have matched.

Number of active trackers

The environment that is provided in this experiment covers a very usual occurrence in real world surgeries which is having surgical tools interfering in the field of view of the camera, thus removing visibility from the algorithm. In such interference, even being close to the tracked AC may cause problems due to the addition of shade and directly changing the overall lighting. These are the main reasons for great loss of trackers from LK tracking algorithm. In *Figure 5.28* , the number of active trackers for LK method are shown at all times.

Table 25 Number of active trackers in experiment 4

	Mean	Standard deviation	Minimum	Maximum
No. of active trackers	567.699	125.059	195	869

Like the previous experiment, here the number of active trackers is not always decreasing. Re-initialization happens when they drop below the threshold defined as 80% of the number of initialized LK trackers, which reactivates a new set of trackers. It is easily noticeable when does re-initialization happens, as it is the only way to have an increase in the number of active trackers. The mean average of active trackers and standard deviation are presented in *Table 25 Number of active trackers in experiment 4*.

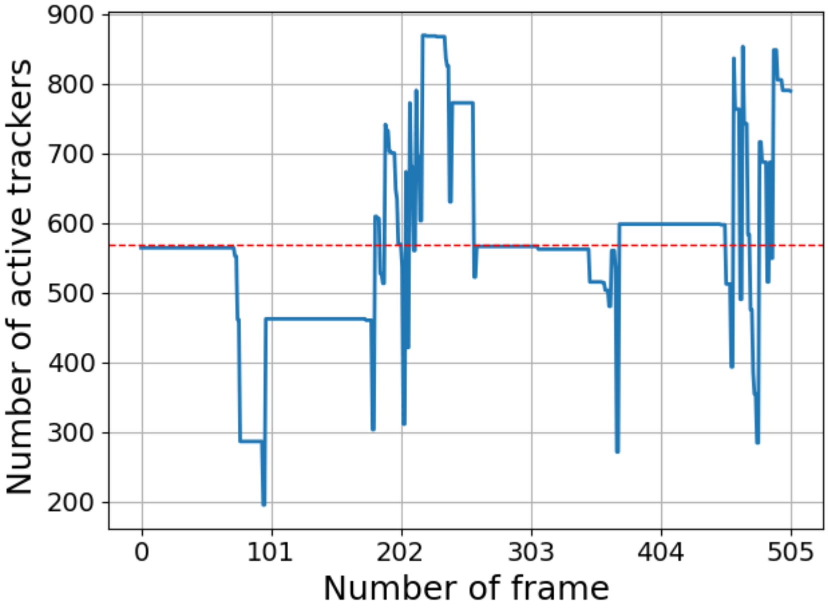


Figure 5.28 Number of active trackers in experiment 4

Program execution speed

In this experiment, as shown in *Table 26 Program execution speed in experiment 4*, the computational speed had an average value 36.739 Hz, with standard deviation of 3.388 Hz, with a minimum of 0.655 Hz and maximum of 43.964 Hz.

Table 26 Program execution speed in experiment 4

	Mean	Standard deviation	Minimum	Maximum
Program execution speed [Hz]	36.739	3.388	0.655	43.964

These numbers lead to the conclusion that real-time application in this scenario is plausible. The average computational speed on the desktop computer for this experiment of 36.739 Hz is more than the minimum required for real-time image processing which is 25 Hz.

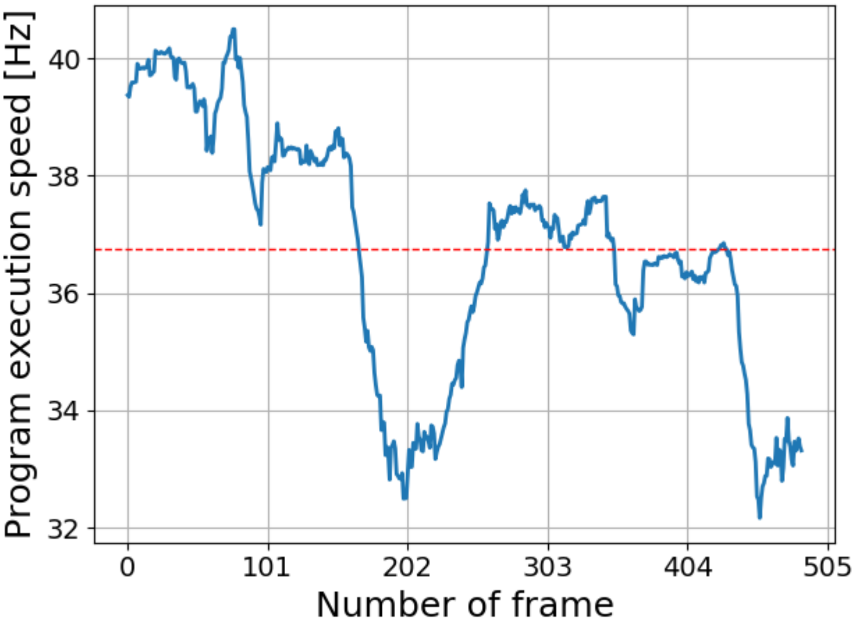


Figure 5.29 Program execution speed in experiment 4

Reliability index

When re-initialization happens, localization and AC redefinition techniques are applied as described in the previous chapter. In order to determine if they are reliable or not, the assumption mentioned in the previous chapter is yet again considered. It states that a well-posed redefinition of AC will not change much after binary erosion is applied. The metric used for this is yet again Jaccard similarity score.

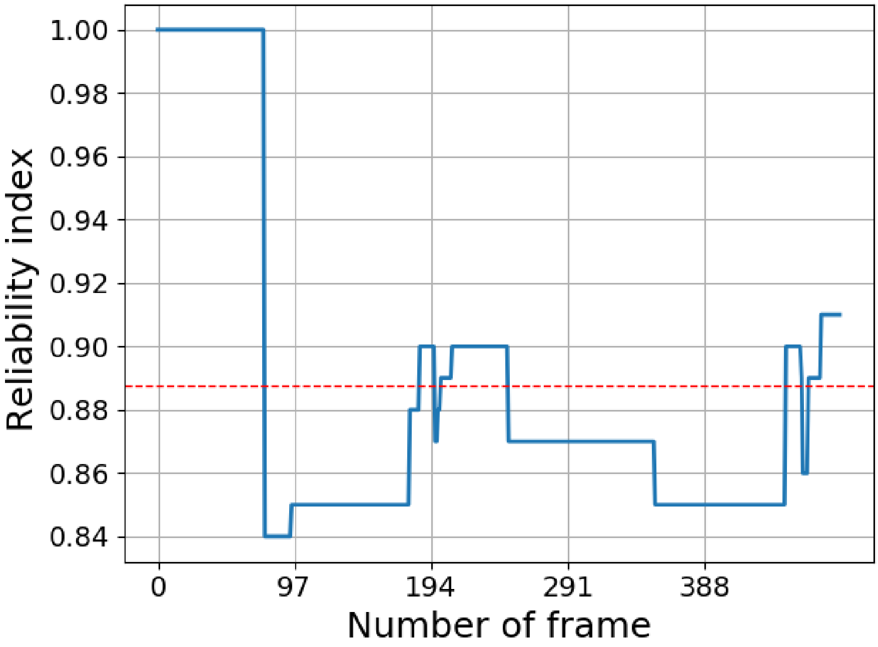


Figure 5.30 Reliability index of experiment 4

From *Figure 5.30 Reliability index of experiment 4* it can be concluded that many tracking failures have happened, causing as many re-initializations. The instability caused by the same reasons discussed in the *AC tracking quality* segment forces many reinitializations in a short time span. According to *Table 27 Reliability index of experiment 4*, a mean of 0.887 from the possible maximum 1 score value is achieved which shows that the proposed method in this thesis tackles the problem of redefinition successfully.

Table 27 Reliability index of experiment 4

	Mean	Standard deviation	Minimum	Maximum
Reliability index	0.887	0.051	0.84	1.0

Length of tracking phases

Similar to the previous experiment, this one also covers re-initializations. With this, the period between successful re-initializations is a period called tracking phase. The length of these phases can give insight on how long it can track an AC definition when visible. This tracking should include deformations which is the goal of this experiment.

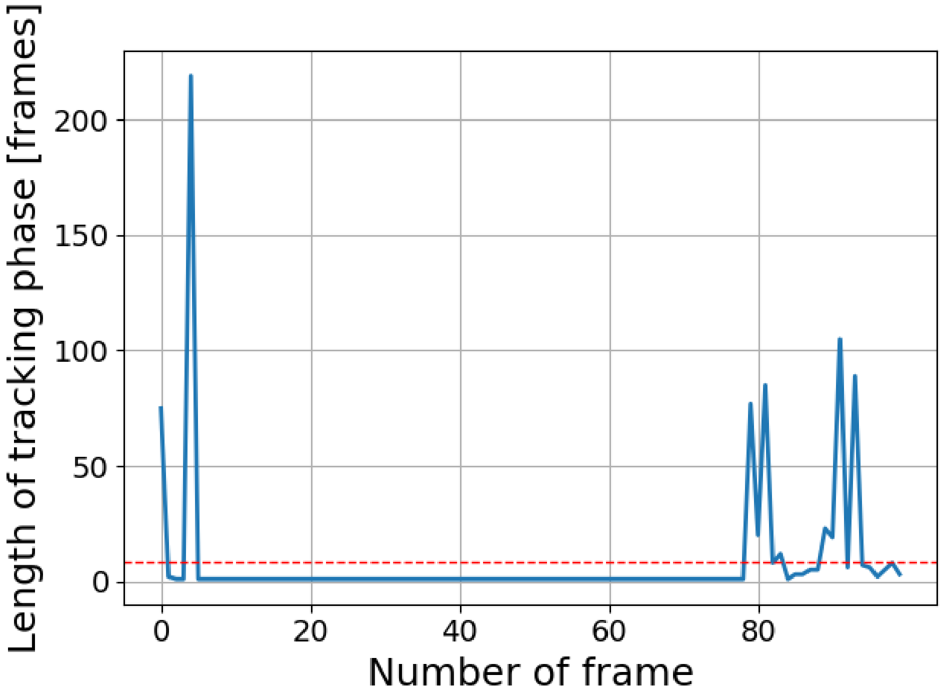


Figure 5.31 Length of tracking phases in experiment 4

In *Figure 5.31* all tracking phases are shown with the length of each of them. There is one maximum of 219 frames length of tracking, which adds up to 8.76 s of tracking without re-initialization. The mean value of all tracking phases is 8.64 frames (See *Table 28 Length of tracking phases in experiment 4*), while the lowest value is 1, when there is no tracking but rather only attempts to re-initialize. Accordingly, the sequential values of 1 are associated to total occlusion.

Table 28 Length of tracking phases in experiment 4

	Mean	Standard deviation	Minimum	Maximum
Tracking length [frames]	8.64	28.300	1.0	219.0
Corrected tracking length [frames]	37.28	52.678	3.0	219.0

If all instances where the value of tracking length phase is below three are discarded, a graph will be acquired representing all phases where there has been effective tracking of at least 3 sequential frames, as shown in *Figure 5.32*. The mean value will rise to 40.045 sequential frames of tracking.

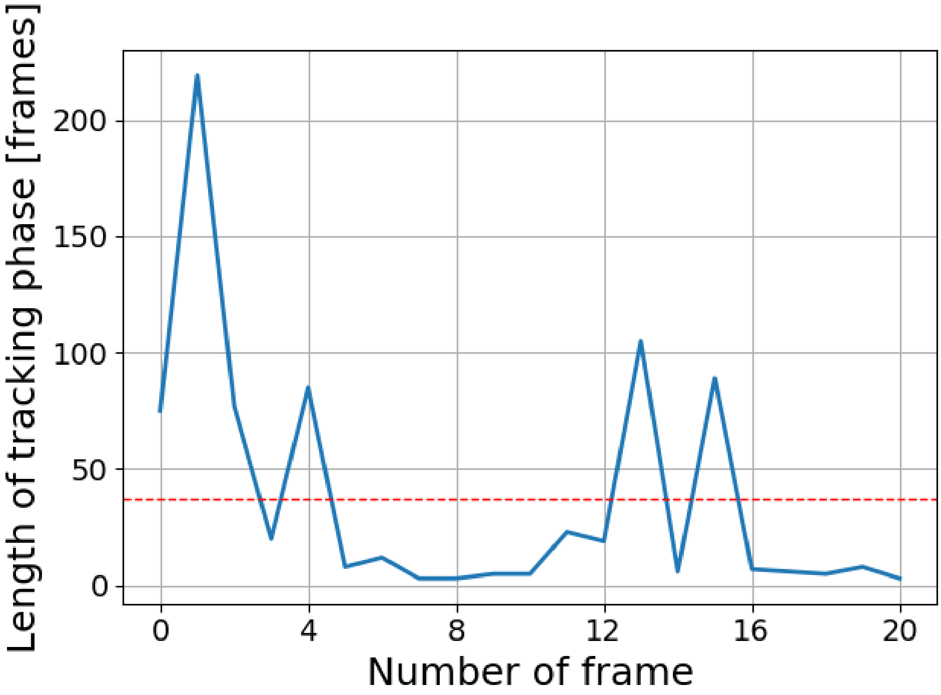


Figure 5.32 'Corrected' length of tracking phases in experiment 4

Visual defects

During this experiment, several visual occurrences were noticed which cannot be inferred from the statistical data, such as:

- Underestimation when AC is re-initialized
- Complete loss of visibility in parts of AC area definition

5.4. Validation of the method in real surgical environment

All previous experiments were in a controlled environment in order to examine behaviors from several aspects. However, this last experiment is a real world situation. A relatively long video sequence of 4 minutes is used, which is an in vivo recording from a pancreatectomy conveyed using da Vinci Surgical system. The video is approximately 4 minutes long.

Out of the TrackVes dataset, only this video was chosen for testing because it is the only *in vivo* video with full results presented in the paper of Penza et al. [25].

Compared to the previous tests, in this sequence there is no dark background, no static organs and no deterministic organ motion or tool manipulation. In such test, there will be many organs showing similar texture, geometry and size, and a very dynamic camera field of view. Camera movement is to be expected in 3D space (left, right, inwards and outwards with respect to the field of view in focus) and rotation (change in projection). Furthermore, one or two surgical tools are always used to manipulate the tissue. Depending on the tool, there may be ocular changes such as smoke from electrocautery. In addition, tools are very likely to appear between the camera (endoscope) and the organ. Movement of the endoscope also will create lighting changes and may increase or decrease lighting and reflections. The endoscope has a very small point lighting which never achieves to provide a homogeneous dispersion of light in the field of view.

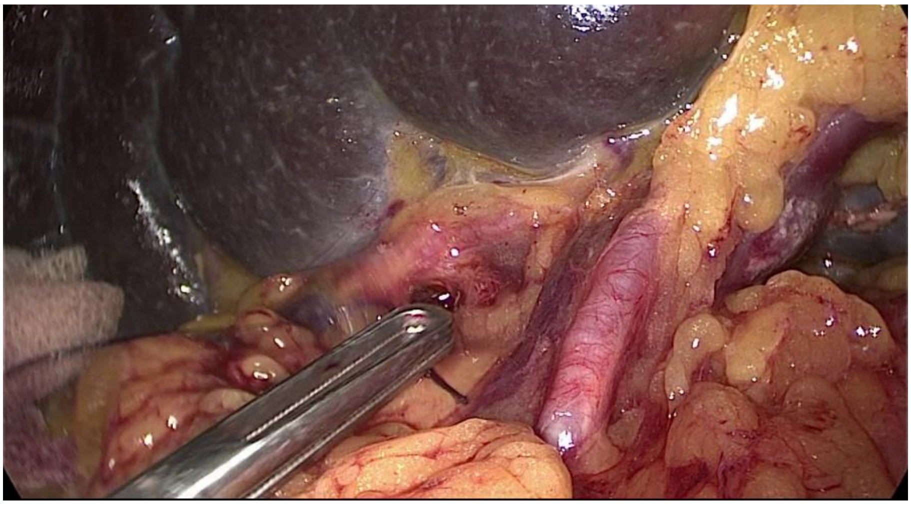


Figure 5.33 Sample frame of experiment 5

This experiment will provide assessment of the proposed method of this thesis in a real world surgical application and it will assess many different situations that were or were not covered in the previous experiments such as:

- Minor and major deformations
- Partial occlusion
- Total occlusion
- Out of field of view
- Partial and total occlusion caused by surgical tool
- Fast paced element in scope of view
- Big motion changes
- Severe lighting changes caused by endoscope lighting
- Inward and outward motion of the endoscope with respect to the tissue being operated

This video sequence has long periods where the AC area is out of field of view. For this reason, the parts where the AC is out of view for more than 20 seconds have been cut out in order to condensate relative information in the results that are to be presented. During these cut out segments it was made sure that there is no occurrence of false positives. Having a long sequence of True Negatives will only cause the significant results regarding tracking and re-initialization to lose on statistical value.

This dataset has a resolution of 1280 x 718 pixels and is 4 minutes long at 25 Hz, which amounts to a total of 6000 frames. The selected AC encircles a region from an artery, as can be seen in *Figure 5.34*.

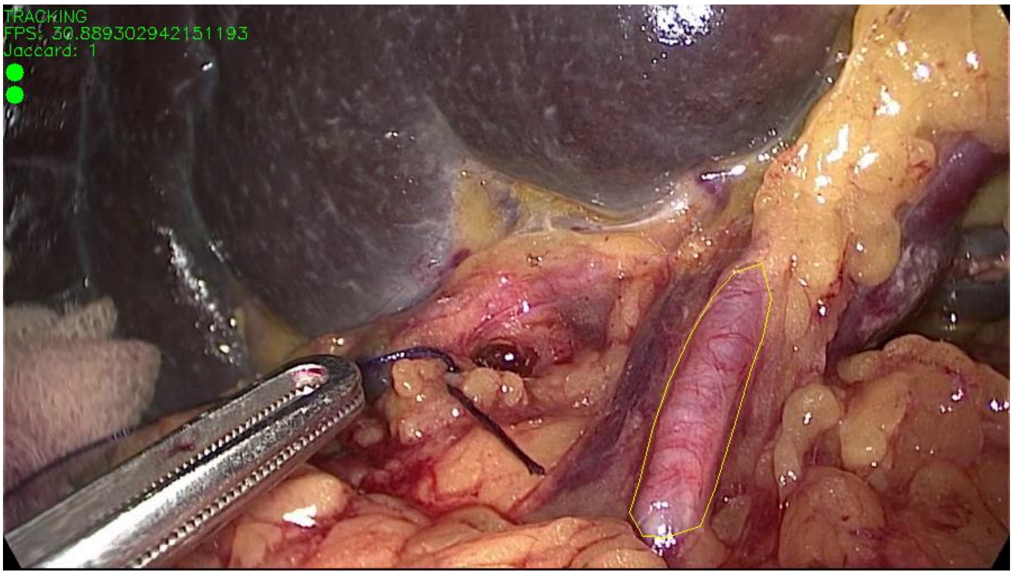


Figure 5.34 AC selection around artery

In this experiment, the tracked AC is not always visible so the table concerning AC visibility is shown in *Table 29*.

Table 29 AC visibility coverage in Experiment 5

SAV	PO	TO	OFV
15.30%	38.76%	16.11%	29.83%

As noted previously, this video sequence has been separated in parts. Three of which are evaluated and will be used in the experiment results, and three parts where the AC area is either totally occluded or out of field of view.

Table 30 Time-wise classification of video sub-sequences in Experiment 5

Timespan	**Length**	**Evaluated or Overlooked**
0 m 0 s – 0 m 22 s	22 s	Evaluated
0 m 22 s – 0 m 44 s	22 s	Overlooked
0 m 44 s – 1 m 17 s	33 s	Evaluated
1 m 17 s – 2 m 48 s	1 m 31 s	Overlooked
2 m 48 s – 4 m 0 s	1 m 22 s	Evaluated
Total	4 m	2 m 07 s evaluated out of 4 m

The overlooked parts do not affect the outcome regarding length of tracking, precision, recall, F score or Jaccard similarity scores. In order to provide clearer results, rather than different sub-chapters for each cut sequence of this video, they were concatenated, and the tests were done on a new video sequence which does not include long parts with AC out of field of view.

Precision, Recall, Accuracy and F1 Score

Precision, recall, accuracy and F1 Score will be calculated and compared in order to have some sense of precision, repeatability and false detection, These values, as previously stated, were calculated by using several thresholds for considering something as True Positive, with respect to the current ground truth (Chapter 5.2.4 Precision, Recall and Accuracy). Ground truth is defined at each 50 frames, which for this video results with 61 labeled frames from which the following calculations are acquired.

Table 31 Statistical results for Experiment 4

Threshold	*Precision*	*Recall*	*Accuracy*	*F1 Score*	*False Positive*	*False Negative*	*True Positive*
0.3	0.948	0.840	0.833	0.891	2	7	37
0.5	0.937	0.810	0.808	0.869	2	7	30
0.7	0.904	0.730	0.750	0.808	2	7	19
0.8	0.857	0.631	0.689	0.727	2	7	12

These results are not to be directly compared with the ones from previous experiments, because they were conducted in controlled environments. The high dynamics and mixture of noises in this experiment prohibits having such high and ‘perfect’ results.

These results only consider if a detection has happened and if it correct, in a very ‘binary’ manner. However, it gives no information on how the tracked AC compares to the GT. The derived Precision value is completely dependent on the chosen threshold coefficient. Using the lowest value, which is 0.3, provides very high scores for all three measures (Precision, Recall and F1 Score).

While using the highest and most realistic threshold for real world application, which is 0.8, provides somewhat lower results as presented in *Table 31*. FN and FP do not depend on this threshold, so the values of the metrics depend only on the threshold of TP here.

These are still comparable to the ones from previous experiments, even though the complexity and extensive outside effects give constant stress on the calculations on each new frame. This is another validation for the proposed method, and how precise and repeatable it is.

Tracked AC quality performance

In *Figure 5.35* it can be seen how the tracked AC differs from the GT. For this purpose, Jaccard similarity score (JSS) is used again to determine how much do the sets differ. It can also be observed that the maximum value is 1.0 and the lowest is 0.0.

Table 32 AC tracking quality in experiment 5

	Mean	Standard deviation	Minimum	Maximum
AC quality JSS	0.458	0.357	0.0	1.0

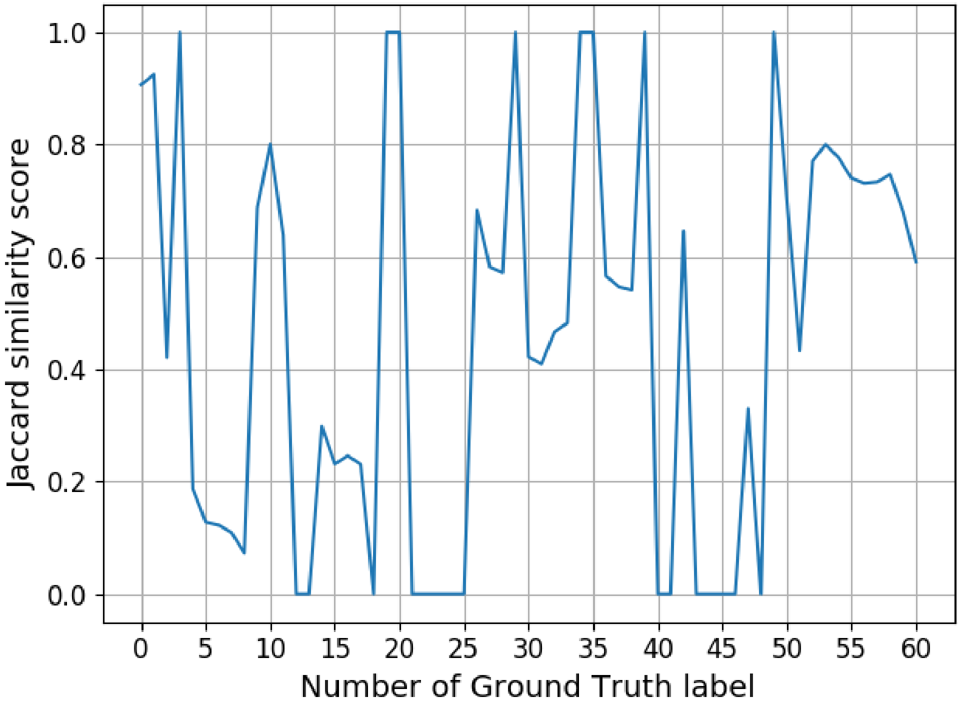


Figure 5.35 AC tracking quality in experiment 5

Scores range from 1 (perfect) to 0 (no common element), with a mean value of 0.458. This is affected largely by False Positive occurrences, because ones it appears it will be tracked by the algorithm until the re-initialization scheme decides to look for the correct AC again. The FP will still be tracked, even though the correct AC has become visible. The two largest portions, lasting 4 labeling time stamps each, with Jaccard score of 0 are due to FP being tracked even though the correct AC area is visible.

Number of active trackers

The environment provided in this experiment, covers every type of occurrence that is plausible to happen in a real-world surgery, such as partial and total occlusion, projection change, lighting change, fast dynamics, minor and major tissue deformations. These are the main reasons for great loss of trackers from LK tracking algorithm. In *Table 33* , the number of active trackers for LK method are shown at all times.

Table 33 Number of trackers in experiment 5

	Mean	Standard deviation	Minimum	Maximum
No. of active trackers	222.691	131.732	66.0	707.0

Similar to previous experiments with occlusions, here the number of active trackers is not always decreasing. Re-initialization happens when they drop below the threshold defined as 80% of the number of initialized LK trackers, which reactivates a new set of trackers.

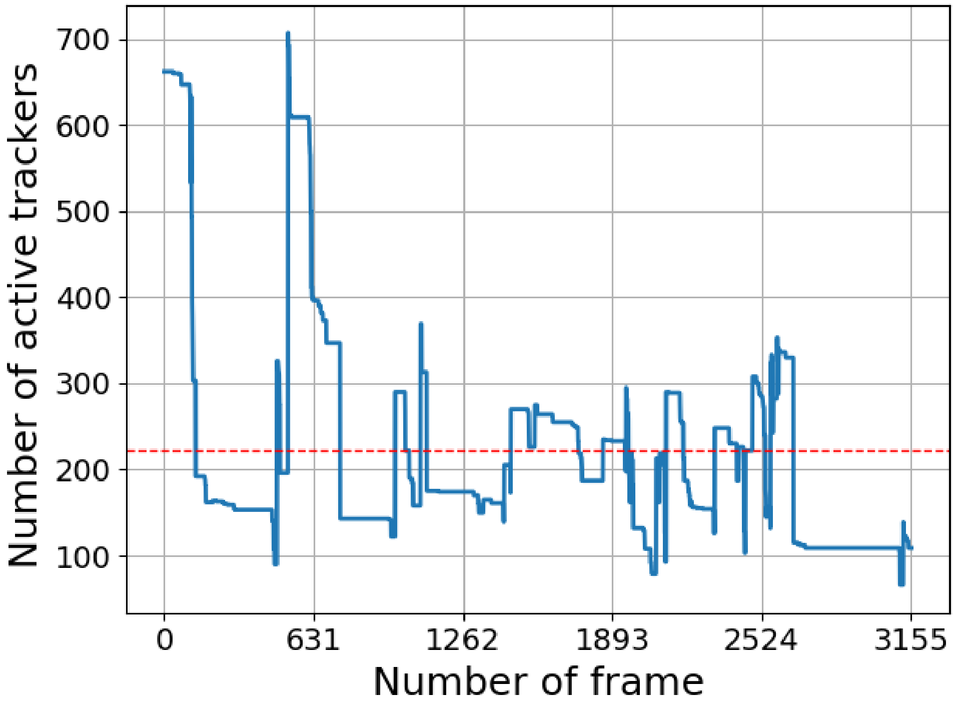


Figure 5.36 Number of active trackers in experiment 5

The evident lowering of the range of active trackers is due to the tracked tissue moving to the background or zooming out. By it becoming smaller, the AC area becomes smaller which eventually means a smaller number of active trackers used.

Program execution speed

The workstation used provided an average of 51 Hz, presented in *Table 34* ,with standard deviation of 8.543 Hz. The live video stream in a real-world surgical robot is at 25 Hz, or 0.04 s per frame sent. By looking at the average computational speeds, the method provides performance which allows a computational cycle to be done in 0.0196 s which is 0.0204 s before the next video frame arrives, thus making real-time implementation possible. However, there are several framerate low points which are directly connected to re-initialization process, as they happen due to that procedure. This low value (minimum of 0.642 Hz) is in a sense the bottleneck of this algorithm, as it will cause loss of intermediate frames.

Table 34 Program execution speed in experiment 5

	Mean	Standard deviation	Minimum	Maximum
Program execution speed [Hz]	51.276	8.543	0.642	71.034

These numbers lead to the conclusion that real-time application in this scenario is plausible.

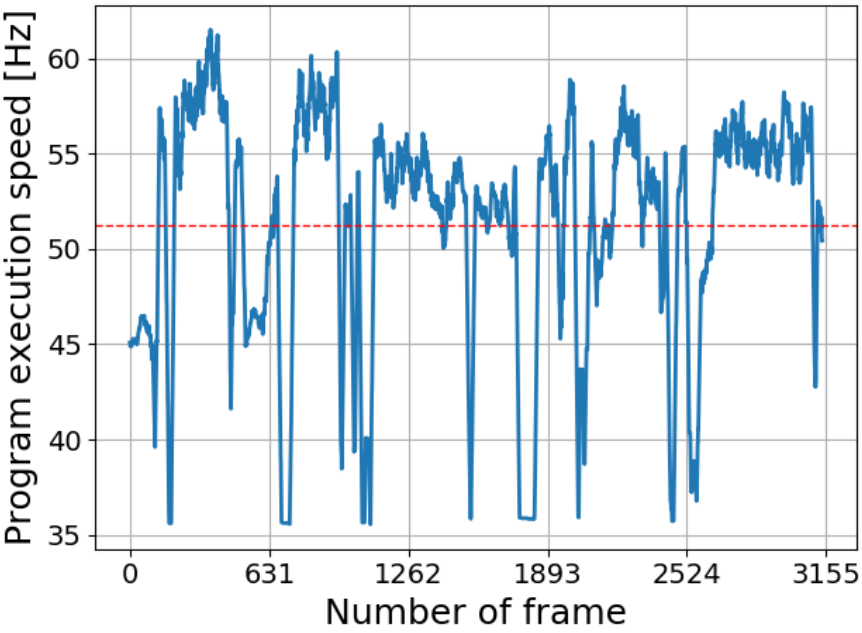


Figure 5.37 Program execution speed in experiment 5

Reliability index

When re-initialization happens, localization and AC redefinition techniques are applied as described in the previous chapter. In order to determine if they are reliable or not, the assumption mentioned in the previous chapter is yet again considered. It states that a well-posed redefinition of AC will not change much after binary erosion is applied. The metric used for this is yet again Jaccard similarity score.

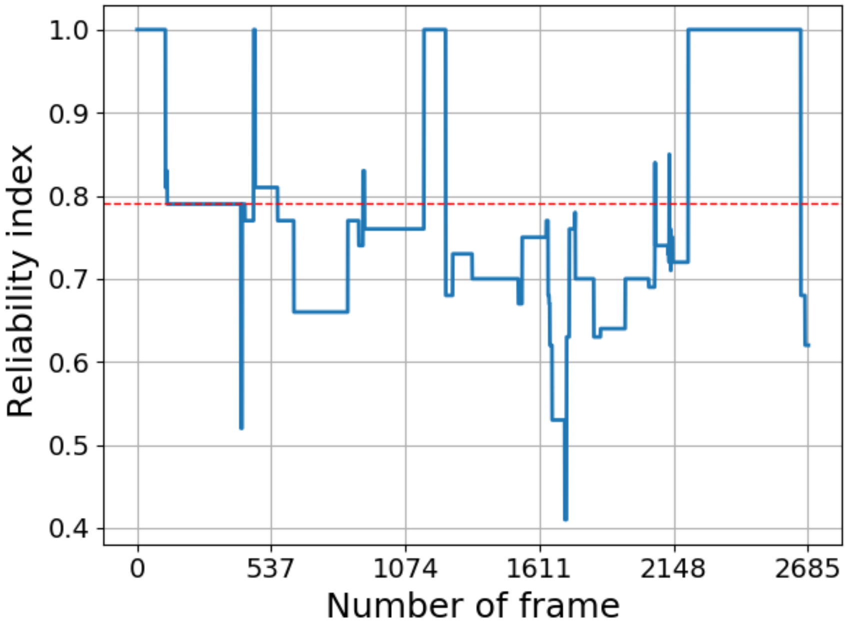


Figure 5.38 Reliability index in experiment 5

The mean value of this reliability index , which is Jaccard similarity score and is presented in *Table 35*, is 0.79 and the lowest achieved value is 0.41, while the standard deviation was at 0.13. These values verify that the proposed method for redefining the AC are working well and provide a steady and stable output for AC redefinition.

Table 35 Reliability index in experiment 5

	Mean	Standard deviation	Minimum	Maximum
Reliability index	0.790	0.131	0.41	1.0

Length of tracking phases

In *Figure 5.39* all tracking phases are shown with the length of each of them. There is one dominant maximum with 450 frames length of tracking, which adds up to 18.08 s of tracking without re-initialization. The mean value of all tracking phases is

6.302frames (See *Table 36*), while the lowest value is 1, when there is no tracking but rather only attempts to re-initialize. Accordingly, the sequential values of 1 are associated to occlusion.

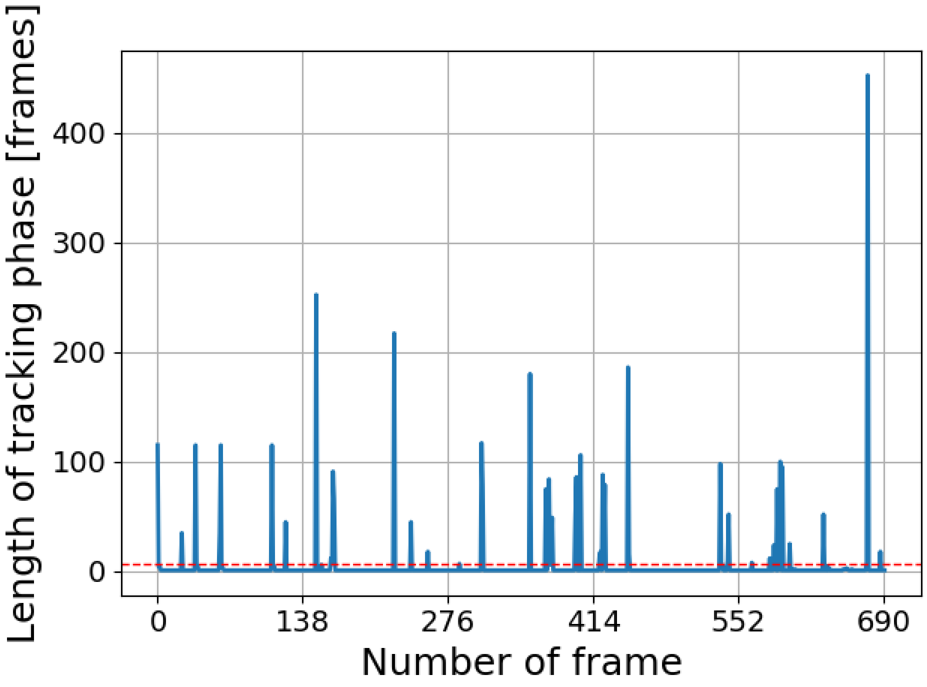


Figure 5.39 Length of tracking phases in experiment 5

Similar to the previous experiment, this one also covers re-initializations. With this, the period between successful re-initializations is a period called tracking phase. The length of these phases can give insight on how long it can track an AC definition when visible.

Table 36 Length of tracking phases in experiment 5

	Mean	Standard deviation	Minimum	Maximum
Tracking length [frames]	6.302	28.020	1	452
Effective tracking lengths [frames]	58.18	74.316	3	452

If all instances where the value of tracking length phase is below three are discarded, a graph will be acquired representing all phases where there has been effective tracking of at least 3 sequential frames, as shown in *Figure 5.40*. The mean value will rise to 58.18 sequential frames of tracking.

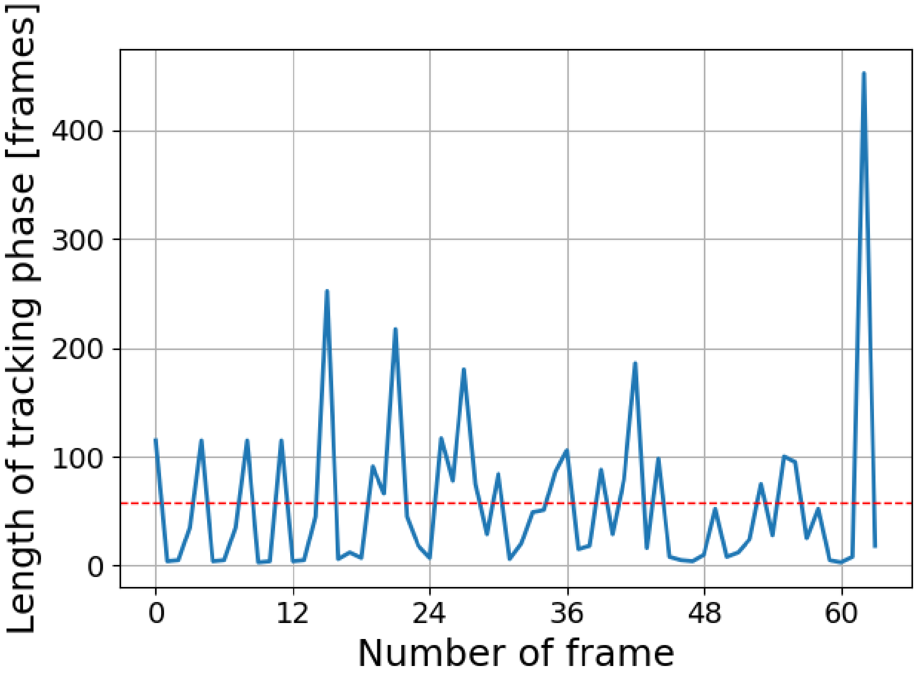


Figure 5.40 'Corrected' length of tracking phases in experiment 5

Chapter 6
Conclusions and discussion

In this thesis work, the development and implementation of real-time intra-operative tracking method of soft tissue during RAS is considered. The main challenges in computer vision for surgical robots were identified, addressed and broadly analyzed. After the base of the framework was proposed, several surveys were conveyed to find the most appropriate algorithm.

The main challenge was to create a small yet effective procedure that does not require pre-operative training. Unlike neural networks, which must be trained prior to application, such training phase cannot be applied to surgery as there is a patient that needs to be operated and it is not possible to train a network at that moment.

An extensive survey of feature detectors and descriptors was done prior to choosing KAZE as the detector of choice. The proposed method required several characteristic models upon which matching is done and series of different methods are applied to first localize, then provide an initial estimate of the AC description, and lastly to finalize the description using active contour snake and some binary filters. A specific law was developed to update the model Buffer in order to have a good model gallery from which re-initialization will happen. Morphological active contours were used in order take into account multiple aspects of the image they are applied on, and not only colors. The environment that is captured by the endoscope is a mixture of repeating colors and textures, and most importantly it difficult to for the computer to distinguish a salient object from the background. This lead to stability issues for Morphological snakes, and the answer to this final and critical step in terms of redefinition of AC was using HSV color space, in particular the Hue channel. The features were put through several tests for repeatability and consistency, where KAZE showed itself as the most reliable for this use. A novel approach of using edge detection mask for feature tracking is proposed, which allowed for significantly longer tracking sequences without the need for re-initialization of the AC. This approach, however, has the drawback of introducing drift effect where texture is lacking.

The choice of filters in the preprocessing phase was done to preserve and enhance edge definitions in order to increase texture where it was hard to be found. This approach proved itself to be highly effective.

As discussed previously, the priority in this thesis is providing an algorithm that would track AC area as long as it can and for when tracking failure happens, a fallback method was developed to redefine the AC.

The proposed framework provides the surgeon with several reliability indexes to assure the surgeon that the AC is being tracked successfully and how trustworthy it currently is, but also when the surgeon either needs to be more cautious because of bad AC tracking, or to manually request AC redefinition. This idea was discussed with the consultant surgeon and was regarded as very helpful and useful in a real-world surgery. The model buffer update scheme is one point where a lot of improvement can be done, in order to make sure that the models stored are always models only of the considered tissue and that they provide enough information for reliable and well-posed re-definition.

An experimental campaign was conveyed to create adaptive parameter definitions, which do not change much between processes, but these small modifications create a domino effect of failures if not chosen correctly.
The experiments were classified in two groups, which leads to a broader set of conclusions to be taken from them.

Table 37 Summary of experimental results [mean value (standard deviation)]

	Exp 1	**Exp 2**	**Exp 3**	**Exp 4**	**Exp 5**	**Total Averages**
Calculations per Second [Hz]	73.5 (5.93)	75.08 (6.81)	27.83 (3.37)	36.7 (3.38)	51.27 (8.54)	52.876
Jaccard Similarity Score	1 (0)	1 (0)	0.866 (0.05)	0.887 (0.051)	0.79 (0.13)	0.9086
Jaccard Similarity Score for AC tracking quality	0.78 (0.42)	0.73 (0.09)	0.711 (0.22)	0.68 (0.078)	0.458 (0.36)	0.67176
Tracking time [frames]	//	//	31.37	34.2	51.2	38.93
No. of active trackers	209.8 (3.16)	170.48 (3.8)	789.285 (261.46)	567.7 (125.06)	222.691 (131.74)	//

Regarding the final results provided in *Table 37*, it can be easily seen that an average computational speed of 52.876 Hz is achieved, while securing AC tracking quality with average of 67% similarity to hand drawn ground truth.

According to the total average, the reliability index of the AC redetection is very high, by estimating an average score of 0.9086 out of 1. Average tracking time through all experiments (excluding experiments 1 and 2, where there is no re-initialization and AC is tracked through the entire video) is 38.93 frames. There are long tracking instances in each dataset applied reaching 400 frames (16.2 seconds) of uninterrupted tracking. Furthermore, in *Appendix A*

Comparison of experimental results, a summary of comparisons between experiments with respect to a particular measure can be found.

Experiment 5, from the TrackVes dataset, was used in order to compare results to the work of Penza et al. [25] , where the high tracking performance Precision (0.85), Recall (0.6) and F1 Score (0.6) values have been calculated and will be considered as reference for assessing the proposed method from this thesis. By using the most strict TP threshold of 0.7, a Precision value of 0.846 is acquired, a Recall value of 0.61 and F1-Score of 0.71. In addition, real-time execution was maintained at average of 48.5 Hz.

Chapter 7
Open problems and future developments

Several issues have risen during the work on this thesis. The primary one is the risk of falsely localizing the considered soft tissue, which may cause injuries and blood loss in the OR. The two most promising approaches to improve this performance are:

- by using forward kinematics and knowledge of the movement of each joint from the endoscope’s robotic arm. The protected tissue is always in a globally static position, which leads to the possibility to use the link transformations[41] and with this to approximate the position of the AC area with respect to the scope of view. Once the localization problem is solved in a more ‘hardware’ approach, it leaves a lot of space to develop more advanced re-definition algorithms and deformation estimating ones.
- By using Infrared emitting markers to mark several points on an organ or region to be tracked. There will be no need of recognition, but it will be a problem of finding an appropriate mathematical description of the change that has happened. This, however, requires a second camera that can record only infrared light emitted from the markers. It has been implemented with great results by [17]. The only drawback is that if only a few of these markers are not visible, the procedure fails.

A combination of these two approaches is also a possibility, which will do most of the work regarding tracking, occlusion detection and deformation estimation without any high-level algorithms, and will also reduce computational load.

Regarding this thesis work, an algorithm for adaptive histogram thresholding was developed to segment out surgical tools from the frame. This algorithm gave promising results, but it was not included in the final version of this thesis due to the fact that it was possible to keep the surgical tools from being mistaken for AC reinitialization only by filtering out some parts of it. However, such segmentation algorithm for surgical tools will be of great help for filtering tools out of the tracking problem.

Further work must be also devoted to the following issues. The camera, though stereo, is a device that is easily affected by noise and can have reduced performance even with small disturbances. Another issue which must be mentioned is computational load. Any AC tracking algorithm must be precise, robust and must be able to have computational speed of above 25 Hz in order to avoid latency. It is usually a trade-off between quality and quantity, and in this case of AC tracking it is usual for proposed methods to have acceptable tracking performance but subpar computation times which makes them not suitable for real-time implementation in RAS. This trade off will be further investigated.

BIBLIOGRAPHY

[1] "SMARTsurg." [Online]. Available: http://www.smartsurg-project.eu/.

[2] J. Shah, A. Vyas, and D. Vyas, "The History of Robotics in Surgical Specialties," *Am. J. Robot. Surg.*, vol. 1, no. 1, pp. 12–20, Jun. 2014.

[3] U. Mezger, C. Jendrewski, and M. Bartels, "Navigation in surgery," *Langenbeck's Arch. Surg.*, vol. 398, no. 4, pp. 501–514, 2013.

[4] Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. Dimaio, "An Open-Source Research Kit for the da Vinci," *Proc. - IEEE Int. Conf. Robot. Autom.*, 2014.

[5] T. H. Tomkinson, J. L. Bentley, M. K. Crawford, C. J. Harkrider, D. T. Moore, and J. L. Rouke, "Rigid endoscopic relay systems: a comparative study," *Appl. Opt.*, vol. 35, no. 34, p. 6674, 1996.

[6] D. Stoyanov, M. V. Scarzanella, P. Pratt, and G. Z. Yang, "Real-time stereo reconstruction in robotically assisted minimally invasive surgery," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6361 LNCS, no. PART 1, pp. 275–282, 2010.

[7] D. Scharstein and R. Szeliski, "[Scharstein02 IJCV] A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithm.pdf," no. 1.

[8] J. Kowalczuk *et al.*, "Real-time three-dimensional soft tissue reconstruction for laparoscopic surgery," *Surg. Endosc.*, vol. 26, no. 12, pp. 3413–3417, 2012.

[9] H. Zhou and J. Jagadeesan, "Real-time Dense Reconstruction of Tissue Surface from Stereo Optical Video," *IEEE Trans. Med. Imaging*, pp. 1–1, 2019.

[10] N. Enayati, E. De Momi, and G. Ferrigno, "Haptics in robot-assisted surgery: Challenges and benefits," *IEEE Rev. Biomed. Eng.*, vol. 9, pp. 49–65, 2016.

[11] A. Abiri *et al.*, "Multi-Modal Haptic Feedback for Grip Force Reduction in Robotic Surgery," *Sci. Rep.*, vol. 9, no. 1, pp. 1–10, 2019.

[12] C. R. Wottawa, "An investigation into the benefits of tactile feedback for laparoscopic, robotic, and remote surgery," p. 165, 2013.

[13] S. A. Bowyer, B. L. Davies, and F. Rodriguez Y Baena, "Active constraints/virtual fixtures: A survey," *IEEE Trans. Robot.*, vol. 30, no. 1, pp. 138–157, 2014.

[14] M. Mohd Ali, N. N. Jaafar, F. Abdul Aziz, and Z. Nooraizedfiza, “Review on non uniform rational B-spline (NURBS): Concept and Optimization,” *Adv. Mater. Res.*, vol. 903, pp. 338–343, 2014.

[15] T. L. Gibo, L. N. Verner, D. D. Yuh, and A. M. Okamura, “Design considerations and human-machine performance of moving virtual fixtures,” no. June, pp. 671–676, 2009.

[16] N. V. Navkar, Z. Deng, D. Shah, K. Bekris, and N. V. Tsekos, “Visual and force-feedback guidance for robot-assisted interventions in the beating heart with real-time MRI,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2012, pp. 689–694.

[17] A. Shademan, R. S. Decker, J. D. Opfermann, S. Leonard, A. Krieger, and P. C. W. Kim, “Supervised autonomous robotic soft tissue surgery,” *Sci. Transl. Med.*, vol. 8, no. 337, 2016.

[18] P. Mountney and G. Z. Yang, “Soft tissue tracking for minimally invasive surgery: Learning local deformation online,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5242 LNCS, no. PART 2, pp. 364–372, 2008.

[19] D. Stoyanov and G. Z. Yang, “Soft tissue deformation tracking for robotic assisted minimally invasive surgery,” *Proc. 31st Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. Eng. Futur. Biomed. EMBC 2009*, pp. 254–257, 2009.

[20] S. Giannarou, M. Visentini-Scarzanella, and G. Z. Yang, “Probabilistic tracking of affine-invariant anisotropic regions,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 130–143, 2013.

[21] G. A. Puerto-Souza and G. L. Mariottini, “A fast and accurate feature-matching algorithm for minimally-invasive Endoscopic images,” *IEEE Trans. Med. Imaging*, vol. 32, no. 7, pp. 1201–1214, 2013.

[22] S. Duffner and C. Garcia, “PixelTrack: A fast adaptive algorithm for tracking non-rigid objects,” *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 2480–2487, 2013.

[23] V. Penza, E. De Momi, N. Enayati, T. Chupin, J. Ortiz, and L. S. Mattos, “EnViSoRS: Enhanced vision system for robotic surgery. A user-defined safety volume tracking to minimize the risk of intraoperative bleeding,” *Front. Robot. AI*, vol. 4, no. MAY, pp. 1–13, 2017.

[24] V. Penza, J. Ortiz, L. S. Mattos, A. Forgione, and E. De Momi, “Dense soft tissue 3D reconstruction refined with super-pixel segmentation for robotic

abdominal surgery," *Int. J. Comput. Assist. Radiol. Surg.*, vol. 11, no. 2, pp. 197–206, 2016.

[25] V. Penza, X. Du, D. Stoyanov, A. Forgione, L. S. Mattos, and E. De Momi, "Long Term Safety Area Tracking (LT-SAT) with online failure detection and recovery for robotic minimally invasive surgery," *Med. Image Anal.*, vol. 45, no. December, pp. 13–23, 2018.

[26] W. Kubinger, M. Vincze, and M. Ayromlou, "The role of gamma correction in colour image processing," *Eur. Signal Process. Conf.*, vol. 1998-Janua, no. 1, pp. 3–6, 1998.

[27] E. S. Gedraite and M. Hadad, "Investigation on the effect of a Gaussian Blur in image filtering and segmentation," *Proc. Elmar - Int. Symp. Electron. Mar.*, no. January 2011, pp. 393–396, 2011.

[28] G. Yadav, S. Maheshwari, and A. Agarwal, "Contrast limited adaptive histogram equalization based enhancement for real time video system," *Proc. 2014 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2014*, pp. 2392–2397, 2014.

[29] S. A. K. Tareen and Z. Saleem, "A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK," *2018 Int. Conf. Comput. Math. Eng. Technol. Inven. Innov. Integr. Socioecon. Dev. iCoMET 2018 - Proc.*, vol. 2018-Janua, pp. 1–10, 2018.

[30] A. B. and A. J. D. Pablo Fernández Alcantarilla, "LNCS 7577 - KAZE Features," pp. 1–14, 2012.

[31] B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision (IJCAI)," in *[No source information available]*, 1981, vol. 81.

[32] J. Bouguet, "Pyramidal implementation of the affine lucas kanade feature tracker," *Intel Corp.*, vol. 1, no. 2, pp. 1–9, 2001.

[33] H. P. Gavin, "The Levenburg-Marqurdt Algorithm For Nonlinear Least Squares Curve-Fitting Problems," pp. 1–19, 2019.

[34] M. Ester, J. Sander, H.-P. Kriegel, and X. Xu, "A Desity-Based Algorithm for Discovering Clusters," *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 169–194, 1998.

[35] V. Yeghiazaryan, I. Voiculescu, V. Yeghiazaryan, and I. Voiculescu, "Department of Computer Science An Overview of Current Evaluation Methods

Used in Medical Image Segmentation CS-RR-15-08 An Overview of Current Evaluation Methods Used in Medical Image Segmentation."

[36] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, Jun. 1986.

[37] Ho-Sik Roh and Jin-Oh Kim, "Manipulator modeling from D-H parameters," in *30th Annual Conference of IEEE Industrial Electronics Society, 2004. IECON 2004*, 2004, vol. 3, pp. 2480-2485 Vol. 3.

[38] N. Otsu, "{A} {T}hreshold {S}election {M}ethod from {G}ray-Level {H}istograms," *IEEE Trans. Syst. Man Cybern.*, vol. 9, no. 1, pp. 62–66, 1979.

[39] R. O. Duda and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," *Commun. ACM*, vol. 15, no. 1, pp. 11–15, 1972.

[40] Y. Wu, J. Lim, and M.-H. Yang, "Online Object Tracking: A Benchmark Supplemental Material," *2013 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1–13, 2013.

[41] S. Kucuk and Z. Bingul, "Robot Kinematics: Forward and Inverse Kinematics," 2006.

Appendix A
Comparison of experimental results

By using error bars (mean value and standard deviation), the results of all experiments will be compared with respect to the type of test.

Program execution speed

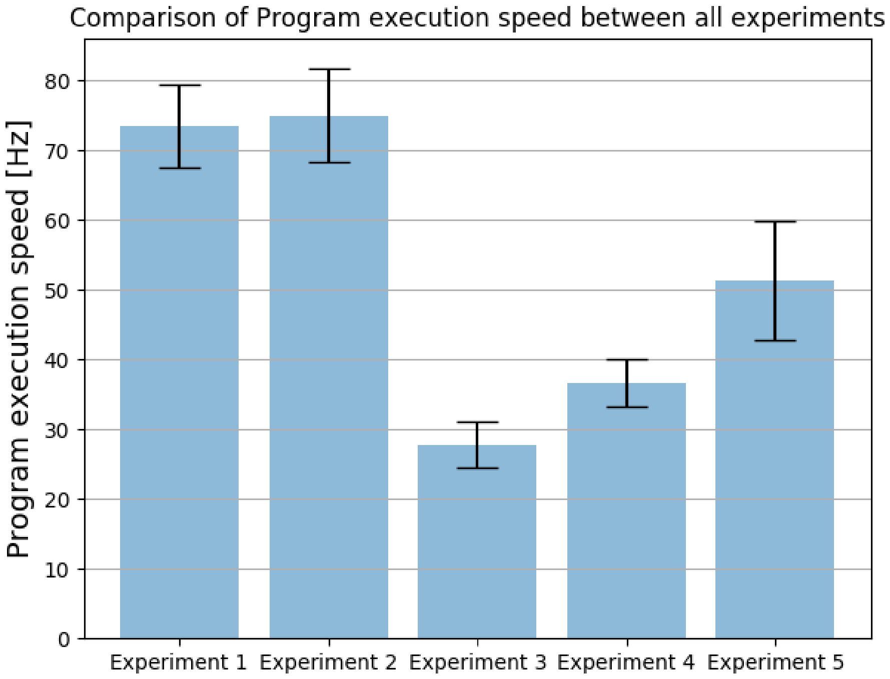


Figure 8.1 Error bar chart of program execution speed

Reliability index

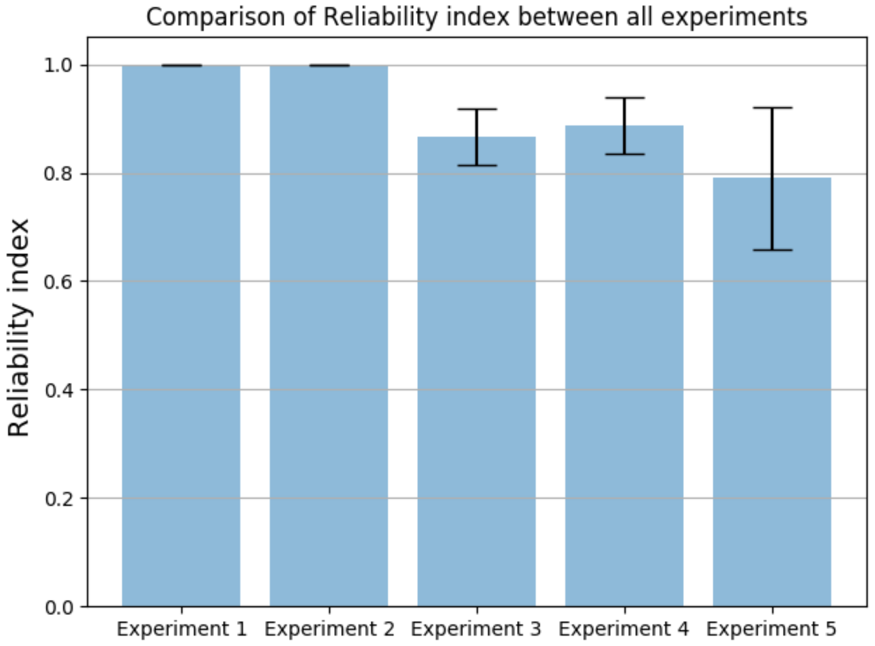


Figure 8.2 Error bar chart of reliability index

Number of active trackers

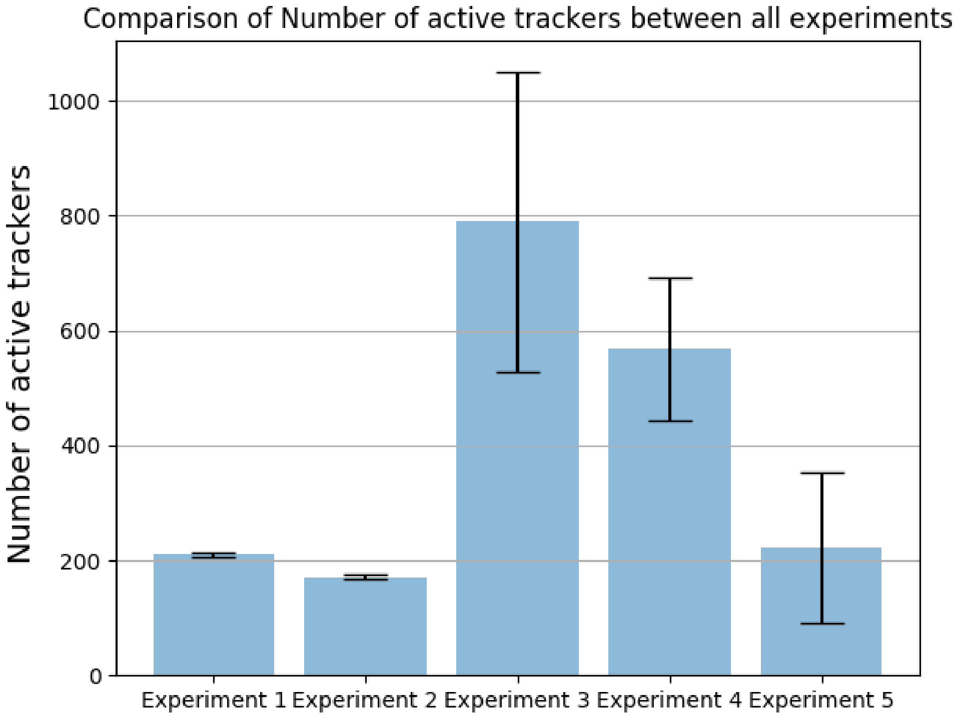


Figure 8.3 Error bar chart for number of active trackers

Length of tracking phase

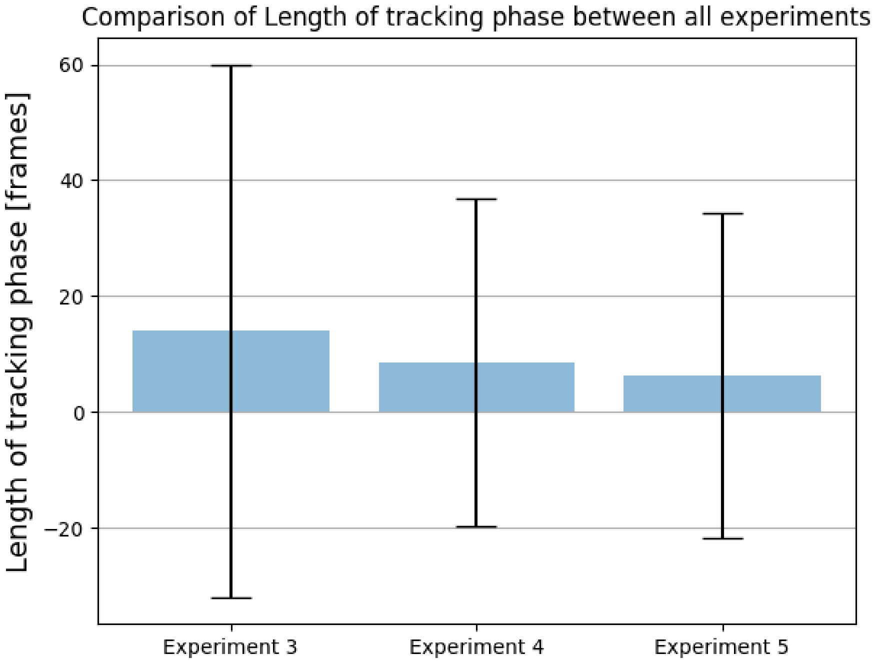


Figure 8.4 Error bar chart for length of tracking

Corrected length of tracking phase

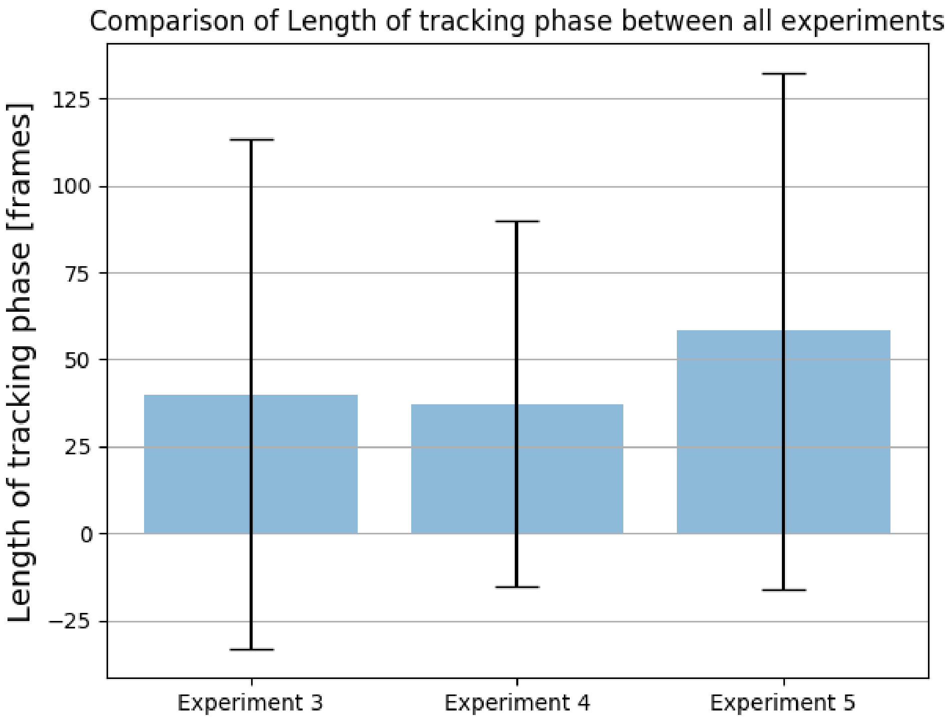


Figure 8.5 Error bar chart for corrected length of tracking