Image-Based Scene Analysis for Computer-Assisted Laparoscopic Surgery

Zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
der Fakultät für Informatik
Karlsruher Institut für Technologie (KIT)

genehmigte
Dissertation
von

Sebastian Bodenstedt
aus Goslar

Tag der mündlichen Prfung:	17. Juli 2017
Erster Gutachter:	Prof. Dr.-Ing. Rüdiger Dillmann
Zweiter Gutachter:	Prof. Dr. med. Beat Müller
Dritter Gutachter:	Prof. Dr.-Ing. Stefanie Speidel

Kurzfassung

Die Ziele von Systemen für die computerassistierte Chirurgie sind vergleichbar mit deren Autonavigation. Solch ein System soll den Chirurgen auf dem Weg eine Operation erfolgreich abzuschließen unterstützen. Eine Assistenz hierfür kann verschiedene Formen annehmen, wie beispielsweise eine Navigationsassistenz die mittels erweiterter Realität eine Tumorposition visualisiert oder ein System zur Messung von Längen im Körper. Weiterhin könnte ein solches System automatisch Geräteparameter im Operationssaal anpassen oder anzeigen welches chirurgisches Werkzeug als nächstes benötigt wird.

In der minimalinvasiven Chirurgie operiert der Chirurg mithilfe kleiner Instrumente und einer endoskopischen Kamera. Diese werden in den Patienten mittels kleiner Schnitte oder natürlicher Körperöffnungen eingeführt. Ein minimalinvasiver Eingriff im Abdominalbereich wird als Laparoskopie bezeichnet. Im Vergleich zu konventionellen chirurgischen Eingriffen bietet die Laparoskopie Patienten erhebliche Vorteile, wie z. B. kleinere Einschnitte und kürzere Krankenhausaufenthalte. Allerdings entstehen für den Chirurgen mehrere Nachteile, wie den Verlust des Tiefensehens und eine erschwerte Hand-Augen-Koordination.

Viele der Assistenzfunktionen bei einem computergestützten Eingriff sind nur während bestimmter Operationsabschnitte erwünscht oder aber müssen ihre Ausgabe im Verlauf der Operation anpassen. Damit ein Assistenzsystem ermitteln kann was für eine Assistenz aktuell benötigt wird, muss es kontextsensitiv sein. In anderen Worten, muss es dazu imstande sein den Fortschritt des aktuellen Eingriffes zu verfolgen.

Um dem Chirurgen eine kontextsensitive Assistenz anbieten zu können ist es erforderlich den aktuellen Status einer Operation in Echtzeit zu ermitteln. Neue Geräte und Sensoren im Operationssaal einzuführen ist schwierig, da bestehende Arbeitsabläufe beeinflusst werden könnten und auch Sicherheitsnormen eingehalten werden müssen. Da laparoskopische Eingriffe mithilfe eines endoskopischen Videosignals durchgeführt werden, kann dieses zur Informationsakquise verwendet werden.

Das Erkennen von Strukturen wie Instrumente und Organe in laparoskopischen Bilddaten ist eine schwierige Aufgabe, da das Aussehen der Instrumente als auch der Patientenanatomie mit einer hohen Varianz behaftet ist. Weiterhin stellen Artefakte wie Glanzlichter, Blut, Rauch und Verdeckungen eine Herausforderung dar. In einem chirurgischen Umfeld ist das Erstellen von großen Mengen an annotierten Trainingsbeispiel schwierig, da oftmals medizinisches Expertenwissen benötigt wird, um aufgenommene Daten korrekt zu annotieren.

Ein Ziel dieser Arbeit ist es deshalb Methoden zu entwickeln die es ermöglichen semantische und quantitative Information aus der laparoskopischen Szene in Echtzeit zu extrahieren, umso eine kontextsensitive Assistenz des Chirurgen zu ermöglichen.

Ein Schwerpunkt wird darauf gesetzt Methoden zu untersuchen, welche die Detektion von Strukturen, insbesondere chirurgischer Instrumente, mittels minimalistischen Annotationen ermöglichen. Weiterhin werden Methoden für die Workflowanalyse, ein wichtiger Bestandteil eines kontextsensitiven Assistenzsystems, vorgestellt.

Semantische Information, beispielsweise welche Instrumente und Organe gerade wo in der Szene sichtbar sind, werden für eine kontextsensitive Assistenz benötigt. Um eine korrekte Assistenz zu gewährleisten müssen die Positionen der Instrumente in Echtzeit extrahiert werden. Durch das Kombinieren eines Random Forest Klassifikators mit Farb- und Gradienteninformationen kann dies erreicht werden. Da sich Random Forests mittels einer GPU parallelisieren lassen, kann so eine Echtzeitverarbeitung des endoskopischen Bildflusses ermöglicht werden.

Diese Methode ist jedoch nicht robust gegen übliche Fehlerquellen in der Laparoskopie, wie z. B. partielle Verdeckungen oder sich überlapende Instrumente. Um den Einfluss dieser Fehlerquellen zu reduzieren, wird die Instrumentendektion mit einer Trackingmethode, basierend auf dem optischen Fluss, kombiniert. Sobald die Instrumente in der Szene erkannt wurden, wird der Instrumententyp festgestellt. Dazu werden Instrumentenregionen erst mittels Histogrammen über Farbwert und Sättigung und einen Bag-Of-Word Ansatz, der auf SURF Merkmalen basiert, beschrieben und daraufhin mittels eines Random Forests klassifiziert.

Ein Nachteil der laparoskopischen Chirurgie ist der Verlust des Tiefensehens. Dies erschwert dem Chirurg Distanzen abzuschätzen. Dies ist jedoch für Eingriffe, beispielsweise die Magenbypasschirurgie, unentbehrlich. Die quantitative Laparoskopie, welche es ermöglicht Größen und Distanzen von Strukturen in 3D zu ermitteln, kann hier zur Unterstützung genutzt werden. In dieser Arbeit wird eine Methode vorgestellt, die es ermöglicht Distanzen entlang einer Organoberfläche, mithilfe eines Stereoendoskopes, zu messen. Dafür werden zuerst das relevante Organ und die Instrumente in Echtzeit detektiert. Die Organoberfläche wird daraufhin rekonstruiert und als Graph modelliert. Die Distanz entlang der Oberfläche wird mittels der Methode von Dijkstra und den Instrumentenpositionen gemessen.

Die chirurgische Workflowanalyse spielt eine wichtige Rolle in der kontextsensitiven Assistenz, da sie gewährleistet, dass der Chirurg die richtigen Informationen zum richtigen Zeitpunkt erhält und so einen Informationsüberfluss vermeidet. Die bildbasierte Analyse des Workflows erfordert, dass Merkmale aus dem laparoskopischen Bildfluss extrahiert werden. Eine manuelle Auswahl der Bildmerkmale hat den Nachteil, dass nur Informationen, die dem Domänexperten bewusst sind, eingebracht werden. Andere relevante Charaktieristiken könnten verloren gehen. Convolutional Neural Networks (CNNs) können lernen welche Merkmale relevant sind um eine Aufgabe zu lösen. Der Nachteil von CNNs ist jedoch, dass sie eine große Menge an annotierten Trainingsbeispielen benötigen. Um dieses Problem zu umgehen wird in dieser Arbeit eine Methode vorgestellt, die es ermöglicht CNNs mithilfe von nichtannotierten Laparoskopievideos vorzutrainieren. Dazu wird einem CNN beigebracht zwei Videoausschnitte in die korrekte zeitliche Abfolge zu sortieren.

Dieses vortrainierte CNN wird erweitert, um eine chirurgische Phasensegmentierung, anhand annotierter Daten durchzuführen. Dabei wird eine rekurrente Netzwerktopolo-

gie verwendet, die es dem CNN ermöglicht bereits gesehene Informationen beizubehalten. Das CNN für die Phasensegmentierungen wurde außerdem abgewandelt, um eine Restdauerprädiktion durchzuführen. Hier konnte gezeigt werden, dass das Vortrainieren der CNNs die Ergebnisse erheblich verbessern konnte.

Um zu zeigen, dass die vorgestellten Methoden auch bei realistischen Szenarien funktionieren, wurden diese mehrfach anhand echter, laparoskopischen Bilddaten evaluiert. Weiterhin wurden die Methoden für die Instrumentendetektion und die Organvermessung in Liveversuchen auf Phantom- und Tierdaten evaluiert. Die Vermeißung wurde außerdem erfolgreich bei einer Magenbypassoperation im Menschen verwendet.

Abstract

The goal of computer-assisted surgery is very similar to that of a navigation system in a car, to provide information that will guide the surgeon towards successfully finishing the operation. This assistance could comes in many forms, such as providing navigation assistance via augmented reality (e.g. position of a tumor or a vital structure that should be preserved), measuring distances in the body, automatically adjusting parameters of devices or providing information on what surgical instruments might be required in the near future.

In minimally-invasive surgery, the surgeon generally operates using small instruments and an endoscopic camera, which are inserted into the patient using small incisions or natural orifices. Minimally-invasive operations in the abdominal cavity are referred to as laparoscopies. Compared to open surgery, laparoscopic interventions offer a great number of benefits for the patient, while the surgeon suffers multiple drawbacks, such as the loss of depth perception and an impeded hand-eye-coordination.

Many of the assistance functions that a system for computer-assisted surgery provides are only required at certain times or change in response to the progress of surgery. To ascertain what assistance is currently required, the system has to be context-aware or, in other words, needs to be aware of the progress of the current intervention and of the task that is surgeon is currently performing.

Providing the surgeon with context-aware assistance requires real-time information pertaining to the current state of the laparoscopic operation. Introducing new hardware and sensors into the operating room can be a difficult task, as existing workflows could be impacted and certain safety standards have to be guaranteed. Since laparoscopic surgeries are performed using the endoscopic view, making it an obvious choice for acquiring information on the current state of the operation. Knowing which instruments and organs are currently visible in the scene is a vital part of the current state.

Detecting structures such as instruments and organs in laparoscopic images is a difficult task, as one has to deal with a high variance in appearance of instruments, patient anatomy and endoscope optics, as well as artifacts such as specularities, smoke, blood and occlusions.

The focus of this work is therefore to develop methods that make it possible to acquire semantic and quantitative information from the laparoscopic scene that can be used for providing context-aware assistance to the surgeon during an operation in real-time. Furthermore, techniques for workflow analysis, a vital part for providing context-aware assistance, are introduced.

For a context-aware assistance, semantic information, such as what instruments and organs are currently visible and where they are located, is a prerequisite. To provide

on-time assistance, the instruments have to be located in real-time. This can be accomplished by classifying pixels based on features such color and gradient information. This method, while fast, is not robust against common errors that occur in laparoscopic videos, such as partially occluded instruments and overlapping instruments. To alleviate these problems, a tracking method is used to stabilize the detection of the instruments. Once the instruments in a laparoscopic image have been detected, the type of instrument are identified.

One drawback that laparoscopic surgeries entail for surgeons is the loss of depth perception. Thus estimating distances becomes difficult. Here quantitative laparoscopy, which provides numerical information on the size and distances of structures in the scene, can help to mediate this effect. In this thesis, a method for measuring distances along the surface of organs using a stereo endoscope was developed.

Surgical workflow analysis plays a vital role for context-aware assistance, as it ascertains that the right information is provided at the right time, avoiding sensory overflow of the surgeon and keeping the amount of required input to a minimum. Analyzing the surgical workflow from the laparoscopic image stream requires features to be extracted from the images in the stream. Manually selecting image features has the drawback that only information that the domain expert is aware of can be captured, other characteristics that might still contribute are possibly lost. Convolutional neural networks (CNNs) instead actually learn features that are relevant to solve a task, such as phase segmentation. The drawback of CNNs though is that they require a large amount of data for training. A method for pretraining CNNs on unlabeled videos from laparoscopic interventions was implemented.

This pretrained network was then extended to perform surgical phase segmentation on labeled interventions using a recurrent neural network topology, which makes it possible for the network to recall information. Furthermore, a modified version of the CNN for phase segmentation was used to predict progress of surgery.

To demonstrate that the proposed methods function in real-world scenarios, multiple evaluations on actual laparoscopic image data recorded from surgeries were performed. The proposed methods for instrument detection and organ measurements were successfully evaluated in live phantom and animal studies and also used during a live gastric bypass on a human patient.

Danksagung

Diese Arbeit entstand während meiner Zeit als wissenschaftlicher Mitarbeiter am Humanoids and Intelligence Systems Lab (HIS) des Instituts für Anthropomatik des Karlsruher Instituts für Technologie. Während meiner Promotion war ich Mitglied des Graduiertenkollegs 1126 “Intelligente Chirurgie” sowie des SFB/Transregio 125 “Cognition-Guided Surgery”. Bei dem Erstellen dieser Arbeit habe ich von vielen Personen Unterstützung erhalten, bei denen ich mich bedanken möchte.

Zuerst möchte ich mich bei meinem Doktorvater Prof. Dr.-Ing. Rüdiger Dillmann für seine Unterstützung, sein Vertrauen und für die hervorragenden und einzigartigen Arbeitsbedingungen am Lehrstuhl bedanken. Auch möchte ich mich bei Prof. Dr. med. Beat Müller, Leiter der Sektion “Minimal Invasive Chirurgie” an der Klinik für Allgemein-, Viszeral- und Transplantationschirurgie des Universitätsklinikum Heidelberg, für die jahrelange Unterstützung auf der klinischen Seite und für die Übernahme des Koreferats bedanken. Weiterhin möchte ich mich bei Prof. Dr. rer. net Bernhard Beckert, Prof. Dr.-Ing. Michael Beigl, Prof. Dr.-Ing. Jörg Henkel und Prof. Dr. rer. nat. Wolfgang Karl für ihre Mitwirkung als Mitglieder der Prüfungskommission bedanken.

Ein sehr großer Dank gebührt vor allem Stefanie Speidel. Sie hatte es schon zu Studienzeiten geschafft, mich nachhaltig für die Thematik der computergestützten Chirurgie begeistern. Dank dieser Begeisterung, ihrer Unterstützung und Führung wurde diese Arbeit überhaupt erst möglich. Auch Sebastian Röhl und Stefan Suwelack danke ich für ihre fachliche Unterstützung, Ideenreichtum und Geduld sowohl zu Studien- als auch Promotionszeiten. Bei Yoojin Azad, Michael Delles, Darko Katić und Enrico Kuhn bedanke ich mich für die super und lockere Büroatmosphäre, die vielen Diskussionen und die ganzen gemeinsamen Mittags- und Teepausen. Auch bei Daniel Reichard möchte ich mich für seine entspannte Art, die gemeinsamen Pausen, den regen Austausch an Ideen und die vielen Versuche bedanken. Ganz besonders möchte ich mich bei meinen “Leidensgenossen” Darko Katić für die ganzen Diskussionen (natürlich immer forschungsrelevant und höchst seriös), für seine Unterstützung und dafür das er mich immer auf den Boden der Tatsachen zurückgeholt hat (“Wolltest du nicht zwei IPCAI Paper schreiben?”) bedanken.

Weiterhin möchte ich mich bei Graziella Barbaro, Diana Becker, Christine Brand und Isabelle Wappler, den guten Seelen des HIS/H2T, für all ihre Geduld und Unterstützung bedanken. Auch allen Kollegen am H2T gebührt mein Dank. Hier möchte ich mich ganz besonders bei Christian Mandery und Ömer Terlemez für die vielen gemeinsamen Mittagessen und Diskussionen bedanken.

Ohne die Unterstützung auf medizinischer Seite durch die Gruppe für minimalinvasive Chirurgie am Uniklinikum Heidelberg wäre diese Arbeit nie geglückt. Hier möchte

ich mich bei Hannes Kenngott für sein endloses Ideenreichtum und Unterstützung, bei Benjamin “Amazing” Mayer für die Geburtshilfe des “Bowel Measurement System” und Patrick Mietkowski für seine Expertise in Sachen Workflow bedanken. Ein ganz besonderer Dank geht an Martin Wagner für sein riesiges Engagement und seine tatkräftige Unterstützung bei allen medizinisch-/chirugischen Themen.

In meiner Zeit am HIS hatte ich das Vergnügen viele großartige Studenten betreuen zu dürfen, die durch ihre harte Arbeit zu dieser Disseration beigetragen haben. Hierfür möchte ich mich bei Davuud Adigüzel, Matthias Eisenmann, Isabel Funke, Jochen Görtler, Carina Hansmann, Heiko Klare, Thomas Kornela, Sabine Kugler, Angie Neumann, Antonia Ohnemuss, Sinan Onogur, Pavo Orepic, Patrick Spengler, Laura Thämer und Pamela Wochner bedanken.

Ein weiterer großer Dank geht an den Stammtisch Baumann (Janine Altschuh, Ruben Baumann, Jannik Steinbring und Matthias Weidemann) für die Aufrechterhaltung meiner Moral und meiner pathologischen Leberwerte in allen Lebenslagen. Auch Fabian Laforet und Lisa Löbnitz möchte ich für die moralische Unterstützung und alle die Diskussionen, Curry-Queen-Besuche, Abendessen und Whiskeys in den letzten Jahren danken.

Last but not least möchte ich mich bei meinen Eltern Bernd und Birgit und meinen Bruder Alexander für ihre jahrelange Unterstützung und Ermutigung bedanken. Ihnen ist diese Arbeit gewidmet.

Karlsruhe, 2018 *Sebastian Bodenstedt*

Contents

1 Introduction

"In 200m, please turn right", nowadays it is difficult to imagine driving an unfamiliar route without the aid of a navigation system, whose directions guide us towards our destination. Similarly, surgeons often also operate in unfamiliar terrain, as each patient varies from others. The aim of computer-assisted surgery is very similar to that of a navigation system in a car, to provide information that will guide the surgeon towards successfully finishing the operation. Such an assistance could take many form, for example the position of a tumor or preoperative planning could be displayed via augmented reality. Further, the surgical staff could be preemptively informed what tools the surgeon might require in the next five minutes, thereby minimizing delays.

Laparoscopy is a form of minimally-invasive surgery performed in the abdominal region. The surgeon generally operates using small instruments and an endoscopic camera, which are inserted into the patient using small incisions or natural orifices. Compared to open surgery, laparoscopic operations offer a great number of benefits for the patient. As smaller incisions are used during laparoscopic surgery than during conventional operations, the postoperative pain experienced by the patient and also the risk of infection are greatly reduced. Furthermore, smaller incisions heal faster, resulting in a shorter stay in the hospital.

Due to the many benefits laparoscopic surgery provides for the patient, it has become the gold standard for many types of surgical procedures, such as gallbladder removal. On the other hand, such a mode of surgery entails multiple drawbacks for the surgeon. As an operation is performed via endoscope and a video screen, the surgeon loses the stereoscopic cues, which leads to a restriction of depth perception. Furthermore, the surgeon incurs an impeded hand-eye-coordination due to the long instruments.

In laparoscopy, the goal of a computer-assisted surgery system is to compensate some of the typical drawbacks. This assistance could come in many forms, e.g. providing navigation assistance via augmented reality, such as the position of a tumor or a vital structure that should be preserved, measuring distances in the body, automatically adjusting parameters of devices or providing information on what surgical instruments might be required in the near future. Many of these assistance functions are only required at certain times or change in response to the progress of surgery. One could easily supply the surgeon with every available parcel of information as soon as it becomes available, but this might lead to a sensory overflow, causing the surgeon to lose track of what is currently of importance. Asking the surgeon to actively select what information is currently required can result in a loss of concentration. A more advantageous approach would be to anticipate the needs of the surgeon and provide the right bit of assistance at the right time. To ascertain what assistance is currently required, the system has to be context-aware or, in other words, needs to be aware of the progress of the current operation and of the task that is surgeon is currently performing.

Providing the surgeon with context-aware assistance requires real-time information pertaining to the current state of the laparoscopic operation. Introducing new hardware and sensors into the operating room (OR) can be a difficult task, as existing workflows could possibly be impacted and certain safety standards have to be guaranteed. But since laparoscopic surgeries are performed using the endoscopic view, a video stream is always available during surgery, making it an obvious choice for acquiring information on the current state of the operation. Detecting structures such as instruments and organs in laparoscopic images is a difficult task, as one has to deal a high variance in appearance of instruments, patient anatomy and endoscope optics. Also, artifacts in laparoscopic images such as specularities, smoke, blood and occlusions can be challenging to deal with. Furthermore, collecting large amounts of training data in a surgical environment can be difficult, especially considering that often the knowledge of medical experts is required to correctly annotate the acquired data.

The focus of this work is therefore to develop methods that make it possible to acquire semantic and quantitative information from the laparoscopic scene that can be used for providing context-aware assistance to the surgeon during an operation in real-time. Here, this thesis places an emphasis on developing methods that allow the detection of structures in the surgical scene, especially surgical instruments using minimally annotated data. Furthermore, techniques that makes workflow analysis, a vital part for providing context-aware assistance, are introduced.

1.1 Research Questions

Context-aware assistance requires information on the current state of the operation to provide the right form of assistance at the right time. Furthermore, different forms of assistance also require information, such as where a certain organ is located and the type of instruments that are currently in use. The endoscopic video stream is a rich and readily available source of information on the progress and current state of a laparoscopic operation [LRBJ12]. A surgical scene can generally be characterized by objects contained in it and their relations, e.g. what organs, instruments and devices are visible and where are they located. In this work, the focus lies on extracting three different forms of information from endoscopic videos, semantic, quantitative and workflow information (figure 1.1) in real-time.

Semantic information in an image describes what objects are contained in the image and where they are located, while quantitative information provides numerical information on the size and distances of structures in the scene. Surgical semantic and quantitative image analysis generally operate on stand-alone images or on a small temporal neighborhood. Surgical workflow analysis, on the other hand, is a more holistic approach, where information from previously seen images is used to estimate the progression of surgery. This can be achieved on the basis of previously computed semantic and quantitative information, but also by considering more low-level image features.

Following research questions arise:

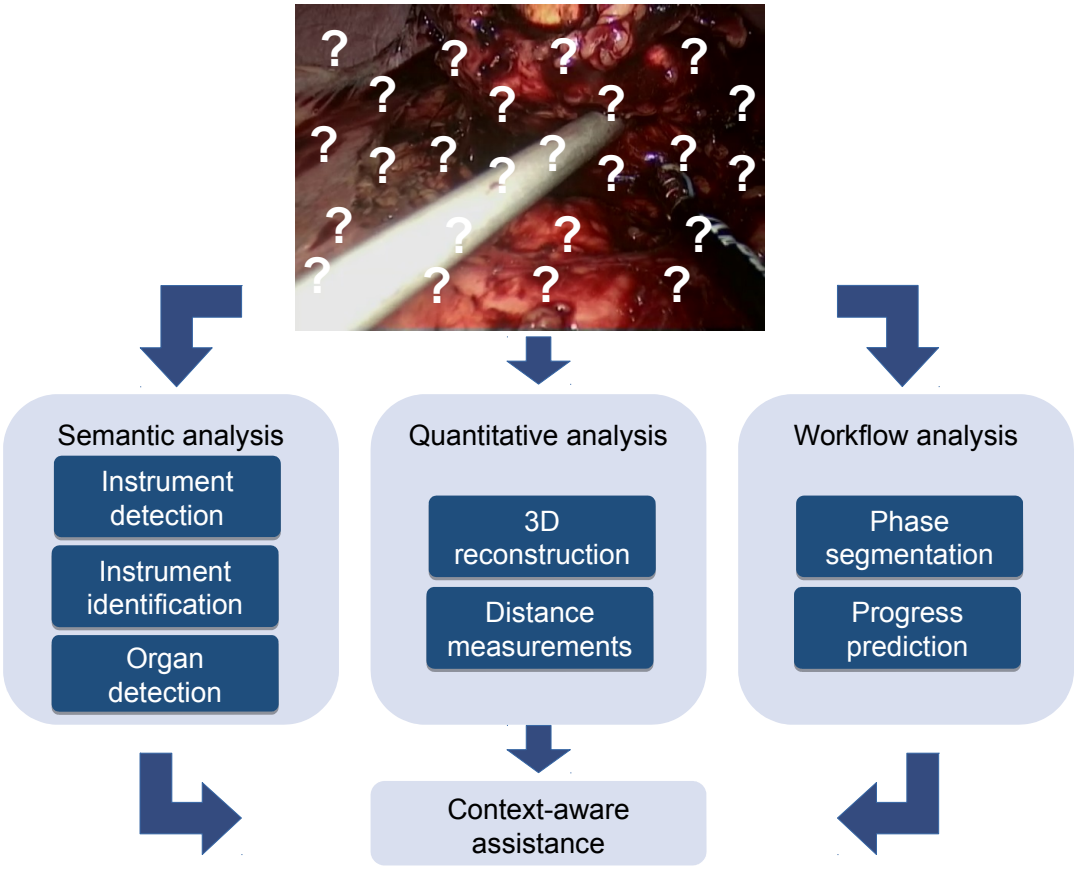


Figure 1.1: Overview of the concepts in this work and how they interact.

- **How can semantic information be extracted from a laparoscopic image?**

  Knowledge about the contents of the surgical scene, such as what instruments the surgeon is using and what organs are currently visible, is imperative to computer-assisted laparoscopic surgery as input to assistance functions. A part of this work is dedicated to determine which organs and instruments are currently occupying a laparoscopic scene. A large focus hereby is placed on detecting and identifying instruments. Depending on the type of assistance, real-time detection is required. As the impact on the surgical workflow is to be minimized, the instrument detection has to function without artificial markers. Furthermore, as a means to increase robustness, methods for propagating detections over time should be investigated.

  Methods for semantic image analysis require annotated training examples, which are time-consuming to generate and, especially in a surgical setting, often require expert knowledge. An emphasis is therefore put on investigating different annotation strategies on different levels of granularity.

- **What quantitative information can be derived from a laparoscopic image?**

  New forms of 3D endoscopes, e.g. stereo endoscopes, are currently becoming more readily available in the OR. These techniques allow 3D information to be

acquired from the surgical scene via 3D reconstruction, making it possible to reconstruct surfaces and compute 3D positions and distances between objects in the laparoscopic scene. The quantitative information collected from laparoscopic scenes can be incorporated into assistance functions. On example of such assistance would be to alleviate the effects the surgeon incurs due to the loss of depth perception, e.g. estimating distances from the laparoscopic video frame can be difficult, by performing automated measurements.

A goal of this thesis is to explore methods for measuring distance in the laparoscopic scene. The question arises how semantic information can be combined with 3D reconstructions, so that measurement on organ surfaces, e.g. hernia size or bowel segment length, in real-time can be accomplished in the surgical environment.

- **How can information on the surgical workflow be directly retrieved from the laparoscopic video stream?**

  Workflow analysis plays a vital role for context-aware assistance, as it ascertains that the right information is provided at the right time, avoiding sensory overflow of the surgeon and keeping the amount of required input to a minimum. Workflow analysis is also an integral part of planning the utilization of the OR, one of the most costly resources in a hospital. In this work, methods for analyzing the surgical workflow directly from the laparoscopic video stream are explored. Such methods generally require large amounts of annotated data, which is not always feasible to acquire in a surgical environment. To accommodate this, methods that require few annotated examples to generalize are explored.

1.2 Contributions

This thesis addresses the above mentioned research questions and introduces novel and efficient methods that allow semantic and quantitative image analysis as well as workflow analysis in context-aware computer-assisted laparoscopy. In particular, the following aspects are examined and solutions are presented:

- **Semantic image analysis: Real-time segmentation of instruments & organs**

  Knowing the locations of structures in the endoscopic environment is of utmost importance to many forms of computer-assisted surgery. To guarantee a timely response and that the provided information is current, important structures have to be located quickly. To this end, this thesis introduces novel methods that segment relevant structures, such as instruments and organs in real-time using simple and quickly computed features [BOK+15] [BGW+16] [BWM+16]. Different granularities of segmentation are explored and compared, with a focus on laparoscopic instruments [BOK+15] [BGW+16].

- **Semantic image analysis: Real-time instrument detection, identification & tracking**

The located instrument regions do not always describe an entire instrument, due to occlusions on the instrument shaft or false positives. A post-processing method to fuse some regions and remove others is implemented. Often, the assistance need differs depending on the type of instrument, therefore, instruments that have been detected in an image are identified with a novel instrument identification step [BOK+15]. Once instruments are detected and identified, tracking methods are used to propagate them into future frames to ensure robustness. Due to challenges specific to laparoscopic surgery, such as specularities, smoke, blood and overlaps, detecting and identifying instruments from scratch in each image can lead to erroneous results. The main idea here is to implement new methods capable of real-time performance.

- **Semantic image analysis: Image content classification on different granularity levels**

  Annotating images is a time-consuming task that often requires experts. Consequently, this thesis explorers different granularity levels of image content classification. For segmenting structures, novel approaches are presented and compared for both accuracy and performance speed [BOK+15] [BGW+16]. Often, knowledge of what objects in a scene and not their position is sufficient for certain tasks. A new method for automatically assigning labels to surgical images, based on the objects contained, is presented and evaluated. The method operates on image-wise labels, making finer, more detailed annotations unnecessary.

- **Quantitative image analysis: Intraoperative laparoscopic measurements**

  Previous works have shown that 3D reconstruction in a laparoscopic environment can be achieved in real-time. In this thesis, a focus is to provide the surgeon with a first intraoperative measurement tool, utilizing 3D reconstruction. Combining previously presented methods for semantic image analysis with a method for 3D reconstruction into one novel measurement tool, allows surgeons to perform 3D distance measurements on organs [BWM+16].

  This measurement system was first thoroughly evaluated in the laboratory, before being put into test in a clinical environment at the University of Heidelberg. Clinicians succesfully tested the system in phantom, ex- and in-vivo porcine trails, before finally utilizing the system successfully during a first-in-human study.

- **Workflow analysis: Reduction of required annotations**

  Correctly interpreting the flow of a laparoscopic surgery is imperative to context-aware assistance, as otherwise the wrong information might be displayed. Methods of workflow analysis generally require large amounts of annotated data. In this thesis, a novel approach for utilizing unlabeled surgical videos to pretrain a convolutional neural network is presented and evaluated on two surgical workflow related problems [BWK+17].

- **Workflow analysis: Laparoscopic phase segmentation & progress predication**

The most common form of surgical workflow analysis is phase segmentation. Here a surgical operation is divided in different phases, which usually correspond to a treatment of a certain structure. To evaluate the pretrained convolutional neural network for workflow analysis, a modification was made to extend it to phase segmentation [BWK+17]. This new approach is evaluated on two types of laparoscopic surgeries. As phase are often exclusive to a certain type of operation, a method for predicting the progress of surgery from different types of operations is presented and evaluated. Here, unlabeled operations are used [BKW+17].

1.3 Outline

The following thesis is divided into seven chapters. These chapters contain the following:

- **Chapter 2** gives an insight into the fundamentals of laparoscopic surgery and the problems surgeons face in the operating room during laparoscopic surgery. This insight is relevant to the following chapters.
- **Chapter 3** provides an overview of the current state of the art relevant to this work. The focus here lies on methods for image-based semantic, quantitative and workflow analysis for context-aware computer-assisted laparoscopic surgery.
- **Chapter 4** introduces methods for detecting objects of interest, e.g. surgical instruments or organs, in laparoscopic images. The main focus in this chapter lies on detecting and identifying laparoscopic instruments. Once detected, the instruments are tracked over time.
- **Chapter 5** builds upon the methods introduced in the previous chapter by combining them with 3D stereo reconstruction, resulting in a system that allows automatic organ measurements in the laparoscopic scene.
- **Chapter 6** focuses on analyzing the surgical workflow. The chapter introduces a method for unlabeled pretraining neural networks to comprehend the surgical workflow and then extends the method for surgical phase and progress predication.
- **Chapter 7** summarizes the presented work and provides an outlook onto future works and challenges.

2 Laparoscopic Surgery

In contrast to conventional surgery, minimally invasive surgery, also called keyhole surgery, is a form of surgery performed through small incisions into the body of the patient or through natural orifices. Through these holes, often called ports, small instruments are inserted via trocars, allowing the surgeon to manipulate organs. Furthermore, through one of the holes a camera, the so-called endoscope, is inserted. The endoscope is connected to a monitor, allowing the surgeon and surgical staff to see the surgical site and to steer the instruments [ARJK+12]. Minimally invasive operations in the abdominal region are generally referred to as laparoscopies or laparoscopic surgeries. In laparoscopy the abdominal region is insufflated with an inert gas, e.g. CO_2, to provide adequate space for the surgery. The medical term for the insufflated abdominal region is pneumoperitoneum. An illustration of an endoscopic setup can be seen in figure 2.1.

Minimally invasive operations offer many benefits, such as minimized scarring, reduced risk of infection, faster recovery and lower costs. This has caused minimally invasive operations to become the gold standard for many forms of surgical treatment, such as gallbladder removal.

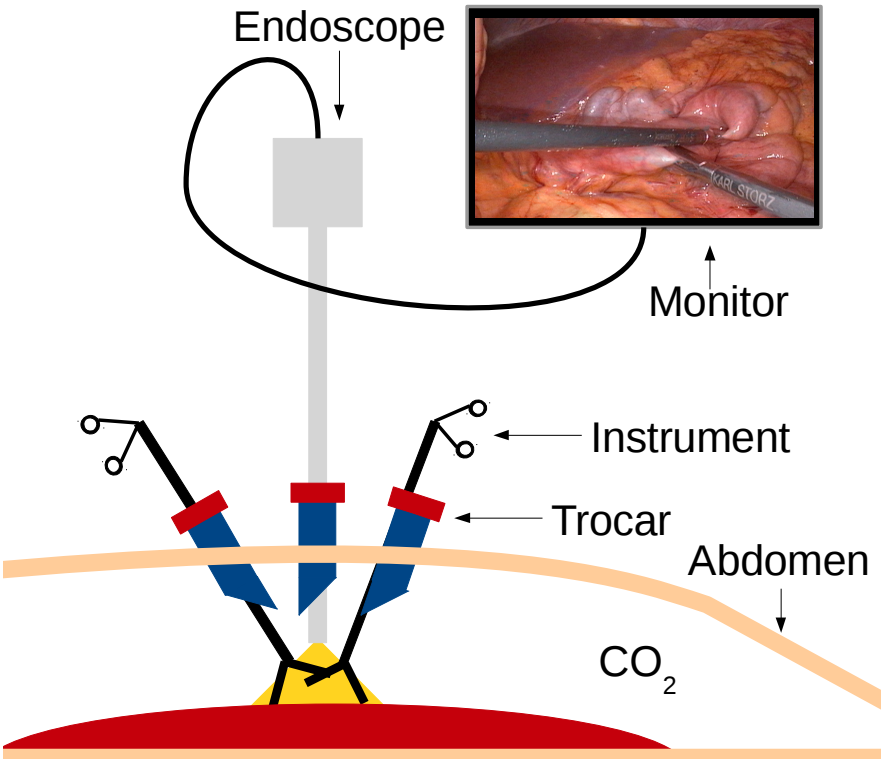


Figure 2.1: Typical setting during minimally invasive surgery.

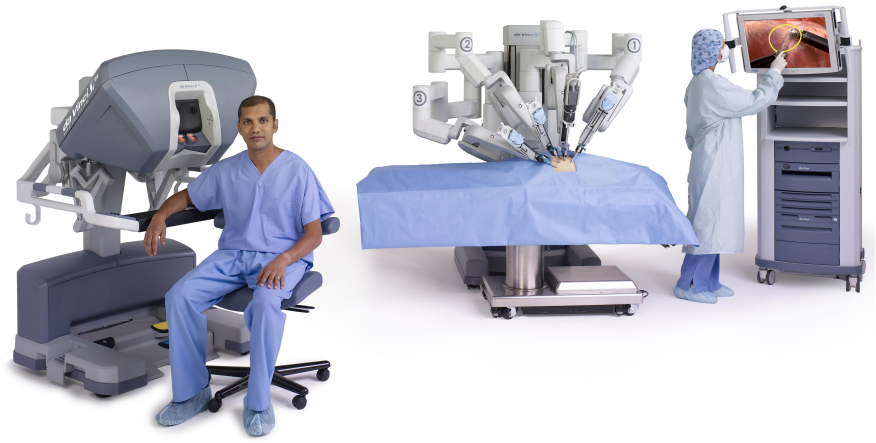


Figure 2.2: The da Vinci surgical system. ©2017 Intuitive Surgical, Inc.

On the other hand, minimally invasive operations can prove to be challenging to the surgeon. A surgeon incurs multiple handicaps, such as a loss of both depth perception and haptic feedback. As the surgeon is operating with tools that are inserted into the body through a port, the direction of motion of the tool is reversed, due to the fact that the port acts as a pivot. The motions of the tools used in minimally invasive surgery are also restricted by the port, reducing the degrees of freedom of movement. Since the surgeon does not have a direct line of sight onto the site of the operation, orientation can be difficult. These issues make minimally invasive surgery more complex for the surgeon to perform than a conventional surgery [GJ00].

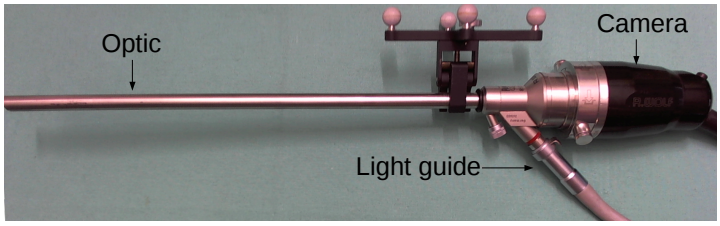
The aim of surgical assistance systems is to address some of the drawbacks previously mentioned. For example, the da Vinci from Intuitive Surgical is a tele-operator that allows the surgeon to operate from a console (see figure 2.2). It transfers the surgeon's hand movements onto the instruments of the tele-operator, allowing a more intuitive control with a wider range of motion. Furthermore, it restores the surgeon's depth perception, due to a stereo endoscope combined with a 3D display.

2.1 Laparoscopic Tools

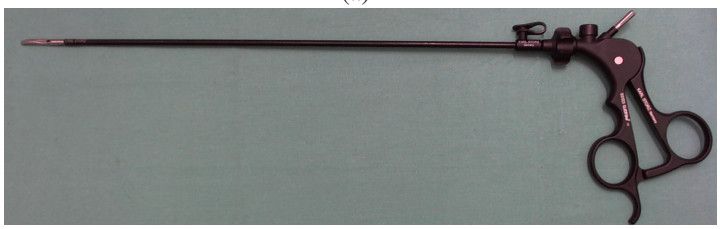
A wide range of surgical tools are used for laparoscopic surgery. The endoscope allows the surgeon to see inside the body, allowing manipulation with other instruments to take place.

2.1.1 Endoscope

Endoscopes are the most important source of information during laparoscopic surgery. They allow the surgeon to see the site of surgery while keeping the sizes of incisions to a minimum. Laparoscopic operations are generally performed using rigid endoscopes (see figure 2.3(a)). Flexible endoscopes exist as well and are commonly used for examining the interior of organs like the intestines or the esophagus.



(a)



(b)

Figure 2.3: Laparoscopic tools: (a) endoscope, (b) laparoscopic instrument

Rigid endoscopes generally have a length of 20 to 40 cm and a diameter of 2 to 12 mm. Up until recently, endoscopes, like the one seen in figure 2.3(a), almost exclusively had a lens in the tip and the camera at the other end of the shaft. The shaft contains a rod lens optic system that projected the image from the tip of the endoscope to the camera in the back. New types of endoscope containing the camera chip on the tip of the endoscope are already available, though the majority of endoscopes still use rod lenses. The endoscope is also connected to a cold light source in order to illuminate the scene of surgery [Röh13].

Stereo endoscope

Stereo endoscopes are a variation of the standard endoscope that contain a second lens on the tip and a second system of rod lenses (see figure 2.4). The camera generally consists of two imaging chips. New stereo endoscope systems with two separate imaging chips on the tip exist as well. The distance between the lenses is usually between 1 to 5 mm [Röh13].

Stereo endoscopes allow the surgeon, in combination with the appropriate display method, to perceive depth while performing the surgery. For example, the surgical console of the da Vinci tele-operator contains two displays, one for each eye of the surgeon. Each display is connected to one camera of the stereo endoscope. 3D monitors are also being used more frequently. These monitors allow the entire surgical staff to view the stereo images using glasses with, for example, polarized filters.

2.1.2 Instruments

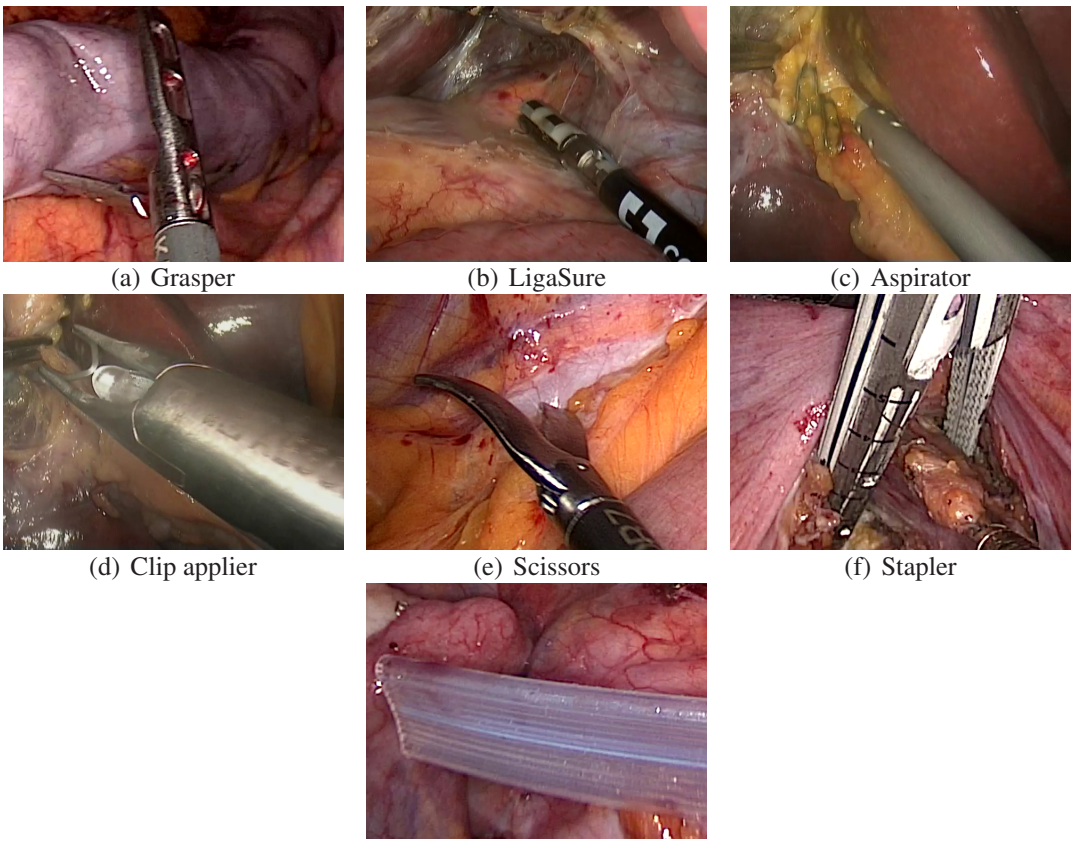
Laparoscopic operations require specialized tools that can be inserted into the abdominal cavity via trocars. Commonly they take a similar form as the example presented in figure 2.3(b). On one end of the instrument is a handle that allows the surgeon to control the clasper on the other end of the tool. A long shaft connects the two ends of

Figure 2.4: The stereo endoscope of the da Vinci surgical system. ©2017 Intuitive Surgical, Inc.

the instrument. The diameter of the instruments generally varies between 1.8 to 12mm [Röh13]. Most instruments allow the clasper to be opened and rotated.

A wide range of instruments for laparoscopic surgery exists. The instruments and other tools mostly commonly used in the types of surgery relevant to this work are:

- **Trocar**: A trocar is a tube with a seal that is inserted into the abdomen via incisions during laparoscopic surgery. They are used to insert instruments and the endoscope into the abdomen. The trocars form an airtight seal that keeps the pneumoperitoneum intact.
- **Grasper**: An instrument used for grasping, fixating and extracting organs, parts of tissue or other objects. Many variation of the grasper exist, which often vary in size, shape and grasping force (figure 2.5(a)).
- **LigaSure**: The LigaSure is an electric, bipolar instrument for dividing and sealing tissue. The instrument is most often used for preparation of tissue and for dissection (figure 2.5(b)). Since an electric current is used for dividing tissue, the LigaSure simultaneously cauterizes the tissue.
- **Aspirator**: An instrument used for irrigation and washing with a liquid, usually a saline solution, inside the body cavity. The instrument can also be used for evacuating debris and liquids (figure 2.5(c)).
- **Clip applier**: Surgical clips are used to seal vessels, such as blood vessels, ducts and other structures, during surgery. Two types of clips, metallic and absorbable, exist. The clip applier is an instrument used to insert clips. The instrument holds a magazine of surgical clips (figure 2.5(d)).
- **Scissors**: Surgical scissors are used for separating and preparing tissue, vessels and ducts, but also for cutting suture materials (figure 2.5(e)).
- **Stapler**: The stapler is an instrument used to insert and place surgical staples into tissue in the abdominal cavity. The staples can be used for connecting divergent



(a) Grasper (b) LigaSure (c) Aspirator

(d) Clip applier (e) Scissors (f) Stapler

(g) Silicone drain

Figure 2.5: Example images of the different instrument types.

vessels, but also to seal vessels. Some staplers contain a knife for separating tissue. This makes it possible to separate vessels, while keeping both halves sealed, e.g. when dividing the stomach during gastric bypass (figure 2.5(f)).

- **Silicone drain**: Drains are usually placed during surgery to drain body fluids that can accumulate during surgery and also recovery (figure 2.5(g)).

2.2 Relevant types of Laparoscopic Surgery

Laparoscopy is a versatile technique for abdominal surgery that is used for many different procedure types. In this section, three different laparoscopic procedure types, which are relevant to the remainder of this thesis, are briefly described. The reader is advised to peruse the listed sources for further details.

2.2.1 Gallbladder removal

The surgical removal of the gallbladder, also called cholecystectomy, is a procedure for which laparoscopy is currently the gold standard. Indications for gallbladder removal are, for example, gall stones and infection.

During surgery, the gallbladder is located and retracted to reveal the cystic duct and artery and the hepatic duct. This area is dissected and the cystic duct and artery are clipped and cut. The gallbladder is then freed from the liver bed and removed via one of the trocars or ports [PEI+91].

2.2.2 Colorectal surgery

Another type of surgery that is frequently performed via laparoscopy are operations of the rectum and the colon. For this work, the proctocolectomy, the removal of the rectum and parts of the colon, is relevant. Indications for a proctocolectomy are, for example, chronic inflammation, due do colitis or Crohn's disease, and cancer.

During surgery, first lymph nodes and the blood vessels connected to the sigmoid colon are dissected. After this, the colon is mobilized by dissecting the tissue that adheres it to the peritoneum, i.e. the inside of the abdominal cavity. Next, the rectum is mobilized and divided. The part of the rectum connected to the intestine is then extracted via a port and divided outside the abdominal cavity. The intestine is then re-inserted and connected to the stoma, a small opening in the abdomen, to facilitate healing. Later, the intestine can be reconnected to the remaining part of the rectum [LRGCM+07][TP95].

2.2.3 Gastric bypass

A gastric bypass is a type of bariatric surgery, a form of surgery that focuses on treating morbid obesity. Common indications for gastric bypasses are morbid obesity and type 2 diabetes.

During laparoscopic Roux-en-Y gastric bypass surgery, the most commonly performed gastric bypass, the stomach is first divided into a small upper pouch, which is connected to the esophagus, and a larger lower pouch, which is connected to the small intestine. The small intestine is then divided approximately 75 cm below the lower stomach pouch. After further 150 cm of small intestine, the so-called Roux limb, the piece attached to the lower stomach pouch is reattached to the remaining small intestine, forming a Y. The Roux limb is then connected to the upper stomach pouch [VS14] [MHT08].

The smaller stomach reduces the amount of food consumed by the patient, while the Y-configuration reduces the absorption of nutrients. Both factors contribute to weight loss.

2.3 Challenges for the Surgeon

As previously outlined, laparoscopic surgery has certain drawbacks for the surgeon. This section will give a brief overview on the most significant challenges [Röh13] [Feu07] [Ken10] [vdPGJD08] [GMM+98]:

- **Restricted range of motion**: Due to the trocar, the degrees of freedom of both instruments and the endoscope are reduced to one degree of freedom for translation and three degrees for rotation. Furthermore, the trocar acts as a pivot, reversing the movement direction. Both these effects make it difficult to reach certain areas of the surgical site.
- **Reduced field of view**: Since the motion of the endoscope is also restricted, only a small area of the surgical site can be seen at once, impeding the surgeon's ability to fully overview the site. The reduced view makes it difficult to ascertain complications if they occur away from the camera center.
- **Loss of haptic feedback**: The tactile feedback the surgeon receives from the instruments is reduced due to the length of the instrument shaft and friction in the trocar. This makes it difficult to distinguish tissue types by differences in tactile sensation.
- **Difficult hand-eye-coordination**: The length of the instruments, the loss of depth perception and the reduced motion make it difficult to successfully direct the instruments. A not ideally aligned sight of the endoscope can also contribute to this.

2.3.1 Depth perception & laparoscopic measurements

A further challenge that the surgeon faces is the loss of depth perception when operating via monitor. The loss of depth perception makes estimating distances somewhat difficult for the surgeon. This can be a drawback as surgeries like gastric bypasses or hernia repairs require an accurate estimation of distances, such as the length of a segment of bowel or the size of a hernia. For instance, during a Roux-en-Y gastric bypass is relocated 70 cm down the small intestine. To measure this distance, the surgeon grasps a length of bowel with two laparoscopic instruments and iteratively moves the bowel past the camera (figure 2.6). For each iteration, the surgeon estimates the length of bowel between the instruments using the camera image. Due to the loss of depth perception, these estimates can be erroneous.

In literature [SIH+03] two more standardized approaches to bowel length measurement are described: First, marks on the instrument at a predefined distance (5 or 10 cm) are used as a ruler. Second, an umbilical tape of the proposed length (70 cm) is introduced into the abdominal cavity as a flexible ruler. These methods, however, integrate poorly into the surgical workflow as they do not reflect the usual way of grasping the bowel and iteratively moving along its length. One focus of this thesis is to propose a method for automatically measuring the relevant distances during surgery.

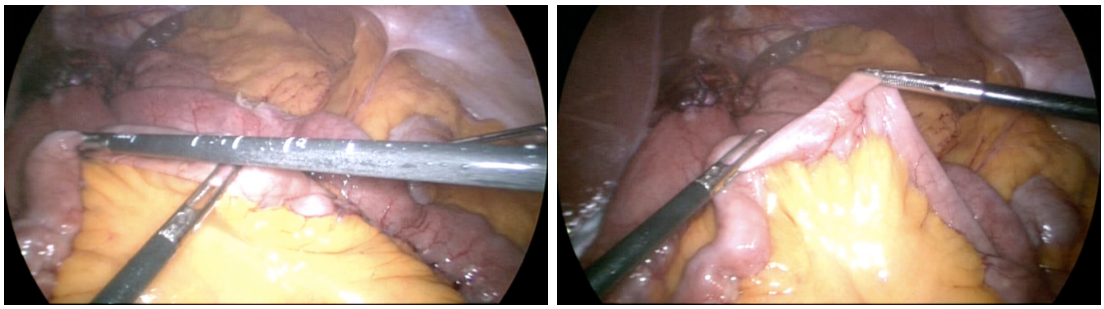


Figure 2.6: During a Roux-en-Y gastric bypass, the surgeon uses two instruments to pass the bowel iteratively past the camera, estimating the distance between instruments at each iteration, to measure a distance of 70 cm starting from the duodenum so that the stomach can be reconnected to the small intestine at that location [BWM+16]

.

3 State of the Art

The aim of a context-aware computer-assisted surgery system (CAS) is to provide the surgeon with the right type of assistance at the right time. In laparoscopic surgery, such systems aim to compensate for some of the drawbacks typical to laparoscopy, such as the limited field of view or difficult orientation in the abdominal cavitiy (see section 2.3), by e.g. providing assistance during navigation. Often, this assistance is provided via augmented reality [SSF+07] [KWG+13], which in laparoscopic surgery does not require a complicated setup as it does in open surgery, such as using mirrors [FDM+05] or see-through glasses [KCCO+11].

Context-aware implies that the system has to be aware of the progress of surgery. Knowledge about the progress can be derived from the laparoscopic video feed. Semantic image analysis methods are one possibility to describe the content of laparoscopic images. An overview of current methods, with a focus on detecting and identifying surgical instruments, is given in section 3.1.

Quantitative image analysis methods, on the other hand, provide 3D distances and measurements from image and video data. 3D distances provide information on the spatial relations between objects, e.g. the distance between an organ and an instrument, which also allows an insight into the current task the surgeon is performing. The spatial relations also can be used for alleviating the surgeon's loss of depth perception via intraoperative measurements. Further, 3D information can be used for registering pre- and intraoperative data, which can be vital for certain assistances, such as displaying a tumor's position. In section 3.2 methods and applications of quantitative surgical image analysis are presented.

Surgical workflow analysis methods provide an estimate on the current state of a surgical operation. Such an estimate is imperative to computer-assisted surgery, as it provides the context-awareness that controls what information is displayed and when it is displayed. Surgical workflow analysis generally operated on surgical sensor data, e.g. laparoscopic images or surgical device data, or on semantic and quantitative information already extracted from the data. Section 3.3 introduces such methods.

3.1 Semantic Surgical Image Analysis

In semantic image analysis, computer vision methods are used to extract information on the content of the scene, e.g. visible objects, from given video frames. Knowing the content of the scene is relevant for context-aware computer-assisted surgery, as surgical instruments and relevant anatomic structures provide insights into the progress of surgery. Furthermore, their locations and specific type is often relevant to certain assistance functions.

Before being identified, relevant regions in the image are usually segmented via endoscopic image analysis. For this, segmentation methods, e.g. [MHMK+14a] [AOT+13] [TKK+01] [CBMC14], generally assign a class label to each pixel and then fuse similar regions together. Information that is potentially contained in the neighborhood of a pixel is hereby disregarded.

Not many segmentation methods in literature focus on laparoscopy. In [CBMC14] a method for segmenting and detecting laparoscopic instruments and the uterus in real-time is introduced. This method though does not generalize from offline training data, but rather learns patient specific models through manual input at the start of a given operation. A method for segmenting endoscopic images into regions of similar hue and homogeneity is proposed in [TKK+01], though the resulting region do not necessarily correspond to the same semantic object. A large community is focusing on developing methods polyp segmentation in colonoscopy [BTS+17], though these methods are not real-time capable.

3.1.1 Detecting, tracking & identifying surgical instruments

During laparoscopic surgeries, the instruments are the actuators of the surgeon and assistants. To allow context-aware laparoscopic assistance, it is therefore imperative to determine their locations in the laparoscopic video feed during the course of a surgery. This knowledge makes it possible to determine what tools are being used and what is being manipulated. Knowing where each tool is used gives clues to the intentions of the surgeon. Furthermore tool positions can be used for measurements (see chapter 5).

Multiple ways of detecting surgical tools are known in literature. Common approaches utilize external tracker and markers, e.g. electromagnetic [FKG+97] [LLG16] or optical [EdlFR10]. If the operation is performed with a surgical robot, the kinematics could also be used for tool detection [RAZ12]. These methods though require additional, specialized hardware in the OR that needs to be calibrated.

Detecting laparoscopic images directly in the endoscopic video feed would be a more preferable approach, as it keeps the impact on the surgical workflow to a minimum and no additional hardware is required. Such a method requires the surgical tools currently in use to be located and identified in the images in real-time. The problem of image-based instrument detection is well known and different approaches can be found in literature, e.g. [AOT+13] [SAR+12] [VLC07]. A thorough review of the current state of the art in instrument detection can be found in [BASJ17]. These methods often rely on heuristics to detect instruments, which do not always generalize to other datasets, or they are not able to process images in real-time.

Instrument type identification on the other hand is a lesser known problem. The types of instruments being used though are of relevance to a content-aware surgical assistance, as the type posses a constraint on the actions that could be performed by the surgeon. In [SBF14] a method for identifying instrument parts, but not explicitly the instrument type, is introduced. There are methods in literature that identify the types of the currently visible instruments in a laparoscopic frame [TSM+16] [BFN10], but they

do not provide the location of the identified visible instruments. [SBK+09] introduces a model-based approach for identifying surgical instruments, but does not evaluate the approach on actual surgical videos.

3.1.2 Sparse annotations

One common problem in surgical image processing is collecting sufficient annotated data that can be used for training computer vision algorithms. Crowd sourcing has been successfully used for tasks such as pixelwise instrument annotation [MHMK+14a] and labeling corresponding image points in laparoscopic image pairs [MHMK+14b]. Often though domain knowledge from surgical experts is required, for example when identifying structures in laparoscopic images. The time of these experts is often restricted and expensive, making it necessary to explore methods for sparse annotations.

Methods for sparse annotations, such as superpixels [VdBBR+12] and image-wise labels [SJC08] have been used for many tasks in the computer vision community, though few works exist that apply these methods to tasks in the surgical domain. Superpixel have been used for medical image retrieval [HDB+11] and MRI image segmentation [JWY+14]. In [TSM+16], sparse labels are used to detect instruments visible in laparoscopic frames, though organs have not been explored.

3.2 Quantitative Surgical Image Analysis

Quantitative endoscopy is a broad term that refers to methods that perform calculations on 3D relations extracted from an endoscopic scene, e.g. measurements of lengths or area. The 3D relations can also be used to reconstruct surfaces for building intraoperative organ models [RBS+15]. These models can serve as boundary condition for biomechanical registration [SRB+14] of preoperative data. As a first step, these methods generally require at least a partial 3D reconstruction of the surgical scene

3.2.1 3D reconstruction

Thorough overviews of 3D reconstruction methods that can potentially be used intraoperatively is given by Maier-Hein et al. [MHMB+13] and Mirota et al. [MIH11]. Both reviews divide the presented methods into two groups, passive methods that only require images for reconstruction and active methods that require controlled light.

Passive methods

Passive methods for 3D reconstruction rely only on one or more images taken from one or more cameras to recover depth information from a viewed scene. Generally it is assumed that the cameras in use are calibrated (for more details, please see section 5.3), meaning that parameters such as the focal length and the potential offset between multiple cameras are known.

Stereo

Stereo camera systems consist of two cameras that view the same scene from slightly varying perspectives. A more detailed explanation of stereo cameras can be found in section 5.3.

Depth retrieval from stereoscopic image pairs requires corresponding points in both images. Finding a large number of correspondences for a dense reconstruction in real-time is still a focus in the computer vision community. Multiple approaches for computing such a dense correspondence map exist. For example, in [Hir08], the authors propose a real-time method for matching corresponding pixels in stereo image pairs using mutual information to minimize an energy function, while [AKS04] propose a method for finding correspondences on a block-level and a pixel-level in real-time, resulting in a temporally and spatially consisted correspondence map.

Currently, the trend in correspondence analysis is heading the direction of convolutional neural networks (CNN) (for more details see section 6.1.1). For example, in [MIH+16] the authors presented a first fully CNN-based approach for correspondence analysis, which is an extension of a previously presented CNN for optical flow analysis [DFI+15]. Furthermore, on the public Middlebury dataset [SS02, SS03, SP07, SHK+14, HS07], which provides datasets for the comparison of correspondence analysis methods, 6 of the 10 methods with the lowest average error are CNN-based [1].

A further dataset for evaluating correspondence analysis methods, the KITTI dataset [GLU12, FKG13, MG15], is aimed at reconstructing scenes common to traffic situations from the perspective of a car. The authors of the set found that methods that performed well on datasets like the Middlebury dataset did not necessarily perform well on the KITTI dataset[2]

It can be assumed that the same is true for the laparoscopic domain as laparoscopic images contain specific challenges like specular reflections and textureless regions. On of the first to address this problem in a laparoscopic environment were Lo et al. in [LCS+08] where they propose a method for reconstructing organ surfaces in 3D using correspondence analysis and Markov Random Fields. A quantitative evaluation is not provided though. Stoyanov et al. [SSPY10] propose a method for computing semi-dense correspondence maps from stereoscopic laparoscopes in real-time and evaluated it quantitatively on phantom data. Another method for real-time semi-dense correspondence analysis is presented in [TTS+14]. A phantom evaluation was performed here as well, though no in-vivo results were provided. In [RBS+12], the authors present a real-time method for dense correspondence analysis in a laparoscopic setting, based on [AKS04], and evaluated it quantitatively on phantom and in-silico data. Furthermore, a qualitative in-vivo evaluation was performed. Chang et al. [CSD+13] proposed a method for dense correspondence analysis using convex optimization. They evaluated their method on the same phantom dataset as [SSPY10] and [RBS+12], outperforming both methods.

[1] See: http://vision.middlebury.edu/stereo/eval3/ (accessed: July 2, 2018)
[2] See: http://www.cvlibs.net/datasets/kitti/index.php (accessed: July 2, 2018)

Structure from motion

Structure from motion is a similar problem as stereo correspondence analysis. It also uses two images to retrieve depth, but, in contrast to stereo, these image are recorded by a monocular camera over time. Therefore Structure from motion has the added problem that not only the correspondences need to be located, but also the spatial offset between these two frames. Assuming a rigid environment, this offset can be computed up to a scale from image observation [HZ03], making it possible to apply the same methods as previously outlined for stereo.

Laparoscopic scenes are generally not rigid, due to soft-tissue deformation. While methods that handle deformable surfaces exist, e.g. [MBC11] and [HPF+12], none operate in real-time [BNSD17].

Shape from Shading

Shape from shading, in contrast to the previous methods, generally only needs one image for depth estimation [PF06]. It analysis how the lighting affects the shading of the scene to estimate the depths of visible objects.

In [CB12], Collins et al. propose and evaluate such as method on laparoscopic data, but arrive at the conclusion that the cues provided by shape from shading alone are not sufficient in a laparoscopic settings. Instead they propose a combination with structure from motion [MB14].

Active methods

Active methods generally use some form for controlled light to retrieve depth information from a scene.

Structured light

Methods that rely on *structured light* attempt to retrieve 3D information by projecting a pattern of light with known geometry onto the viewed scene. The projected pattern is observed with a camera and, once the pattern has been successfully detected, depth information can be extracted via triangulation. The geometry between the light projector and the camera has to be known [RAR04]. Due to features being introduced externally, structured light is robust when reconstructing homogeneous, textureless regions.

The main change in introducing structured light to laparoscopic surgery is pattern projection. In literature, most methods use one of two solutions for this problem. One solution is to use separate devices for the camera and for the projector, such as in [MADdM12] where two endoscopes, one for vision and one for the pattern projector are used. The methods outlined in [AHR13] and in [EPY+15], both utilize miniaturized projectors for projecting the pattern, which is then observed with a standard endoscope. In [CSMH+11] and [LCE15], a separate probe is used for the projector. In contrast, in [SFSA12] a sensor integrating both camera and projector is introduced,

though the resolution of the camera is rather small with 400×400 pixels. Similarly, in [FMM+15], a standard endoscope is used and the projector is fitted into the instrument channel of the endoscope.

Time of flight

Another active method for visually measuring distances and thereby performing 3D reconstruction is *time of flight*. Time of flight devices emit light and measure the travel time of the light from the emitter to an object and back to the detector [STDT08].

While multiple systems for time of flight in laparoscopy have been introduced, e.g. [PHS+09], [MMS+11] and[BSH+13], these system still incur a systematic distance error and suffer from a low signal to noise ratio [MHMB+13].

Comparison of reconstruction methods

In [MHGB+14], Maier-Hein et al. performed a comparative study of available reconstruction methods in the laparoscopic domain. Contained in the study were three stereo reconstruction methods, [RBS+12], [SSPY10] and [CSD+13], a structured light approach [CSMH+11] and an experimental time of flight endoscope developed by Richard Wolf GmbH [MHGB+14]. These methods were evaluated on data collected from explanted porcine organs, containing challenges common to laparoscopic images. Here, the authors showed that the stereo-based methods outperformed the other methods. Furthermore the two dense stereo methods, [RBS+12] and [CSD+13] outperformed the sparse method [SSPY10]. [CSD+13] has a higher reconstruction accuracy and robustness, while [RBS+12] has a higher surface coverage.

When considering surgical availability, the monocular-based methods are clearly in the advantage. But, while it is a specialized hardware, stereoscopic endoscopes are becoming more frequent in the operating room, thanks to the daVinci surgical teleoperator and new 3D monitors [MHMB+13].

3.2.2 3D measurement

As previously mentioned in section 2.3.1, one of the challenges that surgeons face during laparoscopic surgery is the lose of depth perception. Using depth information from 3D reconstructions makes it possible to alleviate this by allowing intraoperative measurements.

In [KBR+95] the authors present a method for computing the area of Barrett's metaplasia, though the reported error of 2 cm is rather large. A method for measuring the size of objects from two different image was presented in [TAKW00], though the method required knowledge of the distance traveled between two images. In [FCSS09] a method for measuring distances, e.g. the size of a hernia, using a calibrated stereo endoscope is introduced. For this, the tips of two surgical instruments are located in

a stereo image pair using optical markers attached to the instruments and then triangulated. The shortest distance between the two tips is then presented as result. The surface of the organ is not taken into account, which would cause the method to underestimate distances on curved surfaces. Furthermore, using markers for instrument detection requires extra modifications to the surgical setup and the surgical workflow.

3.3 Surgical Workflow Analysis

For many applications in CAS, such as providing the position of a tumor, specifying the most probable tool required next by the surgeon or determining the remaining duration of surgery, analyzing the surgical workflow is a prerequisite.

3.3.1 Surgical workflow segmentation

One common method to describe the surgical workflow is via surgical phases. Therefore, to asses the progress of surgery, automatic phase segmentation is required. Often, laparoscopic tool usage [BJD11] [SOP+14] [PBA+12] or surgical activities [KWG+14] [NSM+06] [FRJ15] are used as features for such a segmentation, but currently this information is usually derived through additional hardware (e.g. RFID tags in the case of [SOP+14]), which is not always available in the OR or through manual annotation, which is not feasible for online workflow segmentation or large datasets. The kinematic data from a robotic system, such as the daVinci can be used for providing tool usage information and tool trajectories [DLM+16] [ZBHV13], but this information is only available for robotic surgeries and not the majority of laparoscopic surgeries.

While methods for automatically extracting information on tool usages from endoscopic images do exist [BFN10][SBK+09] there are few publications with a purely image-based approach for workflow analysis [BFN10] [DBH+16] [LRBJ12] [TSM+16] [LCRH16]. The authors in [BFN10], [DBH+16] and [LRBJ12] utilize a combination of manually selected image features to describe the content of single video frames. Manually selecting image features has the drawback that only information that the domain experts are aware of can be captured, other characteristics that might still contribute are possibly lost.

In [TSM+16], the authors propose EndoNet, a combination of a CNN and a hybrid hidden markov model (HHMM). The CNN here is used to automatically learn image features that can be used to distinguish different surgical phases in laparoscopic gallbladder removals, which are then fed into a HHMM to determine the most probable phase for each image frame. On the dataset of the Endoscopic Vision 2015 Workflow Challenge[3], EndoNet outperforms the method outlined in [DBH+16], which uses manually selected image features.

The drawback of EndoNet is that a large amount of annotated data is used for training, 40 videos of laparoscopic gallbladder removals in which not only the surgical phases,

[3] http://endovissub-workflow.grand-challenge.org/

but also the laparoscopic instruments are annotated for each frame. This amount of annotated data is difficult and costly to collect. If one takes into consideration that laparoscopic gallbladder removals are simple and standardized operations, one can assume that more complex types of surgeries, such as colorectal or pancreatic surgery, would require even more labeled data. In [LCRH16], the authors present a CNN-based approach for offline phase segmentation that outperforms EndoNet on the EndoVis15Workflow dataset, which uses only 6 operations for training.

Offline phase segmentation means that data from the entire surgery is used for assigning a phase to each frame retrospectively. The approaches makes usage of spatio-temporal information to capture object motion during the course of a laparoscopic surgery. The features extracted with the CNN are then combined with either a linear model, a semi-markov model or a time-invariant model, based on dynamic time warping, with the latter two models outperforming [TSM+16], leading to the conclusion that including temporal information during workflow analysis improves classification outcome.

3.3.2 Surgical progress prediction

While surgical phase segmentation methods can be used to approximate the duration of surgical procedures, these methods generally require a sufficient amount of labeled examples as training input. Furthermore, seeing that phase models are generally specified to a certain type of surgery, multiple classifiers would need to be trained. Therefore, using a phase-based method as a general solution to determine the remaining duration of surgeries would require an unfeasible large amount of labeled training data. Currently, not many methods for unlabeled surgical progress prediction are available. In [GPM+16], the authors propose a system that determines the remaining time of surgery during laparoscopic cholecystectomies without surgical phases, but directly from the usage of the electrosurgical device.

3.4 Discussion

In summary, while multiple methods for semantic image analysis of endoscopic images exist, they generally do not focus on real-time performance, which is a necessity for bringing computer assistance into the operating room. Other methods that are capable of real-time performance either require user interaction, apply heuristics that don't always generalize or only operate on single pixels.

In quantitative laparoscopy, while multiple methods for 3D surface do exist reconstruction exist, stereo endoscopy is currently the most promising technology, as monocular reconstruction methods are either not fast enough or accurate enough for clinical purposes. Other promising technologies are still in development and not available in a clinical environment. While image-based surgical measurement tools do exist, they have not been in large focus in literature. Currently, no method that allows the surgeon to measure distance along organ surfaces, as some operation types require, exist.

Laparoscopic workflow analysis has been a focus of research, with multiple published approaches for surgical phase segmentation. These approaches often rely on information provided by semantic image analysis methods, often beyond the scope of the current state of the art, or hardware not generally available. Recently, the trend has been to extract low-level features from images for image-based workflow analysis. These methods require a large amount of training data. Surgical progress detection without phase segmentation has recently been addressed in literature. Such a method has the advantage that no annotations by experts are required. Currently no purely image-based method exists.

4 Semantic Surgical Image Analysis

This chapter proposed methods for semantic surgical image analysis. Semantic information in an image describes what certain parts of an image represent, such as what objects are contained in the image and where they are located. First a real-time segmentation method for laparoscopic images is introduced and extended to perform surgical instrument detection (section 4.1). Once an instrument has been detected, a further method identifies the type of instrument (section 4.2). A tracking method makes it possible to propagate detected and identified instruments into future frames, thereby increasing robustness (section 4.3). A superpixel-based segmentation method is introduced and evaluated for instrument segmentation (section 4.4). A approach for labeling laparoscopic image content based on sparse annotations is also presented (section 4.5).

4.1 Real-Time Instrument Detection

During laparoscopic surgeries, the instruments are virtually the main manipulators of the surgeon and assistants. To allow context-aware laparoscopic assistance, it is therefore imperative to determine their positions during the course of a surgery. Detecting the positions of the instruments makes it possible to determine what tools are being used and what is being manipulated. Furthermore their positions can be used as input information for assistance functions, such as measurement tools.

The appearance of laparoscopic tools is influenced by many factors, such as illumination, optic, and partially also patient physiology. It is therefore important for a method that detects these tools, to learn from a wide variety of data. Depending on the scenario, e.g. a warning about a potential risk situation, real-time detection is also a requirement.

In this section, we present a method for segmenting laparoscopic instruments in real-time, building on simple features and a random forest-based classifier, which are both easily parallelizable. Based on this segmentation, a postprocessing step is used to detect instrument regions. To clarify, segmenting instruments refers to the task of locating pixels or pixel regions in the images that probably belong to laparoscopic instruments. Instrument detection refers to the task of using this segmentation to cluster segmented regions into different instruments.

An overview of the different components of the method can be found in figure 4.1.

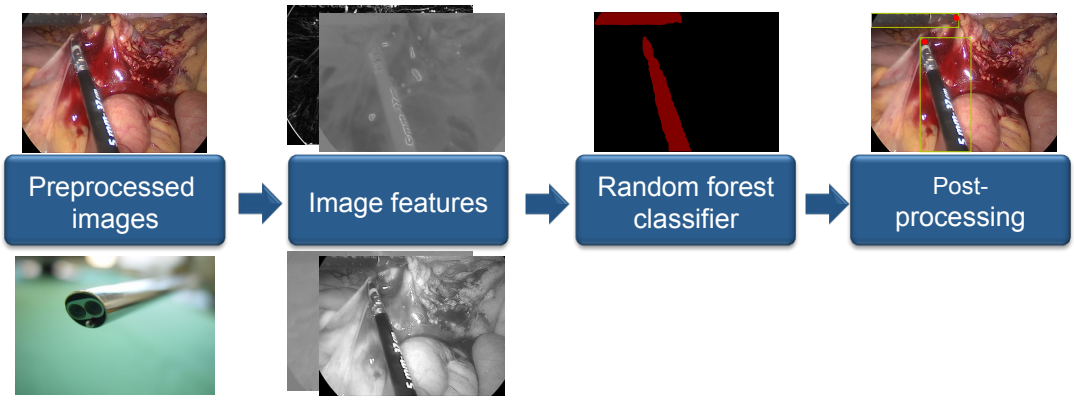


Figure 4.1: Overview of the components used for instrument detection.

4.1.1 Methods used for instrument detection

To detect laparoscopic instruments in images, we first have to detect regions of interest, which is a segmentation task, or, in other words, for each pixel p in an image I we need to determine, if it belongs to the class instrument or the class background. For this, we require a labeling function:

$$L : p \in I \rightarrow \{0,1\} \tag{4.1}$$

We first preprocess a given image to reduce the search space and to rule out false positives. In a next step, we need to determine features f_p that describe each pixel. Based on that description, we can then train a classifier to label each pixel in an image.

Preprocessing

As presented in section 2.1, endoscopes generally have a circular shaft and optic, which results in a border on the resulting camera image, as endoscopic cameras contain a rectangular image chip. Since this circular border common to many laparoscopic images has a similar color as most instruments, it can interfere with color features. Furthermore, laparoscopic instruments, due to the placements of the port, always enter the field of view of the endoscope from the side, therefore knowing where the exact border the endoscopic view is located can help with detecting and eliminating false positives. As its size and orientation varies from image to image, due to zooming or rotation of the camera in reference to the optic, we have to detect and remove it automatically. This is accomplished by applying a threshold filter (threshold = 3) to a grayscale version of the image and then traversing from each corner of the image along the diagonals towards the center until a non-black pixel is found. The two points p_1 and p_2 on the same diagonal with the furthest distance from each other are used to define a circle with center $c = \frac{p_1+p_2}{2}$ and radius $r = ||p_1 - p_2||$. The two points on the same diagonal with the furthest distance from each other are used to define a circle. This circle can then be used to mask the image.

Image features

Once the image has been preprocessed, features for segmenting regions of laparoscopic instruments have to be selected. To streamline the detection process, a multitude of simple features that can be compute quickly and in parallel for each pixel p_i in the entire image were used instead of detecting more complicated features (e.g. SIFT [Low99], SURF [BTVG06], ORB [RRKB11]). The features used consist of values taken from different color spaces and gradient information.

There are numerous color spaces that each allow different representations of pixel values. Different color spaces provide different advantages, e.g. one color space might allow a linear classifier to differentiate between red and green or a representation that is less sensitive to changes in illumination. The following color spaces were examined and used in this work:

RGB

One of the most commonly used color spaces is the RGB color space. The RGB color space is an additive color space, consisting of the primary colors **r**ed, **g**reen and **b**lue (see figures 4.2(b)-4.2(c)). Generally colors in RGB are represented as a tuple of three values (R,G,B) where $R,G,B \in [0,\ldots,255]$. An alternative convention is $R,G,B \in [0,1]$, but, unless stated otherwise, the first notation will be used from here on. RGB is the color space used throughout computer graphics, as it is based on the way humans perceive colors. RGB is commonly used in cameras and computer screens [Jäh12].

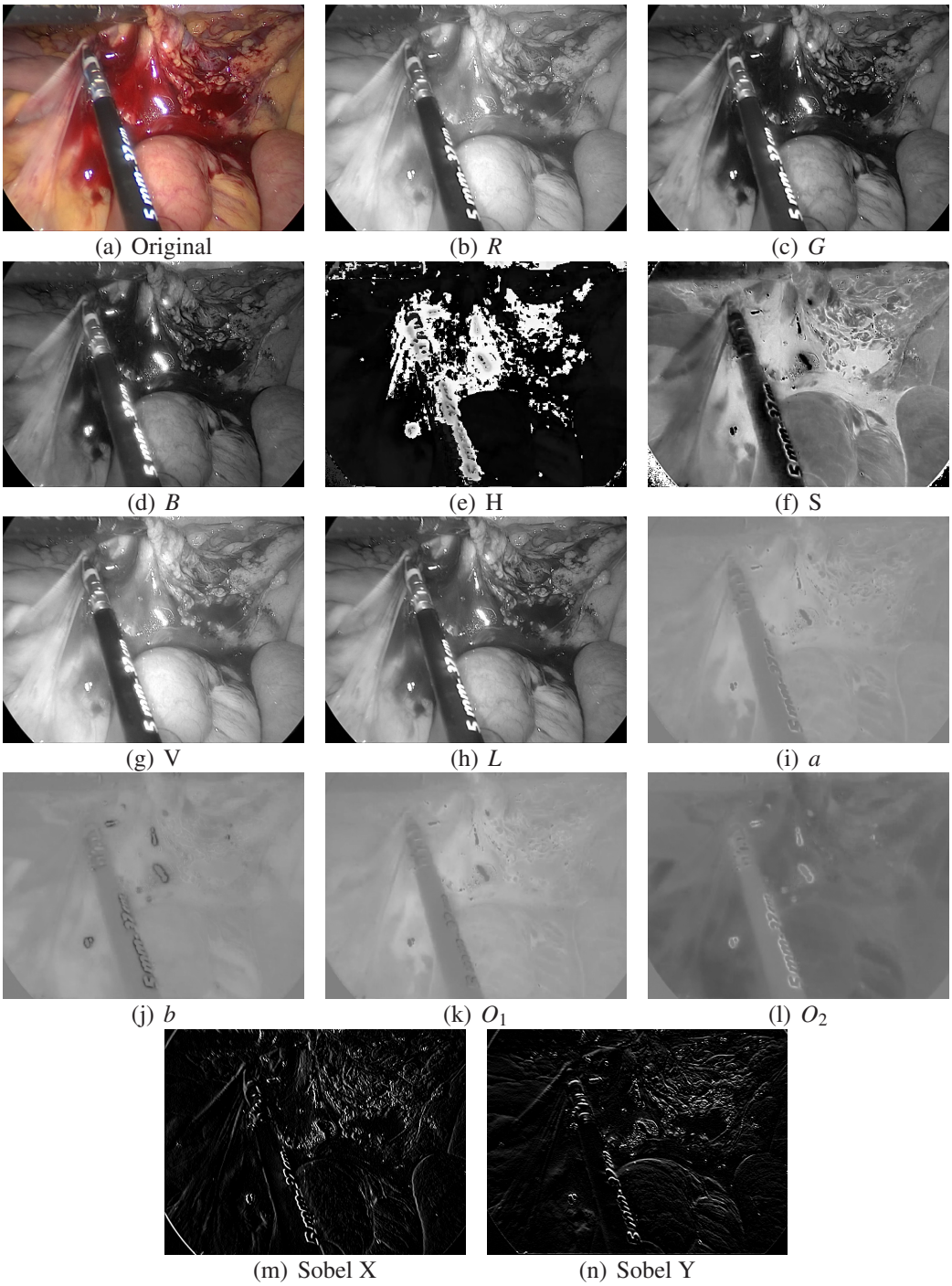
HSV

In the HSV (**h**ue, **s**aturation and **v**alue) color space, colors from the RGB space are represented as points in a conical coordinate system (see figure 4.3). Hue specifies the angular dimension with red at 0°, green at 120°and blue at 240°. The HSV color space emulates the way human perception conceptualizes colors in terms of hue, brightness and chroma [JG78, BKB82].

By separating brightness from the hue, the color spaces becomes less sensitive to changes in light (see figures 4.2(e)-4.2(g)). This is especially important in laparoscopic images as the illumination can vary greatly in even areas of the same image, due the camera being placed in close proximity to the light source. Furthermore, it allows different hues (green, yellow, red, blue, ...) to be easily distinguished using just the hue value. A given color value in the RGB format can be converted to HSV by the following transformations:

$$V = \max(R,G,B) \tag{4.2}$$

$$S = \begin{cases} \frac{V-\min(R,G,B)}{V} & \text{if } V \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$



(a) Original (b) R (c) G

(d) B (e) H (f) S

(g) V (h) L (i) a

(j) b (k) O_1 (l) O_2

(m) Sobel X (n) Sobel Y

Figure 4.2: A laparoscopic image (a), its representation in RGB (b)-(d), HSV (e)-(g), CIELab (h)-(j), opponent (k)-(l) and results of the Sobel operators (m)-(n).

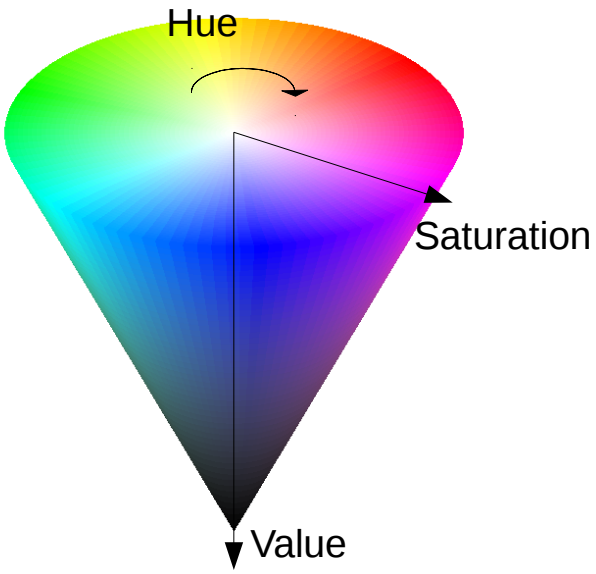


Figure 4.3: HSV color space

$$H = \begin{cases} 60 \cdot \frac{(G-B)}{V-\min(R,G,B)} & \text{if } V = R \\ 120 + 60 \cdot \frac{(B-R)}{V-\min(R,G,B)} & \text{if } V = G \\ 240 + 60 \cdot \frac{(R-G)}{V-\min(R,G,B)} & \text{if } V = B \end{cases} \tag{4.4}$$

CIELab

The CIELab color space is based on opponent color theory, which states that colors can be described as a mix of red, green, yellow and blue. Furthermore, red and green form an opposing pair, as do yellow and green and white and black, or, in other words, humans can perceive red or green, but not both [Her64]. The three components (L,a and b) of the CIELab color space mirror this observation (see figures 4.2(h)-4.2(j)). Here L describes how close a color is to black or white, a how close it is to red or green and b how close a color is to yellow or blue. To transformations a color given in RGB to CIELab, one first performs a basis transformation to the XYZ space [Jäh12]:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.490 & 0.310 & 0.200 \\ 0.177 & 0.812 & 0.011 \\ 0.000 & 0.010 & 0.990 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{4.5}$$

From there, we can directly convert to CIELab [Bra00]:

$$L = \begin{cases} 116 \cdot \sqrt[3]{Y} - 16 & \text{if } Y > 0.008856 \\ 903.3 \cdot Y & \text{otherwise} \end{cases} \tag{4.6}$$

$$a = 500 \cdot (f(X) - f(Y)) + 128 \tag{4.7}$$

$$b = 200 \cdot (f(Y) - f(Z)) + 128 \tag{4.8}$$

with

$$f(t) = \begin{cases} \sqrt[3]{t} & \text{if } t > 0.008856 \\ 7.787t + \frac{16}{116} & \text{otherwise} \end{cases} \tag{4.9}$$

Opponent

Similar to CIELab, the opponent color space is also based on opponent color theory. Here the transformation from RGB to the first 2 values is defined as the following [GS03]:

$$o_1 = \frac{R-G}{2} \tag{4.10}$$

$$o_2 = \frac{B}{2} - \frac{R+G}{4} \tag{4.11}$$

o_3 is defined identically to the value in the HSV color space (see figures 4.2(k), 4.2(l) and 4.2(g)).

Sobel

The image gradients also contain information about pixel properties. The quick and common way to approximate the gradient of an image is provided by the two Sobel operators S_x and S_y, which approximate the derivatives of the image in the horizontal and vertical directions [AGD08][SF68].

$$S_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \tag{4.12}$$

$$S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{4.13}$$

These operators use two 3×3 kernels that can be convoluted with the original image, resulting in the gradient images G_x and G_y (see figures 4.2(m) and 4.2(n)).

$$G_x = S_x \star I, G_y = S_y \star I \tag{4.14}$$

Based on these, we can compute the magnitude and the orientation of the gradient:

$$M = \sqrt{G_x^2 + G_y^2} \tag{4.15}$$

$$\Theta = \text{atan2}(G_y, G_x) \tag{4.16}$$

As can be seen in figure 4.2, different contents of the example image become easier to distinguish in different channels of certain color spaces. For example, the pixels belonging to the instrument are more discernible from the background in the o_1 channel (figure 4.2(k)). While these features are simplistic, they still take a large amount of time to compute, especially with increasing image resolution. To achieve real-time capabilities, parallelization paradigms are necessary. All of the features mentioned previously can be computed for each pixel independently, most don't even require information from neighboring pixels. This makes it possible to port the computation of the features onto a Graphics processing unit (GPU), which allow simultaneously executing the same function (or kernel) by thousands of threads. We therefore ported the feature computation onto a GPU using CUDA from NVidia [NBGS08].

This provides us with a function F that computes a feature mape F_I, which contains feature vectors f_p for every pixel p in image I.

$$F_I = F(I) = \{f_p | p \in I \wedge f_p \in \mathbb{R}^N\} \tag{4.17}$$

Here, N is the number of features (e.g. $N = 3$ if hue, o_1 and the Sobel magnitude are used).

The feature map F_I is then used as input for a classifier to determine the most probable class of each pixel.

Random forest classifier

To classify each pixel as either background or instrument, we selected random forests [Bre01] for this task, as they are easily parallelizable, fast to compute and not prone to overfitting. A random forest is an ensemble of binary decision trees [DHS01], which independently classify a given input and cast a vote for the resulting class. The output of the random forest is determined via majority voting over all trees.

A binary decision tree is a hierarchal collection of weak classifiers, also called split functions (see figure 4.4). Each node j in the tree is associated with a different split function h and each leaf contains a label.

$$h(x, \theta_j) : \mathbb{R}^N \times \tau \rightarrow \{0, 1\} \tag{4.18}$$

These split functions provide a binary output for a given input vector x, deciding which child node to visit next. $\theta_j \in \tau$ are the split parameters associated with node j [CS13]. In our case, we use axis-aligned linear split functions, i.e.

$$h(x, \theta_j) = [x \cdot e_{\tau_d} < \tau_v] \tag{4.19}$$

where $\tau_v, \tau_d \in \theta_j$ indicate the split value and the split dimension and e_i is a N-dimensional vector containing zeros at every position except i, where it contains a one. $[\cdot]$ is the indicator function that returns 1 if the argument is true and 0 otherwise [CS13].

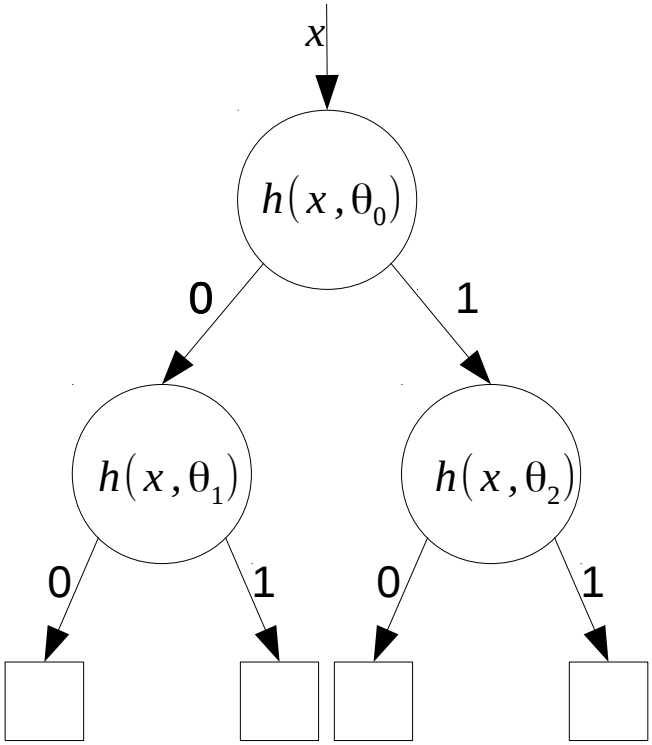


Figure 4.4: Example decision tree

Given a set of training examples, a decision tree is trained by maximizing the information gain by the split at each node. information gain is defined in the following manner:

$$= H(S_j) - \sum_{i \in \{0,1\}} \frac{|S_j^i|}{|S|} \cdot H(S_j^i) \tag{4.20}$$

where S_j is the training data arriving at node j and S_j^0 and S_j^1 the two subsets in which S_j will be split. S_j^0 will be passed to the left child node and S_j^1 to the right child node.

$$H(S) = -\sum_{c \in C} p(c) \log(p(c)) \tag{4.21}$$

is the Shannon entropy, with C being the set of all possible classes and $p(c)$ the distribution of class c in S [CS13].

During training, at each node, the split parameters for each node j are determined in the following manner:

$$\theta_j = \underset{\theta \in \tau}{\operatorname{argmax}} I(S_j, \theta) \tag{4.22}$$

the depth of a decision tree is increased until either a maximum is reached or the samples after a split are pure, meaning they are all of the same class. A leaf contains the label corresponding to that of the majority of the samples arriving there.

Training a random forest consists of training multiple decision trees. During training, randomness is introduced in two manners. First, each tree t is not trained on the whole

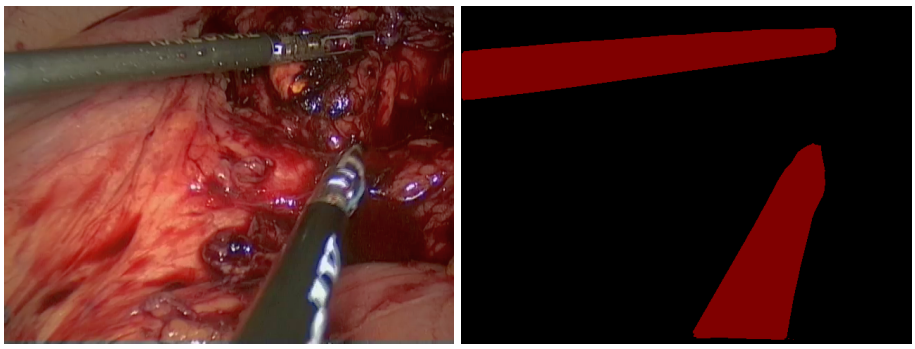


Figure 4.5: An example of a training image used for the instrument detection and its annotation

dataset S, but a random subset $S_t \subset S$, an approach called bagging. Bagging helps to reduce overfitting, as each tree only specializes on a subset of the entire dataset. A second method to introduce randomness is to randomly subsample τ when training a node. In other words, equation 4.22 becomes:

$$\theta_j = \underset{\theta \in \tau_j}{\operatorname{argmax}} I(S_j, \theta) \tag{4.23}$$

with $\tau_j \subset \tau$. Reducing the parameter space has been shown to increase classification efficiency [CS13].

During testing, each tree in the ensemble processes the input x and votes on a class, the class that receives the majority of the votes is then the final classification.

To apply a random forest to our segmentation problem, we have to classify each feature vector f_p that we computed for each pixel in the original image. This takes a large amount of time, therefore parallelization is required. Since the trees in the forest operate independently and each pixel can also be processed independently, we ported the random forest classification part onto the GPU using NVidia CUDA [NBGS08].

To train the random forest, laparoscopic images with previously labeled instruments and background are used as input (figure 4.5). Example outputs are provided in figure 4.6. In this chapter, the segmentation method is only applied and evaluated on surgical instruments. A further evaluation for bowel segmentation can be found in section 5.2.

Postprocessing

The segmentations are refined with a morphological closing [AGD08], before detecting connected contours. To detect connected regions and their borders in a segmentation, we apply the method outlined in [SA85] to the segmentation, resulting in a list of contours. Artifacts, like blood, on the instrument shaft can lead to gaps in the segmentation (see figure 4.6(b)), which have to be closed. Here we compute the principal direction of each contour by eigen-decomposition of the covariance matrix of all contour points. Contours that lie in close proximity and share a similar principal direction are fused together. We repeat this process until no further fusions take place. For each contour, we compute the tip of the instrument by finding the point on the contour that lies furthest

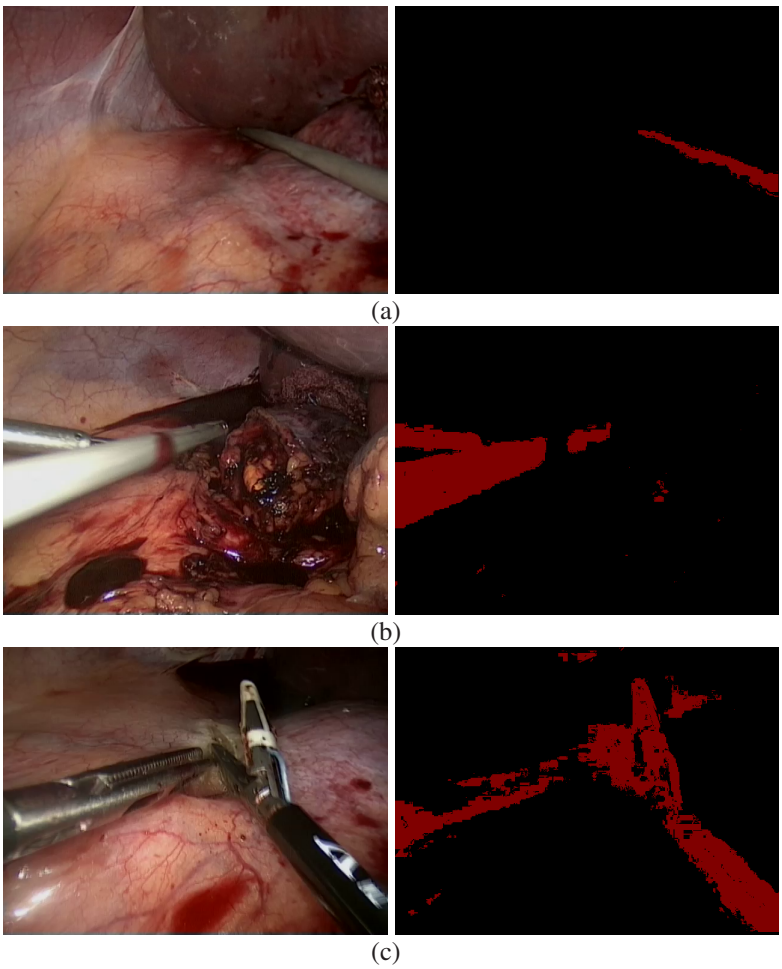


Figure 4.6: Example segmentation results

from the image border and is detected on a line formed by the contour mean point and its principal axis.

4.1.2 Evaluation of the instrument detection

The basis for evaluation of the random forest and the instrument detection are five laparoscopic videos of two different operation types (two adrenalectomies and three pancreas resections). 20 images from each video (100 overall) were selected manually, taking care that the variance in backgrounds and in instrument type was reflected in the data. Image resolution was 640×480 pixels. These images were then annotated pixel-wise by means of crowd sourcing [MHMK+14a][1] (see figure 4.5). Each instrument in each image was annotated by 10 independent participants. The resulting annotations were then fused pixel-wise via majority voting. This data set will be refered to as **Crowd** from here on forward.

[1] Data available from: http://www.open-cas.org/?q=InstrumentCrowd (accessed: July 2, 2018)

	Precision	Recall	DICE	Accuracy
Crowd	64.8%±12.0%	75.6%±18.6%	68.0%±10.4%	89.7%±2.5%
EndoVisRigid	73.5%±21.5%	44.0%±22.8%	50.4%±20.0%	93.1%±4.6%
EndoVisRobotic	85.7%±9.1%	72.1%±7.4%	77.9%±6.3%	95.3%±1.5%

Table 4.1: The performance of the random forest-based segmentation methods on multiple data sets.

In addition to **Crowd**, we also evaluated the random forest segmentation on the data set from the EndoVis instrument challenge from MICCAI 2015[2]. Two challenges for instrument segmentation, one for standard laparoscopic and one for robotic instruments, were held. The data set for the standard laparoscopic instruments (**EndoVisRigid**) consists of 6 sets of 50 images taken from videos captured from 6 different laparoscopic surgeries. The data set is divided into a training set, consisting of 40 images from 4 of the sets, and a testing set, consisting of the remaining 10 images of the 4 sets and the other 2 sets completely. The images were extracted equidistant from the videos at a resolution of 640×480 pixels. The reference annotations for these images were acquired in the same manner as **Crowd**, via crowd sourcing.

The data set for the robotic instruments (**EndoVisRobotic**) was collected from 6 one minute videos recorded with the daVinci tele-operator. The recorded videos contain interactions between the daVinci's instruments and ex vivo organs. The data set is also divided into a training set, containing the first 45 seconds of 4 of the 6 videos and a testing set, consisting of the remaining data. Each video had a resolution of 720×576 pixels The reference annotations in this data set were extracted directly from the kinematics of the tele-operator. An overview of the datasets used can be found in appendix B.

Results

To train the random forest for segmentation, the following parameters were used: As features for each pixel hue, A and B from the LAB color space, o_1 and o_2 from the opponent color space, and gradient orientation and magnitude were selected. For training the random-forest, 50 trees and a maximum depth of 10 were used. All parameter values were determined empirically through experiment.

For the **Crowd** data set, we performed a leave-one-surgery-out evaluation, the results of which can be found in table 4.1. An explanation of the used metrics can be found in appendix A. While evaluating the **EndoVisRigid** and the **EndoVisRobotic** data sets, we trained as specified by the challenge, using the training data set in a leave-one-surgery-out fashion. The results for the two data sets can also be found in table 4.1. The results of the postprocessing step will be presented and discussed in section 4.2.2.

[2] Data available from: https://grand-challenge.org/site/endovissub-instrument/ (accessed: July 2, 2018)

	Pre-processing	Feature extraction	Feature classification	Post-processing	Σ
CPU	2.4 ms	113.1 ms	552.7 ms	16.3 ms	684.6 ms
GPU		0.1 ms	5.1 ms		23.9 ms

Table 4.2: Comparison of the run-time between CPU and GPU-based random forest segmentation. Times were averaged over 100 runs.

Run-time

The evaluation was run on a workstation PC with an Intel Core i7-5820K, 32GB of RAM and an NVidia Titan X. We ran two experiments to determine the run-time of the different components: Preprocessing, image feature extraction, random forest-based feature classification and postprocessing. In one experiment, we used the CPU-based version of each component, in the second experiment, we used GPU-based feature extraction and classification. The results can be found in table 4.2.

4.1.3 Discussion

In this section, we presented a method for segmenting and detecting laparoscopic images in real-time. The results show that the method produces the best results on the **EndoVisRobotic** data set. This can be partially attributed to the image quality, as the images contained less compression artifacts than those of the other two data sets. A major difference between **EndoVisRobotic** and the other two data sets is the variance, as all videos in **EndoVisRobotic** were recorded with the same optic and similar instruments, while the optics in **Crowd** and **EndoVisRigid** varied. Furthermore **EndoVisRobotic** did not contain a large amount of specularities and no blood or smoke, which are artifacts common to laparoscopic videos.

Results from two other groups, one from the University College of London [AOT+13] and one from the University of Bern, on the **EndoVisRigid** data set were made publicly available[3]. Here our method was ranked second, following the method from Bern, an AlexNet-based approach [KSH12], which, at the time of writing, had not been published.

The proposed CPU-method achieves an average frame rate of almost 1.5Hz at a resolution of 640×480 pixels. Moving the most processing time consuming components, feature calculation and the random forest classification, onto the GPU led to a vast improvement in processing speed, achieving average frame rates of almost 42Hz.

4.2 Instrument Identification

For many applications, not only the position of an instrument is relevant, but also its type. For example, if a tool that is located in the vicinity of the liver, can be used for

[3] https://grand-challenge.org/site/endovissub-instrument/results/

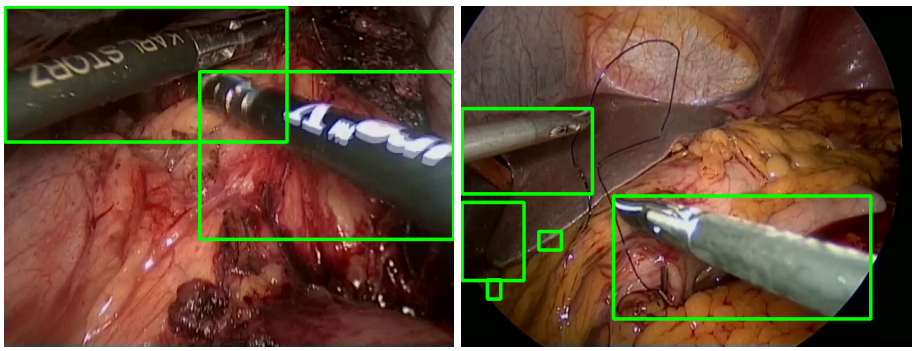


Figure 4.7: Example output of the instrument detection postprocessing. The boxes describe regions that were detected as instruments.

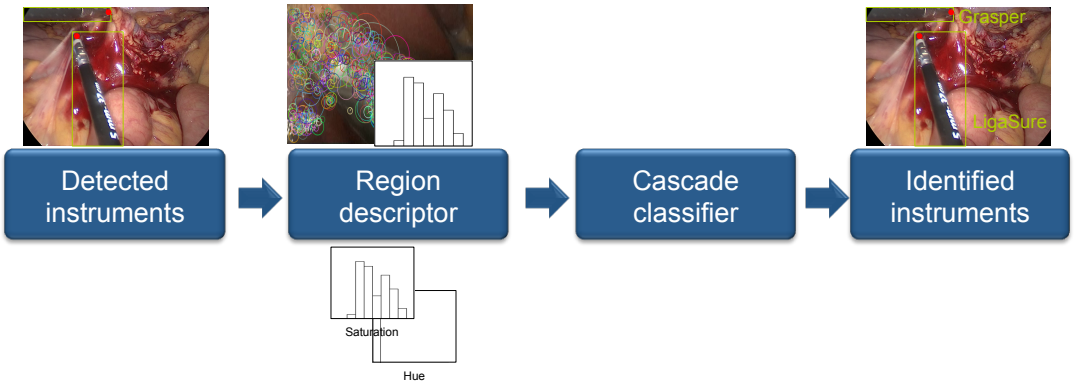


Figure 4.8: Overview of the components used for instrument identification.

cutting, a resection line might be shown in augmented reality. The types of instruments currently in use can also serve as an indicator for the surgical phase.

In this section, we present a method that uses the previously detected contours (section 4.1) and determines whether a contour really belongs to an instrument and, if it does, what type of instrument it contains. Figure 4.7 contains examples of contours found by the instrument detection method. As can be seen, contours that are falsely assumed to be instrument regions are a possibility. To achieve a high throughput, we first filter out falsely detected contours with a classifier using features that can be computed quickly. The remaining contours are then closely examined to determine instrument type using features that require more time to be computed. Further details can be found in [Ohn15]. This work was published in [BOK+15] An overview of the different components of the method can be found in figure 4.8.

4.2.1 Methods for instrument identification

To determine if and which instrument is occupying the region of interest returned by the instrument detection method, we require features that make such a distinction possible. As some of these features, e.g. SURF, can be computationally expensive, we use a cascade of two classifiers that cheaply rejects false positives based on quickly

computed features. The more costly features can then be applied to a smaller selection of instrument candidates.

Region descriptors

The following features are used to described regions of interest:

SURF detector

Speeded Up Robust Features (SURF) [BTVG06] is a scale- and rotation-invariant interest point detector and descriptor modeled after SIFT [Low99]. The object of SURF is to be a faster approximation of SIFT without penalizing its performance.

SURF detects "interesting" points in the image using an approximation of the Hessian matrix (second order derivative):

$$H = \begin{bmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma) & L_{yy}(x,\sigma) \end{bmatrix} \approx \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} = H_{approx} \tag{4.24}$$

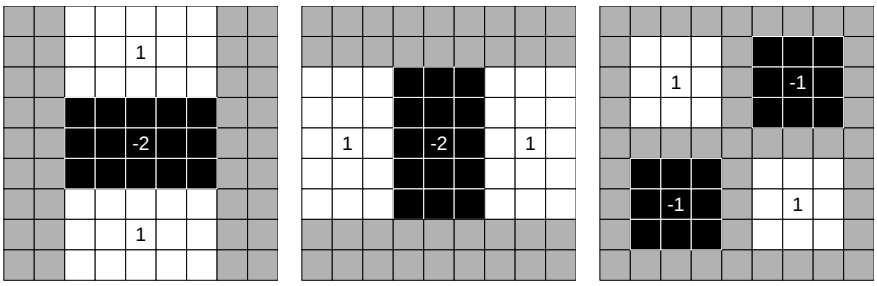
where $L_{xx}(x,\sigma)$ is the result of the convolution of the Gaussian second order derivate $\frac{\delta^2}{\delta x^2} g(\sigma)$ with the image (L_{xy} and L_{yy} analog). D_{xx} is the approximation of this derivative using simple box filters, as seen in figure 4.9 (D_{xy} and D_{yy} analog). These box filter have the advantage that they, in combination with the integral image [VJ01], can be computed in constant time, regardless of size. The determinate of H_{approx}, a cornerness measure is computed as follows:

$$\det(H_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2 \tag{4.25}$$

SIFT implements scale pyramids by downsampling the original image. SURF, instead, achieves similar results by increasing the size of the box filters. A non-maximum suppression is then applied in a $3 \times 3 \times 3$ neighborhood. The maxima are then interpolated in scale and space, resulting in the interest points.

SURF descriptor

SURF [BTVG06] also provides a feature descriptor, which allows comparison and matching of detected interest points. Similar as the SIFT descriptor, the SURF descriptor provides information on the intensity distribution around the interest point. Given a fixed interest point, a reproducible orientation is computed to achieve rotational invariance. This is done by calculating the Haar-wavelet responses in x and y directions in a circular neighborhood with radius $6s$, s being the scale at which the point was detected. These responses are weighted with a Gaussian of $\sigma = 2.5s$, centered at the interest point. The circular neighborhood is then divided into windows covering $\frac{\pi}{3}$. For each window, we sum the x and y Haar-wavelet responses separately ($\sum x, \sum y$) and then construct a vector $\begin{bmatrix} \sum x \\ \sum y \end{bmatrix}$. The longest of these vectors provides the orientation



(a) D_{xx} (b) D_{yy} (c) D_{xy}

Figure 4.9: Approximation of the Gaussian second order derivatives [BTVG06].

of the interest point. To extract the descriptor, a square region of side length $20s$ centered on the interest point and with the previously determinant orientation is examined. This square is divided into 4×4 subregions, where for each region at 5×5 intervals, the Haar-wavelet responses d_x and d_y (filter size of $3s$) are computed and then weight with a Gaussian of $\sigma = 3.3s$, centered at the interest point. For each subregion the description vector $v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$ is computed. These 16 vectors are then concatenated into one 64 dimensional descriptor of the interest point.

Bag of words

The SURF detector makes it possible to extract distinctive features from a region of interest. Though for each examined region of interest, we receive a varying number of these features, making it necessary to combine their SURF descriptors into one descriptor, representing the contents of a given region of interest, that can be directly entered into a classifier. We therefore constructed a vocabulary over all the SURF descriptors found in our training data, making it possible to use a bag of words approach, similar as in [SZ09]. Given a training set of N regions of interest I_i, we assume here that a region of interest is given as a rectangular image excerpt, we apply the SURF detector to each I_i, collecting altogether M interest points or rather descriptors $D_m \in \mathbb{R}^{64}$. To reduce the dimensionality, we perform a principal component analysis [Pea01]. We retain the first λ components, where λ is the smallest value that satisfies

$$\sum_{i=1}^{\lambda} e_i \geq 0.9 \cdot \sum_{i=1}^{64} e_i \tag{4.26}$$

$e_i \in \mathbb{R}$ are the singular values of the covariance matrix

$$\Sigma = \frac{1}{M} \sum_{m=1}^{M} (D_m - \bar{D})(D_m - \bar{D})^T \tag{4.27}$$

with $\bar{D} = \frac{1}{M} \sum_{m=1}^{M} D_m$. Each D_m is then projected onto its reduced form $\tilde{D}_m \in \mathbb{R}^{\lambda}$. We distill a vocabulary from all D_m via K-Means clustering [M$^+$67], with $K = \sqrt{\frac{M}{2}}$. The centers were initialized via K-Means++ [AV07]. These K centers C_i allow us to construct a descriptor for a region of interest in the following manner:

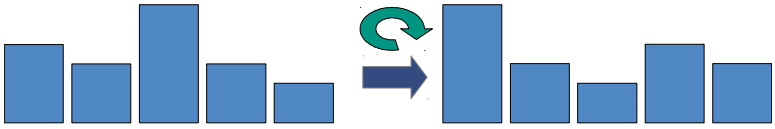


Figure 4.10: From gradient, an orientation histogram is computed and then aligned according to the highest peak.

1. Detect SURF interest points D_i and compute descriptors
2. For each D_i, compute $L_i = \text{argmin}_L \|C_L - D_i\|$
3. Compute K bin histogram: $H_k = |\{L_i = k | \forall L_i\}|$
4. Normalize histogram: $H_k = \frac{H_k}{\sum_i H_i}$

The normalized histogram can then be used as descriptor for the entire region of interest.

Color and gradient histograms

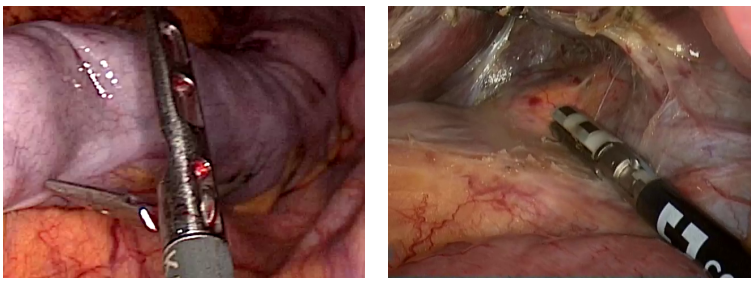
In addition to the interest point vocabulary, features on the color and gradient distributions in the regions of interest are also computed. To incorporate the color distribution, a region of interest was first transformed into the HSV color space (as outlined in section 4.1.1). Two normalized 10-bin histograms are then constructed, one over the hue channel and one over the saturation channel. The value channel is ignored in order to minimize the effects of varying light conditions.

For the gradient information, two further histograms are constructed. First, the Sobel operators in x and y are applied to the region of interest (see section 4.1.1). The orientation and magnitude of the gradient for each pixel are calculated from the results and used to construct two 5-bin histograms. Each orientation is mapped onto the interval $[0°, 180°]$ and, to achieve rotation invariance, the bins of the resulting histogram are rotated so that the bin with most entries is at the first position (figure 4.10).

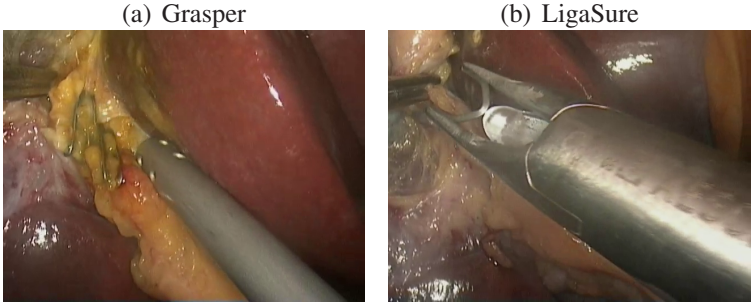
Cascade classifier

To speed-up the identification process, a cascade of two random forests [Bre01] is used. The first random forest is used to solve the binary problem of determining if a region of interest contains an instrument or not. Only the hue and saturation histogram are used as feature vector, as they can be computed quickly. To avoid a large number of false positives, only after more than a certain percentage of the trees in the random forest (threshold α) classify the region as not containing an instrument, is the region discarded.

If the region is not discarded, a second random forest will determine if the region contains an instrument and distinguish between instrument classes. Here, in addition to hue and saturation histograms, the gradient histograms and the bag of words are used. These extra histograms are computed only if the region of interest clears the first cascade.



(a) Grasper (b) LigaSure



(c) Aspirator (d) Clip applier

Figure 4.11: Example images of the instruments used.

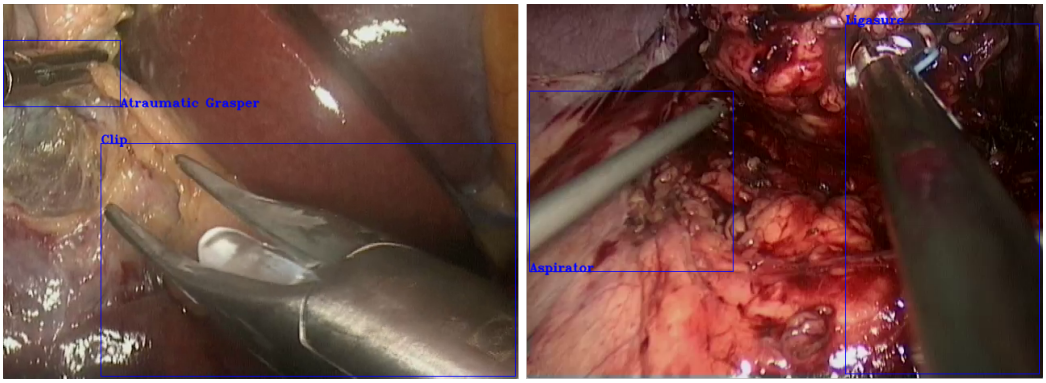


Figure 4.12: Example annotations of the instrument type

4.2.2 Evaluation of the instrument identification

From the five videos used for the**Crowd** dataset, we constructed two datasets, the first, **TypeManual**, consisting of manually selected frames, here it was ascertained that at least 10 instances of each class from each video were selected. For the second data set, **TypeAutomatic**, 100 frames were taken automatically at a fixed interval from each video. In each frame, the instruments were labeled with an axis aligned bounding box, furthermore the type of instrument was also annotated. We labeled 4 different types of instruments: *LigaSure*[4], an electric vessel sealing and dividing tool, *grasper*, *aspirator* and *clip applier* (see figure 4.11). Furthermore, we selected regions in images in both data sets were no instrument was visible and labeled these as well. Example annotation can be seen in figure 4.12. The data sets were labeled by an expert, as we deemed a medical background necessary for selecting the right tool from a sin-

[4] http://www.medtronic.com/covidien/products/vessel-sealing (accessed: July 2, 2018)

	# Instruments	# detected	# identified as instrument
TypeManual	411	386 (94%)	360.5 (88%)
TypeAutomatic	650	554.6 (85%)	517 (80%)

Table 4.3: Results of the combination of instrument detection and identification showing how many instruments were found and how many of these were correctly identified as an instrument.

gle frame. These two data sets were used to determine how many instruments were successfully detected.

For training purpose, only **TypeManual** was used. A leave-one-surgery-out evaluation was performed, training on the data from four operations and evaluating on the fifth. We performed two experiments: In the first experiment, the manually provided bounding boxes were used as input for the random forest cascade and the results were compared to the labels provided by experts. In the second experiment, the instrument detector introduced in section 4.1 was used to first locate regions of interest in given images. These automatically located bounding boxes were then identified.

For all experiments, 300 trees and a maximum depth of eight were selected for both random forests. During the cascade, $\alpha = 60\%$ was used.

Results

In table 4.3, the number of correctly detected instrument bounding boxes and the number of those correctly identified as a type of instrument (but not necessarily the right type) are listed. The results of the identification method on the manually and automatically selected bounding boxes can be found in tables 4.4 and 4.5 respectively, in form of a confusion matrix per set. We also computed the average percentage of correctly identified classes (table 4.6).

Run-time

The evaluation was also run on a workstation PC with an Intel Core i7-5820K, 32GB of RAM and an NVidia Titan X. The average run-time was at approximately 6.3 ms per given region of interest or on average 30 ms for all regions in a given frame (32.2Hz).

4.2.3 Discussion

In this section we presented, to our knowledge, the first approach for a real-time image-based identification of surgical tools in a laparoscopic setting. We are currently able to correctly detect 80% of all instruments in a realistic data set. Furthermore we are able to correctly determine the type of instrument in 48% of all cases in the same data set. The proposed combination of instrument detection and tracking has also been successfully used as input for a camera robot in laparoscopic surgery [WBM+15].

(a) **TypeManual**

		Actual class				
		No ins.	*LigaSure*	*Grasper*	*Aspirator*	*Clip applier*
Predicted	*No ins.*	69%	3%	16%	10%	1%
	LigaSure	1%	67%	18%	8%	6%
	Grasper	10%	26%	44%	15%	5%
	Aspirator	10%	13%	11%	38%	28%
	Clip applier	4%	5%	11%	10%	71%

(b) **TypeAutomatic**

		Actual class				
		No ins.	*LigaSure*	*Grasper*	*Aspirator*	*Clip applier*
Predicted	*No ins.*	58%	4%	25%	10%	3%
	LigaSure	2%	54%	18%	15%	11%
	Grasper	13%	21%	43%	14%	9%
	Aspirator	2%	14%	13%	50%	21%
	Clip applier	18%	17%	30%	0%	35%

Table 4.4: Confusion matrices illustrating the identification performance on **TypeManual** (a) and **TypeAutomatic** (b) on manually drawn bounding boxes

Some of the major problems that occur, especially on realistic data, can be seen in figure 4.13. If part of an instrument, e.g. the tip (figure 4.13(a)), is occluded by tissue or blood, ambiguities are possible, since different types of instrument share a similar shaft and can therefore only be reliably distinguished by the tip. Motion blur (figure 4.13(c)) can also cause ambiguities. If two instruments overlap (figure 4.13(b)) they can be detected as one instrument. Further error sources are differences in illumination and white balance, which can vary between different operations, especially if a different optic is used. The most common confusions were between *LigaSure*, *grasper* and *aspirator* as they share a similar formed shaft, which, under different lighting conditions, can be difficult to distinguished. Also, when only a small portion of the instrument could be found, either due to occlusions or due to it just entering the field of view, a confusion with the *no instrument* class was frequent.

4.3 Instrument Tracking

A solution to the majority of the problems discussed in the previous section would be a tracking method that propagates successfully detected and identified instruments over time. This would mitigate the effect of occluded instrument tips, assuming the tip was visible in previous frames.

(a) **TypeManual**

		Actual class				
		No ins.	*LigaSure*	*Grasper*	*Aspirator*	*Clip applier*
Predicted	*No ins.*	50%	4%	21%	25%	0%
	LigaSure	1%	46%	16%	25%	11%
	Grasper	15%	20%	34%	18%	12%
	Aspirator	6%	7%	9%	58%	20%
	Clip applier	3%	1%	7%	17%	71%

(b) **TypeAutomatic**

		Actual class				
		No ins.	*LigaSure*	*Grasper*	*Aspirator*	*Clip applier*
Predicted	*No ins.*	51%	4%	18%	26%	1%
	LigaSure	4%	43%	17%	27%	9%
	Grasper	14%	17%	33%	19%	17%
	Aspirator	1%	7%	8%	62%	23%
	Clip applier	11%	7%	33%	16%	33%

Table 4.5: Confusion matrices illustrating the identification performance on **TypeManual** (a) and **TypeAutomatic** (b) on automatically detected instrument bounding boxes

	Identification	Detection & Identification
TypeManual	58%	49%
TypeAutomatic	52%	48%

Table 4.6: The average percentage of correctly identified tools in each data set. The first column contains the results based on manually annotated instruments and the second column the results based on automatically detected instruments.

In this section, we propose a method for tracking laparoscopic instruments that have been previously detected and identified by the methods described in the previous sections over time. Further details can be found in [Woc15]. An overview of the different components of the method can be found in figure 4.14.

4.3.1 Methods for instrument tracking

Our proposed method for instrument tracking relies on optical flow [HS81], which describes the movement of objects in image sequences. We rely on a method that computes the optical flow for only a given amount of pixels in an image (sparse optical flow). This, in contrast to dense optical flow, can be achieved quickly, allowing real-time applications. Such a sparse method though requires features that can be reliably tracked over time.

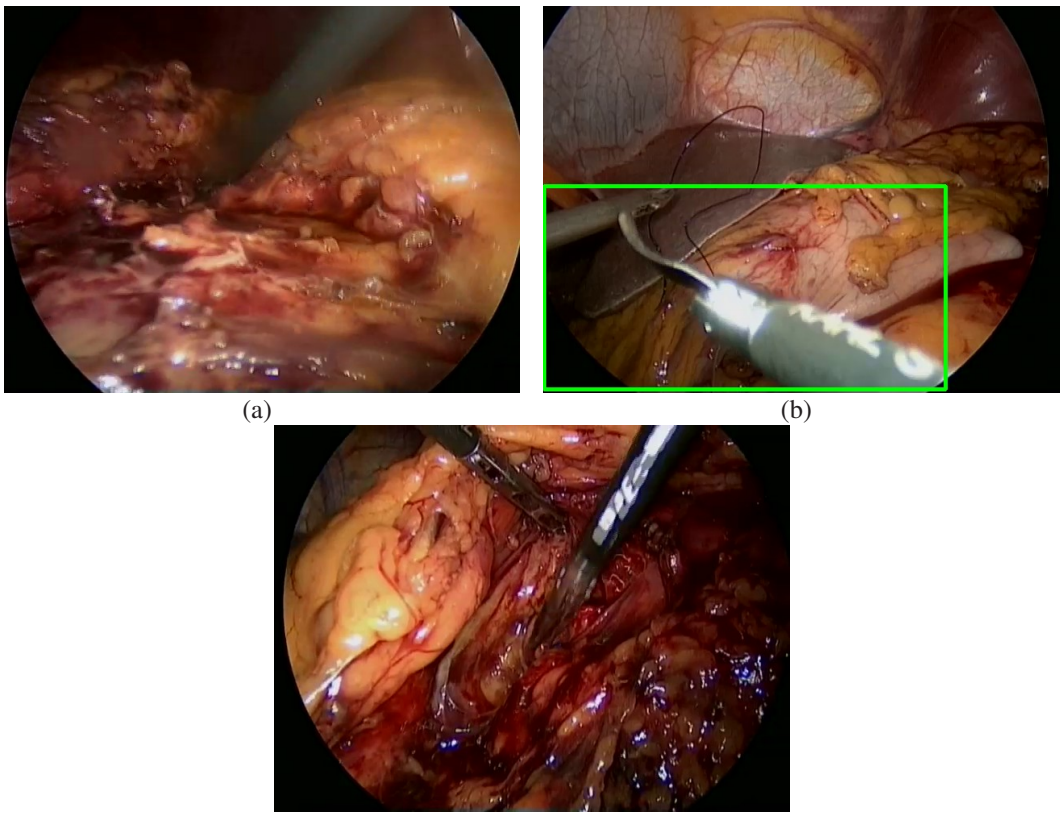


Figure 4.13: Potential error sources: (a) Instrument tip not visible, (b) two overlapping instruments detected as one and (c) motion blur.

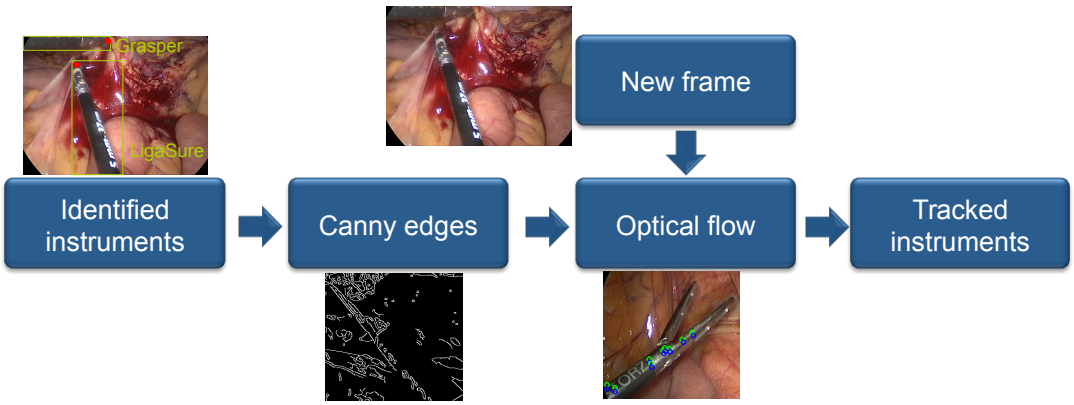


Figure 4.14: Overview of the components used for instrument tracking.

Optical flow

According to [HS81] optical flow is "the distribution of apparent velocities of movement of brightness patterns in an image", which can arise due to relative motion between the viewed objects and the camera. Therefore, optical flow can provide information on the arrangement of objects and the rate of change of this arrangement. To allow the optical flow to be calculated, assumptions have to be made. One of these assumptions is brightness consistency. In other words, given the image brightness $I(x,y,t)$ of a

pixel $p = (x, y)$ at time t brightness consistency implies that the brightness of a moving object at p does not change for a small time period δt:

$$I(x,y,t) \stackrel{!}{=} I(x+\delta x, y+\delta y, t+\delta t) \tag{4.28}$$

A further assumption is that the motion in the image that occurs in a δt period is small. This allows us to use first order Taylor expansion:

$$\begin{aligned} I(x+\delta x, y+\delta y, t+\delta t) &= I(x,y,t) + \delta x \cdot \frac{\delta I}{\delta x} + \delta y \cdot \frac{\delta I}{\delta y} + \delta t \cdot \frac{\delta I}{\delta t} + \dots \\ &\approx I(x,y,t) + \delta x \cdot \frac{\delta I}{\delta x} + \delta y \cdot \frac{\delta I}{\delta y} + \delta t \cdot \frac{\delta I}{\delta t} \end{aligned} \tag{4.29}$$

Plugging equation 4.28 into equation 4.29 results in:

$$\begin{aligned} &\delta x \cdot \frac{\delta I}{\delta x} + \delta y \cdot \frac{\delta I}{\delta y} + \delta t \cdot \frac{\delta I}{\delta t} = 0 \Rightarrow \delta x \cdot \frac{\delta I}{\delta x} + \delta y \cdot \frac{\delta I}{\delta y} + \delta t \cdot \frac{\delta I}{\delta t} = \\ &\frac{\delta x}{\delta t} \cdot \frac{\delta I}{\delta x} + \frac{\delta y}{\delta t} \cdot \frac{\delta I}{\delta y} + \frac{\delta t}{\delta t} \cdot \frac{\delta I}{\delta t} = u \cdot \frac{\delta I}{\delta x} + v \cdot \frac{\delta I}{\delta y} + \frac{\delta I}{\delta t} = 0 \\ &\Rightarrow u \cdot \frac{\delta I}{\delta x} + v \cdot \frac{\delta I}{\delta y} = -\frac{\delta I}{\delta t} \end{aligned} \tag{4.30}$$

where $u = \frac{\delta x}{\delta t}$ and $y = \frac{\delta y}{\delta t}$ are the x and y component of the velocity vector or the optical flow $V = \left[\begin{smallmatrix} u \\ v \end{smallmatrix}\right]$. Since this equation has two unknowns, further assumptions are required for solving for u and v [HS81].

Lucas-Kanade optical flow

The approach of Lucas and Kanade [LK+81] attempts to solve for u and v of a given pixel p by applying local constraints. They assume that all N pixels $q_n = (x_n, y_n)$ in a window around p undergo a similar displacement, leading to the following system of equations:

$$\begin{aligned} u \cdot \frac{\delta I(x_1,y_1,t)}{\delta x} + v \cdot \frac{\delta I(x_1,y_1,t)}{\delta y} &= -\frac{\delta I(x_1,y_1,t)}{\delta t} \\ u \cdot \frac{\delta I(x_2,y_2,t)}{\delta x} + v \cdot \frac{\delta I(x_2,y_2,t)}{\delta y} &= -\frac{\delta I(x_2,y_2,t)}{\delta t} \\ &\vdots \\ u \cdot \frac{\delta I(x_N,y_N,t)}{\delta x} + v \cdot \frac{\delta I(x_N,y_N,t)}{\delta y} &= -\frac{\delta I(x_N,y_N,t)}{\delta t} \end{aligned} \tag{4.31}$$

This can then be solved via least squares: $V = (A^T A)^{-1} A^T b$, with

$$A = \begin{bmatrix} \frac{\delta I(x_1,y_1,t)}{\delta x} & \frac{\delta I(x_1,y_1,t)}{\delta y} \\ \frac{\delta I(x_2,y_2,t)}{\delta x} & \frac{\delta I(x_2,y_2,t)}{\delta y} \\ \vdots & \\ \frac{\delta I(x_N,y_N,t)}{\delta x} & \frac{\delta I(x_N,y_N,t)}{\delta y} \end{bmatrix}, V = \begin{bmatrix} u \\ v \end{bmatrix}, b = \begin{bmatrix} -\frac{\delta I(x_1,y_1,t)}{\delta t} \\ -\frac{\delta I(x_2,y_2,t)}{\delta t} \\ \vdots \\ -\frac{\delta I(x_N,y_N,t)}{\delta t} \end{bmatrix} \tag{4.32}$$

Selecting the window around p is not simple, as too small a neighborhood might lead $A^T A$ to not be invertible due to noise. Also large motions might not be accurately accounted for. If the window is too large, local accuracy will degrade. To compensate, the authors in [Bou01] suggest a pyramid-based approach.

Feature selection

Feature selection places an important role in tracking and also for computing the optical flow. In [NVBR12], the authors examined how well different feature types could be tracked over time using Lucas-Kanade optical flow. Their results show that Canny edges [Can86] were quickly computed, provided a large number of features and performed comparable to state of the art methods such as SIFT [Low99], SURF [BTVG06], Harris corners[HS88] and Good Features To Track [S$^+$94].

Canny edges

The Canny edge detector [Can86] is a four step method that extracts edges from a given image. As input, a grayscale image and two thresholds, t_1 and t_2 are required [BK08].

1. Noise suppression using a Gaussian filter.
2. Compute gradients of x and y using the Sobel operators and the magnitude and orientation of each gradient (see section 4.1.1).
3. Non-maximum suppression: Gradients that are not the maximum when examining its two neighbors in gradient direction are discarded.
4. Hysteresis thresholding: If the gradient magnitude of a pixel is larger than t_2, it is kept. Afterwards, the 3×3 neighborhood of kept pixels are examined and pixels with magnitudes larger than t_1 are also kept (iterative).

An example of detected edges can be seen in figure 4.15.

Tracking detected instrument

To track laparoscopic instruments, we first have to detect candidates for tracking. This is accomplished with the instrument detection method outlined in section 4.1. Whenever no instrument is currently tracked, the instrument detector is applied to a new

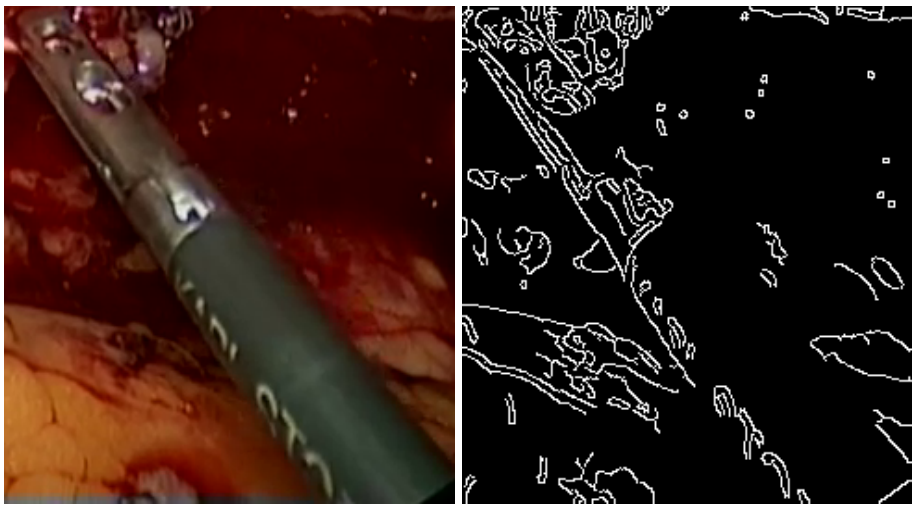


Figure 4.15: Edges detected by the Canny edge detector

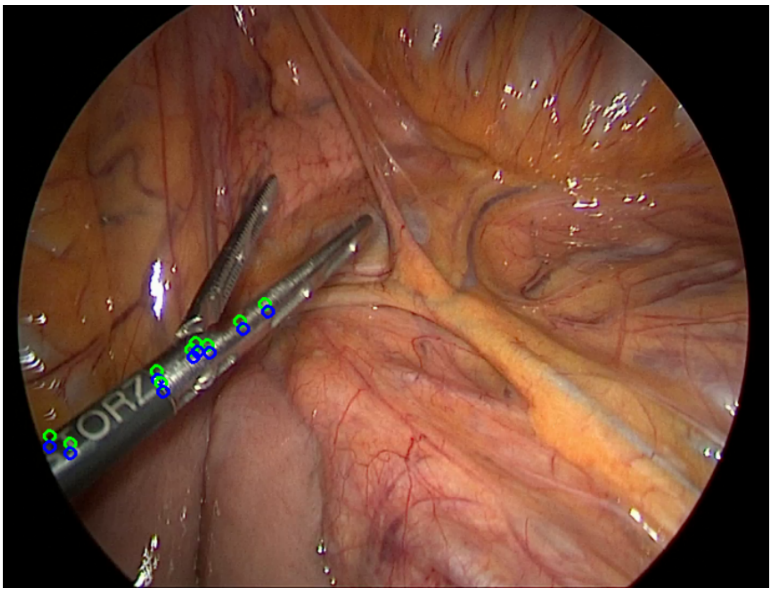


Figure 4.16: Original features (green) and their new positions as estimated using Lucas-Kanade (blue)[Woc15]

frame to extract regions of interest, which are then identified via the method in section 4.2. Once a region of interest containing an instrument has been successfully identified, we use the Canny edge detector to detect potential features to track. Only edge points that correspond to pixels previously identified as instrument pixels are considered. If no features were found, the region of interest is discarded.

For every new frame, we then update the positions of the features, of the region of interest and of the instrument tip. Given a new frame, the Lucas-Kanade is used to determine the optical flow vector $V_i = \left[\begin{smallmatrix} u_i \\ v_i \end{smallmatrix}\right]$

Furthermore, for each propagated feature, we compute a matching error ε_i by calculating the L1 distance between the patch around the original point and the matched point. We then determine the actual tool displacement via quadrant voting. Each V_i is sorted

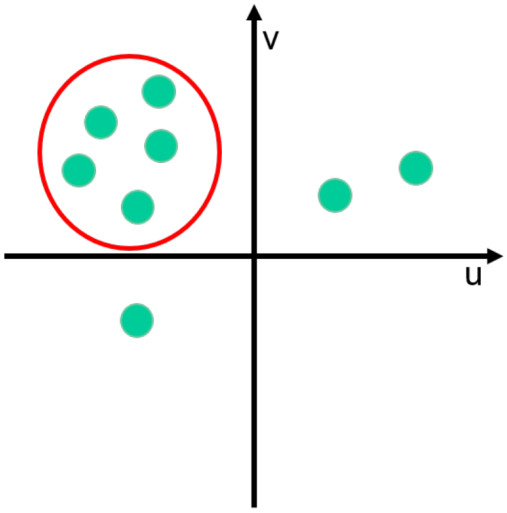


Figure 4.17: Quadrant voting: Each feature is sorted into one quadrant according to the signs of the components of V_i.[Woc15]

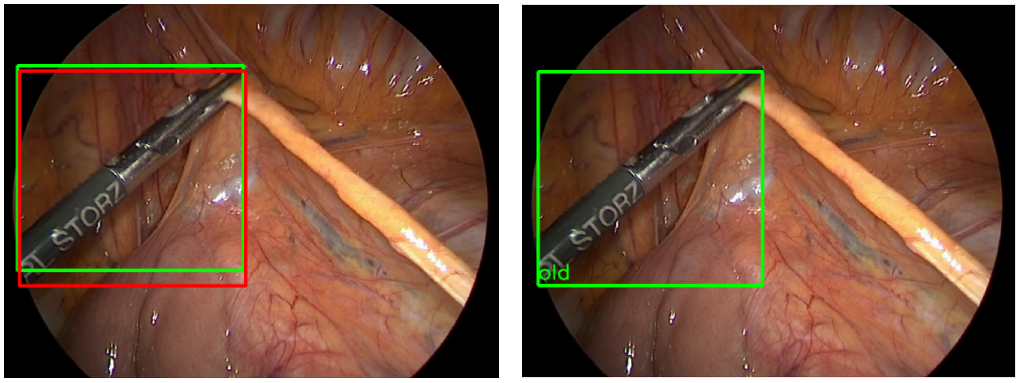
into one of four bins, according to the signs of its two components. The quadrant containing the largest amount of features is then kept, the rest discarded (see figure 4.17). This method increases the tracking's robustness towards outliers. We then sort the features in the majority quadrant according to their value of ε_i and average the 50% with the smallest error. This averaged displacement error is then used to propagate the region of interest and the instrument tip.

When few or no features are detected in a region of interest, we apply the instrument detector on an enlarged region of interest to reinitialize the tracking. If this fails, we mark the instrument as lost.

When no instruments are being tracked, we run the instrument detector on the whole image until new instruments are detected. To detect new instruments and to avoid feature drift, the detector is used in regular intervals to detect new instruments in the image, but also to supply currently tracked instruments with new features (see figure 4.18). If two instruments are touching or in close proximity to one another, the instrument detection method sometimes merges region of interests belonging to multiple instruments. To avoid this situation, we perform a sanity check (see figure 4.19). When a newly detected region of interest intersects with multiple tracked regions of interest, we examine the ratio of detected instrument pixels to the area of each region of interest. If the ratio is higher for the multiple tracked region of interest, the result of the detector are rejected.

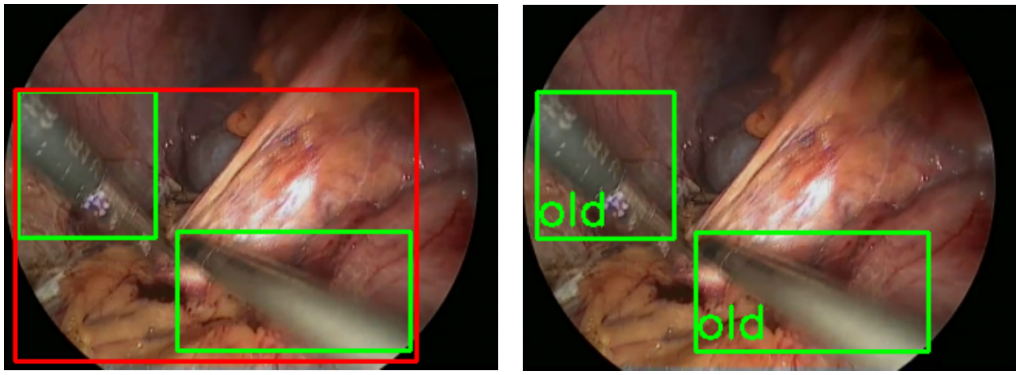
4.3.2 Evaluation of the instrument tracking

We performed two separate evaluation to examine the accuracy of the tracking method. First, we applied the tracking method to the datasets **TypeManual** and **TypeAutomatic** introduced in section 4.2.2. Since the the tracking requires temporal information, we



(a) (b)

Figure 4.18: We regularly updated tracked instruments (a) (green: results tracking. red: newly detected region of interest) with new features by re-detecting instruments (b) (green: merged region of interest) [Woc15].



(a) (b)

Figure 4.19: To avoid wrongly merged region of interests from the detector (a) (green: results tracking. red: newly detected region of interest) a sanity check is performed and the newly detected region is rejected (b) [Woc15].

used the entire videos as input and compute the accuracy of the detected instrument type on the annotated frames.

Furthermore, we evaluated the tracking method on another dataset of 6 laparoscopic colorectal surgeries, from which we extracted 14 separate video sequences, each lasting one minute (1500 frames). Each sequence had a frame resolution of 960×540. In every 25*th* frame, the positions of the tips of the visible instruments were manually annotated (figure 4.20). On this dataset, we applied the standalone instrument detector and the proposed tracking method on the sequences. We then compared the error in detecting the instrument tips between the two methods. For this, the euclidean distance between the tracked and the annotated instrument tips is computed. Furthermore, we compared the number of instruments found.

Results

Table 4.7 shows the results of running the instrument tracking on the previously introduced **TypeManual** and **TypeAutomatic** datasets (see section 4.2.2). The numbers

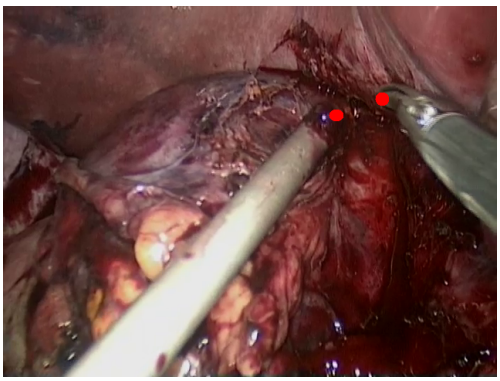


Figure 4.20: Example annotation of instrument tips

(a) **TypeManual**

		Actual class				
		No ins.	*LigaSure*	*Grasper*	*Aspirator*	*Clip applier*
Predicted	*No ins.*	76.5%	4.4%	5.9%	7.4%	5.9%
	LigaSure	4.3%	78.3%	8.7%	4.3%	4.3%
	Grasper	15.8%	15.8%	52.6%	15.8%	0.0%
	Aspirator	5.3%	10.5%	15.8%	57.9%	10.5%
	Clip applier	12.5%	6.3%	6.3%	0.0%	75.0%

(b) **TypeAutomatic**

		Actual class				
		No ins.	*LigaSure*	*Grasper*	*Aspirator*	*Clip applier*
Predicted	*No ins.*	70.4%	7.0%	7.0%	9.9%	5.6%
	LigaSure	4.8%	62.9%	32.3%	0.0%	0.0%
	Grasper	3.7%	29.6%	59.3%	7.4%	0.0%
	Aspirator	9.5%	9.5%	4.8%	71.4%	4.8%
	Clip applier	22.2%	11.1%	5.6%	5.6%	55.6%

Table 4.7: Confusion matrices illustrating the identification performance in combination with tracking on **TypeManual** (a) and **TypeAutomatic** (b) on automatically detected instrument bounding boxes (Compare to table 4.5).

should be compared to the results of the combination of instrument detection and identification shown in table 4.5. Table 4.8 compares the number of actual instruments found by the tracking method and by the detector, as well as the accuracy of the detected tip. Figure 4.21 shows an example sequence of a tracked instrument and figure 4.22 gives an example on how both detector and tracker handle a situation were instruments start to overlap.

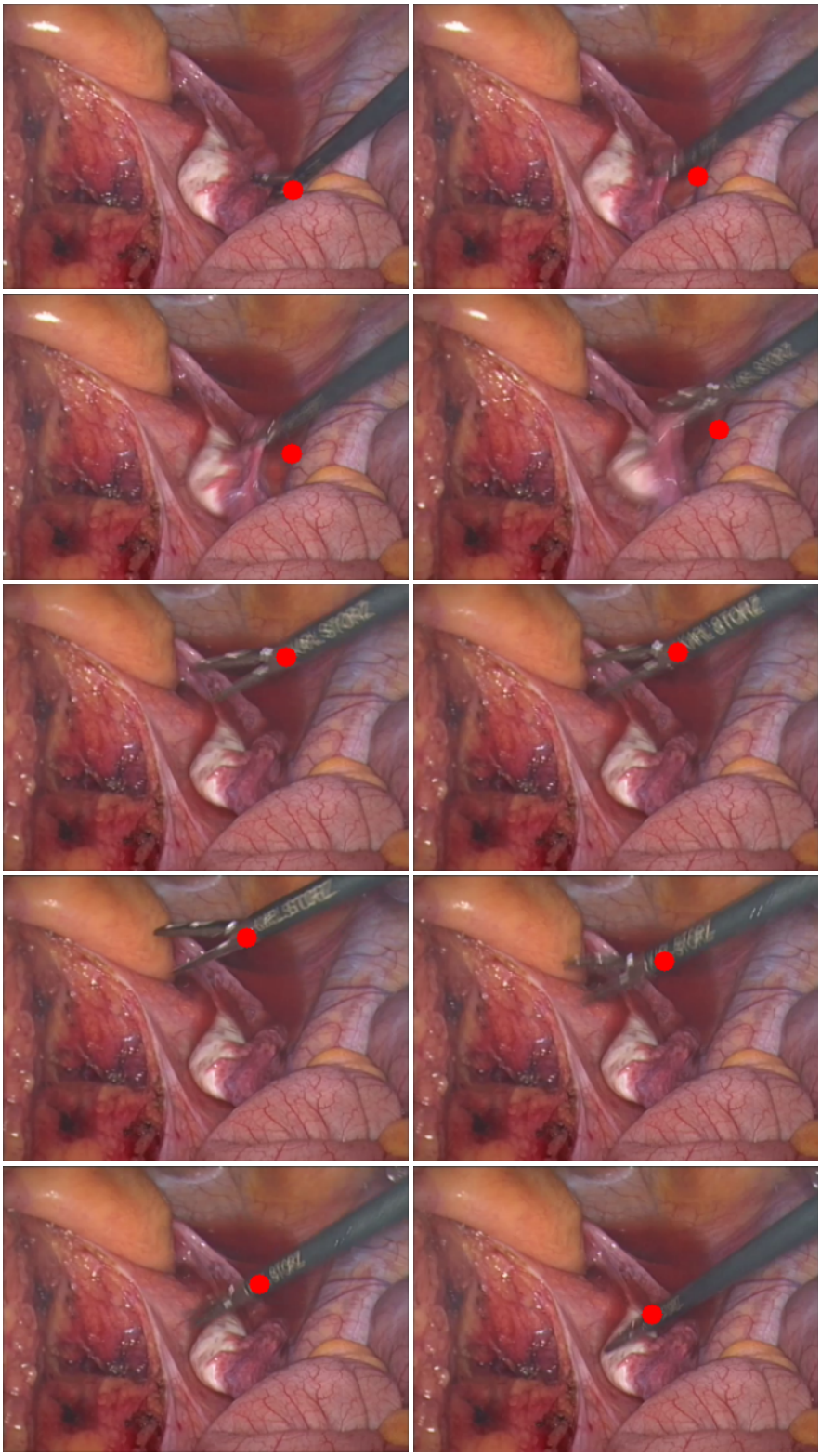
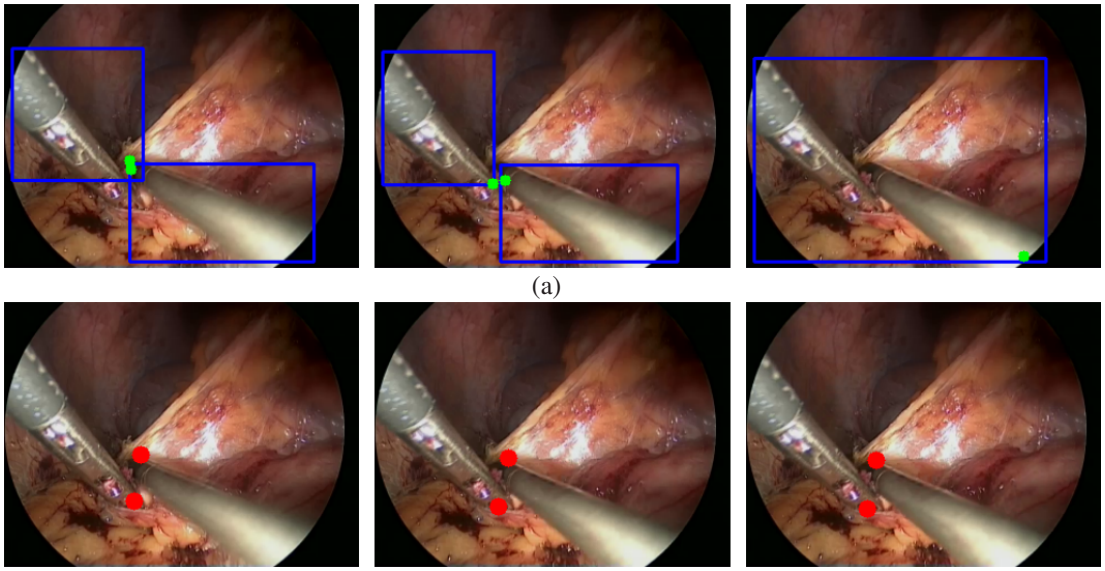


Figure 4.21: Example sequence of tracking results [Woc15]. The red circle symbolizes the propagated instrument tip.

	Instrument Detector	Instrument tracking
# instruments found (of 1038)	649	812
Avg. tip error	66.5px	65.9px

Table 4.8: Comparison of the instrument detector and the instrument tracking. The Tip error is the euclidean distance between the tracked instrument tip and the annotated instrument tip.



(a)

(b)

Figure 4.22: A comparison of how the detector (a) and the tracker (b) handle a situation with overlapping instruments [Woc15].

Run-time

During the above mentioned experiment the tracking method averaged a run-time of 10.1 ms on images with a resolution of 960×540. The detector achieved a performance of 33.9 ms on the same dataset. The evaluation was run on a workstation PC with an Intel Core i7-5820K, 32GB of RAM and an NVidia Titan X.

4.3.3 Discussion

In this section, we introduced a method for tracking laparoscopic instrument in real-time. The approach builds and incorporates on the previously introduced methods for instrument detection (section 4.1) and identification (section 4.2). The proposed method was able to increase the accuracy of the instrument identification (table 4.7) by propagating information from past frames into future frames. The most distinguishable feature of laparoscopic instruments is often the tip, which can be covered by, for example, tissue or other instrument. The tracking method makes to possible to propagate information from a frame where the tip is sufficiently visible into the future.

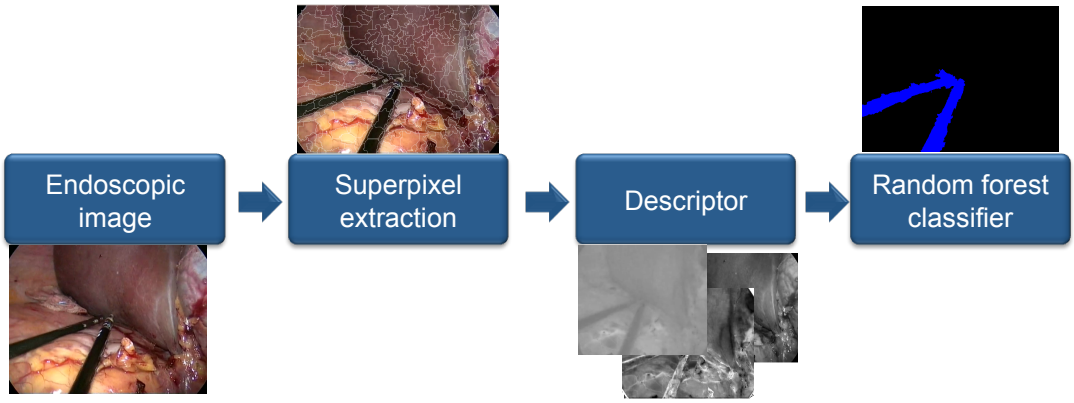


Figure 4.23: Overview of the components used for superpixel-based surgical object segmentation.

Furthermore the method increase the number of tools found in comparison to the detector (table 4.8). The instrument tip detection error does not vary significantly between the detector and the tracker (table 4.8), which serves as an indicator for the robustness of the method, as the tip candidate is only propagated from detection results. The instrument tip is not always correctly detected by the detector due to tissue reflections. An example of this can be seen in figure 4.21. The tracking approach is also able to augment the detection results in situation were the detector erroneously merged instruments due to proximity (figure 4.22).

The tracker is also able to furthermore reduce the average time required to detect instruments.

4.4 Superpixel-Based Surgical Object Segmentation

The segmentation method outlined in section 4.1 assigns each pixel a label solemnly based on its own value. The properties of neighboring pixels are largely ignored. Only when we by considering the gradient of a pixel do we slightly take some neighborhood information into consideration

In this section, we present a method that, instead of classifying each pixel separately, takes neighborhood information into consideration. For more details, please see [Gör15]. The work presented here was published in [BGW+16]. An overview of the different components of the method can be found in figure 4.23.

4.4.1 Methods for superpixel-based segmentation

We first segment the image into superpixel or, in other words, regions of connected, similar pixels. The content of these superpixels is then described using information regarding the color and texture of the pixels contained in it. The superpixels and manually annotated laparoscopic images are then used to train a random forest classifier, which will then be used to assign a label to each superpixel.

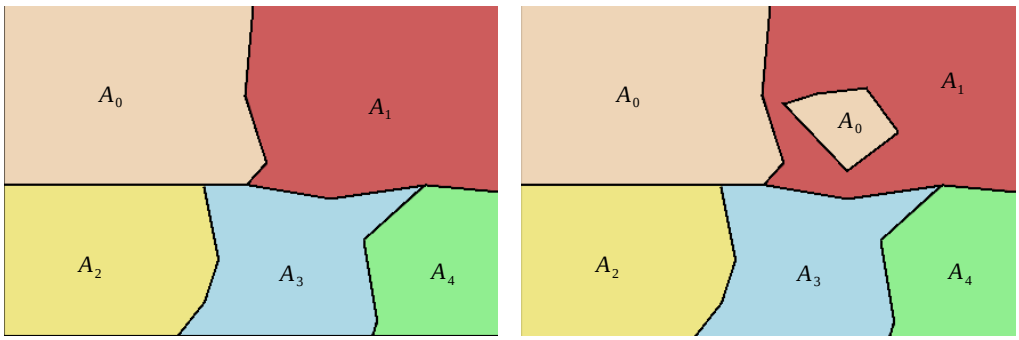


Figure 4.24: Example of a valid superpixel configuration (a) and an invalid configuration (b) [Gör15]

Superpixels

Given an image with N pixels and let K be the number of wanted superpixels, then the division of one image into superpixels can be described with the following mapping [VdBBR+12]:

$$s : \{1,...,N\} \rightarrow \{1,...,K\} \tag{4.33}$$

where $s(i)$ is the ID of the superpixel to which the pixel i is assigned.

A superpixel A_k can therefore be described in the following manner:

$$A_k = \{i : s(i) = k\} \tag{4.34}$$

Also let for two superpixels A_k and A_l with $k \neq l : A_k \cap A_l = \emptyset$, meaning that each pixel i is mapped to exactly one superpixel. Furthermore, each superpixel should only contain neighboring, i.e. spatially connected, pixels. Figure 4.24 contains an example of a valid and invalid superpixel configuration.

SEEDS

To divide a given image into superpixels, we make use of the SEEDS algorithm [VdBBR+12], which, at the time of writing, was the state of the art. SEEDS distinguishes itself to other methods due to its speed, achieving up to 30Hz on 640×480 images [VdBBR+12]. Other methods, such as SLIC [ASS+10] or Felzenszwalb et al. [FH04] achieve significantly lower frame rates.

The method proposed in [VdBBR+12] maximizes an energy function to divide an image I into K superpixels.

$$\hat{s} = \underset{s \in S}{\operatorname{argmax}} E(s) \tag{4.35}$$

where S is the set that contains all valid superpixel partitions. The proposed energy function consists out of two components:

$$E(s) = H(s) + \gamma G(s) \tag{4.36}$$

Figure 4.25: Example SEEDS segmentations of laparoscopic images [BGW+16]

$H(s)$ is a color distribution term, which describes the homogeneity of the colors in the superpixel, for this the color distribution in each superpixel is examined its histogram. The term penalizes larges differences in color value in a given superpixel. $G(s)$ is a boundary term, which describes the shape of the superpixel. To compute the term, a window around each pixel in the image is examined for the number of superpixels having pixels in the window. The term penalizes large numbers of superpixels in a given neighborhood.

The energy function is optimized via hill-climbing. Here it iteratively performs pixel-level and block-level updates.

Figure 4.25 shows the results of applying SEEDS to two laparoscopic images.

Descriptor

Once an image has been divided into K superpixels A_k, we need a representation of their content to determine the most probable label for each superpixel. For this, we construct a feature vector (or descriptor) from color and texture information.

Color descriptor

To describe the color information contained in a superpixel, we convert the original image into different color spaces (HSV, LAB and Opponent). Since the number of pixels per superpixel is not constant, a histogram with 25 bins is constructed for each channel of the aforementioned color spaces, as outlined in section 4.2.1. The combination of channel histograms used in the final descriptor is determined empirically.

Texture descriptor

To incorporate texture information, we utilize local binary patterns (LBP) [OPH94]. A LBP describes for each pixel i the differences between its intensity value compared to those of the pixels in its 3x3 neighbor hood via an 8D binary vector. If the intensity of i is larger than that of its n-th neighbor, the n-th entry in the binary vector is 1,

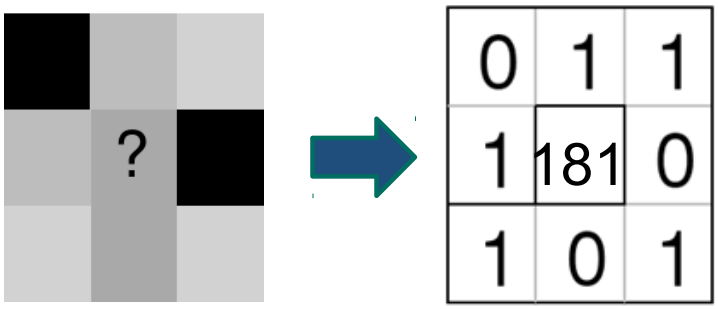


Figure 4.26: Example of a local binary pattern

otherwise 0 (see figure 4.26 for an example). The binary vector can be interpreted as an 8-bit integer, which would lead to an rotation-variant representation. This is undesirable, as objects with a certain texture should be correctly identified regardless of their orientation in the scene. Instead, we use the method outlined in [AHP06] to enter each binary vector into a 10-bin histogram according to the number and positions of 0 to 1 or 1 to 0 flips.

We then combine the color and the texture descriptor into one overall descriptor of a given superpixel A_k. To assign a label to each descriptor in an image, a random forest classifier (section 4.1.1) is used.

4.4.2 Evaluation of the superpixel-based segmentation

We evaluated our proposed superpixel-based segmentation method on the previously described **Crowd**, **EndoVisRigid** and **EndoVisRobotic** datasets. For each image, SEEDS was then used to partition the image into superpixels. As the superpixels were not always "pure", meaning not every pixel contained in a superpixel had the same annotation, we assigned each superpixel the label belonging to the majority of pixels contained in it. The following parameters were used for SEEDS: 1000 superpixels, variance of 3, 1 level, 7 bins and 10 iterations. For a detailed description of these parameters please see [VdBBR+12]. For the descriptor, hue saturation, o1 and o2 of the Opponent color space and LBP were selected. The random forest classifier was trained with a maximum depth of 16 and a maximum number of 200. We then performed a leave-one-out cross-validation, meaning we trained on 5 surgeries and tested on the 6th. The results of the leave-one-out evaluation can be found in table 4.9.

Run-time

The evaluation was run on a workstation PC with an Intel Core i7-5820K, 32GB of RAM. On the datasets, our method averaged a run-time of 118 ms per image with a resolution of 640×480 pixels.

(a) **Crowd**

	Precision	Recall	DICE	Accuracy
Pixel-based	64.8%±12%	75.6%±18.6%	68%±10.4%	89.7%±2.5%
Superpixel-based	69.2%±9.5%	73%±15.8%	69.4%±5%	91.2%±1.6%

(b) **EndoVisRigid**

	Precision	Recall	DICE	Accuracy
Pixel-based	73.5%±21.5%	44%±22.8%	50.4%±20%	93.1%±4.6%
Superpixel-based	62.3%±19.36%	76.3%±12.5%	66%±8.9%	92.1%±3.9%

(c) **EndoVisRobotic**

	Precision	Recall	DICE	Accuracy
Pixel-based	85.7%±9.1%	72.1%±7.4%	77.9%±6.3%	95.3%±1.5%
Superpixel-based	71.9%±20.9%	89.9%±5.1%	75.5%±17%	94.4%±4.8%

Table 4.9: Comparison of the superpixel-based segmentation method and the pixel-based method introduced in section 4.1 on **Crowd** (a), **EndoVisRigid** (b) and **EndoVisRobotic** (c).

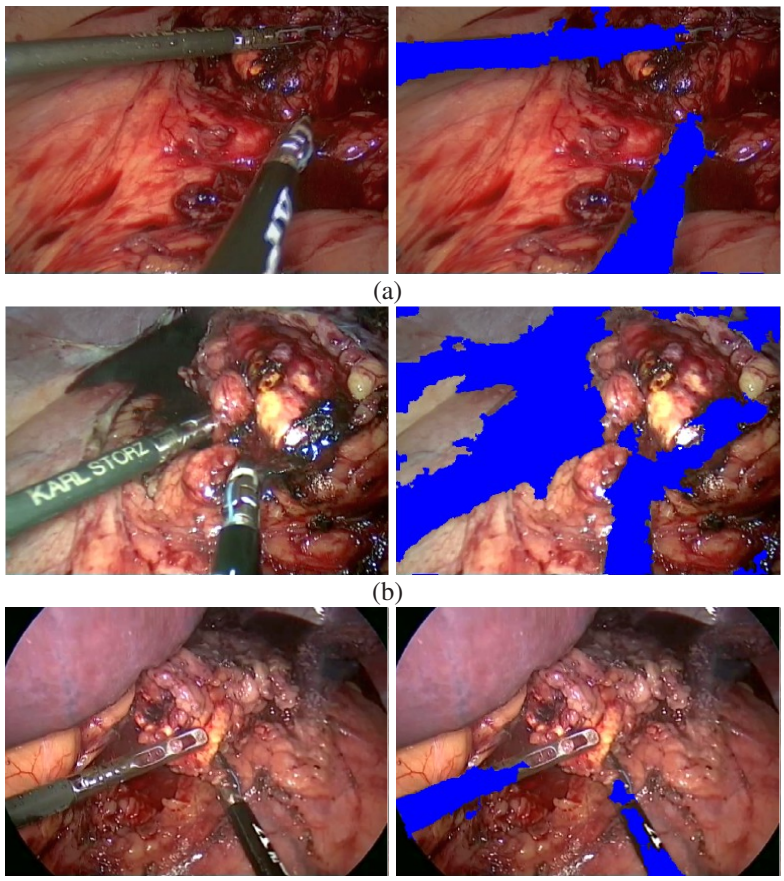
4.4.3 Discussion

We were able to show that an improvement in segmentation results can be achieved using a superpixel-based method in comparison to the previously described pixel-based method. Table 4.9 shows that we were able to achieve a significant improvement in precision on the **Crowd** dataset and a significant improvement in recall on the **EndoVisRigid** and **EndoVisRobotic** datasets. The DICE coefficient increased for both **Crowd** and **EndoVisRigid**. Only for **EndoVisRobotic** did the DICE coefficient decrease slightly.

Some of the decreased in the metrics can at least be partially contributed to the error entailed by the superpixel segmentation (figure 4.27(a)). In figure 4.27 further examples of common error sources can be found. Bleeding (figure 4.27(b)) can lead to false positives, which leads us to the conclusion that more training data is required. Also instrument tips with openings can be falsely segmented (figure 4.27(c)). A possible cause is that the structure is too small in order to be successfully segmented with a superpixel. While the results of the segmentation of the superpixel-based method are an improvement to the pixel-based method, the pixel-based method outperforms it significantly in terms of run-time. This makes the superpixel-based method currently unsuitable for realtime instrument segmentation.

4.5 Random Texton Forests for Image Content Classification

The previously presented random-forest-based segmentation method (section 4.1) and superpixel-based segmentation method (section 4.4) both required extensive annotation for training. Often, knowing only what objects are in an image and not their positions is sufficient, e.g. for content-based image retrieval or also phase detection.



(a)

(b)

(c)

Figure 4.27: Examples of common error sources: (a) small leakage of the superpixels at the edge of the instrument, (b) Misclassification of a bloody region, and (c) Missing instrument tip [BGW+16].

In this section, we will present an approach for classifying the content of laparoscopic images using weakly supervised learning. Weakly supervised in this case means that in contrast to the previous approaches, which required pixel-wise or superpixel-wise labels, this approach only requires image-wise labels, i.e. the visible object classes have to be annotated. For more details, please see [Neu16]. An overview of the different components of the method can be found in figure 4.28.

4.5.1 Methods for image content classification

In literature, the semantic texton forest [SJC08], a modified version of the random forest (see section 4.1.1) has been successfully used to build texture descriptors, or textons, from image regions, given only image-wise labels. The textons can then be used to build an image descriptor, which in turn can be used to assign semantic labels to an image.

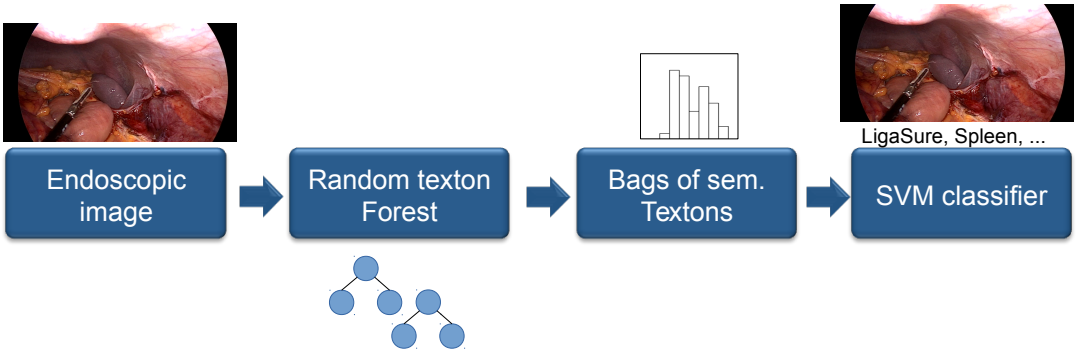


Figure 4.28: Overview of the components used for laparoscopic image content classification.

Random texton forest

A random texton forest is a modified version of a random forest specialized in describing the texture of a given image region of size $d \times d$. Random texton forests have the advantage that no complex feature selection, such as filter selection, is required beforehand. Instead, they operate directly on the pixels of a given image region [SJC08].

The random texton forest differs from the random forest in split functions. When training a standard random forest, an axis-aligned linear split function is used (see equation 4.19), where the axis and the threshold are selected via maximizing the information gain at each node. Instead, the split function, given an image region $x \in R^{d \times d \times c}$ with c being the number of channels, takes the following form:

$$h(x, \theta_j) = [h_{\tau_d}(x) < \tau_v] \tag{4.37}$$

with

$$\begin{aligned} h_1(x) &= x_{i_1,j_1,k_1} \\ h_2(x) &= x_{i_1,j_1,k_1} + x_{i_2,j_2,k_2} \\ h_3(x) &= x_{i_1,j_1,k_1} - x_{i_2,j_2,k_2} \\ h_4(x) &= |x_{i_1,j_1,k_1} - x_{i_2,j_2,k_2}| \end{aligned} \tag{4.38}$$

and $x_{i,j,k}$ being a random pixel value selected from x at position i, j and channel k [SJC08].

The random texton forest is trained in the same manner as the standard random forest. Each region x from an image has the label of the given image. In our case, we train a separate random texton forest for each class in our training set. The authors in [SJC08] recommend to train the forest on the CIElab color space. Instead, we combine channels from multiple color spaces described in section 4.1.1.

Bag of semantic textons

Once the random texton forest is trained, an image can be described using its textons. For this, the image is split into $d \times d$ regions, each region centered around a pixel in the image. Each region is then passed down each tree in the forest, taking note of the nodes the patch passes in each tree. The paths taken down each tree are then used to create a bag of semantic textons [SJC08].

For this, a histogram $H_I(n,t)$ with $N \cdot T$ entries is constructed, with N being the maximum number of nodes per tree and T the number of trees in the forest. Every time a region x from a given image I arrives at a node n in a tree t, $H_I(n,t)$ is incremented by one. After each region has been processed, the non-normalized histogram $H_I(n,t)$ can be used as a descriptor of the texture distribution in I [SJC08].

Classification

To assign a label to a given histogram, a non-linear support vector machine is used. As kernel, a modified version of the pyramid match kernel [GD05] is used. For one tree t with depth D, the distance between two histograms is computed in the following manner [SJC08]:

$$K_t(H_{I_1}, H_{I_2}) = \sum_{d=0}^{D-1} \frac{1}{2^{D-d}} (\iota_{t,d}(H_{I_1}, H_{I_2}) - \iota_{t,d+1}(H_{I_1}, H_{I_2})) \tag{4.39}$$

with ι being the histogram intersection over the portion of the histogram at depth d:

$$\iota_{t,d}(H_{I_1}, H_{I_2}) = \sum_{n=2^d}^{2^{d+1}-1} \min(H_{I_1}(t,n), H_{I_2}(t,n)) \tag{4.40}$$

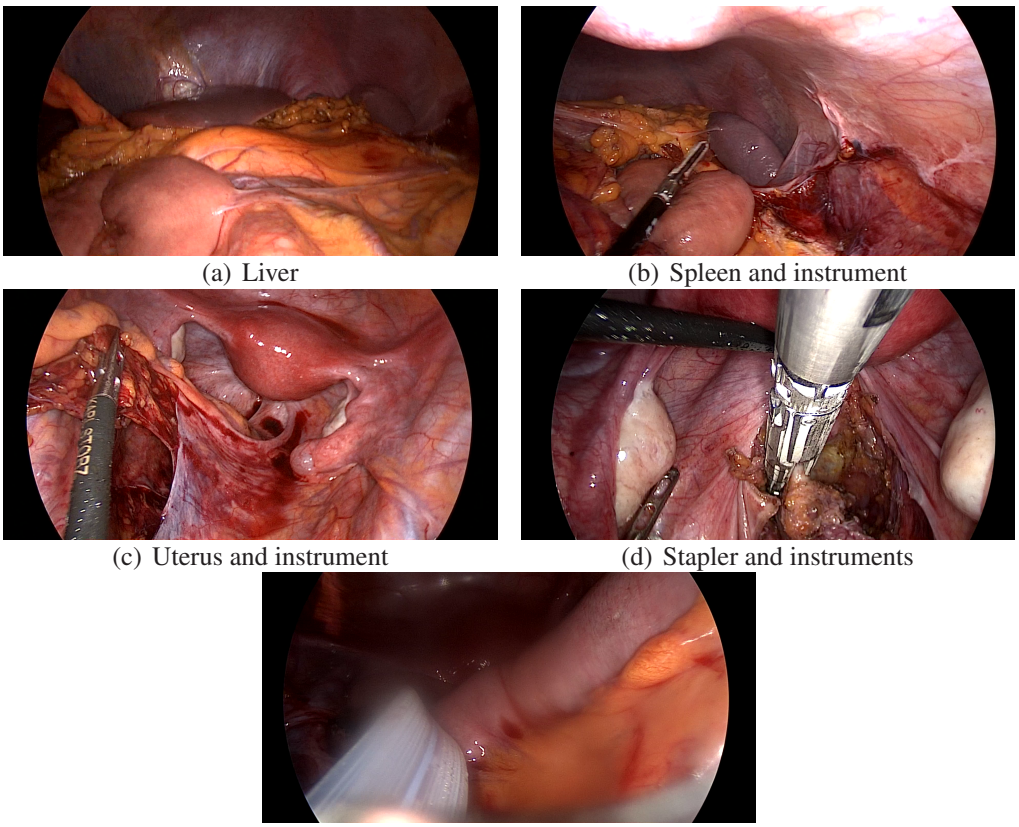
The kernel over all trees is computed as:

$$K(H_{I_1}, H_{I_2}) = \frac{1}{T} \sum_{t=1}^{T} K_t(H_{I_1}, H_{I_2}) \tag{4.41}$$

We train one SVM for each class in our training set.

4.5.2 Evaluation of the image content classification

We evaluated the method for weakly supervised image classification on a data set collected from videos of four different colorectal laparoscopies. Six different classes, spleen, liver, uterus, instrument, silicone drain and stapler were annotated. Examples of each class can be found in figure 4.29. For each class, 50 images were selected from each video and downsampled to 160×90 pixels.



(a) Liver (b) Spleen and instrument

(c) Uterus and instrument (d) Stapler and instruments

(e) Silicone drain

Figure 4.29: Classes used for training the weakly supervised texton forest.

We then performed a leave-one-surgery-out evaluation with the following parameters: 8 trees, a maximum depth of 8 and a combination of all color spaces. The results can be found in table 4.10.

	Precision	Recall	Accuracy
Liver	88.0%	66.0%	78.5%
Spleen	81.7%	76.0%	79.5%
Uterus	84.0%	68.0%	77.5%
Instruments	90.0%	85.7%	87.5%
Silicone drain	78.9%	74.7%	77.9%
Stapler	87.5%	77.0%	83.0%
Average	85.4%	74.8%	81.0%

Table 4.10: The average performance of the proposed method for labeling surgical images.

Run-time

The evaluation was performed on a workstation PC with an Intel Core i7-5820K, 32GB of RAM. The method requires on average 0.4s to classify an image.

4.5.3 Discussion

In this section, a method for labeling surgical images according to their contents was introduced. In contrast to previously introduced methods, only sparse labels were required for training the classifier. Overall, the method performed best on detecting instruments and staplers, the performance for the organs wasn't much lower though. No large difference in recognizing the different types of organs was noted.

Currently, the method requires 0.4s to classify an image, this is still not fast enough for real-time performance. The run-time could be significantly improved by porting the feature extraction onto the GPU.

5 Quantitative Surgical Image Analysis

This chapter introduces methods for performing quantitative image analysis in a laparoscopic setting. In a surgical environment, quantitative image analysis provides numerical information on the size and distances of structures, such as instruments and organs, in the scene. First, an overview of the methods used for 3D reconstruction are given in section 5.1. By combining semantic information with 3D reconstructions, measurements along organ surfaces become possible. A method for 3D organ measurements in a laparoscopic setting is described in section 5.2.

5.1 3D Reconstruction

As previously outlined in section 3.2.1, endoscopes are able to extract depth information from a surgical scene [MHMB$^+$13]. Stereo endoscopes currently provide the reconstructions with the highest accuracy and are already available in the surgical setting. This section gives an overview over stereo camera systems and what techniques can be used to extract depth information from stereo endoscopes.

5.1.1 Stereo camera system & calibration

A stereo camera system generally consists out of two standard cameras that view the same scene from slightly varying perspectives. When viewing a scene with a single camera, a perspective transformation maps 3D points onto a 2D image plane. This transformation can be described mathematically by the pinhole camera model.

Pinhole camera model

The model presumes that every point of the original scene is projected onto the plane along a straight ray that runs through an infinitesimally small point, the projection center (Figure 5.1(a)). As the projection center is generally in front of the image plane, the image projected onto the plane is rotated by 180°. To simplify the model, the position of the projection center is transferred behind the image plane, reversing the rotation of the projected image (figure 5.1(b))[AGD08]. The line from the projection center that is perpendicular to the image plane is called the principle axis and the point where it intersects with the image plane is called principle point [HZ03]. With the

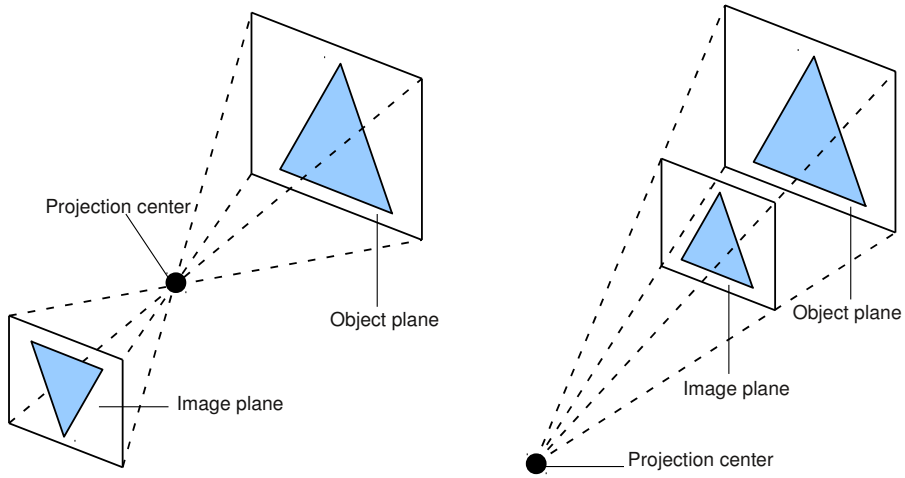


Figure 5.1: The pinhole camera model (a), Pinhole camera with image plane in positive position (b) [Bod12]

image in front of the projection center P, the relation between world and image plane points can be expressed using the following equation:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{Z} \begin{bmatrix} X \\ Y \end{bmatrix} \tag{5.1}$$

Here x,y represent coordinates in the image plane, while X,Y,Z are the coordinates of the corresponding point in the world coordinate system, which has the projection center **P** as origin (figure 5.2). The projection center P lies at a distance f from the image. f is the focal length of the camera. If the image plane is behind the projection center, the sign of the image coordinates changes [AGD08].

Extended pinhole model

The standard pinhole model does not account for non-square pixels and that the image and the camera coordinates usually do not share the same unit of measure, typically images will be measured in pixels and in the camera coordinate system in millimeters. Therefore two new parameters m_x and m_y are introduced, they represent the number of pixels per unit of measure in x and in y directions [HZ03]. Furthermore, it was also assumed that the origin of the image plane lies on the principle point, but usually the origin of an images is in the upper left corner. To account for this, the coordinates of the principle point $C = (c_x, c_y)$ are added onto the image point. Using these new parameters, equation 5.1 becomes:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{Z} \begin{bmatrix} m_x X \\ m_y Y \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} f_x X \\ f_y Y \end{bmatrix} + \begin{bmatrix} c_x \\ c_y \end{bmatrix}, f_x = m_x f, f_y = m_y f \tag{5.2}$$

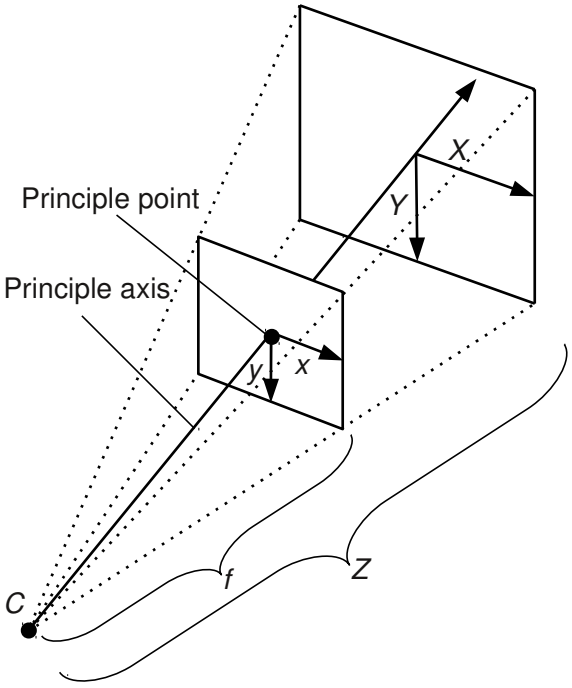


Figure 5.2: Coordinate systems in the pinhole model. Coordinates in the scene, represented by X, Y, Z, are projected through the projection center C onto the coordinates x, y in the image plane, which lies at a distance of f from C [Bod12]

Or, using homogeneous coordinate, the relationship can be expressed via matrix multiplication:

$$\begin{bmatrix} x' \\ y' \\ Z \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = K \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{5.3}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x'/Z \\ y'/Z \end{bmatrix} \tag{5.4}$$

K is the so-called camera coefficient matrix, containing four values called intrinsic camera parameters, since they are independent from the world coordinate system[AGD08].

As the camera coordinate system generally differs from the world coordinate system, the model is extended with a rigid transformation, consisting of a rotation matrix $R \in \mathbb{R}^{3\times3}$ and a translation vector $t \in \mathbb{R}^3$. These are the so-called extrinsic camera parameters. This makes it possible to express the entire transformation from world coordinate system into that of the image with one matrix P, called the projection matrix.

$$P = K \left[\begin{array}{c|c} R & t \end{array} \right] \tag{5.5}$$

Lens distortion

The pinhole camera model does not account for artifacts introduced by the curvature of the lens or by imperfections introduced during the manufacturing process of the camera. The function that describes the effects of lens curvature δ_r, so called radial distortions, can be approximate by using the even coefficients of its Taylor series, as the function is assumed to be symmetrical [BK08]:

$$\delta_r(r) = 1 + k_1 r^2 + k_2 r^4, r = \sqrt{x'^2 + y'^2}, \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \frac{x - c_x}{f_x} \\ \frac{y - c_y}{f_y} \end{bmatrix} \tag{5.6}$$

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = \begin{bmatrix} f_x(x' + \delta_r(r)) + c_x \\ f_y(y' + \delta_r(r)) + c_y \end{bmatrix} \tag{5.7}$$

where k_i are the coefficients of the Taylor series, x and y are the image coordinates before distortion and x_d and y_d after accounting for distortion. Normally the first 2 coefficients suffice to satisfactorily model the radial distortion of a lens. The tangential distortion function, which occurs when lens and camera chip are not perfectly parallel to one another, can be modeled using two coefficients d_1 and d_2.

$$\delta_t \left(\begin{bmatrix} x_d \\ y_d \end{bmatrix} \right) = \begin{bmatrix} 2d_1 x'y' + d_2(r^2 + 2x'^2) \\ d_1(r^2 + 2y'^2) + 2d_2 x'y' \end{bmatrix} \tag{5.8}$$

For further details, please refer to [Bro66].

Calibration

The intrinsic and extrinsic camera parameters as well as the distortion parameters can be computed by the method outlined in [Zha00]. Using images of a chessboard pattern with known geometry, a homography that describes the transformation from object to image plane can be computed for each image. Multiple homographies are then used to solve for the intrinsic and extrinsic camera parameters analytically. Estimating the distortion parameters is achieved iteratively, using Levenberg-Marquardt optimization [Zha00].

A stereo camera system is a setup consisting of two cameras that are observing the same scene. With a stereo camera system, the original three-dimensional point of two corresponding image points in two images can be calculated, if the calibration of the system, consisting of the two internal matrices K_1 and K_2 and the extrinsic parameters R_1,t_1 and R_2,t_2, is known.

The cameras in a stereo endoscope are approximately orientated in the same direction, meaning their image planes are almost parallel, and the line between the two optical centers only differs slightly from their x axises. With the y axis defined as up and looking in the direction of z, we will from now on refer to the camera lying to the left

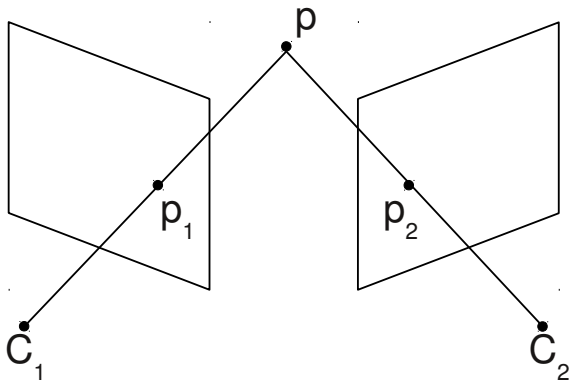


Figure 5.3: The original 3D point is found using stereo triangulation [Bod12]

of the other camera on this line as the left camera. The other camera will be referred to as the right camera. From now on, in order to simplify the model, the world coordinate system will be the same as the coordinate system of the left camera, meaning that only one pair of rotation matrix R and translation vector t describes the transformation from the coordinate system of the right camera into that of the left camera, is required. From now on R and t will be referred to as the extrinsic parameters of a stereo camera system.

Stereo triangulation

When two corresponding points p_1 and p_2 in the left and right image are known, their original 3D point **p** can be found by drawing a line for each camera through its projection center and the corresponding point in the image. The point in which the two lines intersect is the original 3D point (figure 5.3).

To construct these lines, first a vector from the projection center to the image point has to be calculated for each camera. This is done with the inverse of the camera matrix:

$$v_1 = K_1^{-1} \begin{bmatrix} p_1 \\ 1 \end{bmatrix} \tag{5.9}$$

$$v_2 = K_2^{-1} \begin{bmatrix} p_2 \\ 1 \end{bmatrix} \tag{5.10}$$

Since the projection center of the first camera is also the origin of the world coordinate system, v_2 needs to be transfered into the world coordinate system. The two lines are defined:

$$l_1 : x - r \cdot v_1 = 0 \tag{5.11}$$

$$l_2 : x + R^T t - s \cdot (v_2 + R^T t) = 0 \tag{5.12}$$

To find the point of intersection, the two lines have to be equated:

$$r \cdot v_1 - s \cdot (v_2 + R^T t) = -R^T t \tag{5.13}$$

Due to inaccuracies in measurement and discretization, the two lines usually don't intersect, but instead are skewed [AGD08]. To find the point with the shortest distance to both lines, equation 5.13 can be expressed as an over-determined system of linear equations and then solved using the method of least squares:

$$\begin{bmatrix} v_1 & -v_2 + R^T t \end{bmatrix} \begin{bmatrix} r \\ s \end{bmatrix} = A \begin{bmatrix} r \\ s \end{bmatrix} = -R^T t \tag{5.14}$$

$$\begin{bmatrix} r \\ s \end{bmatrix} = -\left(A^T A^{-1}\right) A^T R^T t \tag{5.15}$$

The original 3D point p can then be calculated by plugging r and s into the two lines l_1 and l_2 and averaging the two resulting points[AGD08]:

$$p = 0.5 \cdot \left(r \cdot v_1 - R^T t + s \cdot \left(v_2 + R^T t\right)\right) \tag{5.16}$$

To accurately calculate the original 3D point, the correctness of the calibration is of the utmost importance. Small errors in parameters, e.g. the focal error, can lead to larger errors when calculating the 3D point.

Epipolar geometry

Before a 3D point can be calculated, a pair of corresponding points has to be found. This is not a trivial matter, especially in real-time. One way to reduce the search space is to make use of properties of epipolar geometry, which can reduce the correspondence search space from 2D to 1D. Assume p is a point in an image belonging to a stereo image pair. If the calibration of the stereo camera system is known, the ray of all three-dimensional points that would be projected onto p, can be calculated. This ray can then be backprojected as a line onto the other image in the stereo pair (figure 5.4). If the image contains the corresponding point to p, and if the calibration is correct, the point must lie on this line. The projection of the projection center of one camera into the image of the other camera is called an epipole, giving this geometry its name [HZ03]. This property is expressed mathematically via the fundamental matrix F:

$$F = K_2^{-T} E K_1^{-1} \tag{5.17}$$

where K_1 and K_2 are the calibration matrices of the two cameras. E is the essential matrix:

$$E = R\,[t]_\times \tag{5.18}$$

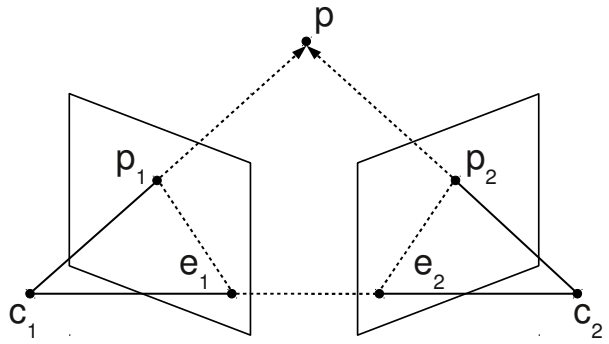


Figure 5.4: An example of how the search for a corresponding point can be reduced to one line using epipolar geometry. e_1 and e_2 are the epipoles [Bod12]

Using homogeneous representation, epipolar lines can be calculated with the fundamental matrix:

$$l_2 = F\tilde{p}_1 \text{ and } l_1 = F^T\tilde{p}_2 \tag{5.19}$$

The left and right null-space of the fundamental matrix are the epipoles, meaning that every epipolar line intersects with one of the epipoles:

$$Fe_1 = 0 \quad \text{and} \quad F^T e_2 = 0 \quad \text{with } e_1 = K_1 R^T t \quad \text{and} \quad e_2 = K_2 t \tag{5.20}$$

Rectification

In the case were the two image planes are coplanar, finding point correspondences is simplified, since every line that runs parallel to the x-axis of the image is an epipolar line, whose conjugate in the other image lies on the same y-coordinate. The question now arises, how this knowledge can be applied to the average stereo camera system, where the image planes aren't coplanar. In [FTV00], the authors present a method for rectifying calibrated stereo image pairs. The idea is to introduce two new projection matrices, one for each camera, that rotate the image planes around their optical centers, making them coplanar. In rectified images, two matching points p_l and p_r should lie on the same y coordinate, i.e. $y_l = y_r$. The difference in the x coordinate, $d = p_l - p_r$ is called disparity.

5.1.2 Hybrid recursive matching & 3D reconstruction

Feature detectors and descriptors, such as SURF [BTVG06] or SIFT [Low99], would make it possible to match features in a given stereo pair. Such a sparse matching would not be sufficient for reconstructing organ surfaces though.

In [AKS04], the authors propose the hybrid recursive matching algorithm (HRM), which they use to perform dense correspondence matching in real-time for video conferencing. Dense correspondence matching implies that the method attempts to find a correspondence for every pixel in both images. As the HRM uses rectified images as input, the dense correspondence matches are given as a disparity map, relative to

the left image. The HRM was extended to deal with problems specific to stereo endoscopic imaging, such as the small baseline between cameras, specularities and smoothness constraints in [RBS+12] and [Röh13]. The HRM is a mix of a pixel-based and a global approach. On one side, for every pixel in the left image, a corresponding pixel in the right image is selected out of a pixel of potential candidates. For this, a similarity measure is used. On the other side, already computed disparities are propagated recursively through the image to generate correspondence candidates in a meander scan fashion. This establishes a local constraint on the disparities. In addition to the spatial constraints, the HRM uses temporal constraints by using disparities from the previous time step as correspondence candidates. These constraints reduces the amount of mismatches in homogeneous regions and enforces smoothness [Röh13] [RBS+12]. A two-stage process is used by the HRM to select a disparity value for a given pixel from four candidates in the other image:

Block recursion

In the block recursion step, three candidates for a disparity value are generated from the temporal and the spatial neighborhoods. For a given pixel, the temporal disparity value d_t is taken from the previous time step. The two spatial disparity values d_v and d_h are taken recursively from the vertical and horizontal neighboring pixel disparities respectively. The resulting disparity d_b for the current pixel is then computed via similarity measure $S(d)$:

$$d_b = \underset{d \in \{d_v, d_h, d_t\}}{\operatorname{argmin}} \; S(d) \tag{5.21}$$

$S(d)$ calculates the Hamming distance between the pixel neighborhoods [Röh13].

Pixel recursion

The block recursion enforces smoothness over in global and spatial terms. Block recursion cannot compute completely new disparity values, which can cause problems in regions were discontinuities in the image occur, e.g. at the boundary of an organ or the edge of the image. If the similarity measure of the block recursion step is larger than a set threshold, it is assumed that the the pixel recursion is p In the pixel recursion step, a simplified, recursive version of the optical flow is used to compute a further disparity candidate d_p. If $S(d_p) < S(d_b)$, d_p is used as disparity value for the current pixel [Röh13].

The resulting disparity map is then smoothed using a bilateral filter [TM98][Röh13]. To speed up computation, the approach was ported onto the GPU in [RBS+12].

5.2 Live Organ Measurement

While minimally invasive operations offer a great number of benefits for the patient, they also pose challenges for the surgeon. One of these drawbacks is the loss of depth perception, without which surgeons, especially less experienced ones, have difficulties

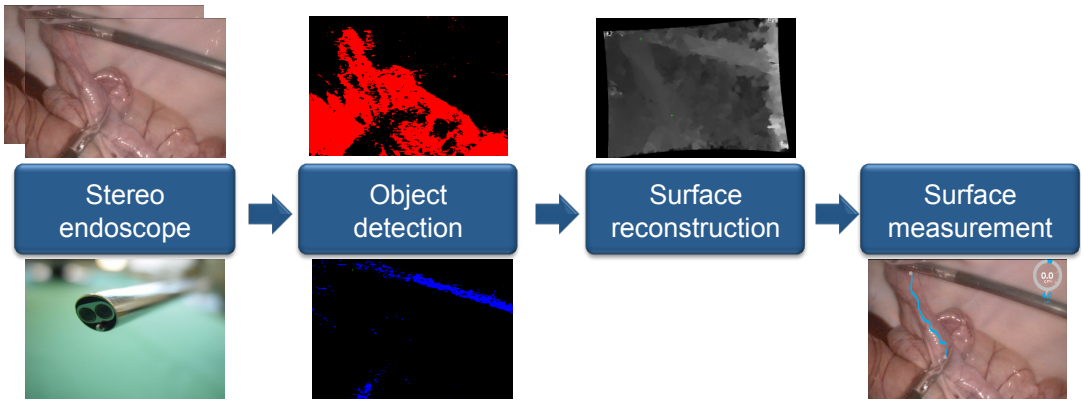


Figure 5.5: Organ measurement system overview. After [BWM+16]

estimating distances. Surgeries such as a gastric bypass or a hernia repair require an accurate estimation of distances, such as the length of a segment of bowel or the size of a hernia. Methods for estimating or measuring distances in laparoscopic surgery exist (see section 2.3.1), these methods often are difficult to integrate into the surgical process or are inaccurate. Automatically and objectively estimating distances during laparoscopic surgery would therefore prove beneficial to the surgeon. Furthermore, distances, for example between organs and instruments, can also provide useful information for context-aware surgery [KWB+11].

In this section, we describe a markerless method for objectively computing distances along organ surfaces directly in a laparoscopic image frame-based on stereo endoscopy. We outline the method in this section and describe how it can be applied to the problem of laparoscopic bowel measurement (see section 2.3.1, though the method is flexible enough to be applied to other measurement tasks involving an endoscope. The work presented here was published in [BWM+16].

5.2.1 Methods

Our approach is divided into four main steps (figure 5.5). Before stereo images can be reconstructed, the stereo endoscope first has to be calibrated to make depth extraction possible. On-the-fly segmentation is then used to locate the organ of interest and the surgical instruments in captured images. We then reconstruct the surface of the organ and the positions of the instrument tips into 3D. Using the positions of the instruments and the reconstruction, we then measure the distance between two instruments along the surface of an organ. In figure 5.6 example outputs from each of these steps can be found.

Object detection

Once the calibrated stereo endoscope has entered the patient, we need to locate the objects that are relevant for taking measurement in the endoscopic video frames (figure 5.6(a)). For locating the laparoscopic instruments, we make use of the methods

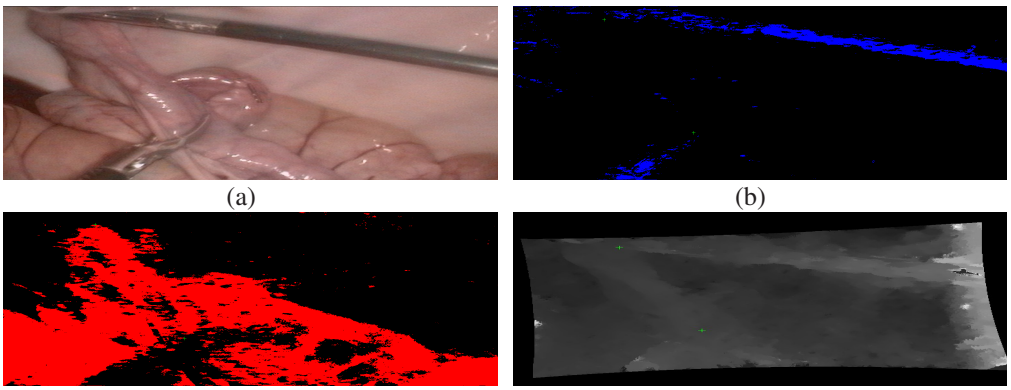


Figure 5.6: Overview of the results provided by the different system components: (a) original image, (b) output instrument segmentation, (c) output bowel segmentation, and (d) output 3D reconstruction [BWM+16].

previously outlined in section 4.1 (figure 5.6(b)). As we are focusing on bowel measurement, only the two instruments grasping the bowel are of interest, we therefore discard every tool region of interest found in the image, except the two largest. The tool tips T_1 and T_2 are located as described in section 4.1.

To extract the organ of interest, in this case the bowel, we also apply extract features from a given frame as outlined in section 4.1.1, though we use different features than for instrument detection. Empirically we found that a combination of G from RGB, a and b from CIElab and o_2 from the opponent space provided the best segmentation results. The resulting features were processed with a random forest as presented in section 4.1.1, though here we selected a maximum depth of 10 and 100 trees. The binary image computed with the random forest was post-processed using a morphological closing operation [AGD08]. Using the method presented in [SA85], contours were located in the binary image. Contours close to one another were combined. Taking the previously located instrument tips, allows us to locate the contours closest to the instruments, discarding all others (figure 5.6(c)).

Isolating only a particular bowel segment reduces the time required for 3D reconstruction and taking the measurement. Please note that the instruments and organ were only detected in the left image of the stereo image pair provided by the stereo endoscope.

Surface reconstruction

Once the relevant structures have been located in the left image, a 3D reconstruction is required. For a 3D reconstruction it is necessary to locate matching pixel correspondences between the stereo image pair. Though even with rectified images, matching correspondences is not a trivial matter.

We therefore apply a dense matching method, the hybrid recursive matching as outlined in section 5.1.2, to our stereo image pair. As only reconstructing the surface of the organ of interest and the positions of the tips of the laparoscopic instruments are

relevant in this scenario, the correspondence search can be sped up further by limiting the HRM only to the previously segmented areas of the stereo image pair.

Once computed, the disparity map (figure 5.6(d)), in combination with the calibration parameters, is used for stereo triangulation (see section 5.1.1), resulting in a 3D point cloud. This point cloud was then post-processed as proposed in [Röh13]. First a least-squares smoothing was applied to the point cloud, removing sharp edges due to matching artifacts and noise. Next, a triangular mesh is computed from this point cloud. As the correspondence between 3D and 2D points is known, we assume that two 3D points are neighbors if their 2D correspondences are also neighbors. If the euclidean distance between the two 3D points lies below a certain threshold, we integrate them into a triangle [Röh13]. This results in a triangular mesh representation of the surface of the bowel. Furthermore, we also extract the 3D positions of the instruments from the disparity via the same process.

Surface measurement

Given the 3D surface mesh and the 3D tip positions of the instruments as input, multiple ways of actually determining distances are possible (figure 5.7). For more details, please see [Kor13].

Direct path

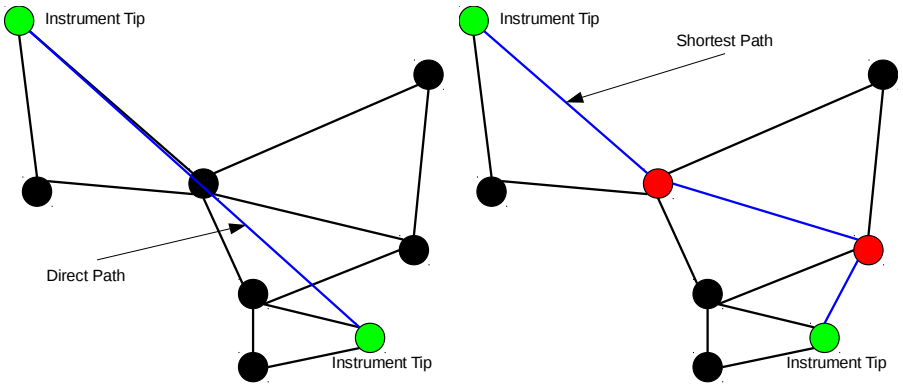
The simplest way of estimating the length of a segment of bowel would be to just use the positions of the instrument tips T_1 and T_2 (figure 5.7(a)). The euclidean distance of the two 3D points gives a good estimate of the length, assuming no large curvatures in the surface of the bowel.

$$d_{\text{direct}} = \|T_1 - T_2\| \tag{5.22}$$

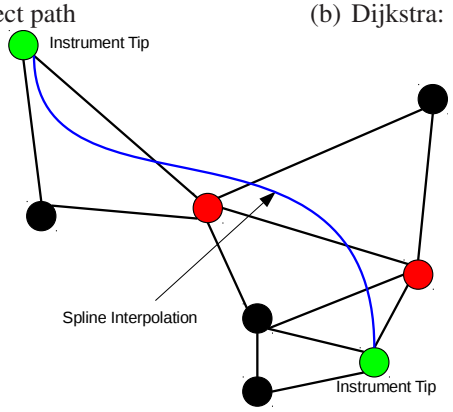
This method has the added bonus that no segmentation or reconstruction of the actual organ is required. The bowels, however, are not a rigid structure, meaning they can be curved, making an estimate formed strictly from the position of the instruments inaccurate.

Dijkstra: shortest path

To deal with the curvature of the bowel, its surface 3D reconstruction has to be taken into consideration. For this, the surface mesh previously described can be used. This mesh can be viewed as a graph, which makes the problem of finding the shortest distance between T_1 and T_2 along the surface of the bowel analog to the shortest path problem in graph theory. A graph is an ordered pair $G = (V,E)$ consisting of a set of vertices V and a set of edges E with $e \in E, e = (v_1, v2), v_1, v_2 \in V$. To transform the mesh, we add a vertex v_i to V for every point p_i in the mesh. If points p_i and p_j are connected in a triangle, we add an edge e_{ij} to E. Furthermore, we define a weight function $w(e_{ij}) = \|p_i - p_j\|$. A few methods for locating the shortest path are known in literature, e.g. [SSK$^+$05][LCDF10], we rely on the Dijkstra algorithm [Dij59], due



(a) Direct path (b) Dijkstra: Shortest path



(c) Dijkstra + spline

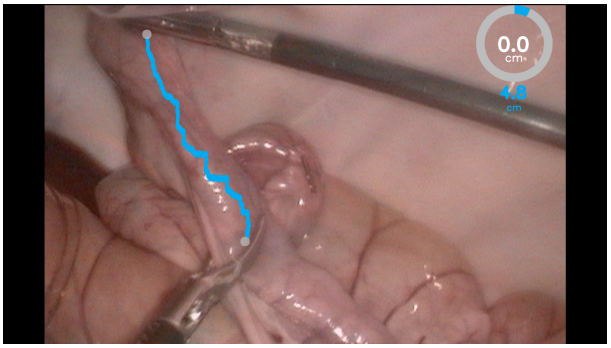
Figure 5.7: Illustrated examples of the three proposed measurement methods: (a) direct path between the two instrument tips without regards for the organ surface, (b) shortest path along the organ surface between the instrument tips, and (c) spline interpolated version of the shortest path. Red vertices indicate vertices along the shortest path. After[BWM+16].

to speed ($O(|V|+|E|)$) and readily available implementations. An example path can be seen in figure 5.7(b). Let p_i be the shortest path with N ordered points along the surface mesh, with $p_1 = T_1$ and $p_n = T_2$. We then estimate the distance along the surface by measuring along the edges:

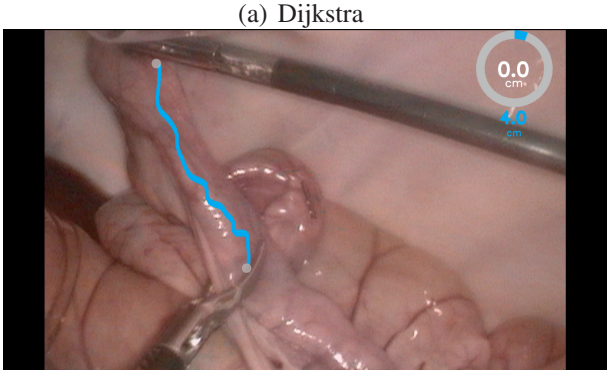
$$d_{\text{dijkstra}} = \sum_{i=1}^{N-1} \|p_i - p_{i+1}\| \tag{5.23}$$

Dijkstra & spline interpolation

The shortest path on the surface of the organ zig-zags across the organ surface due to it being confined to the edges of the graph. An illustrated example of this can be seen in figure 5.7(b) and in figure 5.8(a) a real example. The accumulation of these detours can cause a taken measurement to overestimate the actual distance. In order to find a smoother representation of the shortest path, a spline interpolation [Kno12] with a cardinal cubic B-spline was used. A spline is a piecewise-defined function $C : [0,1] \to \mathbb{R}^N$, consisting of multiple basis functions. In our case, $N = 3$ and we use cardinal cubic basis functions. By using every 10^{th} point on the triangular mesh visited by the shortest path, we are able to fit a spline C that closely follows the shortest path



(a) Dijkstra



(b) Dijkstra + spline

Figure 5.8: (a) the path computed with **Dijkstra** and (b) the path computed with **Spline**. The second path is smoother and shorter than the first, which is longer due to multiple detours along the edges of the surface mesh [BWM+16].

along the mesh while avoiding detours using the method presented by de Boor [De 78]. For an illustrated example see figure 5.7(c) and figure 5.8(b) for a real world example. To compute the distance along the spline, we sample it at 0.02 intervals:

$$d_{\text{spline}} = \sum_{i=1}^{50} \|C(0.02 \cdot (i-1)) - C(0.02 \cdot i)\| \tag{5.24}$$

Into the OR

The presented workflow shows how, starting from a stereo image pair, measurements along an organ surface can be acquired. The system is a one-shot, meaning a signal to begin a measurement is required. During development and offline evaluation, the measurement signal was given via a simple mouse click. In the OR, the surgeon has to decide when to measure. As the hands of the surgeon are sterile and generally not free, a USB foot switch for triggering the measurement was introduced (figure 5.10(a)). Once a distance has been successfully calculated, we present it to the surgeon via augmented reality, as can be seen in figure 5.9. Here we would like to thank Simon Mayer of the Pforzheim School of Design, who kindly provided the design for the augmented reality screen we used.

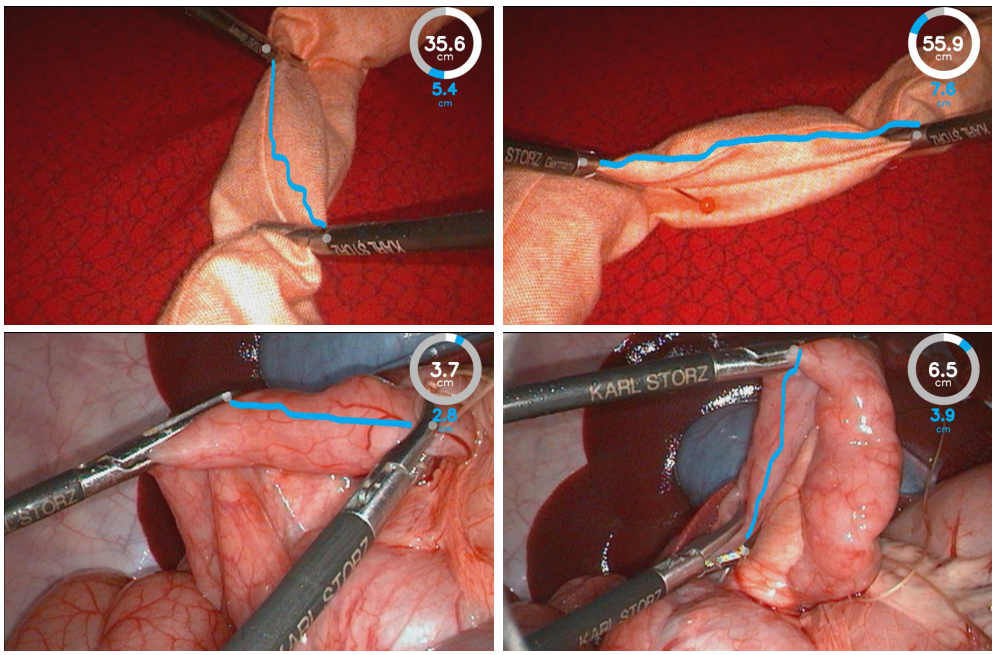


Figure 5.9: Examples on how measurements are presented to the surgeon via augmented reality. Top: Phantom intestine, bottom: Porcine intestine

The augmented reality display shows the surgeon the path along along the surface that was measured, the measured distance, the accumulated distances over multiple measurement and the measurement progress, e.g. 35 cm bowel of 70 cm wanted bowel distance have been measured. This view was made available to the surgeon with a second monitor (figure 5.10(b)), to assure that any failure in the measurement software did not hinder the surgeon's view. The computer system used for the live measurement system was a standard Workstation-PC (Intel Core i7-5820K CPU, GeForce GTX 970 GPU and 32GB RAM).

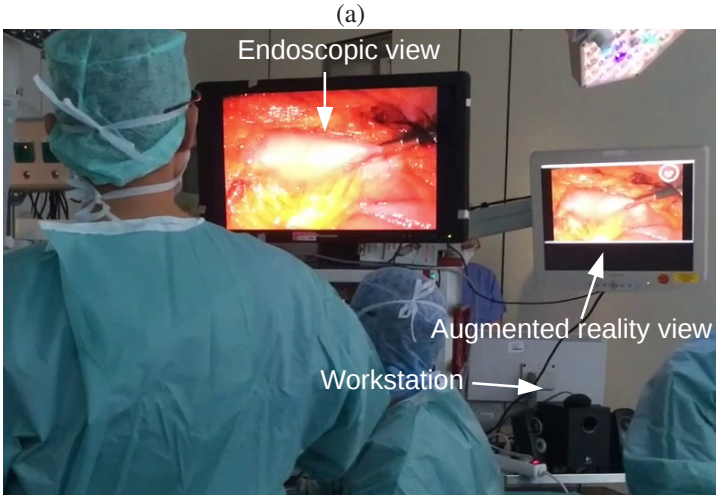
5.2.2 Evaluation

The presented measurement system was evaluated in two stages. First the system was evaluated offline on stereo image pairs collected during phantom and porcine trials. The focus of the offline evaluation was to compare the different measurement methods (**Direct**, **Dijkstra** and **Spline**) and to ascertain the accuracy of single measurements. The online evaluation, performed at the University Hospital of Heidelberg by Wagner et. al [WMB+17b][WMB+17a], focuses on the accuracy of the overall measurement and the comparison to other measurements and is subject of the medical doctorate thesis of Benjamin Mayer [May17].

Offline evaluation

To compare the different measurement modes that we introduced and to determine the accuracy of the system, we collected data from phantom bowels and porcine intestine in the abdominal cavity. For each trial, a 3D TIPCAM 1 HD-stereo endoscope from

(a)



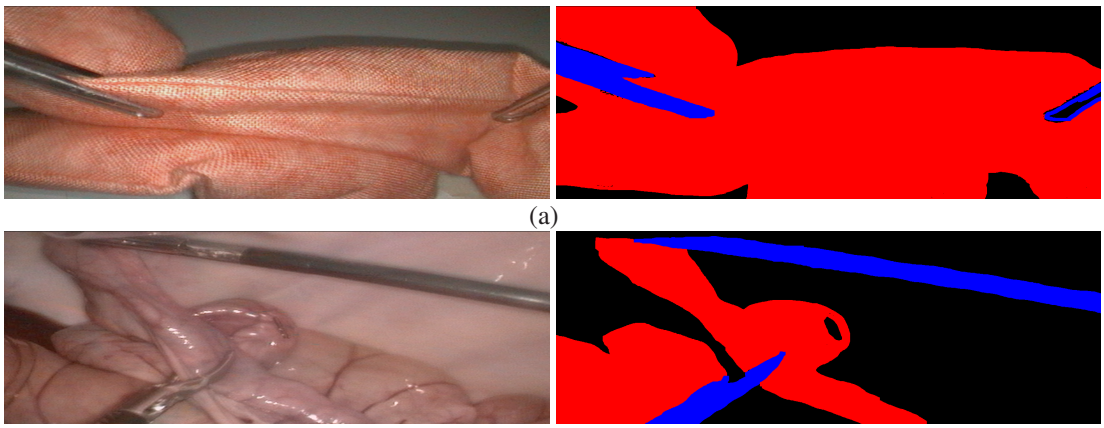
(b)

Figure 5.10: (a) foot switch used to trigger live measurements, (b) setup in the operating room. Images courtesy of Benjamin Mayer and Martin Wagner (University Hospital of Heidelberg)

Karl Storz GmbH Tuttlingen, Germany with chip-on-tip cameras and a 0° optic was used to capture the scene. To handle the bowels, two laparoscopic graspers from Karl Storz GmbH, were used.

Phantom data

A phantom intestine made out of cloth was placed inside a laparoscopic box trainer. Fourteen medical experts were instructed to perform multiple laparoscopic bowel measurements. First a visual guideline for estimation, a tape measure showing 5 cm or 10 cm was placed briefly in front of the camera before each expert started the first measurement. They then had to iteratively pass the cloth intestine in front of the endoscope using laparoscopic instruments. After each iteration, the positions of the instruments were marked with pins by the instructor. We also instructed the experts to estimate a bowel length of either 5 cm or 10 cm during each iteration. These estimates were used as a baseline to evaluate if our proposed method could perform accurately enough to provide an actual benefit to a surgeon. As a reference, the distances between the pins were measured with a standard tape measure after each expert finished a measurement. Pins were used due to the confined space inside the boxer trainer, which would have made measurements more prone to error. The fourteen medical experts were asked to



(a)

(b)

Figure 5.11: Examples of training images used to train the instrument and the bowel detector: (a) phantom data, (b) porcine data [BWM+16]

perform three complete measurements of 70 cm. In total, 504 iterations were available for evaluation.

Furthermore, we separately recorded 45 endoscopic images during measurement. These images were then labeled pixel-wise for instruments and bowel (figure 5.11(a)). The annotated images were used to train the instrument and the bowel detector.

Porcine data

We collected similar data in a porcine trial. Here, four medical experts were asked to perform one laparoscopic bowel measurement, each one consisting of six iterations, resulting in a total of 24 iterations. As reference, the positions of the instruments were marked using an electrocauter. The distance was measured after laparotomy (opening of the abdominal cavity) with a standard tape measure.

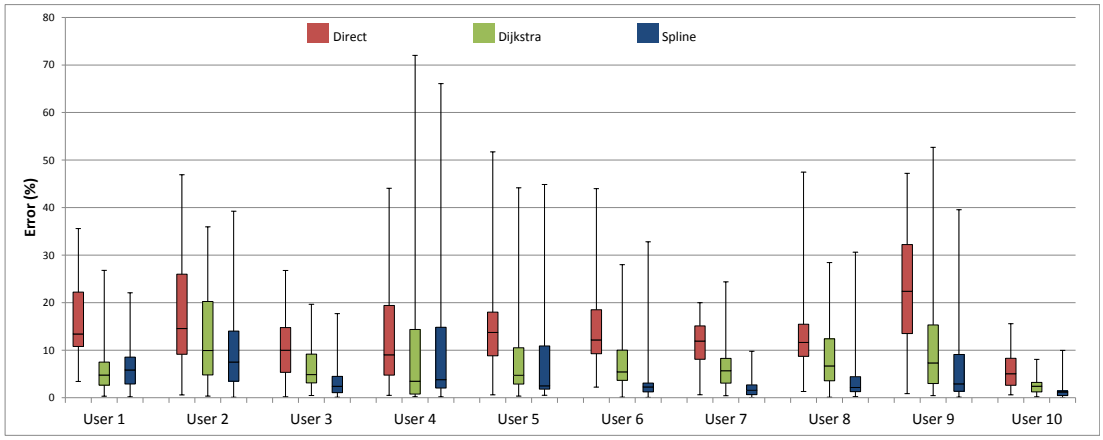
Similarly to the phantom data, we separately recorded 40 endoscopic images during measurement. These images were also labeled pixel-wise for instruments and bowel (figure 5.11(b)). The annotated images were used to train the instrument and the bowel detector.

Calibration

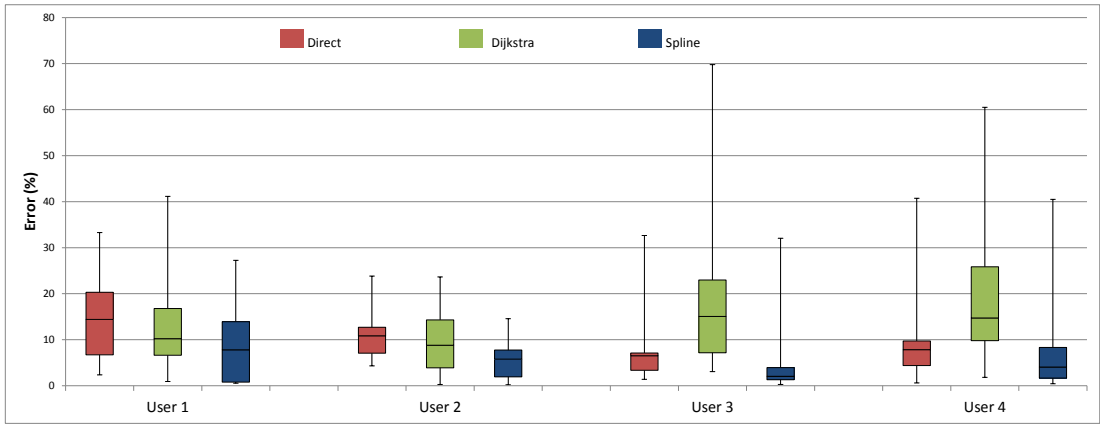
Before evaluating the different measurement methods, the stereo endoscope was calibrated using the method outline in section 5.1.1. Fifty-five different views of a chessboard pattern were recorded for calibration. The resulting calibration had an average 3D reconstruction error of 0.5mm and a rectification error of 0.2px.

Measurement

With the calibration, we compared the results of the three proposed measurement methods (**Direct**, **Dijkstra** and **Spline**) to the collected references. To determine to what



(a)



(b)

Figure 5.12: Relative errors for **manual** measurement on phantom bowel. In (a) the medical experts were told to estimate 5 cm at each iteration and 10 cm in (b) [BWM+16].

extent the automatic instrument detection influences the measurement results, we performed two experiments for each dataset. In one experiment, the positions of the instrument tips were manually provided (**manual**). In the other experiment, the automatic instrument detection was used (**automatic**). During each experiment, all three measurement methods were computed.

Results phantom data

The proposed methods were able to successfully compute distances on 397 out of 504 iterations (79%) of the phantom intestine. The most common reason for failure was that two instruments were not always visible in one or both camera images, making a measurement impossible (figure 5.16(a)).

The results of the **manual** approach can be found in table 5.1 and figure 5.12, while the results of the **automatic** approach can be found in table 5.2 and figure 5.13. We divided the data into two subsets: one contained the experts that were asked to estimate 5 cm and the other the ones that were asked to estimate 10 cm.

(a)

	Relative mean error (%)			Absolute mean error (mm)		
	Direct	**Dijkstra**	**Spline**	**Direct**	**Dijkstra**	**Spline**
User 1	16.1±8.5	6.8±6.2	6.4±5.0	8.6±4.8	3.7±3.5	3.6±3.3
User 2	17.6±10.7	12.4±10.0	8.5±8.3	9.0±6.3	5.9±4.9	4.2±4.7
User 3	10.3±6.1	6.4±5.4	3.7±3.6	4.9±3.1	3.0±2.8	1.7±1.7
User 4	13.2±12.1	11.7±18.7	10.8±15.5	6.4±6.4	4.9±7.3	4.8±6.3
User 5	14.6±10.4	8.5±9.7	7.7±10.4	7.9±7.2	4.8±6.5	4.5±6.9
User 6	15.7±11.0	8.0±7.2	6.3±9.5	8.5±6.6	4.4±4.2	3.6±5.6
User 7	12.0±5.7	7.1±5.9	2.6±3.4	6.5±3.8	3.8±3.3	1.4±2.2
User 8	13.3±8.4	8.6±6.9	4.1±6.0	6.3±4.9	4.0±3.6	2.1±3.5
User 9	23.3±12.3	11.9±12.2	7.8±10.6	11.5±7.6	5.7±5.9	4.0±6.2
User 10	5.4±3.7	2.5±1.9	1.4±2.1	5.4±3.7	2.5±1.9	1.4±2.1
All	13.9±10.0	8.2±9.2	5.7±8.3	7.4±5.8	4.2±4.5	3.0±4.6

(b)

	Relative mean error (%)			Absolute mean error (mm)		
	Direct	**Dijkstra**	**Spline**	**Direct**	**Dijkstra**	**Spline**
User 1	14.7 ± 9.8	15.5 ± 16.5	10.2 ± 10.7	16.3 ± 10.7	16.6 ± 17	11.4 ± 11.8
User 2	11.1 ± 6.3	10.1 ± 8.1	5.8 ± 4.3	12.9 ± 9.4	11.2 ± 8.9	6.7 ± 5.7
User 3	6 ± 3.1	16.1 ± 11.3	2.9 ± 2.2	5.9 ± 3.2	15.4 ± 11.5	2.8 ± 2.2
User 4	8.6 ± 7.1	22 ± 22.9	6.5 ± 7.4	8.6 ± 9.6	21.9 ± 31.2	6.5 ± 9.4
All	9.5 ± 6.9	15.9 ± 16	5.8 ± 6.4	10.2 ± 9.1	16.2 ± 19.6	6.2 ± 7.7

Table 5.1: The results for **manual** measurement on phantom bowel. The tables show the absolute mean error and standard deviation (mm) and the relative mean error and standard deviation. Table (a) shows the results for the 5 cm estimate and (b) for the 10 cm estimates [BWM+16].

As a baseline for comparison, we also evaluated the estimates provided by the medical experts. In figure table 5.3 the accuracies of the estimates provided by the experts are listed. A comparison between the accuracies of the expert estimates and the **manual** and **automatic** instrument detection modes computed with **spline** can be found in figure 5.14

Results porcine data

On the porcine data, the **automatic** method successfully retrieved distances from 13 out of 24 (54.2%) iterations. similar to the phantom dataset, the failures of the method can often be attributed to one or both instruments missing in at least one camera frame. Another problem encountered during the porcine trial was loss of image details due to

(a)

	Relative mean error (%)			Absolute mean error (mm)		
	Direct	**Dijkstra**	**Spline**	**Direct**	**Dijkstra**	**Spline**
User 1	15.2±11.1	13.6±12.5	10.2±9.7	8.2±6.2	6.9±5.9	5.2±4.7
User 2	17.8±13.1	15.0±11.1	12.4±11.4	8.6±6.5	7.2±5.1	5.8±5.1
User 3	13.6±10.6	13.2±10.3	9.7±6.6	6.3±5.1	6.0±4.9	4.5±3.1
User 4	20.4±15.0	15.4±10.7	11.5±8.4	9.8±8.2	6.8±4.6	5.0±3.3
User 5	17.0±12.0	17.8±12.9	15.4±9.5	8.3±5.8	8.8±6.4	7.6±4.6
User 6	13.4±13.1	19.2±9.6	14.9±9.3	7.0±7.1	10.1±5.2	7.8±4.8
User 7	10.1±7.8	20.1±15.4	14.9±12.4	5.4±4.6	10.4±8.0	7.7±6.6
User 8	13.0±11.4	18.4±9.9	13.6±9.1	5.9±5.2	8.4±4.7	6.2±4.2
User 9	26.6±11.3	14.0±7.7	13.7±8.4	12.9±6.7	6.7±3.8	6.6±4.3
User 10	7.1±6.0	12.3±8.3	9.5±6.8	7.1±6.0	12.3±8.3	9.5±6.8
All	15.1±12.2	15.8±11.2	12.5±9.4	7.8±6.4	8.4±6.1	6.6±5.1

(b)

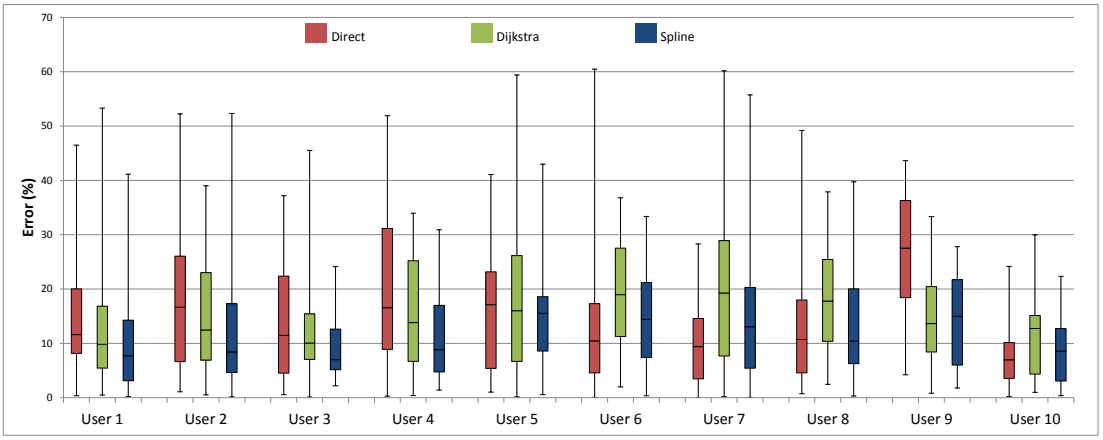
	Relative mean error (%)			Absolute mean error (mm)		
	Direct	**Dijkstra**	**Spline**	**Direct**	**Dijkstra**	**Spline**
User 1	9.2 ± 8.9	15.5 ± 11.1	6.4 ± 6.6	10.4 ± 10.4	17.3 ± 13	7.2 ± 7.4
User 2	8.4 ± 6.8	21.5 ± 11.6	7.2 ± 5.6	9.9 ± 9	23.1 ± 12.2	8.1 ± 6.8
User 3	12.4 ± 11.3	23.4 ± 19.7	9.7 ± 11.1	11.4 ± 9.9	22 ± 17.5	8.9 ± 9.6
User 4	10.2 ± 9.9	20.4 ± 13.1	7.6 ± 10	9.5 ± 9.3	17.8 ± 10.2	6.8 ± 8.3
All	10.1 ± 9.3	20.9 ± 14.4	7.8 ± 8.7	10.2±9.3	20.5±13.3	7.8±8

Table 5.2: The results for **automatic** measurement on phantom bowel. The tables show the absolute mean error and standard deviation (mm) and the relative mean error and standard deviation. Overall 397 iterations were evaluated. Table (a) shows the results for the 5 cm estimate and (b) for the 10 cm estimates [BWM+16].

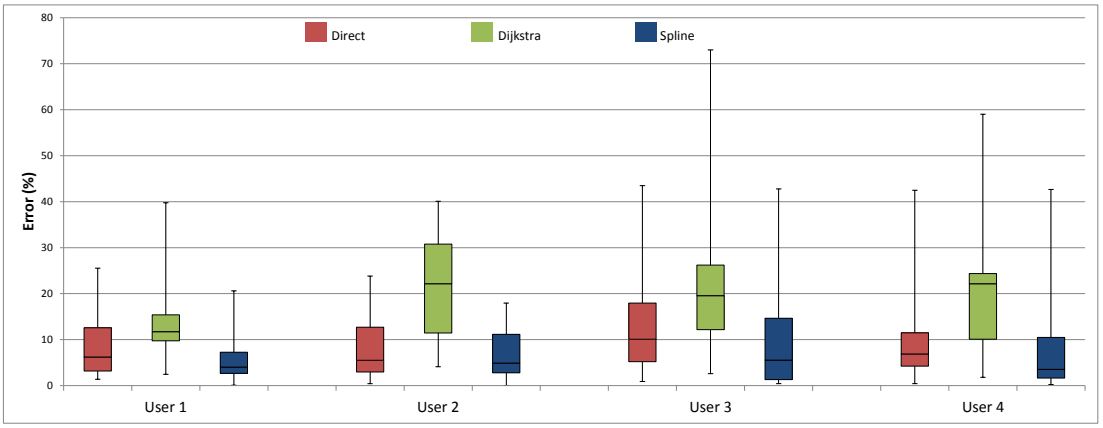
noise added by smoke or fluids on one camera (figure 5.16(c)). Table 5.4 and figure 5.15 show the results of the porcine trial.

Run-time

The offline evaluation was performed on a standard Workstation-PC (Intel Core i7-2700K CPU, GeForce GTX 650Ti GPU and 16GB RAM), were a measurement required 190 ms on average.



(a)



(b)

Figure 5.13: Relative errors for **automatic** measurement on phantom bowel. The medical experts were asked to estimate 5 cm at each iteration in (a) and 10 cm in (b) [BWM+16].

Discussion offline evaluation

On the phantom dataset, **Spline** outperforms the other two methods. As previously predicted, **Direct** tends to underestimate distances, while *Dijkstra* overestimates them. The **manual** approach performs slightly better than the *automatic* approach. For the **manual** instrument detection, **Spline** generally also produces the best estimate for bowel length per iteration. Furthermore, the data shows that the absolute measurement error increases only slightly when increasing the distance measured from 5 cm to 10 cm, while the relative error actually decreases.

When comparing the estimates of the measurement methods to the estimates provided by the experts, it becomes clear that both modes (**automatic** and **manual**) achieve a a higher accuracy than the user estimates. The absolute estimation error of the user actually increases with the distance being estimated.

On the porcine dataset, **Spline** had the lowest rate of error using both the **automatic** and the **manual** instrument detection modes. The measurements performed with the **automatic** instrument detection had a slightly higher accuracy than with the **manually** labeled instrument tips. This might be explained by inaccuracies in the reference data

(a)

	Relative mean error (%)	**Absolute mean error (mm)**
User 1	14.7±13.3	7.3±6.7
User 2	11.3±10.1	5.6±5.0
User 3	11.4±8.5	5.7±4.3
User 4	14.2±11.0	7.1±5.5
User 5	15.6±11.3	7.8±5.7
User 6	14.5±9.2	7.2±4.6
User 7	8.2±9.2	4.1±4.6
User 8	16.1±11.7	8.1±5.9
User 9	14.6±10.7	7.3±5.4
User 10	17.2±13.2	8.6±6.6
All	13.6±11.1	6.8±5.5

(b)

	Relative mean error (%)	**Absolute mean error (mm)**
User1	11.8±9.5	11.8 ± 9.5
User2	12.7±14.2	12.7 ± 14.2
User3	7.9±4.8	7.9 ± 4.8
User4	12.8±9.1	12.8 ± 9.1
All	11.1±9.9	11.1 ± 9.9

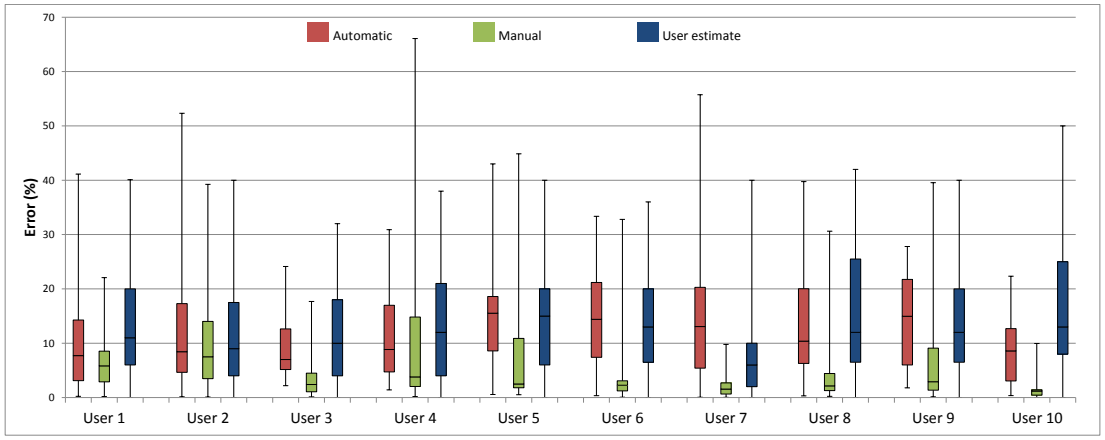
Table 5.3: The absolute mean error and the relative mean error for experts' estimates. The experts were told to estimate either 5 cm (a) or 10 cm (b) at each iteration [BWM+16].

due to the stretchability of the intestine, meaning its length changes when forces are applied.

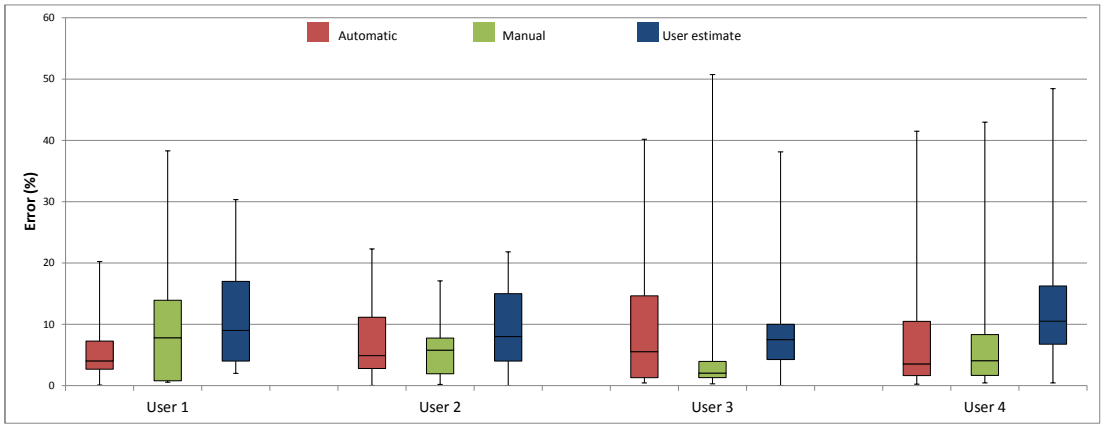
When comparing the results of the porcine and the phantom dataset, the porcine dataset had a significant lower accuracy This can be attributed to multiple sources: First, generating reference data inside the abdominal cavity is significantly more difficult, as a

	Relative mean error (%)			Absolute mean error (mm)		
	Direct	**Dijkstra**	**Spline**	**Direct**	**Dijkstra**	**Spline**
Manual	27 ± 13	24.5 ± 18	21 ± 13	16.22 ± 10.7	13.41 ± 10.3	11.72 ± 7.9
Auto.	20.8 ± 18	34.5 ± 17	17.7 ± 13	13.24 ± 11.9	18.32 ± 10.5	10.41 ± 8.5

Table 5.4: The results on the porcine dataset using both **manual** and **automatic** instrument detection. The table shows the relative mean error and standard deviation, and the absolute mean error and standard deviation (mm) [BWM+16].



(a)



(b)

Figure 5.14: Comparison of the relative error rates of the experts when estimating 5 cm (a) and 10 cm (b), the **automatic** and the **manual** method. **Spline** was used as measurement mode [BWM+16].

laparoscopic cauter had to be used, compared to a pin that could easily be placed by hand. Further, porcine intestine is elastic, which can lead to differences in length during image acquisition and reference data collection, since it cannot be guaranteed that the degree of elongation was similar. Finally, the bowel segmentation proved to be less

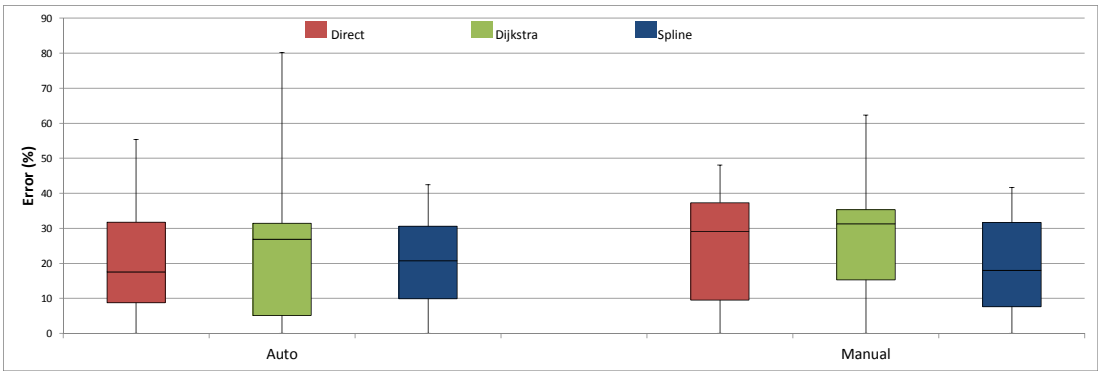


Figure 5.15: Relative error **automatic** and **manual** measurements on the porcine dataset [BWM+16].

	Precision	Recall	DICE
Phantom bowel	97%	94%	95%
Porcine bowel	73%	78%	76%

Table 5.5: Performance of the bowel segmentation on the labeled images collected. A leave-one-out evaluation was performed [BWM+16].

robust on the porcine frames (see table 5.5), which can be attributed to similarities in color and texture of the bowel and the surrounding tissue.

One potential error sources for instrument detection failure in both the phantom and the porcine dataset, is that instruments were not always fully visible, making detection difficult. Further, even if the instruments were visible in the left image, which was presented to the experts during data recording, it does not guarantee the both instruments could be seen in the right image (see figure 5.16(a)). Some failures in the porcine dataset can be attributed to abdominal fluid on the right camera (figure 5.16(b), which can lead to errors in the 3D reconstruction. One source of discrepancy between reference and measured distance are incorrectly placed instruments after completing an iteration as can be seen in figure 5.16(b).

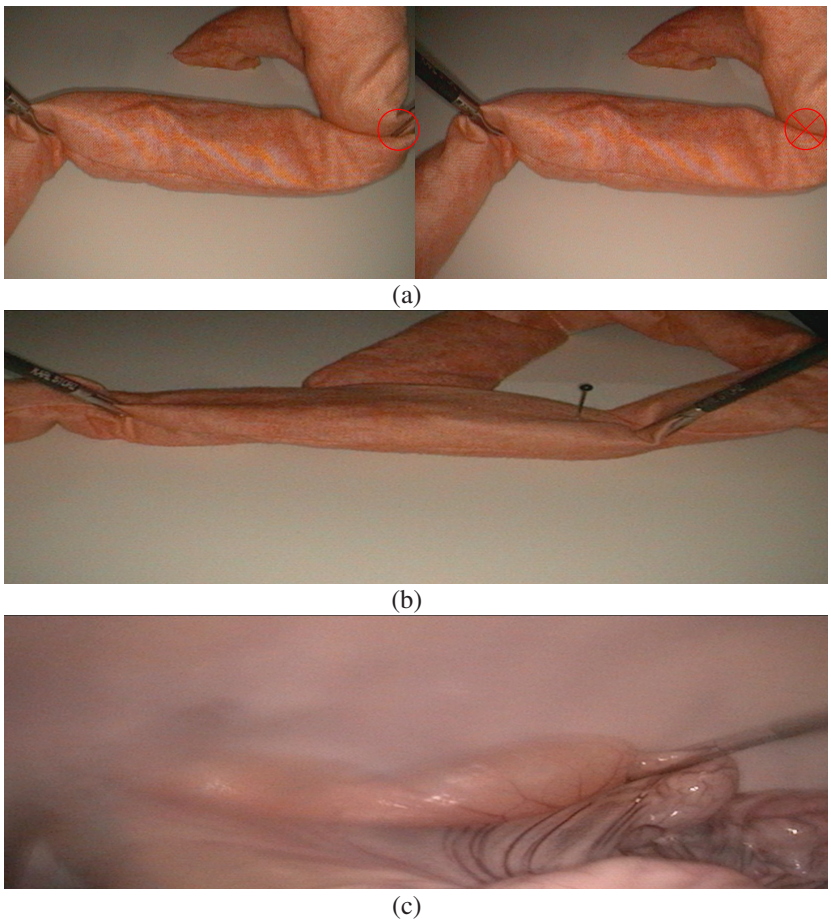
The results on both the phantom and the porcine dataset show that our proposed method can be applied to automatically extract distances along bowel segments. The **Spline** method produces the most accurate results than the other two proposed methods. It even outperforms estimates of medical experts.

Online evaluation

Due to the promising results of the offline evaluation, Wagner et al. further evaluated the proposed online measurement system in a clinical setting at the University Hospital of Heidelberg. For more details please see [WMB+17b] and [WMB+17a]. Please note that the details and results discussed in this section are part of the medical doctorate thesis of Benjamin Mayer of the University of Heidelberg [May17].

Preclinical evaluation & first in-human

In a first study [WMB+17b], a preclinical evaluation, leading up to a first-in-human trial, was performed. The trial consisted of four stages of increasing complexity or rather realism in regards to clinical usage. During the first three stages, a quantitative evaluation was performed on phantom bowel, ex-vivo porcine bowel and in-vivo porcine bowel respectively. For each bowel type, multiple test persons were asked to measure bowel segments with the proposed live measurement system. The reported length of each measurement and actual length were noted and recorded. The authors came to the conclusion that the measurement system generally overestimated distances, but that this error was small enough to warrant a clinical trial.



(a) (b) (c)

Figure 5.16: Sources of error during measurement: (a) instrument is not visible in both images of the stereo pair (b) the user did not place the second instrument near the pin, (c) right camera of the stereo endoscope is covered by fluid [BWM+16].

Due to the success in the in-vivo porcine stage, a first in-human feasibility study was performed. During a laparoscopic roux-en-Y gastric bypass operation, the attending surgeon was asked to measure the bowel using the proposed live system. The setup used in the operating room can be seen in figure 5.10. The first measurements with the proposed system failed, due to a third instrument being visible in the background. Once the third instrument was removed, the surgeon was able to successfully perform measurement on the human bowel. Screen captures of the screen presented to the surgeon during the live measurement can be seen in figure 5.17.

Comparative evaluation

In [WMB+17a] Wagner et al. performed a comparative validation of four different methods for estimating length of bowel segments. The methods consisted out of three methods that are already in clinical usage: visual judgment of the test person, instruments with markings at 5 cm increments and a premeasured tape with 35 cm length. Furthermore the proposed live measurement system was also evaluated. During the study, participants were asked to measure 70 cm of phantom bowel in a laparoscopic setting using all four methods, in a randomized order.

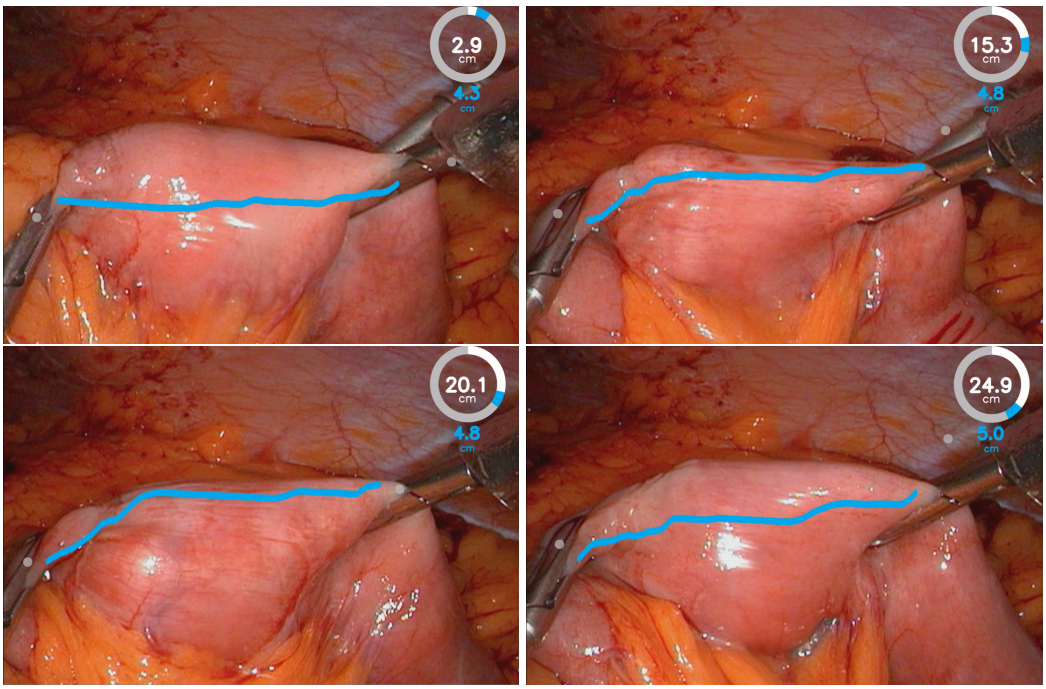


Figure 5.17: Screen captures from the first in-human feasibility study of the live measurement system.

Wagner et al. found that the participants achieved the lowest mean relative error with the proposed measurement system. Furthermore, they noted that the participants needed to grasp the bowel phantom fewer times with the measurement system than with the other measurement methods apart from the visual judgment. Wagner et al. recorded the time required for each measurement and found that on average, the participants required more time with the measurement system.

Discussion online evaluation

The proposed live measurement system has been successfully evaluated in a clinical setting. The evaluations by Wagner et al. [WMB+17b][WMB+17a] demonstrate that the system is able to successfully perform accurate measurements on phantom bowel, ex-vivo porcine bowel and in-vivo porcine bowel. Furthermore, they demonstrated the feasibility of performing live measurements on in-vivo human bowel, though no reference data could be collected here.

It has also shown that a higher accuracy in measurement could be achieved than with conventional methods. The participants in the study required longer with the live measurement system than with the conventional methods, which can possibly be attributed to failed measurements due to instruments not being visible in both frames. Furthermore, the evaluation found that the system reduced the potential for instrument reduced trauma when compared to using marked instruments or a premeasured tape for measurement.

5.2.3 Conclusion

In this section, we proposed and evaluated methods for intraoperatively measuring distances along organ surfaces in a laparoscopic setting. Apart from a stereo endoscope, no further hardware or modification to the surgical workflow is required, making integration easily achievable. The proposed method was evaluated in the scenario of estimating bowel length during laparoscopic roux-en-Y gastric bypass. In an offline evaluation using phantom data, we were able to demonstrate a higher accuracy than human estimates. Furthermore, we evaluated the method on porcine in-vivo data. During these two experiments, we were able to show that our **Spline** measurement method achieves that highest accuracy.

A live implementation of our method was evaluated in a clinical setting at the University Hospital of Heidelberg by Wagner et al. [WMB+17b][WMB+17a]. They showed that the system is able to perform accurate measurements on phantom, ex-vivo and in-vivo bowel. Furthermore, they showed during a first in-human study that measuring on in-vivo human bowel is also possible. Collecting reference data during the first in-human study was not possible though. In a comparative study on phantom bowel, it was also shown that the live measurement system on average achieves a higher accuracies during measurement than three other methods, though measurement took longer. It further required fewer interactions between instruments and bowel than two of the other methods. These interactions can result in trauma to the bowel.

While only measurements using stereo reconstruction was covered in this section, the system is flexible enough that the stereo reconstruction component could be replaced by any other method for 3D reconstruction, such as time of flight. Once such systems become available for intraoperative usage, the system can easily be adapted.

6 Laparoscopic Workflow Analysis

In this chapter, we focus on purely image-based laparoscopic workflow analysis. Analyzing the surgical workflow is a prerequisite for many applications in computer-assisted surgery, such as providing the position of a tumor, specifying the most probable tool required next by the surgeon or determining the remaining duration of surgery, .

Many workflow analysis tasks, e.g. phase recognition, skill assessment, automatic reporting, video indexing or automatic annotation, require a method for providing a temporal representation of video frames, or rather their content. To tackle this problem, we introduce in section 6.1 a method that utilizes unlabeled data to learn more general features that allow supervised workflow analysis. By first extracting these features, we hope to reduce the number of labeled training samples required for further workflow analysis tasks. In section 6.2 we apply these features on the task of surgical workflow segmentation, where we attempt to divide given laparoscopic videos into surgical phases. Here we examine the suitability of our method on two types of laparoscopic surgery of varying complexity. Furthermore, in section 6.3, we show that our features can also be used to estimate the progress of laparoscopic surgeries using unlabeled data.

6.1 Temporal Context Learning

In this section, we introduce a method for learning visual features from unlabeled videos of laparoscopic surgeries by sorting frames into the correct temporal order. The work presented here was inspired by the work in [DGE15], in which the authors train a convolutional neural network (CNN) to develop an understanding of the spatial context of different excerpts from a given image. For this, they divide unlabed images into multiple 3x3 box grids and train a CNN to arranged the outer blocks correctly in relation to the center block. Part of this trained CNN is then modified and retrained to partake in an object detection challenge, achieving state of the art results.

Inspired by this, we extended the idea of pretraining a CNN with spatial context information to pretraining a CNN with temporal context information provided by videos. The task we propose for training the CNN is illustrated in figure 6.1: Given two frames from the same laparoscopic surgery, what is the most probable relative order of the two frames, i.e. which frame comes first? For this, we uniformly sample two random frames from the video of a laparoscopic surgery and feed it into our CNN. The label describing the order is taken directly from the order of the frames in the video. The CNN must decide the relative order of the two frames in the original video, i.e. which frame comes first. We assume that the features learned while solving the sorting task

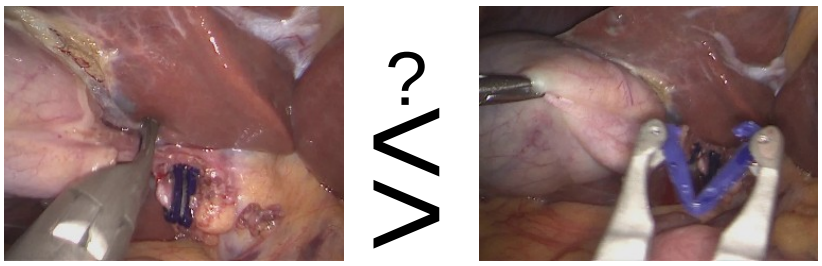


Figure 6.1: Task for pretraining a CNN for workflow analysis. Which is the most probable temporal order of the two images? (Answer: the right image is the first frame, the clip has to be inserted into the body, before being placed.) [BWK+17]

enable the CNN to distinguish frames based on their temporal context. Such a CNN can be used as starting point for many applications were these visual cues would be beneficial, e.g. online and offline video segmentation, automatic annotation, indexing and generating surgical reports. This temporal context learning task is performed using unlabeled laparoscopic videos. The work presented in this section was published in [BWK+17].

6.1.1 Methods for temporal context learning

Training methods for workflow analysis generally requires labeled data, which, especially in the medical field, is not always feasible to obtain. The strength of our proposed method is that relies solemnly on unlabeled data, which makes it possible to just use large amounts of videos collected directly from the OR for training. Here, we first describe the dataset that was used for training. We then go into detail in regards to the methods and the CNN architecture used.

Dataset

A large dataset consisting of 324 laparoscopic surgeries was recorded anonymously at the University Hospital of Heidelberg. The dataset contains videos of 30 different types of laparoscopic surgeries, performed by multiple surgeons with varying endoscopes and optics, providing a diverse range in training data. The surgeries were recorded in the same operating room using the integrated operating room system OR1™ (Karl Storz GmbH & Co KG, Tuttlingen, Germany). From these videos, we extracted frames at intervals of one frame per second, resulting in approximately 2.2 million images. Since the videos were recorded automatically, we had to ensure to exclude sequences that did not contain any large changes (e.g. black screens), from the dataset. This was accomplished by excluding a video frame f from the dataset, if for the last video frame g from the same video that was included in the dataset

$$||I(f) - I(g)|| < 8000 \tag{6.1}$$

with $I(f)$ and $I(g)$ being the respective pixel values for each image.

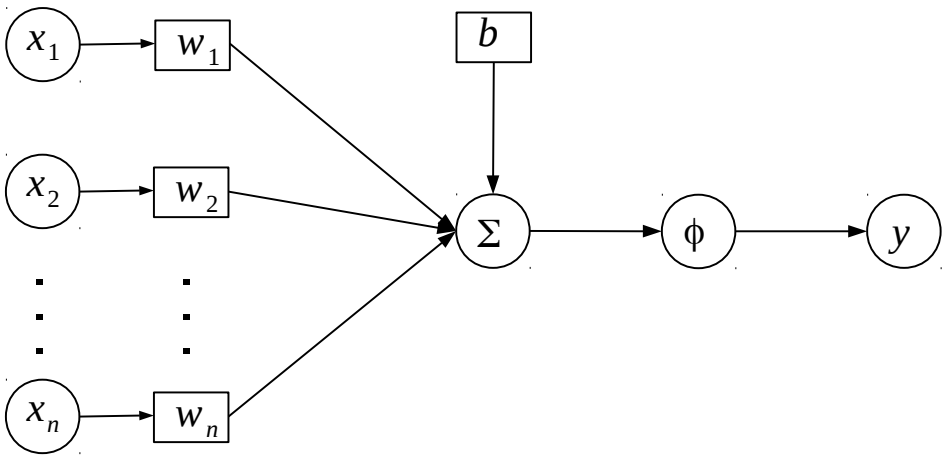


Figure 6.2: Illustrated example of a perceptron. x_i are the inputs, which are weight with w_i. The output is then the weighted sum plus the bias b processed by an activation function $\phi(x)$.

Artificial neural networks & Deep Learning

The foundation of our proposed method for workflow analysis is a convolutional neural network (CNN), a specialized form of an artificial neural network. In the section, we will provide a brief overview of artificial neural networks and training techniques.

Perceptron

The perceptron is the building stone of artificial neural networks. A perceptron is a linear discriminator inspired by the neuron: only if the weighted input is larger than a certain threshold does the perceptron emit an output (see figure 6.2). Mathematically, it takes the form of:

$$y = \phi(\sum_i w_i \cdot x_i + b) = \phi(w \cdot x^T + b) \tag{6.2}$$

where x_i are the inputs to the perceptron, w_i are weights, b is the activation threshold or bias and $\phi(x)$ is the activation function. Traditionally, the perceptron is described as a binary classifier with the $\phi(x) = \text{sign}(x)$ [DHS01]. Multi-layer perceptrons though are usually trained using gradient descent, which requires $\phi(x)$ to be derivable. Therefore, in this work, we generally use one of the following functions as activation function [DSR+15]:

- Sigmoid: $\phi(x) = \frac{1}{1+e^{-x}}$
- Tanh: $\phi(x) = \tanh(x)$
- ReLu: $\phi(x) = \max(0,x)$
- Softmax: $\phi(x)_j = \frac{e^{x_j}}{\sum_{k=1}^{K} e^{x_k}}$, j is the current output perceptron and K is the number of output perceptrons.

Multi-layer neural networks

A single layer of perceptrons is limited in the type of functions it can approximate, as each perceptron can only separate classes that are actually linearly separable. Therefore, calculating certain functions, e.g. the XOR function, is impossible with a single layer of perceptrons. A muli-layer neural network, on the other hand, consists of multiple layers of perceptrons, which can, at least in theory, compute the optimal solution to any classification problem. A multi-layer neural networks generally consists out of an input, an output and at least one hidden layer of perceptrons.

Given an input $x_j \in \mathbb{R}^K = [x_{j0} \dots x_{jK}]^T$ at a layer j with M units, the output $y_j \in \mathbb{R}^M$ is computed in the following manner:

$$y_j = \phi(W_j \cdot x_j + b_j) \tag{6.3}$$

where $W_j \in \mathbb{R}^{K \times M} = \begin{bmatrix} w_j^{1,1} & \dots & w_j^{K,1} \\ \vdots & & \vdots \\ w_j^{1,M} & \dots & w_j^{K,M} \end{bmatrix}$ is a matrix containing the weights of the K inputs to the M perceptrons in layer j and $b_j \in \mathbb{R}^M = [b_j^1 \dots b_j^M]$ is a vector containing the biases of the perceptrons. $\phi(x)$ here is an element-wise function [DHS01].

Training

A multi-layer neural network can basically compute any function, but the question now arises, given a fixed topology of layers and labeled training samples, how to determine the values for weights and biases. Backpropagation, a form of gradient descent, is the most commonly used method to optimize the networks parameters.

In general, the approach starts with an untrained network, with randomly initialized weights and biases, uses training examples as input and then computes the difference between the target value t and the actual output y. The parameters are then optimized in such a way that they minimize the delta. For this, a cost function C is required. Common choices for C are [DSR+15]:

- Absolute error: $C(t,y) = |t-y|$
- Squared error: $C(t,y) = (t-y)^2$
- Categorical cross-entropy (used for classification): $C(t,y)_i = -\sum_j t_{i,j} log(y_{i,j})$

Given a problem-specific cost function C, the weights are modified in a direction that will minimize C:

$$\Delta W = -\eta \frac{\partial C}{\partial W} \tag{6.4}$$

$$W_{\text{new}} = W + \Delta W \tag{6.5}$$

Here η is the learning rate, which controls the size of the step taken to change the weights and W_{new} are the modified weights. This method is called backpropagation, as the error must be propagated backwards from the output layer to the hidden layers, so

that these weights can also be learned. For this, a backward sweep is used to compute the derivate in a recurrent manner. For more details please see [DHS01].

One common modification to the learning rule equation 6.4 is the nesterov momentum [Nes83]:

$$v_{\text{new}} = m \cdot v - \eta \frac{\partial C}{\partial W} \tag{6.6}$$

$$W_{\text{new}} = W + m * v_{\text{new}} - \eta \frac{\partial C}{\partial W} \tag{6.7}$$

where m is the momentum, a parameter supplied by the user. A larger momentum results in a smoothing over more update steps.

Another modification of equation 6.4 is the Adam rule [KB14]:

$$m_{\text{new}} = \frac{\beta_1 m + (1-\beta_1) \cdot \frac{\partial C}{\partial W}}{1-\beta_1} \tag{6.8}$$

$$v_{\text{new}} = \frac{\beta_2 v + (1-\beta_2) \cdot \frac{\partial C}{\partial W}^2}{1-\beta_2} \tag{6.9}$$

$$W_{\text{new}} = W - \frac{\eta}{\sqrt{v_{\text{new}}} + \varepsilon} \cdot m_{\text{new}} \tag{6.10}$$

β_1 and β_2 are user supplied terms, intended to speed up convergence.

Convolutional neural networks

Convolutional neural networks (CNN) are a form of multi-layer neural network inspired by the visual cortex in animals. The idea behind CNNs is that the further away a layer is to the input signal (in this work generally an image) the more complex features it will extract. In the example of image processing: a layer directly behind the input layer might learn to compute image gradients, while the layer after that might use the gradient information to detect edges or lines. These features again might be combined into a more complex construct by the next layer [LBD+89].

Up until now, the neural network architectures introduced worked globally or, in other words, the position of a pattern in the input is highly relevant. CNNs introduce a new form of hidden layer, the convolutional layer, which is able to extract local patterns. These convolutional layers learn multiple feature maps in form of filters (similar to the Sobel operators in section 4.1.1), which are replicated over the entire input, utilizing shared weights. Figure 6.3 shows a 1D illustrated example of such a feature map that is being computed from one input pixel and its left and right neighbors. This principle can be extended to 2D images (see figure 6.4) were 2D filter matrices, also called kernels, are convoluted with different features maps or image channels to extract one feature response [LBD+89].

Apart from the convolutional layers, max-pooling is the other important concept for CNNs. Max-pooling examines rectangular neighborhoods of the image and output their maximum, an example can be figure 6.5. Max-pooling can have multiple benefi-

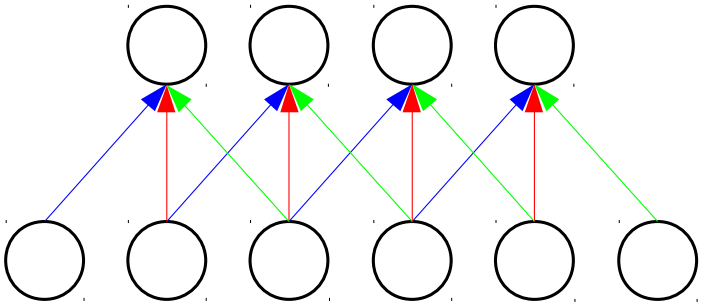


Figure 6.3: Illustrated example of a convolutional layer: A feature map extracts features from an image. Here a pixel and its left and right neighbors are examined. Arrows of the same color share the same weight.

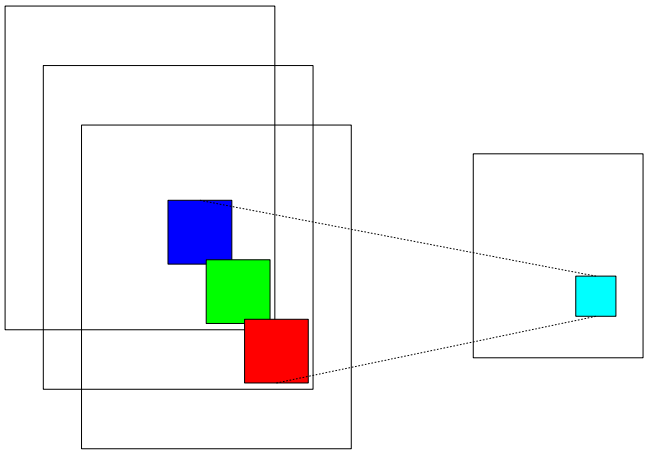


Figure 6.4: Illustrated example of a 2D convolutional layer: A 2D filter combines multiple feature maps or channels into one feature response.

cial effects in a CNN. First it reduces computation time, by eliminating non-maximum values. Furthermore, since it doesn't just downsample the image, but instead takes the maximum, it introduces translational invariance [KSH12].

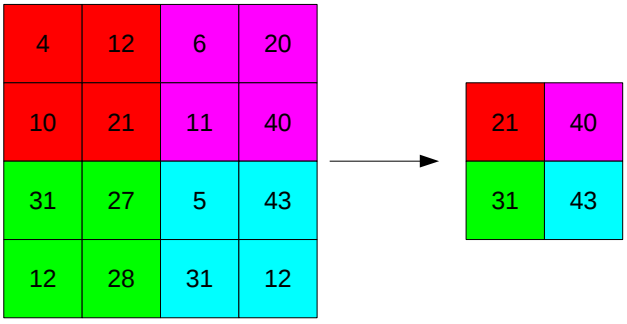


Figure 6.5: Effects of a max-pooling layer with filter size 2×2 and stride 2.

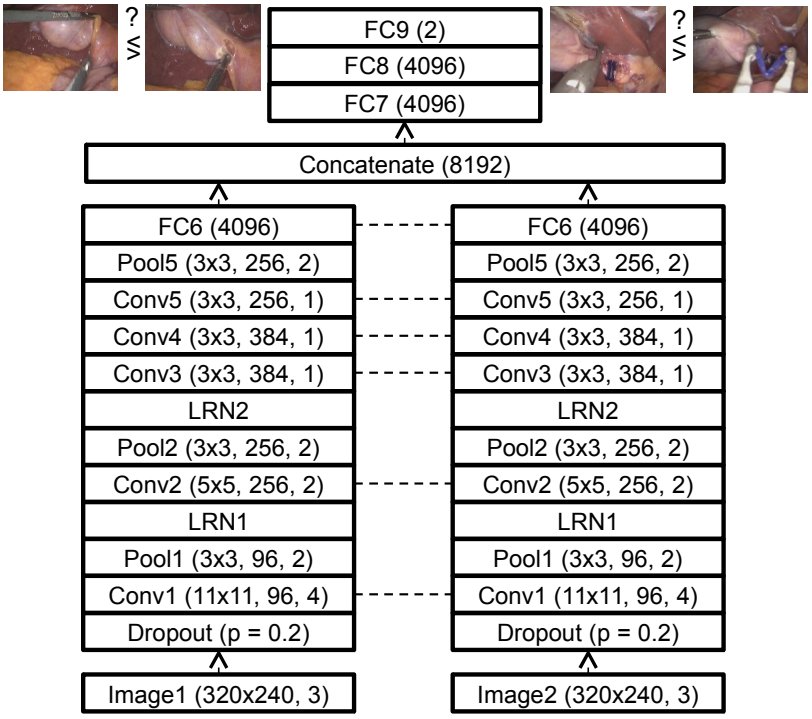


Figure 6.6: CNN Topology for the temporal context prediction task: Dotted lines indicate shared weights. *Dropout* are dropout layers that, with a probability of p, set a value to zero. *Conv* are convolutional layers, *LRN* are local response normalization layers [KSH12], *Pool* are max-pooling layers, *FC* are fully connected layers and *Concatenate* concatenate two input vectors. The numbers in parenthesis indicate size of filter kernel, number of outputs and step size. In the case of fully connected layers, the number of hidden units is listed instead [BWK+17].

CNN architecture & training

Selecting a topology that allows a CNN to predict the relative order of two given video is a difficult task. As our proposed task is similar to the one introduced in [DGE15], we based our model on the one proposed there, as it has been shown to work for spatial context prediction. The topology of our network used can be seen in figure 6.6. Two frames from the same video are fed into the two input layers of the proposed CNN. Each frame is then processed by a chain of convolutional layers (Conv1 to Conv5), each chain with AlexNet-style topology [KSH12]. This results in a reduced representation of the frames in a fully connected layer (FC6). The corresponding layers in both chains share weights. The outputs of the two FC6 layers are then concatenated and then processed using two further fully connected layers. FC9 then decides if either frame 1 (Output: 0) oder frame 2 (Output: 1) comes first in a temporal order in the selected video. For every convolutional and fully connected layer, except FC9, a ReLu nonlinearity [NH10] was used. FC9 uses a softmax nonlinearity instead.

During training, for each epoch 256 operations out of all operations are randomly sampled without replacement. From each of these operations, 3 frames, I_1, I_2 and I_3, were drawn randomly, with $I_t < I_{t+1}$ or, in other words, I_t precedes I_{t+1} in a temporal order. We then crop the borders of the images to a $4:3$ aspect ratio in case they

exhibited a different ratio. The frames are then downsampled to a resolution of 320×240. Furthermore, we normalize each value in the RGB channels by mapping them into the range of $[-0.5, 0.5]$. We then form 6 inequations, i.e.

$$I_0 < I_1,\ I_0 < I_2,\ I_1 < I_2 \tag{6.11}$$

$$I_1 > I_0,\ I_2 > I_0,\ I_2 > I_1 \tag{6.12}$$

resulting in 1536 inequations per epoch. The CNN is then trained for 10000 epochs using stochastic gradient descent (learning rate of 0.0005) combined with nesterov momentum (momentum of 0.9). As loss function, we selected categorical cross-entropy.

6.1.2 Results & discussion

The proposed CNN and its training was implemented in Python, using Theano [The16] and Lasagne [DSR+15], using a NVidia GTX Titan X. The training lasted approximately 2.5 days. The trained network was then able to sort 93.7% of image pairs in the last epoch into the correct order. As the proposed task was only a method for pretraining the CNN, we did not fully evaluate the resulting method of sorting frames using separate testing and training sets. In other words, the above mentioned number is only to be understood as an indicator that the CNN did indeed learn useful features. We more thoroughly investigate the actual suitability of these features in the next two sections. Here we propose two extensions to our pretrained CNN, which make phase detection and surgical progress estimation possible. Furthermore, we evaluate whether the pretrained features influenced the results in a positive manner.

6.2 Temporal Context Learning for Surgical Workflow Segmentation

For a given laparoscopic frame, the method outlined in section 6.1 generates a descriptor, the output of FC6, that can be used for temporal distinction. In this section, we determine the suitability of such a descriptor for surgical workflow segmentation, i.e. dividing a given surgical in coherent and semantic meaningful segments. In our case, we aim to provide a label, in the form of a surgical phases, to each frame in a given video. For this, we propose two CNN architectures, one that decides upon the surgical on a frame-by-frame basis and another one, which takes past frames into consideration when classifying the current frame. Furthermore, we evaluate whether our proposed pretraining and the resulting features increase accuracy during workflow segmentation.

The work presented in this section was published in [BWK+17].

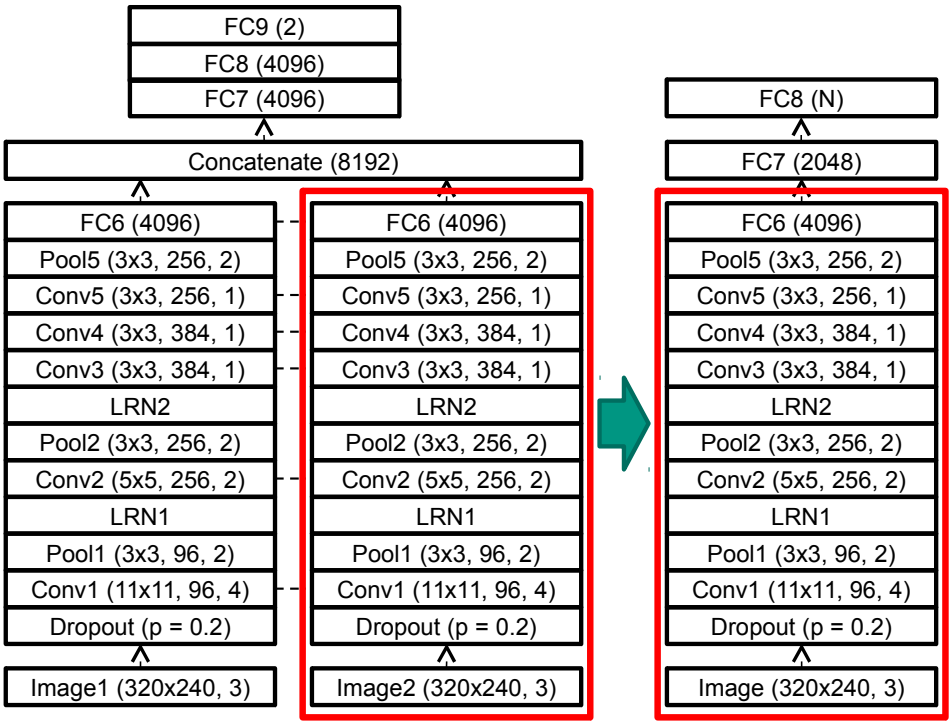


Figure 6.7: For a frame-based approach to workflow segmentation, we take part of the CNN illustrated in figure 6.6 and add two further fully-connected layers to assign a class. Here *N* indicates the number of phases [BWK+17].

6.2.1 Methods for workflow segmentation

Frame-based approach

One simple approach to workflow segmentation would be to extend the CNN introduced in section 6.1. For one of the processing chains (everything before FC6) is paired with further fully-connected layers to assign each frame to the most probable class label. We constructed a frame-based CNN for laparoscopic workflow analysis as can be seen in figure 6.7.

While distinguishing frames certainly is a prerequisites for laparoscopic phase detection, determining the current state from just a single frame seems questionable and prone to ambiguities. We assume that single frames alone do not contain sufficient information to deduce the current phase and therefore propose to extent the CNN to include information seen in previous frames.

Gated-recurrent units

Up until now, we have only examined neural networks that did not contain cycles, so called feedforward neural networks. This implies that the output of a given network only depends on its current input and does not take past information into consideration. Recurrent neural networks (RNN) overcome this limitation by introducing cycles in the topology of the network and thereby allowing the network to process sequences.

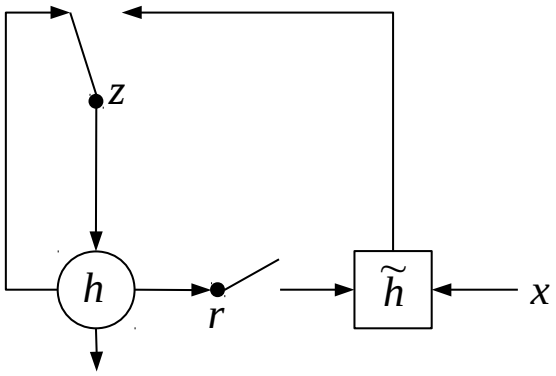


Figure 6.8: An illustrated example of the structure of a gated recurrent unit. After [CVMG+14].

Traditional RNNs suffer from multiple drawbacks, such as gradients that vanish when looking further back during training and therefore recalling only “recent” information [HS97]. Long-term-short-term memory units (LSTM)[HS97], a deep RNN architecture, do not suffer from these drawbacks and, furthermore, are selective about the information they retain and forget.

Similar to LSTMs, gated recurrent units (GRU)[CVMG+14] also do not suffer from the drawbacks of traditional RNN architecture and can learn to recall/forget particular information. As GRUs perform similarly to LSTM while having fewer parameters [CGCB14], this work focuses solemnly on GRUs.

A GRU contains a hidden state h_t, which can recollect previously seen information. It also contains a reset gate r_t, which controls whether the previous hidden state is ignored. Furthermore an update gate z_t decides whether the current hidden state is updated with a new hidden state $\tilde{h}_t$ At each timestep, a GRU outputs h_t. For an illustration, see figure 6.8.

Mathematically, the GRU can be expressed as:

$$z_t = \phi_g(W_z x_t + U_z h_{t-1} + b_z) \tag{6.13}$$

$$r_t = \phi_g(W_r x_t + U_r h_{t-1} + b_r) \tag{6.14}$$

$$\tilde{h}_t = \phi_h(W_h x_t + U_h(r_t \circ h_{t-1}) + b_h) \tag{6.15}$$

$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t \tag{6.16}$$

where W_z and U_z are the weights and b_z the bias of the update gate, W_r and U_r are the weights and b_r the bias of the reset gate and W_r and U_r are the weights and b_r the bias of the hidden update. $\circ$ denotes the Hadamard (entrywise) product.

History-based approach

Recurrent neural network architectures such asGRUs make it possible to integrate previous observations while calculating the current prediction. Therefore, building upon a

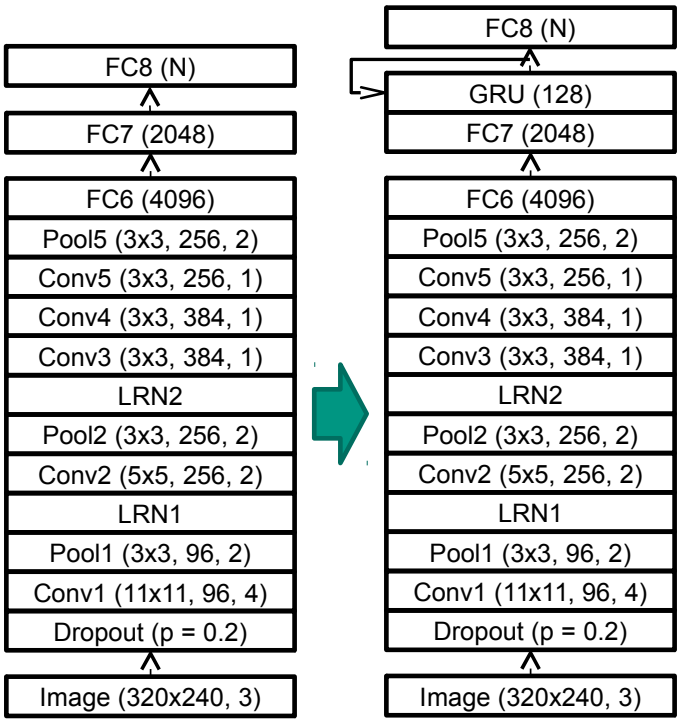


Figure 6.9: History-based workflow segmentation: To incorporate previously seen information into our approach for workflow segmentation, we combine the frame-based approach with a gated recurrent unit (GRU)[CVMG+14], which makes it possible to retain information from previous frames [BWK+17].

RNN would make it possible to design a neural network for a history-based workflow segmentation. For this, we extend the previously introduced CNN for frame-based workflow segmentation with a GRU (figure 6.9) into a recurrent CNN for history-based workflow segmentation. To integrate the GRU, the output from FC6 has to be modified slightly, as RNNs expect sequences as input. For this, the output from FC6, a 2D tensor of the shape $batchsize \times 4096$, is reshaped into a 3D tensor of shape $1 \times batchsize \times 4096$, simulating a $batchsize$ long sequence. The number of frames in a video generally exceeds the batch size, meaning that, instead of one long sequence, the GRU only sees multiple shorter sequences. To compensate for this, we take the contents of the hidden state after the last element of the sequence and use it to initialize the hidden state before processing the next batch.

The CNN for the history-based approach is trained using stochastic gradient descent (initial learning rate λ_0 was set to 10^{-3}) combined with nesterov momentum [Nes83] (momentum of 0.9) for 100 epochs with a batch size of 256. To penalize large weights and thereby prevent overfitting, L1 and L2 regularization are applied during training. For this, we add terms to the cost function, which incorporate the L1 and L2 norm of the weights and thereby penalize large weights. We selected a weight of 10^{-5} for the L1 penalty term and 10^{-3} for the L2 penalty term. To ensure convergence, we reduced the learning rate λ every epoch by a factor of α:

$$\lambda_{t+1} = \alpha \cdot \lambda_t \tag{6.17}$$

Phase ID	Explanation
1	Placement of trocars
2	Preparation of Calot's triangle
3	Clipping and cutting of cystic artery and duct
4	Gallbladder dissection
5	Gallbladder retrieval
6	Hemostasis
7	Attaching drainage, wound closure and end of operation

Table 6.1: Different phases in EndoVis15Workflow.

For α, we selected 0.975 as value. Since we assume that the features previously learned in section 6.1 are well-suited for workflow analysis, we use a smaller learn rate $\lambda_t' = 10^{-1} \cdot \lambda_t$ for FC6 and all layers proceeding it, thereby only fine-tuning the value of these parameters. The value for the parameters specified here were determined empirically.

6.2.2 Evaluation of the workflow segmentation

We evaluated the presented approaches for workflow segmentation on two datasets for laparoscopic phase detection. To compare our proposed method to the state of the art, we first evaluate on the publicly available dataset from the Endoscopic Vision 2015 Workflow Challenge[1] (EndoVis15Workflow). Furthermore, to show that our method translates to longer, more complex surgeries, we evaluate our method on a dataset comprised of colorectal surgeries from the University Hospital of Heidelberg.

EndoVis15Workflow

The public dataset from the EndoVis 2015 workflow challenge consists of 7 laparoscopic cholecystectomies provided by the Technische Universität München. The videos have been segmented into surgical phases, seven phases in total (table 6.1). For each video frame the corresponding label was provided as annotation.

To train both proposed CNNs, we first sampled the provided videos at a rate of one frame per second. We crop the images to achieve a $4:3$ aspect ratio and then resampled the resolution of the selected frames to 320×240. Using this modified data, we perform a leave-one-surgery-out evaluation (training on 6 videos and testing on the 7^{th} video for all seven possible combination of training videos). The progression of the

[1] http://endovissub-workflow.grand-challenge.org/

	Precision	Recall	Accuracy
Frame-based	56.6% ± 7.5%	53.7% ± 8.8%	56.3% ± 8.1%
History-based	79.3% ± 8.1%	73.7% ± 9.7%	74.5% ± 8.4 %
History-based without pretraining	75.4% ± 11.8%	68.8% ± 12.6%	66.0% ± 14.8%
EndoNet (CNN only)[TSM+16]	64.8% ± 7.3%	64.3% ± 11.8%	65.9% ± 4.7%
EndoNet (CNN + HHMM)[TSM+16]	83.0% ± 12.5%	79.2% ± 17.5%	76.3% ± 5.1%
Dergachyova et al.[DBH+16]	72.1% ± 16.4%	71.3% ± 13.6%	68.1%

Table 6.2: Comparison of the results of our proposed methods, EndoNet [TSM+16] (only online results) and the method proposed by Dergachyova et al.[DBH+16] [BWK+17].

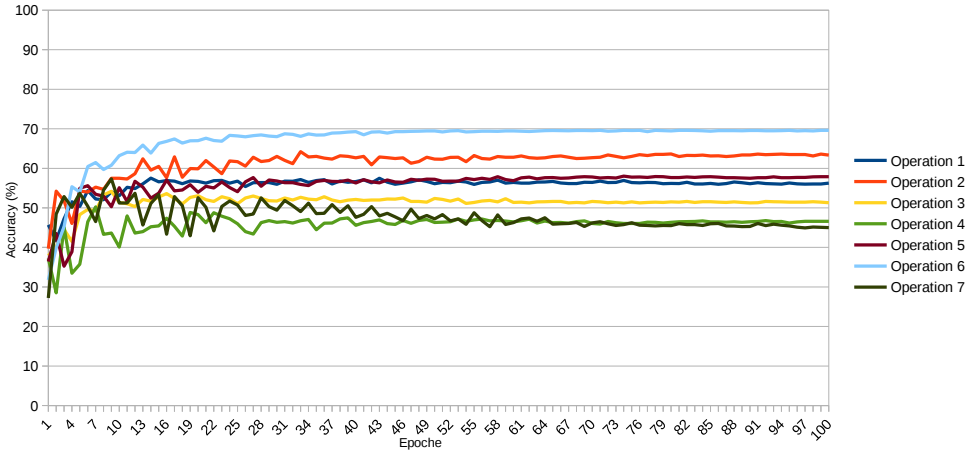
	Precision	Recall	Accuracy
P1	85.2%±12.0%	98.3%±4.4%	98.6%±1.1%
P2	81.8%±8.3%	89.0%±9.6%	94.0%±5.2%
P3	72.0%±25.8%	64.0%±34.0%	89.3%±5.1%
P4	71.7%±34.2%	55.8%±41.6%	88.3%±4.4%
P5	77.5%±23.1%	83.3%±14.1%	92.5%±5.6%
P6	78.4%±23.6%	51.6%±37.9%	88.4%±5.8%
P7	88.4%±15.2%	73.9%±26.7%	97.8%±1.6%

Table 6.3: Performance of history-based workflow segmentation broken down into the different phases [BWK+17].

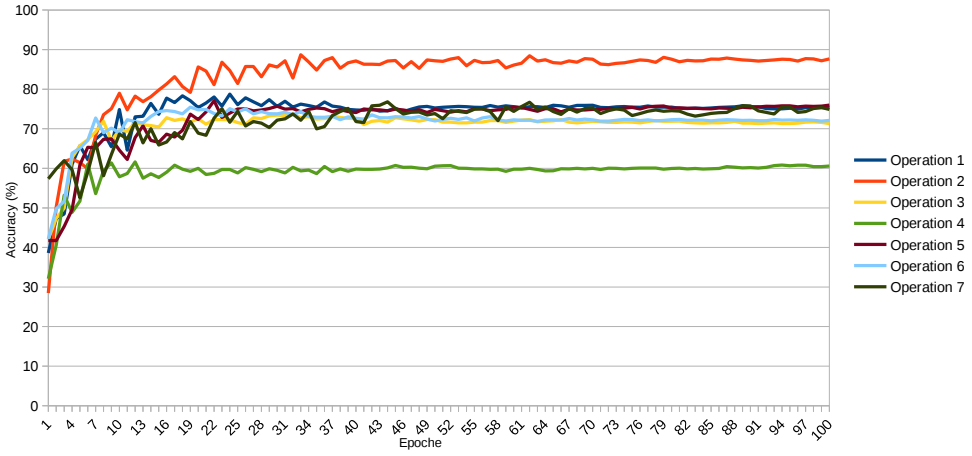
accuracies for each run can be found in figure 6.10. To demonstrate the advantage of the proposed pretraining, we also included results for a version of the history-based approach with randomly initialized weights in figure 6.10(c). Figure 6.10 clearly shows that the GRU-based methods outperform the feedforward-based CNN. Furthermore, we are also able to demonstrate that the pretraining as outlined in section 6.1 increases performance when compared to randomly initialized parameters. Table 6.2 highlights this, as it shows that the history-based approach with pretraining achieves a higher precision, recall and accuracy in comparison to the approach without pretraining. It also outperforms the frame-based approach.

We also compared our results to those published by Twinanda et al. [TSM+16] and Dergachyova et al. [DBH+16] (table 6.2). The history-based CNN outperforms the method presented by Dergachyova et al. and the CNN only version of EndoNet. The CNN + HHMM-based EndoNet outperforms it, which can be attributed the large task specific dataset used for training EndoNet.

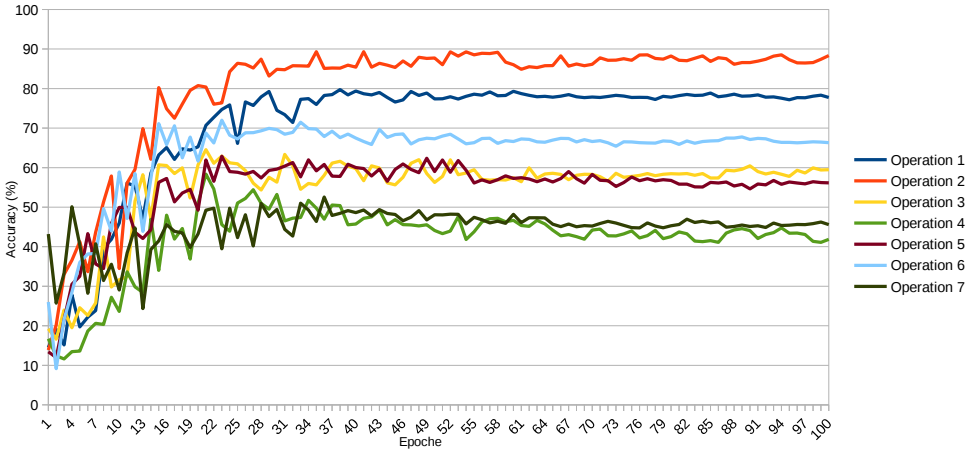
Table 6.3 shows how the history-based workflow segmentation performs for each of the 7 phases individually. The phases closes to the ends of the surgery achieve the highest performance in all metrics, while phase further away perform somewhat worse. Of all phases, phase 6 has the lowest accuracy and recall, which can be attributed to the



(a) Frame-based



(b) History-based



(c) History-based without pretraining

Figure 6.10: Development of the accuracies of the phase detection for each operation and network during the leave-one-surgery-out evaluation on the EndoVis dataset [BWK+17].

Phase ID	Explanation
1	Team Time-Out
2	Preparation and orientation at abdomen
3	Mobilization of colon
4	Dissection of lymph nodes and blood vessels
5	Dissection and resection of rectum
6	Preparation of anastomosis
7	Placing stoma
8	Finishing the operation

Table 6.4: Different phases in the colorectal dataset.

fact that phase 5 and 6 are often intermingled and visually very similar, making them difficult to distinguish. Phase 4 also has a low performance, which could be explained by mix-ups with phases 3 and 6, which are also visually similar.

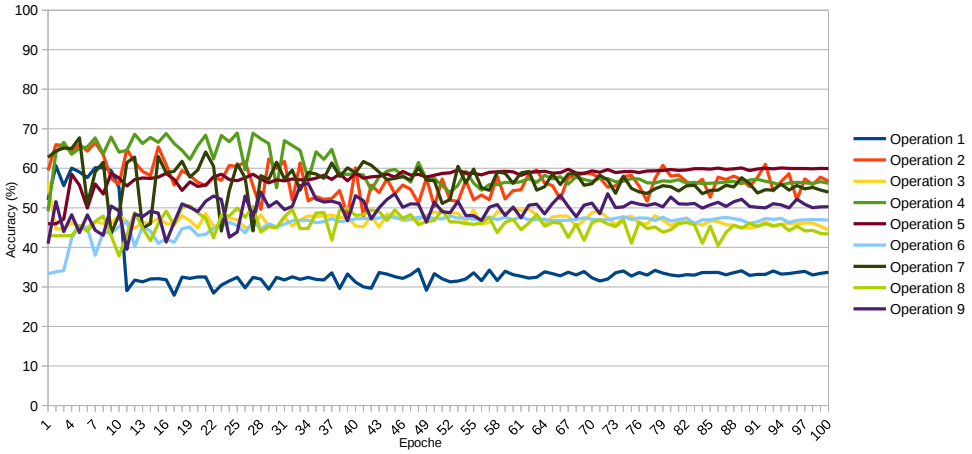
Colorectal laparoscopy

This dataset is made up of 9 colorectal laparoscopies recorded at the University Hospital of Heidelberg. These 9 surgeries contain 6 proctocolectomies and 3 rectal resections. While these surgeries were recorded in the same manner as the dataset outlined in section 6.1, the two datasets are disjunct. The same surgical expert divided each of these laparoscopies into 8 phases (see table 6.4).

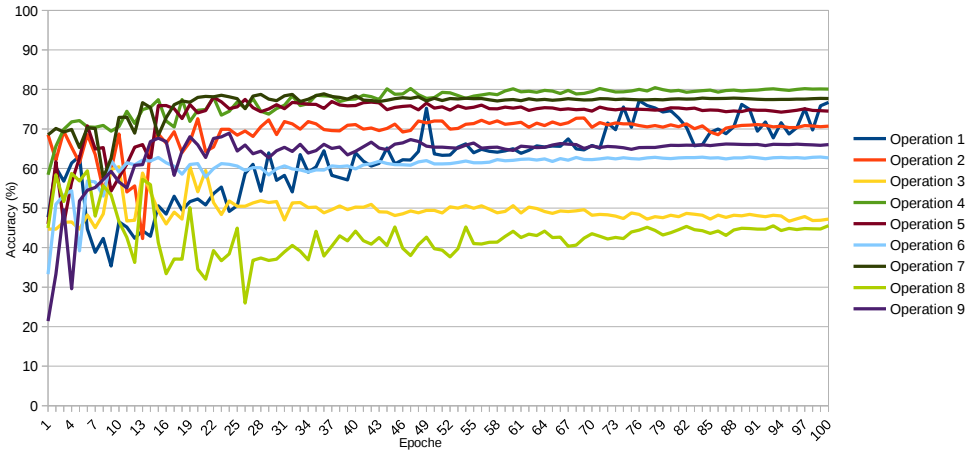
In the same manner as with the EndoVis15Workflow dataset, we extracted one frame per second from the laparoscopic videos and resampled the frames to a resolution of 320×240. With this dataset, we then performed a leave-one-surgery-out evaluation for both the frame-based and the history-based workflow segmentation. The same evaluation was also performed for a version of the history-based method with no pretrained weights. The progression of the accuracies of each test run for each method can be found in figure 6.11. The graphs clearly show that even for this dataset, the GRU-based methods achieve a higher accuracy than the frame-based method. As seen in the previous section, the pretraining also boosts the classification performance on this dataset.

This assumption is confirmed by table 6.5. The pretrained history-based workflow segmentation achieves higher values for precision, recall and accuracy than the version without pretraining and the frame-based method.

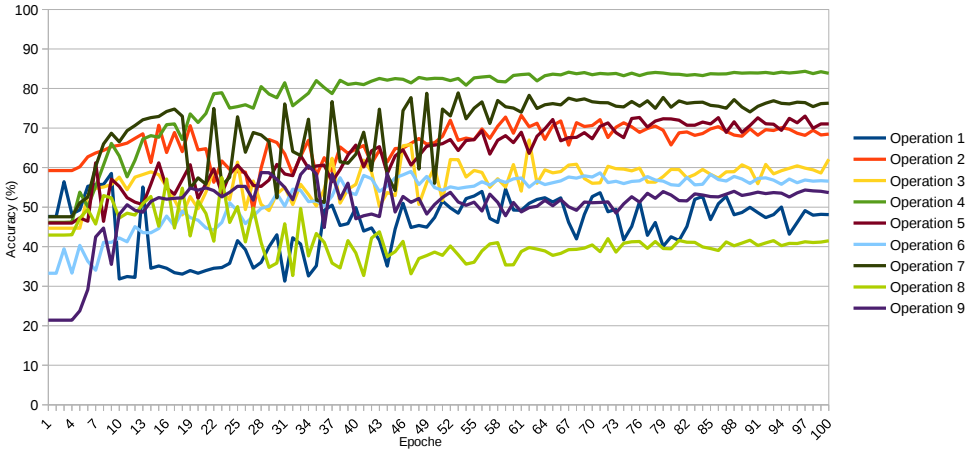
The phase-wise performance of the history-based workflow segmentation is listed in table 6.6. Phases 4 and 7 achieve the lowest performance. Phase 4 is often so confused



(a) Frame-based



(b) History-based



(c) History-based without pretraining

Figure 6.11: Development of the accuracies of the phase detection for each operation and for each network during the leave-one-surgery-out evaluation on the colorectal dataset [BWK+17].

	Precision	Recall	Accuracy
Frame-based	32.0% ± 9.6%	29.7% ± 8.5%	50.4% ± 9.0%
History-based	68.2% ± 15.0%	52.6% ± 9.8%	67.2% ± 13.1 %
History-based without pretraining	53.9% ± 6.7%	43.6% ± 11.2%	62.8% ± 14.1%

Table 6.5: Comparison of the results of our proposed methods on the colorectal dataset from the University of Heidelberg.

	Precision	Recall	Accuracy
P1	88.1%±28.0%	85.8%±30.2%	99.5%±0.7%
P2	72.9%±24.1%	67.0%±33.3%	97.8%±1.4%
P3	72.7%±15.7%	74.8%±31.2%	83.4%±5.5%
P4	58.7%±43.9%	9.3%±17.1%	91.4%±5.8%
P5	76.7%±14.1%	80.3%±18.7%	80.8%±9.7%
P6	57.7%±31.0%	37.0%±37.0%	88.2%±10.2%
P7	55.7%±52.5%	11.5%±33.2%	97.4%±2.3%
P8	62.9%±45.1%	51.3%±42.3%	96.8%±3.5%

Table 6.6: Performance of history-based workflow segmentation on the colorectal dataset broken down into the different phases [BWK+17].

with phase 3, which precedes it and phase 5, which generally follows it. Phase 7 is a rather short phase, meaning only a small number of examples were available for training and visually similar to phase 5 with which it is often confused.

6.2.3 Discussion

In this section, we showed that a pretrained CNN as presented in section 6.1 can be adapted to solve surgical workflow segmentation. We evaluated the method on two datasets: a publicly available dataset of annotated cholecystectomies and a dataset of annotated colorectal surgeries. The evaluation showed that on both datasets a GRU-based approach outperforms a plain feed-forward network. A combination of the GRU-based approach and the pretrained model further increased performance, supporting our hypothesis that the previously described pretraining method would be beneficial.

Our proposed history-based workflow segmentation method, which combines pretraining and a GRU, performs comparable to the state of the art on the public dataset, while the feedforward and the non-pretrained method perform significantly lower. The method outperforms the method of Dergachyova et al. [DBH+16] and the purely CNN-based EndoNet [TSM+16], which did not include temporal information. A second version of EndoNet incorporates temporal information using a hierarchical hidden markov model and thereby achieves a higher performance than the history-based approach.

When comparing the performance of the two methods, one has to take into consideration that EndoNet used 40 further annotated cholecystectomies for training.

Laparoscopic cholecystectomy is a very standardized and simple surgery. To demonstrate that our method can be applied to longer, more complex laparoscopic surgeries, we performed another evaluation on a dataset consisting of colorectal surgeries, which are generally more complex in terms of involved anatomy, vessel resection and required level of surgical expertise. The resulting performance was lower than on the cholecystectomy dataset. This can probably be attributed to the large variance in the dataset, which should be expected with long and complex surgeries. The order of certain phases varied partially between different surgeries, e.g. in operation 7 phase 7 was not performed and in most operations, phase 3 was interrupted multiple times by other phases. This can be partially attributed to the fact that the surgeries were performed by different surgeons, as different surgeons have different preferences when it comes to the order of certain parts of the procedure. The endoscopic optic and the tools used also varied between surgeries. We arrive at the conclusion that in order to mirror this variance, more training examples are required to increase performance. Nevertheless, we were able to show that our pretrained CNN achieves a higher performance on this dataset than a randomly initialized CNN.

6.3 Temporal Context Learning for Procedure Duration Prediction

As time in the operating room (OR) and the time of the operating staff are cost intensive hospital resources and have to be allocated precisely, surgeries have to be accurately planned. For this, the OR schedulers have to be constantly kept in the loop of the progress of ongoing surgeries. An alternative would be to use the endoscopic video stream in combination with machine learning and computer vision techniques to extract information on the progress of surgery.

While surgical workflow segmentations methods can be used to approximate the duration of surgical procedures, training such methods generally require a sufficient amount of labeled examples as training input. Furthermore, seeing that phase models are generally specified to a certain type of surgery, multiple detectors would need to be trained. Therefore, using a phase-based method as a general solution to determine the remaining duration of surgeries would require an unfeasible large amount of labeled training data.

As an alternative, we present in this section a method that extends the pretrained CNN from section 6.1 into a CNN capable of predicting the procedure duration of laparoscopic surgeries based on endoscopic video frames. This does not require any additional labels, as the duration of an surgery can be directly determined from the length of the given video. The work presented in this section was submitted for publication at time of writing [BKW+17].

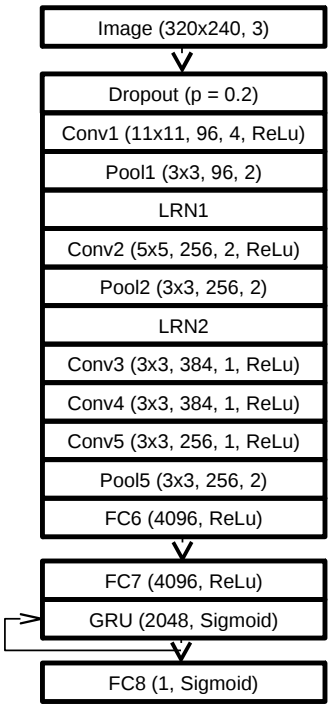


Figure 6.12: CNN topologies for predicting surgery duration from images [BKW+17].

6.3.1 Methods for progress prediction

A requirement for predicting the remaining duration of laparoscopic surgeries is information regarding the current state of the surgery. Laparoscopic surgeries are performed via endoscopic video stream, making it an ideal source of information. To extract this information from the video stream, we propose a CNN that makes predicting surgical duration possible. Seeing as the topology of the proposed history-based workflow segmentation method in section 6.2 performs adequately, we adapted the proposed CNN slightly so that instead of surgical phases, surgical procedure duration is predicted (see figure 6.12).

As input for the CNN, we sampled videos at a rate of one frame per second and downsampled each image to a resolution of 320×240 pixels. This reduction was performed to reduce data size and training time. We assigned each frame a number between 0 and 1 as label, i.e. the label of frame i from a video consisting of N frames is $y_i = \frac{i}{N}$.

Since the layers preceding fc7 are pretrained, we only optimize the weights of the newly added layers. As optimizer, we use Adam[KB14] with an initial learning rate of 10^{-6}. The cost function during optimization is the absolute error between the predicted progress and the actual progress. We trained for 40 iterations with batches of size 256.

6.3.2 Evaluation of the progress prediction

The basis of our evaluation is a large dataset, containing recording of 79 different laparoscopic surgeries of 20 different procedure types, providing a diverse range in data. The procedures were all recorded in the same OR using the integrated operating room system OR1™ (Karl Storz GmbH & Co KG, Tuttlingen, Germany). The average procedure length in the dataset is 99.7 minutes. The dataset used for pretraining in section 6.1 did not contain any of these 79 videos.

To evaluate the proposed method, we divided the dataset randomly into four sets of almost equal size (three sets of 20 procedures and one set of 19) and then performed four leave-one-set-out evaluations. Furthermore, to show the advantage of the pretrained feature, we evaluated a version of the proposed CNN without pretrained weights. In addition to the absolute error between prediction and label progresses during training (see figure 6.13), we compute the duration prediction $\tilde{N}_i$ at each frame i:

$$\tilde{N}_i = \frac{i}{y_i} \tag{6.18}$$

Here $y_i \in [0,1]$ the predicted progress of the procedure. With $\tilde{N}_i$, we can compute the duration prediction error e_i relative to the length N of of each procedure:

$$e_i = \frac{|\tilde{N}_i - N|}{N} \tag{6.19}$$

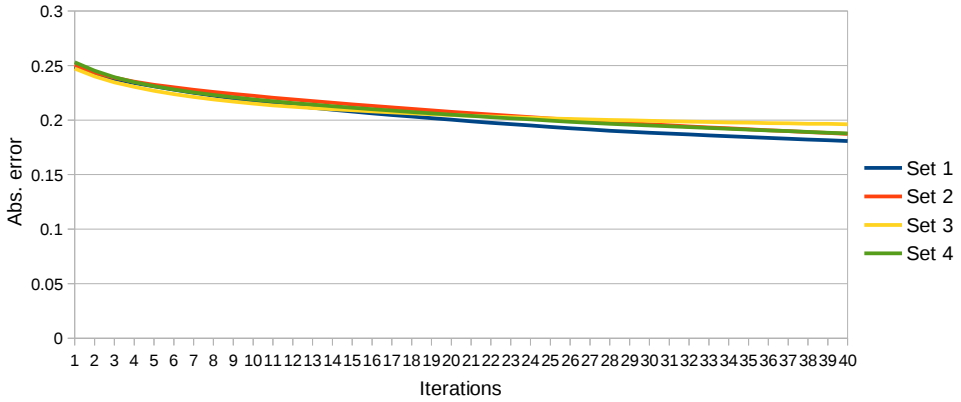
This measure gives a more appropriate impression than the difference between prediction and label on how well the proposed CNN can predict procedure durations. For each of the four sets we provide the average duration prediction error and, to measure how the error progresses during the course of a procedure, we provide the average error during each quarter of the surgery (Q1-Q4) for the proposed method and the version without pretraining (see tables 6.6(b) and 6.6(c)). As baseline, we provide the duration prediction errors that would occur if the average procedure duration were used as value for $\tilde{N}_i$ (see table 6.6(a)).

6.3.3 Discussion

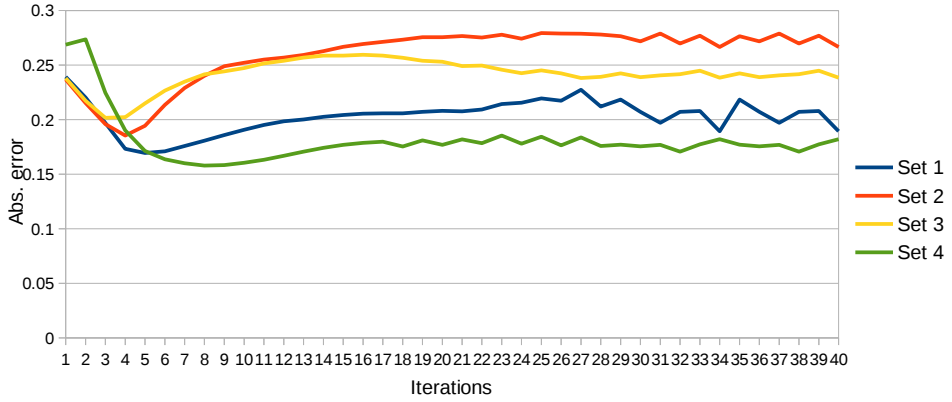
In this section, we presented an extension to the previously pretrained CNN from section 6.1 that makes it possible to predict the remaining duration of surgical procedures. The proposed method outperforms a baseline computed from average procedure duration. Furthermore, we showed that the proposed pretraining increases performance.

The evaluation shows that the presented methods currently produce larger than average errors on procedures with shorter length (shorter than 20min). We assume this is due to a lack of training data as our dataset comprises mostly longer operations. incorporating more data from other surgical devices, such as endoflator, heart rate, blood pressure and drug doses, would provide valuable insights and increase performance further.

Currently, the proposed method does not take the procedure type explicitly into consideration. As the general procedure type is usually known beforehand, this information



(a) With pretraining



(b) Without pretraining

Figure 6.13: Absolute error progression during training for each test set [BKW+17].

could be used to either train type specified CNNs, using the presented network as starting point, or it could be incorporated directly into the CNN. These approaches though would require more training examples per type as our dataset currently provides.

(a) Baseline

	Set 1	Set 2	Set 3	Set 4	Overall
Mean	64.3%	147.2%	126.7%	51.6%	97.9%

(b) With pretraining

	Set 1	Set 2	Set 3	Set 4	Overall
Q1	55.8%±20.4%	54.4%±22.4%	57.3%±19.4%	64.8%±16.5%	57.9%±19.7%
Q2	24.7%±15.5%	39.0%±20.3%	32.2%±16.4%	31.3%±13.3%	31.9%±16.4%
Q3	31.2%±19.6%	48.9%±23.5%	34.6%±19.0%	23.0%±14.3%	34.5%±19.1%
Q4	67.4%±30.4%	79.5%±31.3%	61.5%±27.0%	40.4%±19.8%	62.1%±27.1%
Mean	44.8%±32.0%	55.5%±37.0%	46.4%±31.6%	39.7%±26.6%	46.6%±31.8%

(c) Without pretraining

	Set 1	Set 2	Set 3	Set 4	Overall
Q1	87.6%±33.5%	77.9%±19.4%	81.4%±22.5%	74.0%±18.5%	87.6%±35.5%
Q2	180.9%±61.6%	223.5%±31.4%	226.3%±27.8%	191.4%±24.1%	180.9%±61.6%
Q3	212.0%±84.6%	240.7%±30.8%	206.2%±26.9%	188.0%±34.5%	212.0%±84.6%
Q4	58.1%±25.6%	81.4%±17.6%	85.0%±18.9%	81.6%±21.2%	58.1%±25.6%
Mean	134.7%±51.8%	155.9%±24.8%	149.7%±24.0%	133.7%±24.6%	134.7%±51.8%

Table 6.7: The average duration prediction errors for all four sets.
subreftab:workflow:progress:results:base shows the average baseline error. (b) and (c) show the average error for the proposed method with and without pretraining on the overall procedure and the quarters.

7 Conclusion

This thesis proposed and examined novel methods for extracting information required by a context-aware computer-assisted surgery system from the laparoscopic video stream. The following chapter provides a summary of the most significant contributions and the achieved results. Furthermore, an outlook into possible extension and future focuses of research is given.

7.1 Summary and Discussion

The goal of this thesis was to introduce novel and efficient methods for using image analysis for processing the endoscopic video feed to provide necessary information for context-aware computer-assisted laparoscopy. Seeing that the endoscopic view is a rich source of information and, furthermore, always available during surgery, it became an obvious choice for acquiring information on the current state of the operation.

This insight led to the conclusion that methods for characterizing the surgical scene from image and video data are required. A surgical scene can generally be characterized by the objects contained in it and their relations to each other. The focus was to develop methods that extract semantic, quantitative and workflow information in real-time. The presented methods were thoroughly evaluated in regards to accuracy, robustness and run-time. The following aspects were examined in this work:

- **Semantic image analysis** methods aim to provide information about the locations of structures in the endoscopic environment. To guarantee a timely response and current information, important structures have to be located quickly. This work presented novel methods for semantic image analysis that segment relevant structures, such as instruments and organs, in real-time using simple and quickly computable features. Furthermore, different granularities of segmentation, mainly pixel-based and superpixel-based, were explored and compared, with a focus on laparoscopic instruments and the intestine.
- Knowledge about the location and type of surgical instruments is an imperative **semantic image analysis** task for computer-assisted surgery, as the instruments provide an insight into the intentions of the surgeon and also play a role for many assistance functions. The located instrument regions with the previously introduced segmentation methods did not always describe an entire instrument, due to occlusions on the instrument shaft or false positives, requiring a post-processing method to fuse some regions and remove others. This method allowed instruments to be located in real-time. Since the assistance need often differs depending on the type of instrument, a new method that identified previously detected instruments was introduced and evaluated. To increase robustness

for both detection and identification, a tracking method for propagating successfully detected and identified instruments into future frames was introduced. It was shown that tracking was able to reduce errors due to challenges specific to laparoscopic surgery, such as specularities, smoke, blood and overlaps.

- To train methods for **semantic image analysis** requires annotated training examples. Annotating images is a time-consuming task that often requires experts. It was found that often knowledge about what objects are located in a scene, and not their position, is sufficient for certain assistance tasks. This thesis therefore explored different granularity levels of image content classification. Explicitly, methods for providing pixel-based, superpixel-based and image-based labels were presented and evaluated. Each type of annotation was shown to have its pros and cons, though in consideration of real-time surgical assistance, the pixel-based approach was shown to be the method of choice due to its short run-time.
- In the area of **quantitative image analysis**, a new method for providing intraoperative measurements in real-time was developed and evaluated. The method combined previously presented methods for semantic image analysis with a method for 3D reconstruction. The resulting measurement tool allowed surgeons to perform 3D distance measurements on organs. This measurement system was first thoroughly evaluated in the laboratory, before being put into test in a clinical environment at the University of Heidelberg, where clinicians successfully tested the system in phantom, ex- and in-vivo porcine trails. Furthermore, the system was successfully utilized during a first in-human study.
- To reduce the amount of annotated data required to train methods for **workflow analysis**, a novel method for pretraining using unlabeled data was introduced. A task that required a convolutional neural network to sort laparoscopic video frames in a temporal order was conceived. To solve this task, the network had to learn to extract image features relevant to surgical workflow. This pretrained network was then evaluated on two surgical workflow related problems.
- The most common problem in surgical **workflow analysis** is phase segmentation. Surgical phases generally describe the treatment of a certain structure. To evaluate the previously pretrained convolutional neural network, it was modified to incorporate temporal information from previous frames. This extension was then tested on the task of phase segmentation of two different types of laparoscopic surgeries. As phases are often exclusive to a certain type of operation, a different method for predicting the progress of surgery from different types of operations using unlabeled data was proposed and evaluated. Both evaluations showed that the proposed pretraining was able to increase performance in comparison to untrained models.

One large contribution of this work was to go beyond the laboratory and directly into the surgical environment. Due to this transition, it was possible to show that the proposed quantitative image analysis method, which builds onto described semantic image analysis methods, could be successfully operated by medical personal.

7.2 Outlook

The research presented in this thesis shows the potential of endoscopic image analysis and can be extended and complemented in many ways. While real-time performance was a key aspect in this work, there is still potential for reducing run-time. The presented superpixel-based segmentation for example is currently not suited for GPU parallelization, which could be the focus of future work. The run-time of other methods could be enhanced by utilizing dedicated hardware, such as FPGAs.

The accuracy of the semantic image analysis methods could be further enhanced by combining them with the presented workflow analysis methods and background knowledge. Knowing the phase of an operation, for example, limits the type of instruments and organs an endoscopic scene might contain, and thereby influences class probabilities. Tracking instruments could be instrumental to enable surgical action recognition, which is a requirement for certain workflow analysis methods.

For extracting quantitative information from the surgical scene, the method presented here relies on a stereo endoscope. Since the presented measurement tool is flexible enough to incorporate other methods for 3D reconstruction, further evaluations should be performed, if and when these methods become available in the operating room. Some methods, such as time of flight and structured light, should perform better in a more homogeneous environment and could potentially increase performance and robustness. The presented measurement tool could further be applied to other tasks, such as measuring hernia sizes, given the right training data. A further modification would be to extend the tool for measurements of areas, such as tumor size, or even volume measurements, such as the size and depth of wounds.

In addition to surgical phase segmentation, the pretrained network for workflow analysis could be used for further tasks in laparoscopy. One application could be other segmentation tasks, such as action detection or event recognition. Furthermore, the output of an intermediate layer could be used as a reduced representation of a laparoscopic frame for allowing indexing of surgical videos.

To analyze the surgical workflow, this work focused on the endoscopic image stream, which is only one of many data streams in the operating room. For example, data available to the anesthetist, such as heart rate, blood pressure and drug doses, would provide valuable insights into surgical workflow. Integrated operating rooms, such as the OR1 from Karl Storz, make it possible to access data streams from surgical devices such as cameras, thermoflator, lights, etc. during surgery. As these operating rooms become more prevalent, data collected from them could be integrated into the methods presented here for workflow analysis tasks. The upside to neural networks is that further inputs can often be easily integrated.

Looking beyond the surgical environment, some of the methods presented here could be applied to other problems. The method for endoscopic measurements could be modified for other fields that make usage of endoscopes, such as archeology [Bec15]

or some industrial fields [1]. The methods presented for workflow analysis could be retrained and evaluated on other video segmentation tasks, such as scene detection [TBS14].

[1] Industrial applications of endoscopy: https://www.karlstorz.com/at/en/industrial-group.htm (accessed: July 2, 2018)

A Evaluation Metrics

Many metrics exist to judge the performance of a classification method, this section will give an overview on the metrics used in this work. Here metrics for binary classification are given:

- Number of true positives (TP): The amount of samples that were correctly classified as positives.
- Number of false positives (FP): The amount of samples that were incorrectly classified as positives.
- Number of true negatives (TN): The amount of samples that were correctly classified as negatives.
- Number of false negatives (FN): The amount of samples that were incorrectly classified as negatives.
- Accuracy: The percentage of samples assigned to the correct class:

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

- Precision: The ratio of samples correctly classified as positives to all samples classified as positives:

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

- Recall: The ratio of samples correctly classified as positives to all positive samples:

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

- DICE coefficient or F1 score: The harmonic mean of precision and recall:

$$\frac{2 \cdot P \cdot R}{P + R}$$

The following definitions were used to evaluate the performance of the different workflow segmentation methods:

- Precision: Percentage of frames correctly attributed to a certain phase
- Recall: Percentage of frames attributed to a certain phase that are correctly attributed to that phase
- Accuracy: Overall percentage of frames attributed to the correct phase

B Overview of Datasets

This section will provide an overview of the datasets used in this thesis.

B.1 Datasets for Instrument Segmentation

Name	Origin	Size	Type of annotiation
Crowd	2 adrenalectomies 3 pancreas resections	5×20 frames 640×480px	Pixel-wise via crowd sourcing
EndoVisRigid	6 colorectal laparoscopies	6×50 frames 640×480px	Pixel-wise via crowd sourcing
EndoVisRobotic	6 videos of ex-vivo organs	6×1500 frames 720×576px	Pixel-wise via robot kinematics

B.2 Datasets for Instrument Identification

Name	Origin	Size	Type of annotiation
TypeManual	2 adrenalectomies 3 pancreas resections	5×50 frames	Box around instruments Instrument type
TypeAutomatic	2 adrenalectomies 3 pancreas resections	5×100 frames	Box around instruments Instrument type

B.3 Datasets for Bowel Measurement

Name	Origin	Size	Type of annotiation
Phantom 5cm	10 bowel s measurement	10×12 iterations	Pins after every iteration Distances manually measured
Phantom 10cm	4 bowel measurements	4×7 iterations	Pins after every iteration Distances manually measured
Porcine	4 bowel measurements	4×6 iterations	Coagulation after every iteration Distances manually measured

B.4 Datasets for Workflow Analysis

Name	Origin	Size	Type of annotiation
Pretraining	324 laparoscopies 30 types	various lengths 1Hz 320 × 240px	None
EndoVisWorkflow	7 cholecystectomies	various lengths 1Hz 320 × 240px	Phase annotations for every frame
ColorectalWorkflow	6 proctocolectomies 3 rectal resections	various lengths 1Hz 320 × 240px	Phase annotations for every frame
Progress prediction	79 laparoscopies 20 types	various lengths 1Hz 320 × 240px	None

C Software System MediAssist

MediAssist, a framework for applications for computer-assisted surgery, was developed at the chair of Prof. Dillmann outside of this work [SSF+07] and extended during the course of this work [BRS+11]. MediAssist is written in C++ and builds on the framework provided by the Image-Guided Surgery Toolkit (IGSTK)[GIA+06]. Additionally, it utilizes the Insight Segmentation and Registration Toolkit (ITK)[1] for medical image processing, the Visualization Toolkit (VTK)[2] for displaying 2D and 3D images and scene, and the OpenCV library [Bra00] for 2D camera image processing.

To achieve a high degree of customizability, the components of our system dealing with the acquisition and the subsequent processing of sensor data are designed to resemble highly modular framework component-blocks with a common interface. This interface enables them to interconnect with each other so that information can be passed downstream. The blocks also have the capability to perform their most time-consuming operations that are not connected to the in- and output of data in a separate thread, allowing them to run in parallel by making use of modern multi-core CPUs. We can distinguish between two types of blocks:

- Output only (source blocks): These blocks only provide information, i.e. sensors.
- Input and output (also called pipelines): Here the input data from one or more blocks is collected, fused and modified, e.g. through an ITK mini-pipeline, and is then passed on.

All the blocks in MediAssist are directly or indirectly descended from the class **DataObject**, which provides an interface that allows data (e.g. images, measurments, ...) to be received, processed and sent out. When connecting two objects of class **DataObject**, the interface checks which inputs and outputs the objects support and matches them. Sensors, such as cameras, can be integrated by wrapping the acquisition process in a class inheriting from **DataObject**. A **CameraObject** class exists that extends **DataObject** with an interface for passing along images. The **StereoCameraObject** extends this to stereo images.

To process sensor data, the class **PipelineObject** is used. It inherits directly from **DataObject** and provides an interface that allows data to be received, processed in a separate thread and to pass the processed information on if required. The separate thread makes it possible to simultaneously perform costly computations. **ImagePipelineObject** is an extension that allows image processing.

[1] https://itk.org/
[2] http://www.vtk.org/

The UML class diagram in figure C.1 shows the inheritances in MediAssist based on the organ measurement system. The measurement system implements the interface provided by **StereoCameraObject** with the class **StereoEndoscope**, which allows streaming of the image data of a stereo endoscope. The stereo images are passed to the **HRMPipeline**, which implements the interface of the **ImagePipelineObject**. The **HRMPipeline** performs a correspondence analysis on the stereo images and then passes the disparity maps and the original images to the **MeasurementPipeline**. The **MeasurementPipeline** class uses a pipeline of connected ITK filters to detect organs and instruments, which are then combined with the 3D data to allow 3D measurements.

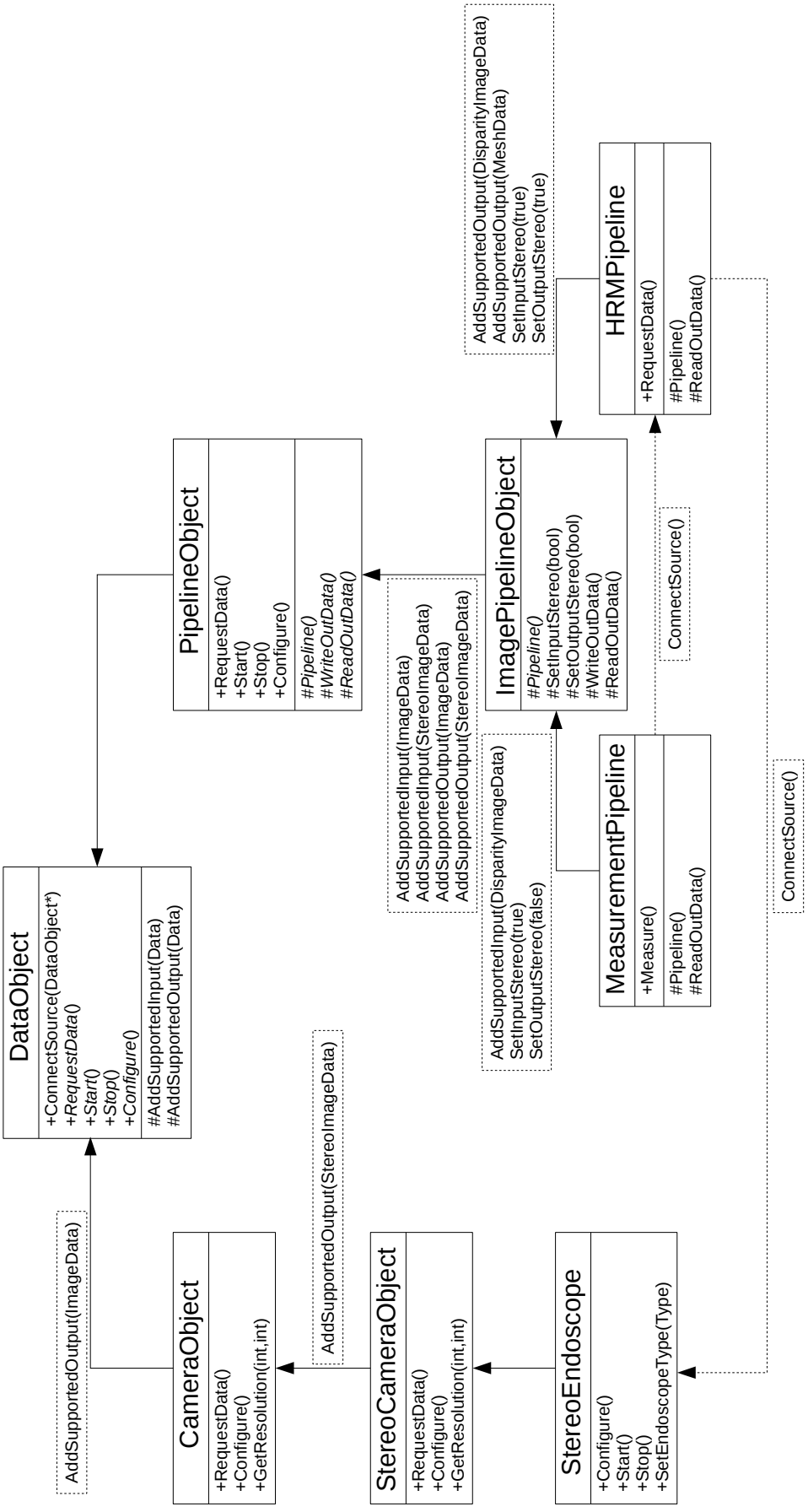


Figure C.1: UML class diagram describing the inheritance hierarchy and class connections in MediAssist, based on the example of the organ measurement system.

List of Figures

List of Tables

Bibliography

[AGD08] Pedram Azad, Tilo Gockel, and Rüdiger Dillmann. Computer vision: Principles and practice. 2008.

[AHP06] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen. Face description with local binary patterns: Application to face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(12):2037–2041, 2006.

[AHR13] Anwar Abdalbari, Xishi Huang, and Jing Ren. Endoscopy-mr image fusion for image guided procedures. *Journal of Biomedical Imaging*, 2013:23:23–23:23, January 2013.

[AKS04] Nicole Atzpadin, Peter Kauff, and Oliver Schreer. Stereo analysis by hybrid recursive matching for real-time immersive video conferencing. *Circuits and Systems for Video Technology, IEEE Transactions on*, 14(3):321–334, 2004.

[AOT+13] Max Allan, Sébastien Ourselin, Steve Thompson, David J Hawkes, John Kelly, and Danail Stoyanov. Toward detection and localization of instruments in minimally invasive surgery. *IEEE Transactions on Biomedical Engineering*, 60(4):1050–1058, 2013.

[ARJK+12] Maurice E Arregui, J Robert Jr, Namir Katkhouda, J Barry McKernan, Harry Reich, et al. *Principles of laparoscopic surgery: basic and advanced techniques*. Springer Science & Business Media, 2012.

[ASS+10] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels. Technical report, 2010.

[AV07] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[BASJ17] David Bouget, Max Allan, Danail Stoyanov, and Pierre Jannin. Vision-based and marker-less surgical tool detection and tracking: a review of the literature. *Medical Image Analysis*, 35:633 – 654, 2017.

[Bec15] Ronald G Beckett. Application and limitations of endoscopy in anthropological and archaeological research. *The Anatomical Record*, 298(6):1125–1134, 2015.

[BFN10] Tobias Blum, Hubertus Feußner, and Nassir Navab. *Modeling and Segmentation of Surgical Workflow from Laparoscopic Video*, pages 400–407. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[BGW+16] Sebastian Bodenstedt, Jochen Görtler, Martin Wagner, Hannes Kenngott, Beat Peter Müller-Stich, Rüdiger Dillmann, and Stefanie Speidel. Superpixel-based structure classification for laparoscopic surgery. *Proc. SPIE*, 9786:978618–978618–6, 2016.

[BJD11] L. Bouarfa, P.P. Jonker, and J. Dankelman. Discovery of high-level tasks in the operating room. *Journal of Biomedical Informatics*, 44(3):455 – 462, 2011. Biomedical Complexity and Error.

[BK08] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.

[BKB82] Toby Berk, Arie Kaufman, and Lee Brownston. A human factors study of color notation systems for computer graphics. *Commun. ACM*, 25(8):547–550, August 1982.

[BKW+17] Sebastian Bodenstedt, Sabine Kugler, Martin Wagner, Lars Mündermann, Hannes Kenngott, Beat Müller-Stich, Rüdiger. Dillmann, and Stefanie Speidel. Predicting procedure duration of laparoscopic surgeries using endoscopic video and surgical device data. *Submitted to MICCAI 2017*, 2017.

[BNSD17] Sylvain Bernhardt, Stéphane A. Nicolau, Luc Soler, and Christophe Doignon. The status of augmented reality in laparoscopic surgery as of 2016. *Medical Image Analysis*, 37:66 – 90, 2017.

[Bod12] Sebastian Bodenstedt. *Diploma thesis: Semi-Automation using Prior Models for Human-Machine Cooperative Tele-Operation*. IAR Dillmann, Karlsruhe Institute of Technology, 2012.

[BOK+15] Sebastian Bodenstedt, Antonia Ohnemus, Darko Katic, Anna-Laura Wekerle, Martin Wagner, Hannes Kenngott, Beat Müller-Stich, Rüdiger Dillmann, and Stefanie Speidel. Real-time image-based instrument classification for laparoscopic surgery. In *MICCAI Workshop on Modeling and Monitoring of Computer Assisted Interventions (M2CAI)*, 2015.

[Bou01] Jean-Yves Bouguet. Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. *Intel Corporation*, 5(1-10):4, 2001.

[Bra00] G. Bradski. Opencv library. *Dr. Dobb's Journal of Software Tools*, 2000.

[Bre01] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[Bro66] D.C. Brown. Decentering distortion of lenses. *Photogrammetric Engineering*, 7:444–462, 1966.

[BRS+11] Sebastian Bodenstedt, Sebastian Röhl, Stefan Suwelack, Darko Katic, Hannes Kenngott, Beat Müller-Stich, Rüdiger Dillmann, and Stefanie Speidel. A flexible framework for multiple sensor integration into a context-aware cas-system. In *Computer Assisted Radiology and Surgery (CARS)*, pages 0–0, 2011.

[BSH+13] Sebastian Bauer, Alexander Seitel, Hannes Hofmann, Tobias Blum, Jakob Wasza, Michael Balda, Hans-Peter Meinzer, Nassir Navab, Joachim Hornegger, and Lena Maier-Hein. *Real-Time Range Imaging in Health Care: A Survey*, pages 228–254. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[BTS+17] J. Bernal, N. Tajbakhsh, F. J. Sanchez, B. J. Matuszewski, H. Chen, L. Yu, Q. Angermann, O. Romain, B. Rustad, I. Balasingham, K. Pogorelov, S. Choi, Q. Debard, L. Maier-Hein, S. Speidel, D. Stoyanov, P. Brandao, H. Cordova, C. Sanchez-Montes, S. R. Gurudu, G. Fernandez-Esparrach, X. Dray, J. Liang, and A. Histace. Comparative validation of polyp detection methods in video colonoscopy: Results from the miccai 2015 endoscopic vision challenge. *IEEE Transactions on Medical Imaging*, PP(99):1–1, 2017.

[BTVG06] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[BWK+17] Sebastian Bodenstedt, Martin Wagner, Darko Katić, Patrick Mietkowski, Benjamin Mayer, Hannes Kenngott, Beat Müller-Stich, Rüdiger. Dillmann, and Stefanie Speidel. Unsupervised temporal context learning using convolutional neural networks for laparoscopic workflow analysis. *ArXiv e-prints*, February 2017.

[BWM+16] Sebastian Bodenstedt, Martin Wagner, Benjamin Mayer, Katherine Stemmer, Hannes Kenngott, Beat Müller-Stich, Rüdiger Dillmann, and Stefanie Speidel. Image-based laparoscopic bowel measurement. *International Journal of Computer Assisted Radiology and Surgery*, 11(3):407–419, 2016.

[Can86] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.

[CB12] Toby Collins and Adrien Bartoli. *Towards Live Monocular 3D Laparoscopy Using Shading and Specularity Information*, pages 11–21. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[CBMC14] Ajad Chhatkuli, Adrien Bartoli, Abed Malti, and Toby Collins. Live image parsing in uterine laparoscopy. In *Biomedical Imaging (ISBI), 2014 IEEE 11th International Symposium on*, pages 1263–1266. IEEE, 2014.

[CGCB14] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[CS13] Antonio Criminisi and Jamie Shotton. *Decision forests for computer vision and medical image analysis*. Springer Science & Business Media, 2013.

[CSD+13] Ping-Lin Chang, Danail Stoyanov, Andrew J Davison, et al. Real-time dense stereo reconstruction using convex optimisation with a cost-volume for image-guided robotic surgery. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 42–49. Springer, 2013.

[CSMH+11] Neil T. Clancy, Danail Stoyanov, Lena Maier-Hein, Anja Groch, Guang-Zhong Yang, and Daniel S. Elson. Spectrally encoded fiber-based structured lighting probe for intraoperative 3d imaging. *Biomed. Opt. Express*, 2(11):3119–3128, Nov 2011.

[CVMG+14] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[DBH+16] Olga Dergachyova, David Bouget, Arnaud Huaulmé, Xavier Morandi, and Pierre Jannin. Automatic data-driven real-time segmentation and recognition of surgical workflow. *International Journal of Computer Assisted Radiology and Surgery*, 11(6):1081–1089, 2016.

[De 78] Carl De boor. *A practical guide to splines*. Springer Verlag., 1978.

[DFI+15] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2758–2766, 2015.

[DGE15] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. In *International Conference on Computer Vision (ICCV)*, 2015.

[DHS01] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern classification*. Wiley, New York, 2. ed. edition, 2001.

[Dij59] Edsger W Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[DLM+16] Robert DiPietro, Colin Lea, Anand Malpani, Narges Ahmidi, S. Swaroop Vedula, Gyusung I. Lee, Mija R. Lee, and Gregory D. Hager. Recognizing surgical activities with recurrent neural networks. *CoRR*, abs/1606.06329, 2016.

[DSR+15] Sander Dieleman, Jan Schlüter, Colin Raffel, Eben Olson, Søren Kaae Sønderby, Daniel Nouri, Daniel Maturana, Martin Thoma, Eric Battenberg, Jack Kelly, Jeffrey De Fauw, Michael Heilman, diogo149, Brian McFee, Hendrik Weideman, takacsg84, peterderivaz, Jon, instagibbs, Dr. Kashif Rasul, CongLiu, Britefury, and Jonas Degrave. Lasagne: First release., August 2015.

[EdlFR10] Robert Elfring, Matías de la Fuente, and Klaus Radermacher. Assessment of optical localizer accuracy for computer aided surgery systems. *Computer Aided Surgery*, 15(1-3):1–12, 2010. PMID: 20233129.

[EPY+15] Philip Edgcumbe, Philip Pratt, Guang-Zhong Yang, Christopher Nguan, and Robert Rohling. Pico lantern: Surface reconstruction and augmented reality in laparoscopic surgery using a pick-up laser projector. *Medical Image Analysis*, 25(1):95 – 102, 2015.

[FCSS09] Matthew Field, Duncan Clarke, Stephen Strup, and W Brent Seales. Stereo endoscopy as a 3-d measurement tool. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 5748–5751. IEEE, 2009.

[FDM+05] Gabor Fichtinger, Anton Deguet, Ken Masamune, Emese Balogh, Gregory S Fischer, Herve Mathieu, Russell H Taylor, S James Zinreich, and Laura M Fayad. Image overlay guidance for needle insertion in ct scanner. *IEEE transactions on biomedical engineering*, 52(8):1415–1424, 2005.

[Feu07] Marco Feuerstein. *Augmented Reality in Laparoscopic Surgery: New Concepts for Intraoperative Multimodal Imaging*. Technische Universität München, 2007.

[FH04] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59, 2004.

[FKG+97] Marvin P. Fried, Jonathan Kleefield, Harsha Gopal, Edward Reardon, Bryan T. Ho, and Frederick A. Kuhn. Image-guided endoscopic surgery: Results of accuracy and performance in a multicenter clinical study using an electromagnetic tracking system. *The Laryngoscope*, 107(5):594–601, 1997.

[FKG13] Jannik Fritsch, Tobias Kuehnl, and Andreas Geiger. A new performance measure and evaluation benchmark for road detection algorithms. In *International Conference on Intelligent Transportation Systems (ITSC)*, 2013.

[FMM+15] R. Furukawa, R. Masutani, D. Miyazaki, M. Baba, S. Hiura, M. Visentini-Scarzanella, H. Morinaga, H. Kawasaki, and R. Sagawa. 2-dof auto-calibration for a 3d endoscope system based on active stereo. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 7937–7941, Aug 2015.

[FRJ15] G. Forestier, L. Riffaud, and P. Jannin. Automatic phase prediction from low-level surgical activities. *International Journal of Computer Assisted Radiology and Surgery*, 10(6):833–841, 2015.

[FTV00] Andrea Fusiello, Emanuele Trucco, and Alessandro Verri. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12:16–22, 2000. 10.1007/s001380050120.

[GD05] Kristen Grauman and Trevor Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1458–1465. IEEE, 2005.

[GIA+06] Kevin Gary, Luis Ibanez, Stephen Aylward, David Gobbi, M Brian Blake, and Kevin Cleary. Igstk: an open source software toolkit for image-guided surgery. *Computer*, 39(4):46–53, 2006.

[GJ00] Gary Guthart and John Kenneth Salisbury Jr. The intuitive^{tm} telesurgery system: Overview and application. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 618–621, 2000.

[GLU12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[GMM+98] AG Gallagher, N McClure, J McGuigan, K Ritchie, and NP Sheehy. An ergonomic analysis of the fulcrum effect in the acquisition of endoscopic skills. *Endoscopy*, 30(07):617–620, 1998.

[Gör15] Jochen Görtler. *Master thesis: Superpixels for identifying structures in laparoscopic surgery*. IAR Dillmann, Karlsruhe Institute of Technology, 2015.

[GPM+16] Annetje CP Guédon, M Paalvast, FC Meeuwsen, David MJ Tax, AP van Dijke, LSGL Wauben, M van der Elst, Jenny Dankelman, and JJ van den Dobbelsteen. 'it is time to prepare the next patient'real-time prediction of procedure duration in laparoscopic cholecystectomies. *Journal of medical systems*, 40(12):271, 2016.

[GS03] Theo Gevers and Harro Stokman. Classifying color edges in video into shadow-geometry, highlight, or material transitions. *IEEE Transactions on Multimedia*, 5(2):237–243, 2003.

[HDB+11] Sebastian Haas, René Donner, Andreas Burner, Markus Holzer, and Georg Langs. Superpixel-based interest points for effective bags of visual words medical image retrieval. In *MICCAI International Workshop on Medical Content-Based Retrieval for Clinical Decision Support*, pages 58–68. Springer, 2011.

[Her64] Ewald Hering. Outlines of a theory of the light sense. 1964.

[Hir08] Heiko Hirschmüller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence*, 30(2):328–341, 2008.

[HPF+12] Mingxing Hu, Graeme Penney, Michael Figl, Philip Edwards, Fernando Bello, Roberto Casula, Daniel Rueckert, and David Hawkes. Reconstruction of a 3d surface from video that is robust to missing data and outliers: Application to minimally invasive surgery using stereo and mono endoscopes. *Medical Image Analysis*, 16(3):597 – 611, 2012. Computer Assisted Interventions.

[HS81] Berthold K.P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1):185 – 203, 1981.

[HS88] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[HS07] Heiko Hirschmüller and Daniel Scharstein. Evaluation of cost functions for stereo matching. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[HZ03] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[Jäh12] Bernd Jähne. *Digitale Bildverarbeitung*. Springer-Verlag, 2012.

[JG78] George H Joblove and Donald Greenberg. Color spaces for computer graphics. In *ACM siggraph computer graphics*, volume 12, pages 20–25. ACM, 1978.

[JWY+14] Shiyong Ji, Benzheng Wei, Zhen Yu, Gongping Yang, and Yilong Yin. A new multistage medical segmentation method based on superpixel and fuzzy clustering. *Computational and mathematical methods in medicine*, 2014, 2014.

[KB14] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[KBR+95] Richard Kim, Brian B Baggott, Suzanne Rose, Albert O Shar, Diana L Mallory, Steven S Lasky, Michael Kressloff, Lynn Y Faccenda, and James C Reynolds. Quantitative endoscopy: precise computerized measurement of metaplastic epithelial surface area in barrett's esophagus. *Gastroenterology*, 108(2):360–366, 1995.

[KCCO+11] D Katic, T Christian, G Castrillon-Oberndorfer, J Hoffmann, G Eggers, R Dillmann, and S Speidel. Calibration of see-through-goggles for a context-aware augmented reality system computer assisted radiology and surgery. 2011.

[Ken10] Hannes Götz Kenngott. Entwicklung und evaluation eines navigationssystems für die weichgewebechirurgie am beispiel der minimal invasiven, transhiatalen, telemanipulator-gestützten ösophagektomie. 2010.

[Kno12] Gary D Knott. *Interpolating cubic splines*, volume 18. Springer Science & Business Media, 2012.

[Kor13] Thomas Kornela. *Master thesis: Bildbasierte 3D Vermessung des Darms*. IAR Dillmann, Karlsruhe Institute of Technology, 2013.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[KWB+11] Darko Katic, Anna-Laura Wekerle, Sebastian Bodenstedt, Hannes Kenngott, Beat Müller-Stich, Rüdiger Dillmann, and Stefanie Speidel. Logic-based situation interpretation with real-valued sensor data for laparoscopic surgery. In *MICCAI Workshop on Modeling and Monitoring of Computer Assisted Interventions (M2CAI)*, 2011.

[KWG+13] Darko Katic, Anna-Laura Wekerle, Jochen Görtler, Patrick Spengler, Sebastian Bodenstedt, Sebastian Röhl, Stefan Suwelack, Hannes Götz Kenngott, Martin Wagner, Beat Peter Müller-Stich, Rüdiger Dillmann, and Stefanie Speidel. Context-aware augmented reality in laparoscopic surgery. *Computerized Medical Imaging and Graphics*, 37(2):174 – 182, 2013. Special Issue on Mixed Reality Guidance of Therapy - Towards Clinical Implementation.

[KWG+14] Darko Katić, Anna-Laura Wekerle, Fabian Gärtner, Hannes Kenngott, Beat Peter Müller-Stich, Rüdiger Dillmann, and Stefanie Speidel. *Knowledge-Driven Formalization of Laparoscopic Surgeries for Rule-Based Intraoperative Context-Aware Assistance*, pages 158–167. Springer International Publishing, Cham, 2014.

[LBD+89] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[LCDF10] Yaron Lipman, Xiaobai Chen, Ingrid Daubechies, and Thomas Funkhouser. Symmetry factored embedding and distance. In *ACM Transactions on Graphics (TOG)*, volume 29, page 103. ACM, 2010.

[LCE15] Jianyu Lin, Neil T. Clancy, and Daniel S. Elson. An endoscopic structured light system using multispectral detection. *International Journal of Computer Assisted Radiology and Surgery*, 10(12):1941–1950, 2015.

[LCRH16] Colin Lea, Joon Hyuck Choi, Austin Reiter, and Gregory Hager. Surgical phase recognition: From instrumented ors to hospitals around the world. *M2CAI 2016*, 2016.

[LCS+08] B. Lo, A. J. Chung, D. Stoyanov, G. Mylonas, and Guang-Zhong Yang. Real-time intra-operative 3d tissue deformation recovery. In *2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1387–1390, May 2008.

[LK+81] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981.

[LLG16] Vasileios Lahanas, Constantinos Loukas, and Evangelos Georgiou. A simple sensor calibration technique for estimating the 3d pose of endoscopic instruments. *Surgical Endoscopy*, 30(3):1198–1204, 2016.

[Low99] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[LRBJ12] F. Lalys, L. Riffaud, D. Bouget, and P. Jannin. A framework for the recognition of high-level surgical tasks from video images for cataract surgeries. *IEEE Transactions on Biomedical Engineering*, 59(4):966–976, April 2012.

[LRGCM+07] F. López-Rosales, Q. González-Contreras, L. J. Muro, M. M. Berber, H. T. Cid de León, O. V. Fernández, and R. R. Veana. Laparoscopic total proctocolectomy with ileal pouch anal anastomosis for ulcerative colitis and familial adenomatous polyposis: initial experience in mexico. *Surgical Endoscopy*, 21(12):2304–2307, 2007.

[M+67] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

[MADdM12] X. Maurice, C. Albitar, C. Doignon, and M. de Mathelin. A structured light-based laparoscope with real-time organs' surface reconstruction for minimally invasive surgery. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5769–5772, Aug 2012.

[May17] Benjamin Mayer. *Computer-assisted bowel measurement in minimal invasive surgery for quantitative laparoscopy (Publication in preparation)*. 2017.

[MB14] A. Malti and A. Bartoli. Combining conformal deformation and cook-torrance shading for 3-d reconstruction in laparoscopy. *IEEE Transactions on Biomedical Engineering*, 61(6):1684–1692, June 2014.

[MBC11] Abed Malti, Adrien Bartoli, and Toby Collins. Template-based conformal shape-from-motion from registered laparoscopic images. In *MIUA*, volume 1, page 6, 2011.

[MG15] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[MHGB+14] L Maier-Hein, A Groch, A Bartoli, S Bodenstedt, B Guillaume, P Chang, N Clancy, D Elson, S Haase, and D Stoyanov. Comparative validation of single-shot optical techniques for laparoscopic 3d surface reconstruction. 2014.

[MHMB+13] L Maier-Hein, P Mountney, A Bartoli, H Elhawary, D Elson, A Groch, A Kolb, Marcos Rodrigues, J Sorger, Suzanne Speidel, et al. Optical techniques for 3d surface reconstruction in computer-assisted laparoscopic surgery. *Medical image analysis*, 17(8):974–996, 2013.

[MHMK+14a] Lena Maier-Hein, Sven Mersmann, Daniel Kondermann, Sebastian Bodenstedt, Alexandro Sanchez, Christian Stock, Hannes Götz Kenngott, Mathias Eisenmann, and Stefanie Speidel. Can masses of non-experts train highly accurate image classifiers? In Polina Golland, Nobuhiko Hata, Christian Barillot, Joachim Hornegger, and

Robert Howe, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2014*, volume 8674 of *Lecture Notes in Computer Science*, pages 438–445. Springer International Publishing, 2014.

[MHMK+14b] Lena Maier-Hein, Sven Mersmann, Daniel Kondermann, Christian Stock, Hannes Götz Kenngott, Alexandro Sanchez, Martin Wagner, Anas Preukschas, Anna-Laura Wekerle, Stefanie Helfert, et al. Crowdsourcing for reference correspondence generation in endoscopic images. In *MICCAI (2)*, pages 349–356, 2014.

[MHT08] Atul K. Madan, Jason L. Harper, and David S. Tichansky. Techniques of laparoscopic gastric bypass: on-line survey of american society for bariatric surgery practicing surgeons. *Surgery for Obesity and Related Diseases*, 4(2):166 – 172, 2008.

[MIH11] Daniel J. Mirota, Masaru Ishii, and Gregory D. Hager. Vision-based navigation in image-guided interventions. *Annual Review of Biomedical Engineering*, 13(1):297–319, 2011. PMID: 21568713.

[MIH+16] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016.

[MMS+11] Sven Mersmann, Michael Müller, Alexander Seitel, Florian Arnegger, Ralf Tetzlaff, Julien Dinkel, Matthias Baumhauer, Bruno Schmied, Hans-Peter Meinzer, and Lena Maier-Hein. Time-of-flight camera technique for augmented reality in computer-assisted interventions. In *SPIE Medical Imaging*, pages 79642C–79642C. International Society for Optics and Photonics, 2011.

[NBGS08] John Nickolls, Ian Buck, Michael Garland, and Kevin Skadron. Scalable parallel programming with cuda. *Queue*, 6(2):40–53, March 2008.

[Nes83] Yuri Nesterov. A method for unconstrained convex minimization problem with the rate of convergence o(1/k2). *Doklady an SSSR*, 1983.

[Neu16] Angie Neumann. *Master thesis: Klassifizierung und Segmentierung laparoskopischer Aufnahmen durch Weakly Supervised Learning*. IAR Dillmann, Karlsruhe Institute of Technology, 2016.

[NH10] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814. Omnipress, 2010.

[NSM+06] Thomas Neumuth, Gero Strauß, Jürgen Meixensberger, Heinz U. Lemke, and Oliver Burgert. *Acquisition of Process Descriptions from Surgical Interventions*, pages 602–611. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[NVBR12] Navid Nourani-Vatani, P Borges, and Jonathan M Roberts. A study of feature extraction algorithms for optical flow tracking. In *Australasian Conference on Robotics and Automation*, 2012.

[Ohn15] Antonia Ohnemus. *Bachelor thesis: Bildbasierte Klassifikation von Instrumenten in der Laparoskopie*. IAR Dillmann, Karlsruhe Institute of Technology, 2015.

[OPH94] Timo Ojala, Matti Pietikainen, and David Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on*, number 1, pages 582–585, 1994.

[PBA+12] Nicolas Padoy, Tobias Blum, Seyed-Ahmad Ahmadi, Hubertus Feussner, Marie-Odile Berger, and Nassir Navab. Statistical modeling and recognition of surgical workflow. *Medical Image Analysis*, 16(3):632 – 641, 2012. Computer Assisted Interventions.

[Pea01] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[PEI+91] Jeffrey H Peters, E Christopher Ellison, Jeffery T Innes, Jonathan L Liss, Keith E Nichols, JACK M Lomano, Sheri R Roby, ME Front, and LC Carey. Safety and efficacy of laparoscopic cholecystectomy. a prospective analysis of 100 initial patients. *Annals of surgery*, 213(1):3, 1991.

[PF06] E. Prados and O. Faugeras. *Shape From Shading*, pages 375–388. Springer US, Boston, MA, 2006.

[PHS+09] Jochen Penne, Kurt Höller, Michael Stürmer, Thomas Schrauder, Armin Schneider, Rainer Engelbrecht, Hubertus Feußner, Bernhard Schmauss, and Joachim Hornegger. *Time-of-Flight 3-D Endoscopy*, pages 467–474. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[RAR04] Alan Robinson, Lyuba Alboul, and Marcos Rodrigues. Methods for indexing stripes in uncoded structured light scanning systems. 2004.

[RAZ12] A. Reiter, P. K. Allen, and T. Zhao. Learning features on robotic surgical tools. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 38–43, June 2012.

[RBS+12] Sebastian Röhl, Sebastian Bodenstedt, Stefan Suwelack, Hannes Kenngott, Beat P Müller-Stich, Rüdiger Dillmann, and Stefanie Speidel. Dense gpu-enhanced surface reconstruction from stereo endoscopic images for intraoperative registration. *Medical Physics*, 39:1632, 2012.

[RBS+15] Daniel Reichard, Sebastian Bodenstedt, Stefan Suwelack, Benjamin Mayer, Anas Preukschas, Martin Wagner, Hannes Kenngott, Beat

Müller-Stich, Rüdiger Dillmann, and Stefanie Speidel. Intraoperative on-the-fly organ-mosaicking for laparoscopic surgery. *Journal of Medical Imaging*, 2(4):045001, 2015.

[Röh13] Sebastian Röhl. *Intraoperative Modellierung und Registrierung für ein laparoskopisches Assistenzsystem*. KIT Scientific Publishing, Karlsruhe, 2013.

[RRKB11] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564–2571. IEEE, 2011.

[S+94] Jianbo Shi et al. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on*, pages 593–600. IEEE, 1994.

[SA85] Satoshi Suzuki and Keiichi Abe. Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, 30(1):32–46, 1985.

[SAR+12] Raphael Sznitman, Karim Ali, Rogério Richa, Russell H Taylor, Gregory D Hager, and Pascal Fua. Data-driven visual tracking in retinal microsurgery. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 568–575. Springer, 2012.

[SBF14] Raphael Sznitman, Carlos Becker, and Pascal Fua. Fast part-based classification for instrument detection in minimally invasive surgery. In *Proc. of MICCAI*. 2014.

[SBK+09] Stefanie Speidel, Julia Benzko, Sebastian Krappe, Gunther Sudra, Pedram Azad, Beat Peter Müller-Stich, Carsten Gutt, and Rüdiger Dillmann. Automatic classification of minimally invasive instruments based on endoscopic image sequences. *Proc. SPIE*, 7261:72610A–72610A–8, 2009.

[SF68] Irwin Sobel and Gary Feldman. A 3x3 isotropic gradient operator for image processing. *a talk at the Stanford Artificial Project in*, pages 271–272, 1968.

[SFSA12] Christoph Schmalz, Frank Forster, Anton Schick, and Elli Angelopoulou. An endoscopic 3d scanner based on structured light. *Medical Image Analysis*, 16(5):1063 – 1072, 2012.

[SHK+14] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *German Conference on Pattern Recognition*, pages 31–42. Springer, 2014.

[SIH+03] Philip R Schauer, Sayeed Ikramuddin, Giselle Hamad, George M Eid, Samer Mattar, Dan Cottam, Ramesh Ramanathan, and William Gourash. Laparoscopic gastric bypass surgery: current technique. *Journal of Laparoendoscopic & Advanced Surgical Techniques*, 13(4):229–239, 2003.

[SJC08] Jamie Shotton, Matthew Johnson, and Roberto Cipolla. Semantic texton forests for image categorization and segmentation. In *Computer vision and pattern recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[SOP+14] Ralf Stauder, Aslı Okur, Loïc Peter, Armin Schneider, Michael Kranzfelder, Hubertus Feussner, and Nassir Navab. *Random Forests for Phase Detection in Surgical Workflow Analysis*, pages 148–157. Springer International Publishing, Cham, 2014.

[SP07] Daniel Scharstein and Chris Pal. Learning conditional random fields for stereo. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[SRB+14] Stefan Suwelack, Sebastian Röhl, Sebastian Bodenstedt, Daniel Reichard, Rüdiger Dillmann, Thiago dos Santos, Lena Maier-Hein, Martin Wagner, Josephine Wünscher, Hannes Kenngott, Beat P. Müller, and Stefanie Speidel. Physics-based shape matching for intraoperative image guidance. *Medical Physics*, 41(11):111901–n/a, 2014. 111901.

[SS02] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 2002.

[SS03] Daniel Scharstein and Richard Szeliski. High-accuracy stereo depth maps using structured light. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2003.

[SSF+07] Gunther Sudra, Stefanie Speidel, Dominik Fritz, Beat Peter Müller-Stich, Carsten Gutt, and Rüdiger Dillmann. Mediassist: medical assistance for intraoperative skill transfer in minimally invasive surgery using augmented reality. In *Medical Imaging*, pages 65091O–65091O. International Society for Optics and Photonics, 2007.

[SSK+05] Vitaly Surazhsky, Tatiana Surazhsky, Danil Kirsanov, Steven J Gortler, and Hugues Hoppe. Fast exact and approximate geodesics on meshes. In *ACM transactions on graphics (TOG)*, volume 24, pages 553–560. ACM, 2005.

[SSPY10] Danail Stoyanov, Marco Visentini Scarzanella, Philip Pratt, and Guang-Zhong Yang. *Real-time stereo reconstruction in robotically assisted minimally invasive surgery*, pages 275–282. 2010.

[STDT08] Sebastian Schuon, Christian Theobalt, James Davis, and Sebastian Thrun. High-quality scanning using time-of-flight depth superresolution. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–7. IEEE, 2008.

[SZ09] Josef Sivic and Andrew Zisserman. Efficient visual search of videos cast as text retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 31(4):591–606, 2009.

[TAKW00] Theodore O Truitt, Roger A Adelman, Dan H Kelly, and J Paul Willging. Quantitative endoscopy: initial accuracy measurements. *Annals of Otology, Rhinology & Laryngology*, 109(2):128–132, 2000.

[TBS14] Makarand Tapaswi, Martin Bauml, and Rainer Stiefelhagen. Storygraphs: visualizing character interactions as a timeline. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 827–834, 2014.

[The16] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016.

[TKK+01] MP Tjoa, SM Krishnan, C Kugean, P Wang, and R Doraiswami. Segmentation of clinical endoscopic image based on homogeneity and hue. In *Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Annual International Conference of the IEEE*, volume 3, pages 2665–2668. IEEE, 2001.

[TM98] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846. IEEE, 1998.

[TP95] Claude Thibault and Eric C Poulin. Total laparoscopic proctocolectomy and laparoscopy-assisted proctocolectomy for inflammatory bowel disease: operative technique and preliminary report. *Surgical Laparoscopy Endoscopy & Percutaneous Techniques*, 5(6):472–476, 1995.

[TSM+16] Andru Putra Twinanda, Sherif Shehata, Didier Mutter, Jacques Marescaux, Michel de Mathelin, and Nicolas Padoy. Endonet: A deep architecture for recognition tasks on laparoscopic videos. *CoRR*, abs/1602.03012, 2016.

[TTS+14] Johannes Totz, Stephen Thompson, Danail Stoyanov, Kurinchi Gurusamy, Brian R Davidson, David J Hawkes, and Matthew J Clarkson. *Fast semi-dense surface reconstruction from stereoscopic video in laparoscopic surgery*, pages 206–215. 2014.

[VdBBR+12] Michael Van den Bergh, Xavier Boix, Gemma Roig, Benjamin de Capitani, and Luc Van Gool. Seeds: Superpixels extracted via energy-driven sampling. In *Computer Vision–ECCV 2012*, pages 13–26. Springer, 2012.

[vdPGJD08] E. P. Westebring – van der Putten, R. H. M. Goossens, J. J. Jakimowicz, and J. Dankelman. Haptics in minimally invasive surgery – a review. *Minimally Invasive Therapy & Allied Technologies*, 17(1):3–16, 2008.

[VJ01] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2001.

[VLC07] Sandrine Voros, Jean-Alexandre Long, and Philippe Cinquin. Automatic detection of instruments in laparoscopic images: A first step towards high-level command of robotic endoscopic holders. *IJRR*, 2007.

[VS14] Larissa Vines and Marc Schiesser. Gastric bypass: current results and different techniques. *Digestive surgery*, 31(1):33–39, 2014.

[WBM+15] Martin Wagner, Andreas Bihlmaier, Patrick Mietkowski, Sebastian Bodenstedt, Stefanie Speidel, Heinz Wörn, Beat Müller-Stich, and Hannes Kenngott. Cognitive camera robot for cognition-guided laparoscopic surgery. In *Proceedings of the Hamlyn Symposium on Medical Robotics*, 2015.

[WMB+17a] Martin Wagner, Benjamin Mayer, Sebastian Bodenstedt, Karl Kowalewski, Felix Nickel, Stefanie Speidel, Beat Müller-Stich, and Hannes Kenngott. Comparison of methods for bowel length measurement in laparoscopic surgery: A controlled cross-over trial. *Publication in preparation*, 2017.

[WMB+17b] Martin Wagner, Benjamin Mayer, Sebastian Bodenstedt, Katherine Stemmer, Arash Fereydooni, Stefanie Speidel, Felix Nickel, Lars Fischer, Markus Büchler, Beat Müller-Stich, and Hannes Kenngott. Quantitative laparoscopy for objective bowel length measurement in minimally invasive surgery: First in human. *Publication in preparation*, 2017.

[Woc15] Pamela Wochner. *Master thesis: Image-based tracking of laparoscopic instruments*. IAR Dillmann/IBT Dössel, Karlsruhe Institute of Technology, 2015.

[ZBHV13] Luca Zappella, Benjamín Béjar, Gregory Hager, and René Vidal. Surgical gesture classification from video and kinematic data. *Medical Image Analysis*, 17(7):732 – 745, 2013. Special Issue on the 2012 Conference on Medical Image Computing and Computer Assisted Intervention.

[Zha00] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:1330–1334, 2000.