THESES

Computer Science

Ralf Roland Stauder

Context Awareness for the Operating Room of the Future

TUM.University Press

Lehrstuhl für Informatikanwendungen in der Medizin & Augmented Reality
Fakultät für Informatik
Technische Universität München

Context Awareness for the Operating Room of the Future

Ralf Roland Stauder

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften (Dr. rer. nat.) genehmigten Dissertation.

Vorsitzender:

Prof. Dr. Helmut Krcmar

Prüfer der Dissertation:

1. Prof. Dr. Nassir Navab
2. Prof. Dr. Hubertus A. E. J. Feußner
3. Prof. Heinz U. Lemke, Ph.D.

Die Dissertation wurde am 17.10.2017 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 11.02.2018 angenommen.

The German National Library has registered this publication in the German National Bibliography. Detailed bibliographic data are available on the Internet at https://portal.dnb.de.

Imprint

1. Edition

Layout design and typesetting: Ralf Roland Stauder
Layoutguidelines for cover design: Designbuero Josef Grillmeier, Munich
Cover design: Caroline Ennemoser
Cover illustration: Ralf Roland Stauder

TUM.University Press
Technical University of Munich
Arcisstrasse 21
80333 Munich

DOI: 10.14459/2018md1398068
ISBN printed edition: 978-3-95884-048-5

www.tum.de

Abstract

In 2004, a group of international experts met in the workshop “OR2020” in order to discuss the concept and requirements of the “operating room of the future”. Today, almost ¾ of the projected timeframe later, especially in the area of surgical robotics many of the postulated necessities have already been achieved. Other areas though, like surgical workflow analysis, have proven to still pose open problems, some of which will be addressed in this thesis.

The operating room (OR) of the future requires smart and context aware devices with the ability to actively support the OR team, especially the head surgeon. Context awareness requires machine-accessible knowledge of the situation and the events happening in and around the OR. Therefore, it is necessary to automatically analyze, detect, and understand the surgical workflow. Several methods are presented and compared for this purpose.

Since not all events happening throughout a surgery equally influence the intraoperative work, they need to be filtered by their relevance for smart support systems. To achieve this, the concept of event impact factors is introduced, which can be calculated based on signals available in the OR. These can then be used to rank all events happening during the intervention, and therefore determine key events in a surgery, filter sensor data by importance, and even quantitatively compare very different approaches to the same surgery.

Finally, the surgeon needs to be able to control the plethora of novel devices as directly as possible. While indirect control is already widely available in today’s ORs (e.g. through voice recognition systems or unsterile nurses), these levels of indirection mainly introduce delays and obstacles and ultimately discourage usage of advanced device functions. Through automated knowledge of the surgical context, expected upcoming phases, and relevant data, it is possible to develop a dynamic, unified user interface, which is small enough to be sterile and available directly to the surgeon, while still presenting relevant information and control elements throughout the full intervention.

This field has recently been established as a building block of the newly defined area of Surgical Data Science, to better represent the wide range of applications. This work is among the first theses in this newly defined research area.

Zusammenfassung

Im Jahr 2004 hat sich eine Gruppe internationaler Experten zum Workshop „OR2020“ getroffen, um das Konzept und die Voraussetzungen eines „Operationssaals der Zukunft“ zu diskutieren. Heute, fast ¾ des erwarteten Zeitraums später, wurden viele der geforderten Ziele erreicht, besonders im Bereich klinischer Robotik. Andere Bereiche, wie die chirurgische Workflowanalyse, stellen weiterhin offene Probleme dar, von denen einige in dieser Arbeit angesprochen werden.
Der Operationssaal (OP) der Zukunft erfordert intelligente und kontextsensitive Geräte mit der Fähigkeit, das OP Team aktiv zu unterstützen, insbesondere den behandelnden Chefarzt. Kontextsensitivität benötigt maschinenverfügbares Wissen über die Situation und Ereignisse im OP oder dessen Umfeld. Deshalb ist es nötig, den chirurgischen Workflow automatisiert zu analysieren, zu erkennen und zu verstehen. Verschiedene Methoden, um dieses Ziel zu erreichen, werden vorgestellt und verglichen.
Da nicht alle Ereignisse innerhalb einer Operation die chirurgische Arbeit gleichermaßen beeinflussen, müssen sie nach ihrer Relevanz für intelligente Assistenzsysteme gefiltert werden. Um das zu erreichen, wird das Konzept von „Ereignisbedeutungsfaktoren“ vorgestellt, welche über intraoperativ verfügbare Signale berechnet werden können. Diese können dann genutzt werden, um alle Ereignisse eines Eingriffs einzuordnen, um darauf aufbauend Schlüsselereignisse zu erkennen, Messwerte nach ihrer Wichtigkeit zu filtern, oder um verschiedene Herangehensweisen an denselben Eingriff quantitativ zu vergleichen.
Abschließend sollte der Chirurg in der Lage sein, eine Reihe an neuartigen Geräten möglichst direkt zu bedienen. Indirekte Bedienkonzepte sind in heutigen OPs zwar weit verbreitet (bspw. über Spracherkennungssysteme oder unsterile OP Assistenten), derartig indirekte Mechanismen haben aber größtenteils Verzögerungen und andere Hindernisse zur Folge, und schrecken letztlich nur von der Verwendung fortgeschrittener Gerätefunktionen ab. Durch automatisiertes Wissen über den chirurgischen Kontext, erwartete Folgephasen und relevante Daten ist es möglich, eine dynamische und vereinheitlichte Benutzerschnittstelle zu entwickeln, die klein genug ist, um steril direkt dem Chirurgen zur Verfügung gestellt zu werden, dabei aber alle benötigten Informationen und Bedienelemente während des kompletten Eingriffs anzuzeigen.
Dieses Forschungsfeld wurde kürzlich als Baustein des neudefinierten Bereichs der „Surgical Data Science“ etabliert, um die breiten Anwendungsmöglichkeiten besser darstellen zu können. Diese Arbeit ist eine der ersten Dissertationen in diesem neudefinierten Forschungszweig.

Dedication

This thesis is dedicated to my late father,
Dr. Gerhard Maria Stauder,
who inspired me to pursue a scientific career early on.

Acknowledgments

First of all, I would like to thank my mother, Dr. Ursula Stauder, and my loving wife, Mandana Stauder, for their endless support in all my endeavors, including the composition of this thesis. Without their continuous support, this work would not have been finished by now.

Then I would like to thank my doctoral adviser, Prof. Nassir Navab, for giving me this opportunity, the support throughout the years, and the freedom to create my own path. All this is possible also because of the way he formed his chair into an amazing and ever-growing community, which provides a great foundation for creative and collaborative research.

At the same time, I am grateful for all the great colleagues and friends at the chair, whether they contributed their help scientifically, recreationally, or often both. Although it is impossible to list everyone on a single page (while maintaining a readable font), let me start with my MICCAI 2015 Co-Organizers Aslı Okur Kuru and Philipp Matthies. Other people supporting me throughout my time at CAMP, in no particular order, include Vasileios Belagiannis, Alexandru Duliu, José Gardiazabal Schilling, Benjamin Gutierrez Becker, Loïc Peter, Christian Rupprecht, Christian Schulte zu Berge, and my old “workflow brother-in-arms” Ahmad Ahmadi. Finally, many thanks go to Martina Hilla, who regularly helped me find my way through the administrative jungle.

I spent a significant time in the last years also at MITI in the hospital Rechts der Isar, where everyone has been very supportive, open to new ideas, and generally a pleasure to work with. Specifically, I would like to thank Prof. Hubertus Feußner, Dr. Armin Schneider, Sebastian Koller, Daniel Ostler, Thomas Vogel, Dr. Michael Kranzfelder, and of course Tereza Baude and Sabrina Stoeppke. The people at MITI have always been very focused on research in the OR, so if a method was suitable for tests under realistic conditions, MITI always found a way to do that.

Table of Contents

Table of Figures

Table of Tables

1 Introduction

The average standard of living in the civilized world has been increasing with the level of available knowledge and technology. Important aspects of this general progress are advances in medicine, both in the discovery of drugs as well as in the development of new treatment techniques. Surgery has historically been a key element in the treatment of medical conditions throughout almost all of human civilization [97, 131]. Yet as the surgical routines have recently become highly technological and sophisticated, it has also become one of the most expensive elements, so it is naturally of high interest for research. As such, impactful publications and specialized workshops regularly try to identify and discuss future trends and requirements for further growth.

One of these workshops was the “OR2020: Operating Room of the Future”, held in Ellicott City, MD, USA in 2004, organized be Cleary et al. [33, 34]. Clinical and technical improvements required for improved surgical care were discussed in different areas, including surgical informatics, systems integration, and operational workflow. Many of the proposed changes have become reality by now, such as improved surgical robotics, better integration of diagnostics into the operating room (OR) and connected, integrated OR suites. Several other features, however, like advanced “plug and play” interoperability of devices in the OR, smart tracking of equipment and patient records, and technical standards for defined surgical workflows are still open issues for research and development.

In the following years, Lemke et al. [13, 88–90, 92] refined these discussions and formulated a common “Therapy Imaging and Model Management System” (TIMMS) for use in the operating theatre (see also Figure 1). According to these works, many issues in the OR (like inappropriate data display, poor scheduling of patients, staff, and equipment, or delays due to extensive or late setup of requested devices) can be related to a lack of situational awareness, real-time access to peri-operative information, and standardized interfaces and protocols between surgical devices. To facilitate a seamless data transfer in order to solve these problems, the TIMMS is suggested as communication platform between different surgical “engines” (such as an imaging engine, a modelling engine, and a workflow and knowledge management engine) and their associated data repositories. The TIMMS also has access to additional, independent repositories for various models, e.g. implants, anatomical structures, intervention workflows, and evidence- and case-based medical data. The system must be designed in a highly modular way, so that individual repositories or engines can easily be exchanged or omitted. Therefore, it is mandatory for all components to communicate in a uniform manner. Due to its widespread use in medicine, an extension of the DICOM standard is recommended for use as “lingua franca” inside the OR, specifically the work of DICOM WG 24 “DICOM in

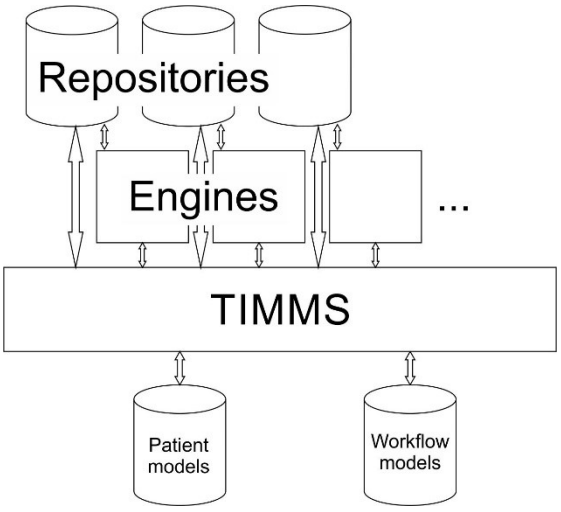

Figure 1: Simplified representation of a TIMMS.

Surgery". Some new aspects have already been introduced and standardized (implant models in PG 131, surface models in PG 132, and implant planning in PG 134), yet many sections still require such standardization. The later works [13, 90] as well as some related publications [13, 69] also describe a digital patient model (DPM) in detail as the core concept of modern, personalized medicine. This model is a flexible congregation of relevant data, from relatively low-level information (including static data such as genetic or anatomical information, or constantly changing data like pulse and respiration) to higher-level information (like environmental and nutritional factors, or interventional data). The model then describes a high-dimensional structure, linking various entries to related records and different properties. Another attempt to solve this issue, even in a way already viable for commercial applications, is the initiative "Integrating the Healthcare Enterprise" (IHE)[1], specifically with its surgery domain, which was discussed since 2011 [29] and founded officially in 2015. Their focus lies on defining integration profiles to enable standardized communication and data exchange across different organizations (such as hospitals in different cities and countries) and manufacturers.

Another influential workshop, called "Surgical Data Science", was held in 2016 in Heidelberg, Germany, organized by Maier-Hein, Speidel, and Jannin [101, 102, 169]. Extrapolating from the history of surgery, three key clinical applications for the future of surgery were identified during this workshop: decision support systems, context-aware assistance, and surgical training. The first two concepts build on the assumption that all perioperative data is available within a surgical network, in order to either provide all required information to the surgeon to aid in clinical decisions, or automatically trigger actions based on contextual cues. The latter takes advantage of the developed patient and procedure models, to simulate a wide variety of interventions and their possible emergency situations for more thorough training opportunities for surgeons. Similarly to the TIMMS approach, a major prerequisite for this is a common communication network. The key applications can even be compared to modernized and more compressed variations of the engines and repositories of a TIMMS. Following a similar motivation was a recent issue of "Innovative Surgical Sciences", focused completely on the future of surgery, also referred to as "Surgery 4.0" [42, 79, 108, 157, 169]. The important parts for this are again identified to be interoperability, availability of information, and smart devices, able to support the human decision-making process, or to come to simple decisions themselves.

Additionally to the presented TIMMS and patient model mentioned above, Lemke et al. [90] also gives a bird's-eye view of the maturity level evolution of the digital operating room (DOR), starting at the situation around 2005 and leading to the expected development until 2025. As per that work, we are currently between maturity levels 2 and 3, with the beginning of model- and workflow-guided interventions, connected (yet still no "smart") display walls, and growing (yet no full) support for DICOM in surgery. Higher levels are characterized by knowledge management, vendor independent interoperability throughout the whole hospital, and patient-specific models (level 4) and surgical cockpit systems, intelligent medical data mining and real-time access, and context aware robotic assistance systems (level 5). Many of these points will be taken into account in this thesis, to hopefully aid the progress of the DOR and further follow the predicted development curve. The chapters 2 and 3 will introduce methods to automatically detect the progress of the surgical workflow and rank events happening inside the OR respectively. Both aspects can be used to analyze the situation in the OR at any point and provide an infrastructure for context aware systems. Chapter 4 will then describe a unified surgical display as central data presentation and control unit. This display builds on the previously established workflow analysis methods in order to provide a smart hub for the surgeon, providing a further step towards the implementation of a full TIMMS.

[1] https://www.ihe.net

2 Detection of Surgical Phases

In a modern operating room, many simple and highly predictable actions still have to be triggered manually, such as switching the lights during a shift of focus from the monitors directly to the surgical site during the extraction phase of a minimally invasive intervention. This seems counter-intuitive for a highly developed environment such as the OR, considering that many aspects in the industrial and even private sector already provide sophisticated levels of automation. Industrial manufacturing has known the technique of process modelling for years by now, where the environment, such as a factory building, is tailored to a specific problem, so that a predefined, well-known workflow can be executed as easily and efficiently as possible. This concept cannot be simply transferred to the medical domain, however, as the surgical environment, which is given by each patient's individual anatomy, condition, and the specifics of their disease, is not necessarily fully known before and cannot be sculpted to fit a given approach. So a theoretically ideal workflow has to be adapted to this varying environment. And while this theoretical workflow is rarely explicitly defined, surgeons, nurses, and other members of the OR team each build their own mental model of each surgery based on formal knowledge and their experiences. This allows a well-practiced team to anticipate and prepare upcoming common tasks for a smooth overall procedure, while new and inexperienced team members, without said internal model, can actually delay and hinder the intervention in the worst case.

Some current approaches to mitigate this problem include input by voice commands in order to give the head surgeon more direct control over peripheral devices (and therefore avoiding the reliance on other team members' ability to anticipate needed actions). Alternatively some devices already offer slightly workflow-oriented user interface (UI) designs, to guide inexperienced users as well as reduce orientation time and possible error sources. Nonetheless, these options exist so far only as solitary and highly limited solutions for few devices, without a common basis or higher understanding of the surgical context.

A key foundation for even partial automation in the OR is digitally accessible situational understanding and context awareness. Similarly to the mental models of experienced surgeons and nurses, context aware devices need to be able to refer to some variation of an explicit or implicit model. Depending on the available data and the intended application, these models and their requirements can vary greatly. A few of the core differences between major methods include the temporal resolution (from the surgery as a whole unit on the coarse end, through phases and activities down to single movement gestures on the fine separation) and the observed input data (e.g. from a limitation to only specific sensors, to information about the used instruments or affected anatomy, to detailed kinematic readings from robotic systems or different video sequences). The nomenclature about these aspects has not yet come to a unified understanding among the involved research community, but a good review about these distinctive approaches is available from Lalys et al. [83].

One approach to provide an explicit model is the manual annotation of every individual action performed by the surgeon. This list of actions is considered a simple model of the specific, record intervention. Applying the general idea of business process modeling [121, 122], this individual model is also known as "individual surgical process model" (iSPM). By collecting several such models of the same intervention type, a "general surgical process model" (gSPM) can be calculated from the set of differing iSPMs [96, 118–120]. On basis of the gSPM it is possible to identify common sequences, which are shared among most or all observed procedures, as well as the sequences, where it is most likely to see high variability. SPMs have been found to be very robust against missing data [95], and approaches exist to transfer them to other structures [49]. Recently, more concepts from the area of business process modeling have also been converted to the surgical domain [28, 117], yet gSPMs so far remain the basis for many different applications, including the prediction of the remaining surgery time or upcoming instrument usage [43–48, 103]. The usage of such

models can be extended to further situations, like flexible training simulations or the evaluation of young surgeons' dexterity and training progress. An attempt to standardize these models by fitting them to existing ontologies, or extending known ontologies to this novel area, has been discussed already quite early [66, 91]. The general idea behind the standardization is a more hierarchical description of the individual or generalized models, to allow for slight changes in the intervention, without the need to extend the model. Therefore, current, workflow related ontologies define the scalpel for example as a "cutting tool", which is itself a subcategory of a "sharp tool", and so forth. Also, the patient's affected anatomy can be described this way in any desired level of detail, starting from the general region of the patient's body, to each individual organ and their vessels and ducts, down to the individual cells, and theoretically even the intracellular components. Different already existing ontologies have been proposed as basis for an extension [17, 73, 76, 77, 111, 115, 116], with the recent development of the "LapOntoSPM" ontology, which extends other, surgical ontologies specifically with the goal to standardize the description of surgical process models [52, 72, 74].

A different approach of surgical workflow analysis tries to minimize the manual annotation costs, by omitting the explicit model and extracting workflow information directly from automatically collected sensor data. These sometimes very rudimentary sensors are processed through various machine learning methods to directly retrieve higher level information, such as the surgical phase or performed activities. A wide range of methods are built on the collection of different external sensors, with a strong focus on instrument usage data. Attaching RFID tags to surgical instruments to detect their usage is a common approach [80, 81], yet several diverse approaches exist on how to extract workflow information from such instrument usage data, often employing dynamic time warping (DTW) [3, 125, 126] or hidden Markov models (HMM) [19–21, 25, 127], but a wide variety of methods exists [24, 154, 155]. Also other sensors than instrument usage are suitable for workflow detection, such as tracking the staff location or movement [2, 12, 105, 113] or recording simple system states and events [41, 104]. Even just predicting surgery difficulty from preoperative data can support numerous applications [26, 144]. Many methods are based instead on the analysis of intraoperative videos, either recorded from external cameras or, due to its general availability in minimally-invasive surgery, from laparoscopic cameras. When analyzing the surgical workflow from external cameras, often 3D information is collected through depth cameras or by reconstructing depth information from multiple cameras. These video streams are then classified using different machine learning algorithms [93, 112, 165], with a large number of methods using HMMs at least for post-processing, due to their ability to preserve temporal information [14, 128, 163, 164]. Occasionally also more exotic modalities like thermographic cameras are deployed [168]. Using external cameras usually allows to segment the process into coarser temporal segments, including the preparation and post-processing of the room, equipment, and patient. Examining the laparoscopic video of minimally-invasive surgery allows a more finely grained segmentation into gestures, activities, or phases, but the approach is limited to the core part of the surgery (from entry of the camera into the patient's body to its exit), and is unsurprisingly unsuitable for open surgery. HMMs are also a common choice to use or include in the analysis of laparoscopic videos [39, 84, 85, 98, 127, 170], although this problem has motivated many different solutions, too [18, 58, 86, 132, 173]. Recently, also so called deep learning approaches have influenced many research areas, including surgical workflow analysis [22, 94, 156, 167]. An interesting approach to combine the usually very reliable results of instrument-based workflow detection with the easy availability of laparoscopic videos tries to identify and occasionally track the surgical instruments in the video images. In some cases additional hardware is used to aid the detection, such as colored markers on the instrument shafts [23], marker rings on the instrument grips for trocar-mounted cameras [162], or scales and infrared cameras [53, 54]. Other approaches directly identify the instruments from mono or stereo cameras [4, 5], or by emulating acceleration data of instruments from visual cues [138]. Recently, with the broader adoption of surgical robotics, the kinematic data of the robotic effectors (often in conjunction with the corresponding video) have also become a valuable source for workflow detection [40, 86, 173, 175]. A

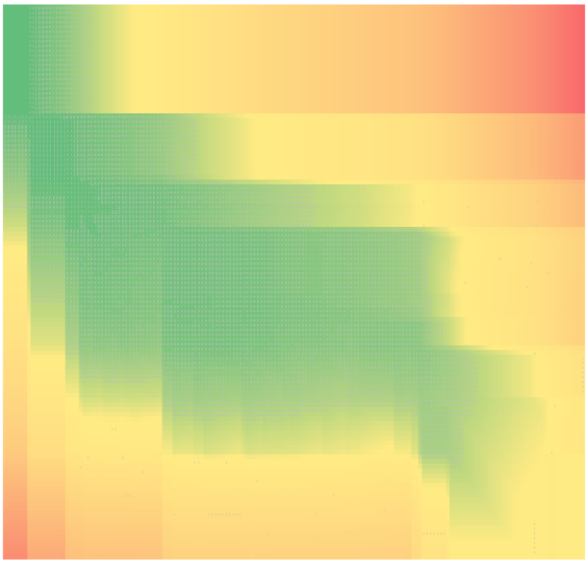

Figure 2: Visualization of a filled DTW matrix for matching two workflow sequences. Green pixels represent small values. A trench of smallest values is clearly visible in the center.

further review about the various methods and input modalities used for sensor-based context recognition is also available from Pernek et al. [130].

In this chapter, different approaches to detect the surgical workflow based on various low-level signal inputs, without the prior collection or definition of explicit models, will be presented and compared. This will include their respective advantages and disadvantages, as well as their recognition performance, given different available input data.

2.1 Methods

Several methods have been applied with the aim to recognize surgical phases from recorded data automatically. These methods, from the areas of dynamic programming and machine learning, will be presented and shortly explained in the following sections. Their respective advantages and disadvantages in regard to surgical phase detection will also be briefly listed.

2.1.1 Dynamic Time Warping

Dynamic time warping [70, 142] is a method to match two related signal series onto each other, compensating for different timing by locally stretching or squeezing the signals to get a best fit between them based on a distance metric. In order to do this, a cumulative distance matrix is calculated by applying a recursive function in a dynamic programming fashion on this matrix.

2.1.1.1 Warping Path Calculation

Given two time series $A = \{a_1, a_2, \dots, a_m\}$ and $B = \{b_1, b_2, \dots, b_n\}$, create a matrix of size $m \times n$: $DTW \in \mathbb{R}^{m \times n}$. As the data series do not have to be of the same length, the matrix will not necessarily be square. For every value $p_{x,y} \in DTW$ first the difference $d(a_x, b_y)$ between the corresponding feature points of both sequences is calculated based on a chosen distance metric (e.g. the Euclidian distance). Then the value of $p_{x,y}$ is given by adding the distance $d(a_x, b_y)$ to the minimal value of the three neighboring values in the direction of the origin. Assume a value of 0 for values outside the matrix (e.g. $p_{0,0} = 0$).

$$p_{x,y} = d(a_x, b_y) + \min\left(p_{x-1,y}, p_{x-1,y-1}, p_{x,y-1}\right)$$

After recursively filling the whole DTW matrix, the warp path $h(t)$ representing best correspondence between both series is achieved by backtracking the "trench" of smallest values (Figure 2), starting at the final corner of the matrix $p_{m,n}$. Every point on this path matches two points of the series, while every point

of each series will be mapped to at least one point of the other series. If both series already match perfectly, this path will be the diagonal line across the square matrix. Otherwise, every deviation from the diagonal indicates that both series are running at different speeds and need to be adjusted to each other.

2.1.1.2 Average Warping Path and Sequence

With DTW it is normally only possible to match exactly two sequences to each other. In order to be able to use the method on several sequences, it is possible to calculate an average, semi-probabilistic sequence based on a collection of sequences through DTW as shown by Ahmadi et al. [3]. In this approach, first one sequence of a collection of sequences is chosen as reference $S_{ref} \in \{S_1, \dots S_k\}$. This selection can be done arbitrarily, though it is advantageous for the calculation to choose the sequence with median length. Every other sequence is mapped to the selected reference sequence to get $k-1$ warping paths. An intermediary warp path h_c is calculated by interpolating and averaging all obtained individual warp paths h_i. This warp path h_c matches the reference sequence to a common, average timing. Then shift functions are calculated for every sequence by concatenating the inverse of the common warp path with their respective individual warp paths: $u_i(t) = h_i(h_c^{-1}(t))$. Finally, all sequences can be matched onto a common, average timeline. The feature values at each time step of the new timeline are calculated as averages of all mapped features. In case of binary input signals, the averaged output feature will therefore have continuous value, which can be interpreted as approximate probabilities of the given feature being active during any specific time in the sequence.

Surgical phases of a newly recorded surgery can be detected by matching a sequence of a known and fully labeled surgery onto the new sequence and transferring the labels for each time step. In order to prevent bias from a single surgery, the best approach is to collect and label several different surgery sequences, and match them to a single, average sequence as shown above. Then this average sequence with matching labels can be used for further labeling of new sequences.

2.1.1.3 Strengths and Weaknesses

The DTW method is easy to apply, as there are no parameters required by the method. The only possible variation is in the choice of the utilized distance metric, while the Euclidian distance is suitable for most cases. It is of course also possible to specialize the recognition by selecting one of several distance metrics based on previous classification as done in [126]. In that work, the distance functions strongly focus on only few feature elements and are exchanged based on the detected phase, to optimize the detection of phase transitions.

Since DTW does not actually calculate predictions based on the observed data, but matches series as a total, it is also very robust against noise and outliers. On the other hand, the DTW algorithm is not generally parallelizable due to its recursive nature. The basic DTW algorithm requires full access to both time series in order to complete the calculation and find a suitable warp path, so an online application is not possible. Newer publications suggest adoptions of the original DTW though, to allow for partial mappings and thereby online applications. Tormene et al. [161] describe an open-ended DTW variant, which is capable of matching a prefix of arbitrary length of a sequence to a full, known reference sequence, while in [8] an unbounded DTW is presented, which is able to match segments of sequences independently of their position in the target sequence.

A systematic disadvantage of DTW is the fact that it can only match sequences onto each other. Therefore, a new sequence can only be described as well as the best fitting ground truth sequence, a true out-of-order labeling of phases is not possible. In addition, a change or reordering of activities within a phase can break the matching, which will also influence the matching of all further samples in the sequence.

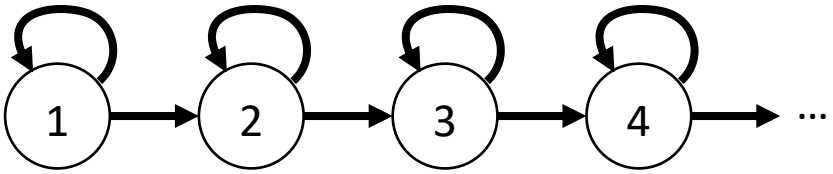

Figure 3: A schematic view of a left-to-right HMM. In every state, the model can either stay in the current state or move to exactly the next state.

2.1.2 Hidden Markov Models

A hidden Markov-model as explained in [133] is a network of *N* distinct states. A system described by this model can be in only one state at any given time, and can change states only between regular, discrete time steps. The transition between states is governed by transition probabilities, with all outgoing transition probabilities a_{ij} for a given state *i* (including the loop to itself a_{ii}) summing up to 1. For every discrete time step, a single observation of a set of known, possible observations is recorded for the system. The probabilities of each observation can be different for every state.

2.1.2.1 Model Description

The model can be described as $\lambda = (N, M, A, B, \pi)$. N is the number of possible states S_n. M is the number of possible observations in the set of all possible observations V, $|V| = M$. $A \in \mathbb{R}^{N \times N}$ describes the transition probability matrix, with $a_{ij} \in A$ being the transition probability from state *i* to state *j* and $\Sigma_j a_{ij} = 1$. The observation probability of observation *k* in state *j* is given by $B = \{b_j(k)\}$. Finally π is the initial state distribution among all states for time step 1. When comparing several possible HMMs for a given scenario, it is common to only describe the models as $\lambda = (A, B, \pi)$, as the states and possible observations are usually given by the problem definition and are not part of the optimization.

In an ergodic HMM all transition probabilities between all states are non-zero, so every state can be reached at any time. As this does not accurately model the surgical modus operandi, it is more common for workflow analysis to restrict the network to so-called left-to-right models (Figure 3). In these HMMs, states can be arranged in a temporal order, and no transition is allowed to go backwards along the model. Therefore, the initial state distribution π is 1 for a single starting state S_1 and 0 for all other states, and the transition probabilities are 0 for all transitions to earlier states, so $a_{ij} = 0,\ j < i$. Several parallel "tracks" are possible, though, including transitions between the tracks, as long as no loops within the system occur.

2.1.2.2 Calculations in Surgical HMMs

Three main questions arise when working with HMMs: What is the probability $P(O|\lambda)$ of a known observation sequence $O = O_1 O_2 O_3 \dots O_T$ given a specific model $\lambda = (A, B, \pi)$? What is the most likely path through the states of a given model λ with a given observation sequence *O*? And how can a model λ be optimized in order to maximize the observation probability $P(O|\lambda)$? For all of these questions, efficient algorithms exist (e.g. the forward-backward-procedure, the Viterbi-algorithm and the Baum-Welch method respectively [133, 172]), though for surgical process modeling, some parameters can simply be obtained by careful examination of the available data.

For surgical interventions a sequence of phases is often apparent or can easily be deducted from monitoring few exemplary surgeries. These phases can in most cases be interpreted directly as states of a left-to-right HMM, in their respective temporal order [20]. The observation space depends on the recorded features (see 2.2) but is predefined and known. The transition probabilities can be approximated through the recorded training data. The probability a_{ij} to switch from a phase *i* to another phase *j* is the number of recorded phase switches in this direction divided by the total number of time steps spent in phase *i* over all recorded datasets. As this ratio depends strongly on the definition of a time step, it is best to either choose a sufficiently large time step, or to also take all neighboring time steps within a certain "transition window"

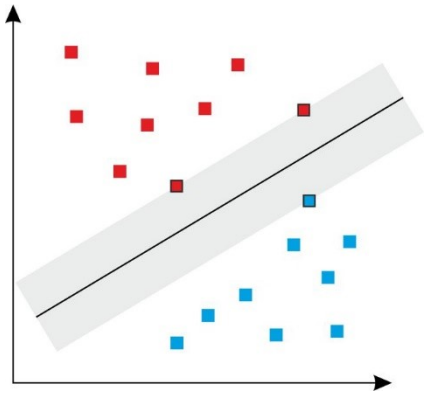

Figure 4: Schematic example of a hyperplane dividing samples of two classes. The corresponding support vectors are highlighted.

into account. The window or step size should be chosen in a way that a phase detection within the defined timeframe is still meaningful for the studied surgery, e.g. 3-10 seconds for instrument prediction or 1 minute for documentation purposes. The transition probability to stay in the current phase, a_{ii}, is then calculated as the complement probability to all outgoing transitions of the phase. The observation probabilities *B* can again be calculated by simply counting each observation per phase over all recorded surgeries and divide them by the total time steps spent per phase. The initial state distribution π is defined to be 1 for the first state, which is the first surgical phase, as for every left-to-right HMM.

2.1.2.3 Strengths and Weaknesses

A strong advantage of (left-to-right) HMMs, which made them very popular for surgical workflow analysis, is their strictly model-driven approach. Domain knowledge about the studied surgeries can be fed directly into the model through the definition of the states and their transition order. The current phase of an ongoing surgery can analogously be identified by calculating the most likely path through the trained model, given the collected observation sequence, and simply reporting the phase corresponding to the final state reached. On the other hand, this explicit model definition can be a problem in case of severe changes in the surgical workflow, to a degree where the trained model does not fit the observations anymore. In such a case, some states would be skipped completely.

2.1.3 Support Vector Machines

Support vector machines (SVMs) [30] are a very common and popular machine learning algorithm for classification into two classes. This classification is achieved by defining an optimal hyperplane, separating the positive and negative data points with maximal margin, through selected support vectors.

2.1.3.1 Calculations and nonlinearity

Given multidimensional input training data $\{x_1, \dots, x_n\}, x \in \mathbb{R}^d$ and their respective ground truth labels $\{y_1, \dots y_n\}, y \in \{-1,1\}$, a SVM finds an optimal hyperplane $wx + b = 0$, which can separate the two classes of the training data. The optimization of the parameters *w* and *b* tries to maximize the margin between the two classes by selecting suitable support vectors in each class, and define the hyperplane to be equidistant from all support vectors (Figure 4).

While the hyperplane according to the definition above would only support linearly separable classes, it is possible through a “kernel trick” to theoretically transfer test and training data into a higher dimensional space, while doing all related calculations through kernel functions in the original space. This way a linear hyperplane in a higher dimension is calculated, which results in a nonlinear separation surface in the original data space.

2.1.3.2 Multiclass classification

SVMs are defined to only separate two classes from each other, though there are several approaches to extend them to multiclass problems. A first approach is to train a separate SVM for every class, to separate between the class (e.g. via label 1) and "everything else" (e.g. as label -1) [30]. The final classification decision is based on the highest positive margin score of all classifiers. A different approach trains classifiers for every pair of classes, to distinguish between them [41]. Then all classifiers vote for a class, and the majority decides the final classification. In this approach, more classifiers need to be trained and evaluated, though the results are more robust.

2.1.3.3 Strengths and Weaknesses

Support Vector Machines are by now widely known and well established, which grants them the de facto status of a gold standard in machine learning. Additionally, and in contrast to other machine learning methods, SVMs have a mathematical proof of obtaining a global, optimal solution for the given training data [30].

SVMs can be extended to regression problems, too, and as shown above, several options exist to extend SVMs to multiclass problems, though they always require additional training or calculations. Due to the definition of a hyperplane, the classification of SVMs tends to have "hard" differentiation between classes. An estimation of a confidence score is only indirectly possible through calculating the distance from an examined sample to the separating hyperplane.

2.1.4 Random Forests

Random forests [27, 35] are generally an ensemble of independent, randomized decision trees. Incoming data are routed to different branches in the nodes of each tree until they reach a terminal leaf node. Each sample is then classified based on a majority vote of the leaf nodes it reached in all trees.

2.1.4.1 Deterministic and Randomized Decision trees

Each decision tree is a binary tree structure, consisting of at least a root node, where each node is either a terminal leaf or branches into two child nodes recursively. For each incoming data point, every node, starting with the root, checks a simple condition (e.g. a specific variable value against a fixed threshold). Based on the check, the sample is sent to a different child node (e.g. if the specified variable value is less than the given threshold, the sample is passed on to the left child node, otherwise it moves to the right child node). This procedure is repeated over multiple layers throughout the whole tree, until a terminal leaf node is reached. All samples, which were passed through the tree, are classified according to the leaf node they reached.

The most important element for a successful decision of the tree is a good choice of the splitting function in each node. In order to achieve this, decision trees need to be trained with fully labeled training data, which consist of observable features and ground truth labels, assigning each sample correctly to the available classes. In deterministic decision trees, all data are tested against all available features with optimized thresholds, and the feature, which achieved the best split, is chosen as parameter for the decision criteria for the given node. Then all data is split into the two subgroups according to the newly defined split function, and passed on to the next child nodes, where the process is repeated on the smaller, split data set. This is repeated until either a predefined, maximal tree depth is reached, or until no splitting function can be found anymore, which could provide a "meaningful split" of the data (see 2.1.4.3). Finally, for each leaf node a classification label is chosen based on a majority voting of all labeled samples that reached this node. Alternatively, a histogram over the labels reaching the leaf node during training can be created to provide further information, such as confidence in the classification or other (less likely, yet possible) classifications.

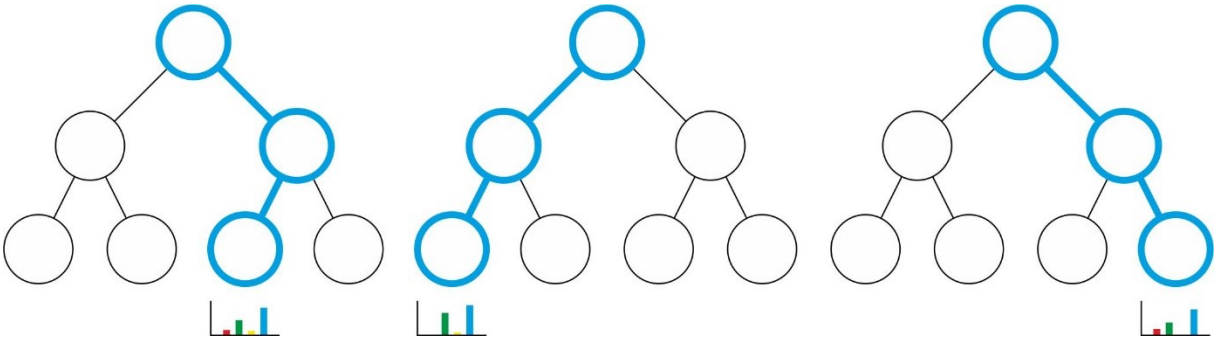

Figure 5: Schematic example of a random forest. A sample can traverse different paths in different trees during classification, the reached leaf nodes then vote for a class based on the training samples, which reached the same leaf.

This process takes all available training data and all available features into account in each node. Due to the fully informed definition of the splitting functions, this method is also prone to overfit the provided data during the training process, which deteriorates the robustness and generality of the approach. While different approaches exist to handle this problem after training (e.g. pruning of decision trees), a better approach is to minimize the risk of overfitting beforehand. Random forests try to avoid risk this by abstaining from using a fully informed training process but creating multiple decision trees in parallel to maintain classification performance.

The training process for decision trees is deterministic, so multiple generated trees would end up with the exact same layout and splitting functions in every node. For randomized decision trees, each tree is only presented with a different, randomly chosen subset of all available input data (so called “bagging”), so that each individual tree receives a different training set. Also, not all features are compared for each node, but again only a randomly chosen subset of all possible features are taken into consideration in each node for choosing the best split. In a related approach for “extremely randomized trees” [51] even the thresholds to compare each feature against are defined randomly, within the value ranges of each feature in the bagged training set.

2.1.4.2 Random Decision Forests

The aim of random forests is to provide better robustness and generalization than single, deterministic decision trees. This is achieved by introducing randomness into the training process and calculating an averaged output. Therefore, not only a single tree is trained and used for classification, but a large number of them. Due to the randomness of the training process, each tree will develop a different structure, with different split functions in each node and differently distributed leaf nodes. For classification all incoming data are given to all trees in parallel, they traverse the trees in different patterns, and they end up in different leaf nodes (Figure 5). Every tree therefore has a separate classification for each input sample. All individual trees participate with their respective output in a majority voting for the final classification of the input data. The majority voting of multiple trees can easily overcome low-confidence misclassifications of few trees with a larger number of correct classifications and thus improves robustness against overfitting.

As added analytical benefit a degree of confidence on a classification can be achieved by calculating the ratio of votes for the classification output to the total number of votes. For example, a classification with 95 votes for class A in a forest with 100 trees has a 95% confidence, while a vote for another input with 60 votes for class B only has 60% confidence.

2.1.4.3 Information Gain as Splitting Function

When choosing one of several possible splitting functions, a clear criterion for defining the “best” candidate is required. Commonly the information gain based on Shannon’s entropy is used, where generally splits are preferred that strongly support the differentiation between classes.

During training, before applying any splitting function, the Shannon entropy *H* [148] is calculated for the dataset that reached the current node:

$$H(S) = -\sum_{c \in C} p(c) \log_2 p(c)\,,$$

where $p(x)$ is the probability, that a random sample of the set S in this node is of class *x*, which is approximated by the ratio of training samples labeled with *x* to the total number of samples.

Candidates for splitting functions are created by randomly choosing a subset of available features (or ad hoc generating a predefined number of features where possible) with appropriate thresholds. These thresholds can also be chosen randomly within reasonable boundaries [51]. Then an available splitting function candidate is applied to all samples, the entropy for both resulting subgroups S^l and S^r are determined, and their weighted average (based on the number of data points in each subgroup) is calculated. The difference between the prior entropy and the calculated posterior entropy is calculated. This represents the information gain I for the tested candidate:

$$I = H(S) - \left(\frac{|S^l|}{|S|} H(S^l) + \frac{|S^r|}{|S|} H(S^r) \right)$$

This is done for all splitting function candidates. The splitting function candidate with the largest information gain is chosen as splitting function for the given node.

Additionally, a minimal required information gain can be defined to further split the node. If no splitting function candidate can reach the required information gain, the node is declared a leaf node, and the classification output of the node is based on the majority of class labels of samples that reached the node.

2.1.4.4 Strengths and Weaknesses

Due to the distribution of the classification of a sample to multiple trees, random forests are able to generalize well over provided training data and achieve good robustness against outliers and atypical data. In addition, the voting mechanism among trees and the basic tree structure with classification histograms in the leaf nodes make random forests inherently well suited for multi-class classification problems. While other classification methods usually rely on a “one vs. all” approach by training separate, independent classifiers per class on the same training data, random forests naturally handle multiple possible classes well within the same structure. A suitable number of classes is not strictly limited by, but related to the maximum tree depth, which in turn should be limited to prevent overfitting. Additionally, the independent nature of the method makes it trivially easy to implement it in a highly parallel fashion, taking full advantage of modern computing hardware of both CPUs and GPUs [149].

Compared to deterministic decision trees another advantage of random forests is the fact that they do not need to evaluate all available features during training or testing. Depending on the examined dataset, there could be millions of possible features present, which would make the training process tedious and slow, or it could even be impossible to calculate all available features beforehand, as they can be defined dynamically through few data-independent parameters. A popular example is image segmentation and pixel classification within given images [4, 36]. Here a very simple feature is used, the difference between brightness values of two pixels in the neighborhood of the examined pixel. For high-resolution images there can easily be millions of pixels (e.g. HD video with a resolution of 1920x1080 pixels has just over 2 million pixels per frame to evaluate), the difference between two pixels squares this feature space (to over 4 billion possible combinations on HD video frames). Exhaustively evaluating all these combinations for every single dataset in every node of every tree is not feasible with limited training time and storage memory. The randomized training phase of random trees on the other hand can easily be adopted to generate a specific

number of random features based on this general idea, so that the feature space presented to each individual node during training has a fixed, predefined, and easily manageable size.

Random forests are not limited to classification, as they can also be extended to handle regression problems. In this case the terminal nodes not only store a single class label or a discrete histogram over class labels, but an averaged function prediction based on the training samples reaching each node. The final regression output is calculated as the average over all individual tree outputs.

Finally, there are different approaches to use fully trained random forests to aid in analyzing the base data and the importance of the calculated features for the trained problem. One approach by Breiman [27] calculates the relative feature importance of all used features in a dataset. To achieve this, first the final classification error of a fully trained forest needs to be calculated on a separate validation set. Then the samples of the validation set are manipulated in a way that all values of an examined feature F_i are permutated throughout all samples, and this manipulated dataset is classified again by the trained forest. The difference between the original classification error and the error on the manipulated dataset indicates the importance of feature F_i for the classification. Another possibility used by [36] is to backtrack the correctly classified samples from the leaf nodes they ended up in to the root nodes of all trees, and keep track of all evaluated features and their values along the way. This allows getting a general idea, which features and feature values are used most often in the forest to determine each class label.

A characteristic of random forests, which can be seen as both advantage as well as disadvantage, is the fact that random forests cannot be supplied with explicitly defined models. While this avoids the need to define a model, especially in situations, where the underlying model of obtained data is unknown, it is also not possible to easily feed model-based knowledge into a random forest without significant changes to the structure or classification workflow, or by embedding this knowledge into elaborate artificial features. On a similar aspect, random forests are also not well suited for situations, where a large-scale exploration of the feature-space is possible and desired.

The proper design of provided features can have a major impact on the classification quality. In complex situations, all non-linear relationships between collected data (e.g. local neighborhood information or temporal dependencies) must be calculated through predefined meta-features. While other models are possible, most random forests use linear splitting functions even for non-linear problems, which leads to the classification behaving as piecewise, linear approximation. Although due to the large number of involved trees, the averaged results can usually provide relatively smooth decision boundaries in most cases.

2.1.5 Convolutional Neural Networks

Artificial Neural Networks (ANN) are among the oldest machine learning methods under research [109, 137], and after some period of little scientific attention have developed to be very powerful tools in recent years, winning recognition challenges of various topics with wide margins.

2.1.5.1 Neurons and Multilayer Networks

The basic building block of a neural network are called neurons (both in the artificial ones described here as well as in the biological original, which inspired the method). A neuron has a number of input signals x_i, for which it calculates the weighted sum, evaluates an activation function f for this sum, and provides the function value as output. Usually every single neuron has a constant bias b as one of its inputs:

$$\vec{x} \to f\left(\sum_i W_i x_i + b\right)$$

Several activation functions have been suggested over time, with the sigmoid function $f(x) = \frac{1}{1+e^{-x}}$ and the hyperbolic tangent function $f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ among the most popular due to their proximity

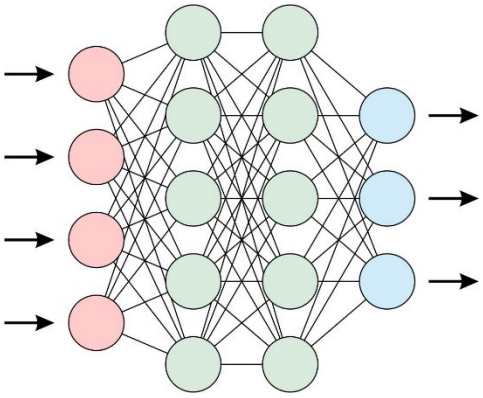

Figure 6: Example of an artificial neural network with 4 input nodes, two hidden layers of 5 nodes each, and 3 output nodes.

to the biological processes. Recently, however, the very simple rectified linear unit (ReLU) $f(x) = \max(0, x)$ is gaining popularity, since it is very easy and fast to compute without deteriorating the overall classification results.

In order to obtain several output values, different neurons can be used in parallel on the same input values and using the same activation function, though with individual weights and biases. This layout is referred to as a neuron layer. While a single neuron layer can already be trained to correctly predict simple, linearly separable problems [109], most modern approaches use networks of several layers. In this setup, all outputs of the neurons of a layer L_i are connected as inputs to all neurons of the following layer L_{i+1}. The actual input data presented to the network is treated as "input layer" L_1, the output of the last layer denotes the final network output, while all layers in between are called hidden layers (Figure 6). The bias terms are defined for each neuron individually, while weights are defined separately between all neurons of two neighboring layers.

2.1.5.2 Feed Forward and Backpropagation

Calculations are done in a trained neural network by feeding the data forward through each layer until the output layer is reached. Since no two neurons in the same layer depend on each other, each layer can be computed in one step, usually implemented as simple matrix multiplication.

Training is done on neural networks through the backpropagation algorithm. A training example is given to the network and the output based on current parameters is calculated. The difference between the desired and the actual output is calculated and multiplied by the derivative of the activation function, evaluated on the same values as during forward calculation. The ReLU activation function is again very easy to compute, as its derivative is always 1 for any positive values and 0 otherwise[2]. This difference score is propagated backwards through the network, by calculating the weighted sum of all outputs per node with the same weights as for the forward step and multiplying by the derived activation function per neuron. Finally, all weights and biases are modified by the partial derivatives in each neuron, weighted by a factor representing the learning rate.

2.1.5.3 Convolutional and Pooling Layers

High resolution images with millions of color pixels can be used directly as input for neural networks, though building a fully connected network directly based on such a large input layer would either result in too many parameters to be feasibly trained or poor prediction accuracy. Therefore, the concept of convolutional layers has been introduced [87]. When taking full images as input, the first few layers are usually trained to

[2] Technically the derivative of the ReLU function is undefined at $x = 0$, though the probability of the output value actually equaling 0 is very low, which makes this an acceptable implementation.

convolve the image with varying kernels of increasing size, while the overall resolution gets down sampled through pooling layers.

Contrary to regular neuron layers, not all neurons in a convolutional layer take all outputs of the previous layer into account, but only the outputs of a small patch of spatially connected neurons. Other neurons with identical patches are repeated in order to cover the whole output of the previous layer. These patches can, but do not have to overlap between neighboring neurons. The input weights of all corresponding patches are shared and applied to all patches alike. A convolutional layer consists of multiple of these sets of corresponding neurons, with weights shared within each set, but independent from the weights of other sets. These patch weights evolve during training to match simple image features in the early layers (like edges or textures), up to human-recognizable structures in later layers (such as wheels, doors, or faces, depending on the training set).

In order to reduce the dimensionality of the data, and therefore the number of parameters requiring training, commonly "pooling" layers are introduced after convolutional layers. These pooling layers choose only a single value within a small patch of neighboring input values, usually the maximal or average value. This value is then output without further calculations and independently of any parameters, so pooling layers are not subject to changes during training. By alternating between convolutional and pooling layers of increasing patch size, it is possible to reduce the size of the input data. This allows for larger, more complex and more numerous kernels to be trained, while keeping the memory footprint and training effort on a reasonable scale.

2.1.5.4 Deep Networks and Memory Units

The increase of powerful and accessible hardware, the development of highly parallel computing on GP-GPUs, as well as the public availability of large-scale annotated datasets (like [67, 156]) have made it possible to significantly increase the number of layers and neurons in neural networks. This evolution to "deep networks" not only made highly performant networks possible, but also enabled the design and application of highly intricate architectures, including "long short-term memory" (LSTM) units [64]. One LSTM unit consists of several cells and a storage vector. The values stored in the vector are provided as additional input to the cells within the same unit. The cells are trained as before, though their outputs are used to trigger different behaviors, such as "forgetting" or overwriting certain parts of the storage vector, as well as calculating a combined output for the LSTM unit as a whole. The application of these LSTM units allows networks to better handle sequential and time-variant data, such as texts and video data.

2.1.5.5 Strengths and Weaknesses

As modern, deep ANNs have a very large number of free parameters, they usually require relatively large datasets for training, which in turn can lead to training times in the magnitude of weeks, even on powerful hardware. Developing completely new network architectures or calculation units is therefore due to practical restrictions mainly limited to institutions with access to large server farms.

Aside from the easily visualizable convolutional layers, it is nearly impossible to analyze the internal workings of a neural network, or trace back decisions to their key contributing features. While it is certainly possible to connect the output of individual neurons to corresponding input data, this is not feasible on a larger scale, and the chances of identifying a simple activation pattern for a single neuron are slim. Therefore, the network has to be treated mainly as a black box.

Classifications can still rely on few visual cues when the training set has some unknown bias towards certain images. After training it is often possible to carefully craft images, which can be very obvious synthetic patterns to a human observer, while the network will identify a trained class in them with high confidence

Figure 7: Three screenshots from a laparoscopic cholecystectomy, during the preparation phase (left), the clipping and cutting phase (middle), and towards the end of the gallbladder dissection (right).

[123]. These generated images can afterwards be used, though, to improve the training set and prevent the network from only relying on simple, unstable patterns in the future.

Most other machine learning approaches require or implicitly assume some kind of feature extraction to be part of the preprocessing of the raw input data. The design and choice of these features is left to the user, though the quality of prediction can depend heavily on a proper selection of features. Neural networks not only train a classification based on features, as part of their training on raw data they develop their own optimal features naturally, which is most apparent in the convolutional layers of CNNs. This might be a major reason for the classification strength of neural networks, which currently outperform all other approaches on public challenges, usually even by a significant margin.

2.2 Surgical Data

All experiments on the recognition of surgical phases have been carried out on anonymized medical data, recorded during real surgeries on various patients. In the following sections the surgical intervention chosen for all experiments, the laparoscopic cholecystectomy will be explained first, followed by a description of the different signals, which were recorded during the surgeries, as well as the annotation of the data.

2.2.1 Surgical Intervention: Laparoscopic Cholecystectomy

The Laparoscopic Cholecystectomy (a minimally invasive removal of the gallbladder) is a well-suited and popular study target in the field of surgical workflow detection [3, 72, 81, 104, 126, 155, 167]. The number of phases and instruments used during this intervention is easily manageable and does not vary too much even across different hospitals. The workflow is mainly linear in the ideal case, though some simple loops and possible exchanges in the order of later phases introduce all challenges relevant for a flexible and robust modeling and detection approach. It is a highly standardized intervention, performed regularly all over the world, with a typical runtime of about 35 minutes, though individual cases can take anywhere from 20 up to 90 minutes. Due to these characteristics, and in order to be able to better compare results, this intervention type was used as basis for all experiments in this work. Some sample views are shown in Figure 7.

This surgery is done under general anesthesia, and without taking the preparation or closure of the patient into account, the core of the surgery can be divided into eight phases [156]. The first phase of trocar placement starts with inflating the abdominal cavity with an inert gas, so that a sufficient working space can be established. Trocars are thin metal tubes with a valve, through which laparoscopic instruments can be inserted into the body easily and without losing the intra-abdominal pressure. The first trocar is inserted blindly after inflation has finished, and the laparoscopic camera is inserted for the first time to assess the surgical site and check for additional, unexpected findings. Three further trocars are then inserted under observation through the camera. Additional trocars may be inserted if complex anatomical structures (e.g. an exceptionally large liver) require additional tools.

After all trocars have been placed, the preparation phase begins. In this usually very short phase, the surgeon approaches the gallbladder and occasionally prepares the surgery, e.g. by detaching tissue obstructing the further approach. Then the preparation of Calot's triangle begins. This area, connecting the

gallbladder to the digestive tract, consists usually of the cystic artery and the cystic duct, surrounded by connective tissue, though in some cases both vessels can be grown together, or a second artery runs as separate vessel in parallel. In this phase, the connective tissue needs to be removed to expose all vessels. In the next phase, each vessel is sealed off through the placement of three clips and separated by cutting between the clips. Most clips used are made of a biodegradable plastic, so they can be reabsorbed by the body over time, though for larger vessels stronger metallic clips can be used. The cut is done between the placed clips so that two clips remain in the body for a safe seal of the vessel, and one clip is removed later with the gallbladder, to prevent leakage.

In the next phase, the gallbladder has to be detached from the liver bed. In most cases the connective tissue, which attaches the gallbladder to the liver bed, is separated by coagulating the tissue with high-frequency monopolar electric current. The detached gallbladder is packaged in a plastic retraction sac in the next phase. This prevents bile liquid from spilling into the abdominal cavity on accidental injuries and eases the later removal of the gallbladder. After the packaging of the gallbladder, the liver, liver bed, detached cystic pathways, and the surrounding tissue are rinsed and checked in the hemostasis phase. If any bleedings are detected, the tissue is coagulated to stop the bleeding and close minor wounds. Finally, in the last phase of the surgery the gallbladder is retracted from the body. This usually happens by pulling the retraction sac closely to a trocar and removing the bag with the trocar from the body, though in some cases, especially when large gallstones are present, the surgeon needs to remove the gallbladder partially. While the retraction sac is pulled out of the body as far as possible, the content of the retraction sac is carefully cut into smaller pieces, liquids are drawn from the bag, and the volume is gradually reduced until the whole bag can be retrieved.

The order of these phases is not necessarily fixed, and the transitions can be fuzzy. If significant bleedings occur at any point during the surgery, a short hemostasis is introduced to stop the bleeding and clean the wound immediately, regardless of the currently ideal phase. During preparation of the surgical site or Calot's triangle, it can become evident that additional tools are needed, so another trocar may be inserted after the initial trocar placement phase. An enlarged and inflamed gallbladder is at a high risk of rupturing during the detachment phase, so the retraction sac may be introduced to partially package the gallbladder very early and minimize the risk of spillage in case of an injury to the gall bladder. The gallbladder may also be removed prior to the final hemostasis phase.

Other phase definitions are undeniably possible and have been used even during some experiments, though the differences between common phase definitions on this surgery are minor, and it is often possible to specify a transformation from other phase definitions to the one given here, or vice versa.

2.2.2 Instrument and Sensor Data

As it has been shown already very early [3, 77, 80], detecting the surgical workflow can be done with high confidence if reliable data about the intra-operative instrument usage are available. Several sensors have been developed and deployed in a hospital OR in order to automatically collect such minimal signals in real-time.

2.2.2.1 Laparoscopic Instruments

The laparoscopic instruments used during the examined intervention are a liver retractor, alligator forceps, PE (or biopsy) forceps, scissors, clip applicator (used with either biodegradable plastic clips or more robust metal clips), irrigation rod, and suction rod (Figure 8). Most of these metallic tools can be attached to the high frequency current generator in order to apply monopolar cutting or coagulation current. Some other, non-laparoscopic instruments are also used, mainly scalpels and needles for preparation of the minimally invasive procedure and sutures for wound closure afterwards, though they are not tracked or used for any further analysis.

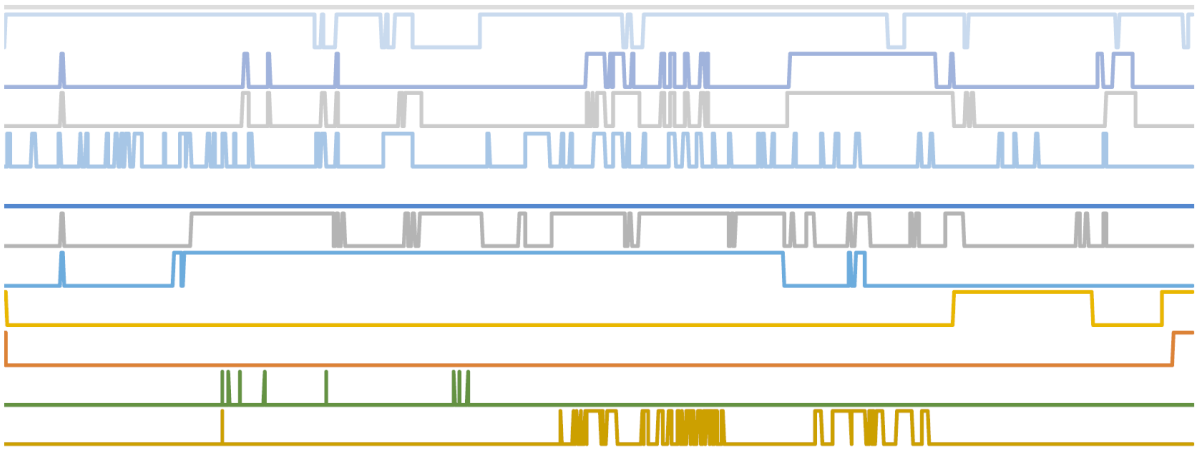

Figure 8: Each line represents a single binary signal over the course of a surgery, active when the line is elevated, and inactive otherwise. From top to bottom: metal clip, suction rod, irrigation rod, scissors, clipping tool, PE forceps, alligator forceps, table light, room light, HF cutting, and HF coagulating.

Several approaches exist in order to detect which instruments are in use at any given time. Many of them rely on visual cues from the laparoscopic or external cameras, which will be discussed in section 2.2.3. Purely sensor-based methods usually attach some non-intrusive markers to the instrument, such as RFID tags [81]. These can be read out from readers included into the trocars, though these trocars are rarely used due to their high costs and minimal immediate advantages. Alternatively, the instrument table can be equipped with large RFID antennas. This way it is not immediately possible to recognize the instruments currently in use in the patient, though by knowing which of all available instruments are stored on the table, it is trivial to extract that information. The disadvantage of this indirect method lies in the fact that experienced scrub nurses tend to predict the next required instrument and prepare it ahead of time, which also removes it from the detected pool of instruments on the table. The signals of the RFID tags are unfortunately not necessarily very robust, and the common usage of metals in the OR and interference from other devices can disrupt the detection quality of the RFID antennas, which can lead to highly noisy data.

2.2.2.2 Other sensors

Other easily deployable sensors can also be used to detect relevant states during the surgery without interfering with the intervention itself or the devices, and without the need for an accessible data interface on devices. By attaching the corresponding bags to weight sensors, it is possible to indirectly measure the amount of irrigation water and vacuumed fluids. The intra-abdominal pressure can be drawn from the insufflation device through an additional pressure sensor, the tilt of the surgical table can be determined

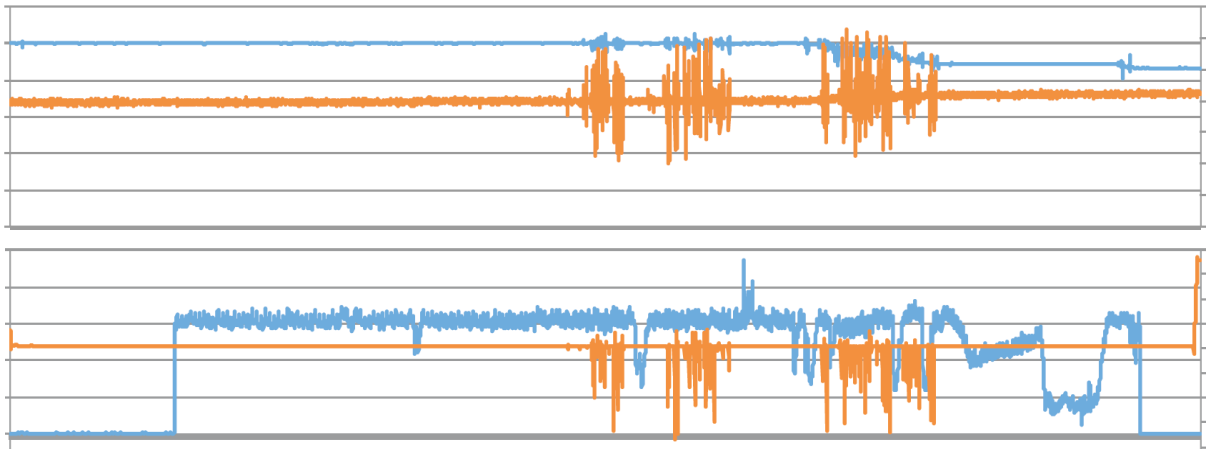

Figure 9: Recorded analog sensors over the time of one surgery. Top row: Irrigation weight (blue) and suction weight (orange). Bottom row: Intraabdominal pressure (blue) and table inclination (orange).

with a rotational sensor (Figure 9). With light-sensing diodes attached to the rim, the status of room and surgical lights can be detected reliably. Additionally, by carefully placing such diodes onto the control lights indicating active usage, it is also easily possible to detect the active modes of the HF generator. Some of these signals are digitally available for some devices or systems (e.g. light status or insufflator pressure), though many closed systems still require external sensing. External sensors also occasionally have the advantage of measuring the actual current state instead of the planned target state, which can provide additional information, as will be described in 2.3.1.1.

2.2.3 Laparoscopic Video

The most critical tool during a minimally invasive surgery are naturally the laparoscopic optics, providing the surgeon with the required field of view and lighting of the surgical site and therefore enabling minimally invasive procedures. In a modern OR, a camera is attached to the optics, so the situs can be viewed in a more ergonomic posture and by all members of the OR team through one or several monitors. Specific optics and cameras, which provide stereo vision and thereby natural depth perception, have become available for laparoscopic interventions, though they are not yet widely adopted.

In most ORs, the camera is operated by the assistant surgeon, though camera control is a field where many lightweight robots are available, which can reduce the necessary surgical team, without the costs, complexity and infrastructure constraints of a full robotic suite. When controlling the camera, the operator must pay attention to the required field of view (e.g. centering the image on specific anatomical structures), the distance from the site, and a level image horizon. The main risk during camera operation is getting liquids like blood or water onto the lenses, which partially or fully obscures the view and usually requires immediate rinsing of the lenses.

The video signal of the laparoscopic camera can easily be recorded and digitized without interfering with the remaining system. Due to its ubiquitous usage and accessibility, analysis of the surgical video is the most common approach in the field of surgical data science [14, 18, 23, 58, 83].

2.2.4 Data Synchronization and Annotation

All recorded data was synchronized, and each synchronized recording was manually annotated. All sensor data was recorded on a single server, so each incoming event was stored with a common timestamp. Different frequencies of various sensors were synchronized by creating a combined measurement for all recorded channels with the highest used frequency and repeating known values for channels with lower frequency. Synchronization between sensor and video data was done by filming the current system time of the recording server with all involved cameras. This way, an offset between the video time and the server clock can be calculated for several frames and averaged, to calculate the offset between each video file and the sensor data. Videos were recorded with a framerate of 25 frames per second, and recorded sensor data was adjusted to match the laparoscopic video frames as closely as possible afterwards, by choosing the closest sensor data frame to each video frame, dropping sensor data in between, or interpolating missing sensor data for unmatched video frames.

An expert, either a surgeon or medical engineer with thorough understanding of both the intervention and surgical workflow detection, manually annotated each synchronized surgery. The annotation always assigned a single-phase label (see phase definitions in 2.2.1) to each frame. Depending on the experiment (see the subsections in 2.3), additionally a ground truth label for each available instrument was given (if no RFID information was available).

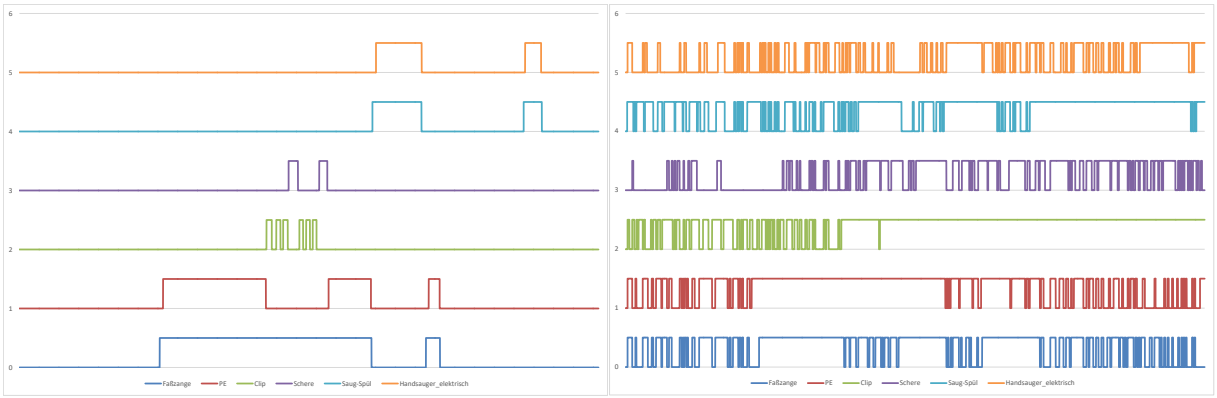

Figure 10: Detected instrument usage signals under heavy noise. The left image shows a theoretically ideal signal progress, the right image an extreme case of the actually recorded data.

2.3 Experiments and Results

Different experiments have been conducted, applying the methods described in 2.1 on various subsets of the available data mentioned in 2.2. An important distinction should be made between methods using only non-video data, those employing video data only, and methods using both data in combination.

2.3.1 Using Instrument and Sensor Data

The following experiments were designed so that the employed methods only relied on sensor data and did not take any visual information from laparoscopic or external video into account. Most of the utilized data are either already digitally available through medical devices (though not necessarily openly accessible at present), or can easily be collected through simple, unobtrusive sensors.

2.3.1.1 Random Forest on Instrument and Sensor Data

Random forests (see 2.1.4) have been used in [155] in order to detect the current of seven surgical phases based on instrument usage and minimal sensor data. For this experiment, a forest size of 50 was chosen due to its good balance between performance and speed. Best results were achieved with a maximal tree depth of 4 nodes, while in each node 4 out of 16 features were randomly selected to choose the best split.

The data used for this experiment consisted of 12 binary and 4 analog signals, recorded as time series throughout the duration of four full cholecystectomy (see 2.2.1), with approximately 60000 samples over all four series. Of the binary signals 8 were indicating instrument usage, 2 documented the status of the HF generator modes (coagulating or cutting), and the remaining 2 gave the status of different light sources (table light and room light). The weight of both the irrigation and the suction bags were recorded as analog signal, as well as the intra-abdominal pressure and the measured inclination of the surgical table. Instrument usage was detected automatically through sterile RFID chips attached to the handles of the laparoscopic instruments and antennas at the instrument table (see also 2.2.2) [81]. Unfortunately, due to severe interferences caused by the multitude of different metallic objects in the OR, the detection was exposed to a high level of noise, which led to a very low signal-to-noise ratio (SNR) for the instrument signals (Figure 10).

Each entry of the time series was labeled with the name of the surgical phase, in which it took place, though in this experiment the phases were defined slightly different. Compared to the phase definitions in 2.2.1, the trocar placement and preparation phases were merged, the gallbladder packaging and retraction phases were always done together and therefore merged, and a new, final drainage and trocar removal phase was introduced, resulting in 7 phases. Since not all phases are of equal length and longer phases are at risk of dominating the dataset and introducing unwanted bias, the samples from shorter phases were artificially boosted by duplicating them in the training set. The experiment was evaluated in a leave-one-surgery-out cross validation, so the forest was trained on 3 datasets and evaluated on the 4th, repeating the

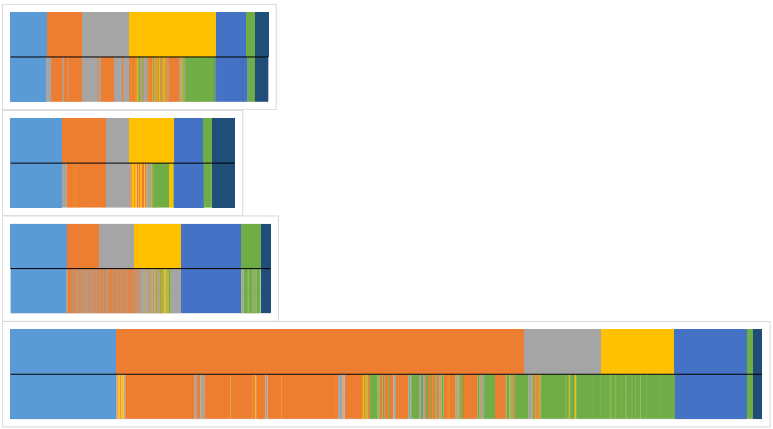

Figure 11: Visualization of four recorded surgeries. Each bar represents one recording, the different lengths are caused by the varying length of the interventions. The top half of each bar denotes the manually annotated phase labels through colors, the bottom half shows the detected phases for each frame.

process so that every surgery was used for evaluation once. Results were calculated by averaging the outcome of all four trials.

The classifier achieved an overall accuracy over all phases of 68.8% and an average Jaccard index of 58.6%. Class specific results are shown in Table 1 and Figure 11. The phases "trocar placement", "gallbladder retrieval", and "drainage and closing" had the most successful detection rates with Jaccard indices of over 97% each, while the phases "gallbladder detaching", "hemostasis", and "clipping and cutting" are the hardest to reliably detect, with Jaccard indices of 11%, 18%, and 26% respectively. It should be noted, that this is caused in part by the fact that bleedings can happen during any phase, so activities to stop the bleeding, which are otherwise typical for the hemostasis phase, can occur at any time. Detaching the gallbladder is also an activity, which is prone to cause minor bleedings of the liver bed, so significant parts of the phase "gallbladder detaching" can produce the same signals (and therefore classification) as the hemostasis phase.

Table 1: Confusion matrix of all classified phases after detection with random forests.

	Trocar pl.	**Prep.**	**Clipping**	**Det. gb.**	**Retr. gb.**	**Stop bl.**	**Drainage**
Trocar pl.	0,995	0,000	0,005	0,000	0,000	0,000	0,000
Preparation	0,000	0,794	0,093	0,016	0,000	0,098	0,000
Clipping	0,000	0,361	0,405	0,013	0,000	0,221	0,000
Detaching gb.	0,000	0,293	0,201	0,113	0,006	0,387	0,000
Retrieving gb.	0,000	0,000	0,002	0,000	0,997	0,000	0,001
Stop bleeding	0,000	0,012	0,105	0,001	0,047	0,835	0,000
Drainage	0,000	0,000	0,000	0,000	0,000	0,001	0,999

A strength of random forests is their ability to allow for in-depth analysis of the data after training the system, like calculating the relative importance of each signal to determine the most and least discriminative for the given task (see 2.1.4.4 for details). As can be seen in Figure 12, the most influential signals were the intra-abdominal CO_2 pressure, the weight of the suction bag, the surgical lamp status, and the table inclination. Both an unused signal as well as a newly introduced random, binary control variable were reliably ranked the lease discriminative. On the other hand, other signals like the usage of the laparoscopic scissors or the clipping device, which tend to be strong indicators for human observers, were also practically excluded from classification, very likely due to the severe noise described above. However, some of the sensor noise in the analog signals are caused by unintended, yet clearly identifiable external sources, such as the surgeons leaning on the surgical table or performing rapid movements. These movements can shake the surgical table, which can be picked up by sensors measuring the exact table tilt, as well as the weight sensors of the fluid bags, as these are often attached to the table. This kind of side-channel information can be used by machine learning systems, hence the high ranking of signals such as the table inclination, which would otherwise only provide very limited information.

2.3.1.2 Random Forest and HMM on Raw and Filtered Instrument and Sensor Data

Several related trials were done and compared for this experiment [154] using random forests (2.1.4) and HMMs (2.1.2). The results of 2.3.1.1 were successfully recreated with a larger dataset first, then the effects of denoising and data augmentation were examined. Finally, HMMs were introduced for their strong modeling capabilities, both directly on the raw data, as well as in combination with the output of the random forest.

Two datasets were used for this experiment. The first dataset consists of sensor and instrument usage data, as well as laparoscopic video, for 5 laparoscopic cholecystectomy surgeries, annotated with phases by medical experts. This is the same dataset and phase definition as used in 2.3.1.1, though one additional surgery has been recorded in the same way after the work of [155]. This dataset is also used later in 2.3.2.1, utilizing the video information. The second dataset consists of sensor and instrument data, recorded in the same manner as the first dataset, of 18 additional surgeries, all annotated by a medical expert, although without the video information. The phase definition of the second dataset was more verbose, including short intermediary phases, which explicitly signaled the completion of the previous phase. For consistency and comparability with other results, these phases have been merged with their respective predecessors

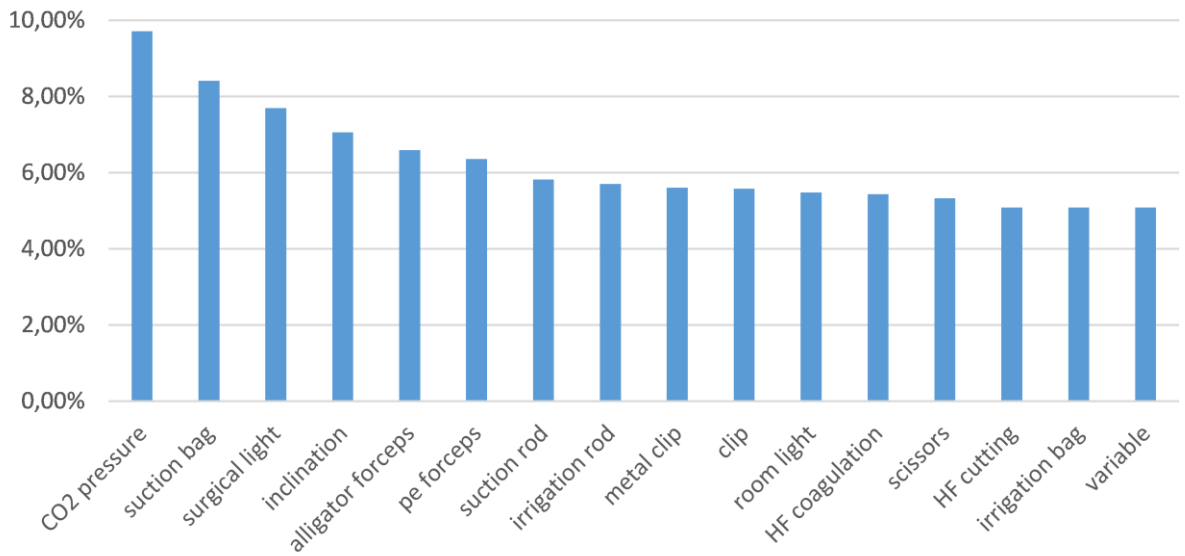

Figure 12: Relative feature importance after training the random forest. Many instruments, such as scissors and clip, only achieve very low importance due to high noise, which makes their signals unreliable.

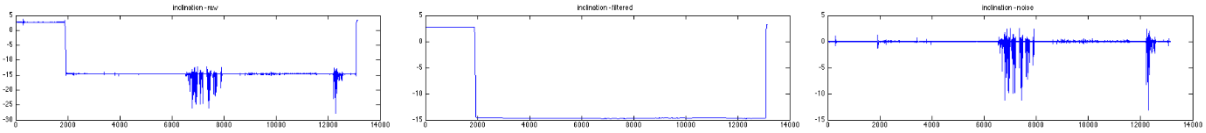

Figure 13: The original, noisy table inclination signal (left), the signal after applying a median filter (middle), and the extracted, pure noise signal (right).

(e.g. “Finished preparation” was merged into “Preparation”). The signals used in this experiment are the same 12 binary and 4 analog sensor signals as described above, with the addition of the elapsed time (in seconds) since the start of the recording.

In an attempt to compensate for noise in the recorded data, the recorded analog signals have been smoothed with a sliding window median filter of size 120. Additionally, the pure noise signal (obtained by subtracting the smoothed signal from the original) is also retained as new feature, as the occurrence of noise can be related to external influences, as described above (see Figure 13). Since median filtering does not improve noisy binary signals, another approach was to augment the binary signals with additional, time-dependent features. Two additional features are calculated per binary signal. The first feature is the cumulative sum of the feature over time, providing a linear and steep increase during usage, and shallow, non-linear increases otherwise due to noise. As second feature, the sum of rising edges is calculated over time, displaying plateaus during continuous usage or noise-free periods of inactivity, and a non-linear increase otherwise.

The optimal parameters for the random forest were determined through an exhaustive parameter sweep. Therefore, the forest was set up to contain 80 trees with a maximal depth of 9 nodes, presenting 8 randomly selected features to each node during training. Impact on training and evaluation speed was also taken into consideration during parameter optimization, although it did not influence the final choice, as all possible parameter combinations were determined to be unambiguously real-time capable. The classification results are evaluated through a leave-one-surgery-out cross validation.

The first experiment practically recreated previous work by applying the random forest on the raw input data, achieving an accuracy of 69.9% and 70.1% respectively on the first and second dataset, and an average Jaccard index of 60.0% and 57.1% respectively. The minor improvement can be attributed to the larger datasets and optimized parameters. Splitting the analog signals of the dataset into a noise-reduced signal and the pure noise signal as described above increases the accuracy on the first dataset slightly to 72.0% and the average Jaccard index to 62.8%. Additionally, providing the time-dependent features on binary signals described above has several ramifications. The overall accuracy changes on the two datasets to 64% and 71.5% each, the average Jaccard index changes to 65.3% and 60.3% on the first and second dataset respectively. This diverging development can be traced back to the fact that the classifications are generally “smoother”, with less rapid changes between neighboring frames, however the short phase “Clipping” has been completely skipped in many tested surgeries.

Since the random forest classifier works on individual frames only, without taking context into account (aside from the augmented, cumulative features), a main reason for the suboptimal performance are “jittering” classifications. Due to their inherent modeling capabilities, a left-to-right HMM was used for classification. In a baseline experiment, where the HMM was trained directly on the raw datasets, the system achieved an accuracy over all phases of 41.8% on the first and 48.1% on the second dataset. The average Jaccard index in these cases were 32.8% and 30.3% each. It is important to note, that during this experiment, many phases were skipped, and the last phase was never reached in many cases. Similar results were obtained when using the filtered datasets, while the augmented dataset decreased the performance on the first, smaller dataset, as the HMM tends to fall into a specific state, from which it will never leave.

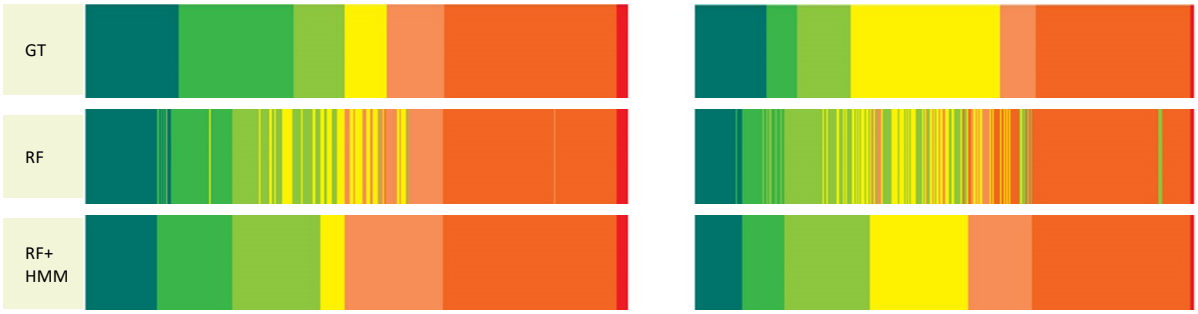

Figure 14: Results of two surgeries (left and right) after classification with the combined RF+HMM approach. The top row shows the manually annotated ground truth, the middle row the preliminary classification as provided by the random forest, and the bottom row depicts the final classification after refinement by the HMM.

The two methods can be combined by classifying the input data with a random forest first, and then use the classifier output as observations in the HMM (Figure 14). The training data has to be split in 3 parts in this case, with the first part used as training for the random forest, the second part being used to evaluate the random forest classifier and train the HMM, and the third part for evaluation of the complete system. Calculating the confusion matrix of the random forest classifier can directly be used as initialization for the emission matrix of the HMM. The combination of these very different methods provides improved results, as the random forest classification prevents the HMM from skipping or overrating phases, while the learned HMM filters the output and provides a smooth final classification. Due to the large amount of required training data, this experiment could only be done on the second, larger dataset. The accuracy of the combined approach on the raw dataset was 80.8%, with noise-reduced signals 82.4%, and with time-dependent features 74.5%. The average Jaccard index on the filtered signals was 71.1%.

2.3.1.3 SVM, HMM and Conditional Random Fields on Full and Reduced Sensor Data

In the work in [41], three related methods to detect phases were compared. A system of regular one-vs-one SVMs with majority voting (2.1.3.2), distinguishing between two classes each, was used as baseline for comparison. The second method also employed an SVM first, but fed the calculated SVM scores as observations into a succeeding HMM (2.1.2). Finally, the third method similarly used an SVM on the signals and feeding the obtained scores as features into a linear-chain conditional random field (CRF). A CRF [82, 158] can be seen as a special case of HMM, but while a HMM systematically tries to model both the observation probabilities and the states simultaneously, a CRF focuses on modeling only the states, given the observations.

The dataset for this experiment consisted of the same binary and analog signals described above, recorded for 42 fully annotated surgeries. In a second trial, the same setup was used, though the data only consisted of the analog signals and the light and HF generator states, while the instrument usage data was removed. This restriction aims to only use sensors, which can be easily deployed and therefore can be used with higher probability in other ORs and hospitals. Each signal of the raw sensor input is augmented in several ways before being presented to the classifier. In an attempt to better identify temporal relationships, the mean, standard deviation, and slopes of linear fits are calculated over varying sliding windows. Four different window sizes of 4, 16, 64 and 256 seconds were used for each metric. Every feature window was defined to end at the current timeframe to avoid using future information, in order to ensure the online capability of the method. Additionally, every feature value was copied from the past to the current sample (using the same periods as for the windowed features) to further enable the detection of temporal relationships.

When using the full dataset (including instrument usage information) the overall accuracies for the three methods are 75.9% for the plain SVM, 73.1% for the combination with HMM and 74.4% for the combination

with CRF. The average Jaccard indices are 61.5%, 58.2%, and 59.0% respectively. As this outcome shows, the additional applications of HMM or CRF do not generally improve performance. The results on the reduced dataset (without instrument data) are highly comparable, with an accuracy of 73.9% for the SVM, 69.6% for the combination with HMM, and 70.4% for the combination with CRF. The respective average Jaccard indices for the three methods on this trial are 58.9%, 50.6%, and 53.9% each. This shows again that no method dominates the others, though reducing the number of available features by focusing on easily deployable sensors only also does not decrease performance to a large degree, while greatly increasing the theoretical availability of the data.

2.3.2 Using Surgical Video

These experiments were aimed utilizing the laparoscopic video, which is by the very concept of minimally invasive surgery always available and easy to access digitally. The raw image data is more challenging to work with, though while they can be augmented by additional sensor data, the goal is to achieve acceptable results without the need for more sensors.

2.3.2.1 Random Forest on Video and Combined Data

The work described in 2.3.1.2 and [154] was also extended to utilize the laparoscopic videos available in the dataset. The random forest was trained and evaluated on image features extracted from the video frames, on rescaled video frames directly, and on the combination of the available sensor data with the extracted video features. The same parameters were used for the random forest as in the other experiments without video information. The dataset of 5 surgeries, the recorded sensors, and the employed phase definition used for these experiments are the same as described above as “first dataset”. The sensor data has not been filtered or augmented in this case. The source videos have been recorded with an image resolution of 352x240 pixel, though due to the legacy recording device, different recordings have black bars of varying sizes at the image borders. To ensure a consistent dataset, the videos have been manually preprocessed by cropping the borders and resizing the resulting frames non-proportionally to a size of 64x64 pixel.

Three different kinds of image features are calculated for each video frame. For all three color channels of both the RGB and HSV representation each, the average pixel value is calculated and used as first feature. As second feature, a bag-of-words (BoW) is created, counting the number of SIFT features [99] detected in equally sized image regions. SIFT key points detected exactly at the image border were ignored in this case. Finally, a HOG descriptor [37] is calculated for each frame and included in the feature vector. The free parameters of the individual image features were again optimized sequentially with a parameter sweep after initial guesses. A grid of 5x5 regions was chosen as base for the BoW, with a SIFT threshold of 0.01. The overlapping cell size used for the HOG was set to 24x24 pixel, with 5 bins used for the histogram. For another experiment, the images were down sampled to a size of either 16x16 or 8x8 pixel, split into the individual RGB color channels, and transformed into single vectors each. Then the three vectors were concatenated and used directly as image representation, without further feature calculation.

Training the random forest on only the calculated image features for each video frame does not yield reliable results. The accuracy over all phases was 38.8%, the best and worst Jaccard index achieved per phase is 29.7% and 5.5% respectively, with an average of 20.4%. Using only the pixel data from the down sampled images produces worse results with an accuracy of only 28.0%. Finally, a random forest was trained with both the extracted image features and the corresponding, unfiltered sensor data. An accuracy of 65.1% and an average Jaccard of 49.8% were reached. While this is a clear improvement compared to using only video features, it does not reach the performance of focusing only on the sensor data.

2.3.2.2 Deep Convolutional Networks on Video Data

The following experiments used deep convolutional networks (as generally described in 2.1.5) on the laparoscopic video stream in order to detect various low-level signals, such as the number, type or location

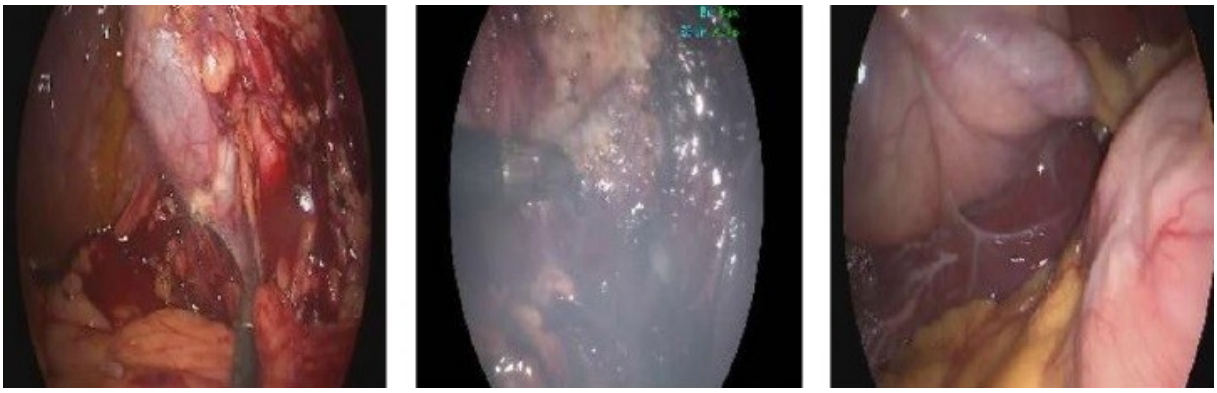

Figure 15: Examplary frames of different classes: blood (left), smoke (middle), and regular (right). The aspect ratio of these images is distorted to achieve a square input as required by the classification framework.

of laparoscopic instruments [153]. No immediate workflow phase detection was done with this approach. However, the methods presented here can be used to improve or even replace other sensors for instrument detection, and as was already shown in [3], reliable information about used laparoscopic instruments can be used directly to accurately estimate the current surgical phase.

The convolutional network VGG-16 [150] was used here, specifically the architectural setup labeled "D" in the original work. This network consists of a total of 13 convolutional layers and 3 fully connected layers, including the output layer. The convolutional layers in this network only utilize small convolutional filters of size 3x3, though larger filter regions can be achieved implicitly by concatenating several convolutional layers directly, without pooling layers in between. This approach has the advantage of introducing additional non-linearities to the convolution, while also reducing the number of parameters compared to a single layer with larger filter size. The convolutional layers are pre-trained on common image classification datasets, while the training and optimization on the laparoscopic data is done only on the final, fully connected layers. The output layer has been resized from originally 1000 output nodes to 3 or 4, to fit the severely reduced classification requirements of these trials. The input images are taken from every 5^{th} frame from the laparoscopic video, downsampled to 224x224 pixel and normalized to a mean brightness value of 0.

The first classification task is to detect the number of surgical instrument visible in the image, in order to support other instrument detection methods. A dataset of 10 recorded surgical videos was split into three parts, containing images with none, one, or two visible instruments each. The correct number was detected in 71.5% of all cases. Another similar experiment was performed to detect the presence or absence of smoke from the HF coagulation device or excessive bleeding from injuries. Again, a dataset was created by splitting video frames into three categories for smoke, blood or none. The accuracy in this case was close to 100%, as this very simple classification mainly depends on very prominent, global color changes (see Figure 15). The absence of both blood and smoke was trained with regular laparoscopic images from inside the body, while external views (e.g. before the first cut or after removing the camera from the body again) were not part of the training set. However, anecdotal evidence suggest that these images can also be correctly identified, as none of the three trained classes received high confidence scores when presented with a sample image taken outside the body. As more challenging task, the network was trained to not only detect the presence of instruments in general, but to also detect the type of instrument. For this setup, images with visible instrument were selected and categorized if they showed either the PE forceps, the irrigation rod, the clip applicator, or the laparoscopic scissors (see 2.2.2.1). On this approach, an accuracy of 73.0% was achieved.

Detecting the instruments not only by presence, but generating bounding boxes around them, can create more meaningful data (e.g. to estimate the interaction between tools and anatomy in future work). For this purpose the VGG was embedded into the Faster R-CNN framework [134]. This introduces an additional region proposal network (RPN) to suggest possible target objects, while the VGG is then used to predict

their corresponding classes. Through an alternating 4-step training process between the RPN and the VGG, their convolutional layers can be shared, which improves the detection of objects fitting the trained categories and drastically reduces classification time. For this experiment, three diverse classes were defined for the gallbladder, surgical clips, and the retrieval bag. A total of 1324 images showing any of the mentioned objects were selected and annotated by drawing bounding boxes around the corresponding objects. The network achieved an overall accuracy of 81.8%, while the gallbladder detection scored highest with 99.8%, and the clip detection scored lowest with 75.5% accuracy. The main source of misclassifications were multiple, overlapping bounding boxes detected for the same object.

2.3.3 M2CAI 2016 Challenge

The advance of deep neural network structures in different fields of image classification have also led to a widespread application of this technique to the medical domain. It is mainly used for variations of instrument or anatomy detection, however inspired by the popular tradition of the computer vision community, in 2016 Nicolas Padoy, Andru Twinanda, and Ralf Stauder organized the combined workshop and challenges on Modeling and Monitoring of Computer Assisted Interventions (M2CAI)[3]. As part of this workshop, two related challenges were offered on two separate datasets, but both of manually annotated laparoscopic videos of cholecystectomies. One challenge was about the automatic detection and identification of surgical instruments in the given videos, while the other was aimed at correctly segmenting the videos into the surgical phases. The focus of this section is on the latter, workflow recognition challenge.

The dataset of the workflow detection challenge consists of 41 laparoscopic videos, collected in two hospitals in Strasbourg and Munich, as described in [156, 167]. The videos were manually annotated into the same eight phases as described in 2.2.1. In order to keep the challenge close to possible clinical applications, all submissions were required to perform an online analysis of the data, meaning that for any given frame of a video, knowledge about all previous frames could be taken into account, but no following frames may influence the classification. On the other hand, a classification was considered correct, if it matched a ground truth label within a 10 second window, since phase transitions can often not be clearly related to single points in time.

A total of five teams participated in this challenge, of which 3 teams employed variations of deep neural networks directly, while a fourth team used a CNN as feature preprocessing for random forests. Only one team did utilize machine learning techniques other than neural networks. During the challenge, only the labels for a subset of 27 training videos were given to the participants, while their classification results on the remaining 14 testing videos were compared to the ground truth by the organizers. The Jaccard index was calculated for each phase and each team, and the average of the Jaccard indices for each team defined their final score[4].

The submission by Sahu et al. uses a multi-step classification process [141]. First, a pre-trained AlexNet CNN with adjusted fully connected layers is used for feature generation. The output of this network is then used as input to a random forest classifier, of which the output is fitted to a time series cluster by Gaussian distributions. Finally, this time series prediction and the initial phase estimates are refined to the final phase prediction by a group of random forests, of which each forest is trained to specifically identify only a single phase. This setup could reach a mean Jaccard of 45.0.

The contribution of Dergachyova et al. was based on an earlier work in [39]. In this approach, first a generic model is built from the observed phase transitions of the annotations. Then different image features are calculated for each video frame, which are used together with the extracted models to train an AdaBoost

[3] For more information visit http://camma.u-strasbg.fr/m2cai2016/.
[4] An updated ranking on this dataset can be found at http://camma.u-strasbg.fr/result-list-m2cai16-workflow.

classifier. As last stage, a generative Hidden semi-Markov Model is trained on the AdaBoost response signatures to conserve the temporal relationship between phases. As only submission without any usage of deep learning methods, they achieved a mean Jaccard of 51.5.

Twinanda et al. also base their contribution on the AlexNet architecture [166]. For the workflow detection challenge, the outputs of the second to last layer of this network are taken as input features for a multi-class SVM to calculate preliminary phase assignments. These are refined through a hierarchical HMM for the final classification. This architecture (called PhaseNet-m2cai16 with HMM in the describing paper) reached a mean Jaccard of 64.1 in the challenge, yet other, improved versions are also presented with mean Jaccard indices up to 69.8.

The team of Cadene et al. placed second with their two-step approach [31]. In a first step, they evaluated several pre-trained CNN architectures on their capabilities to correctly assign video frames from the surgical videos to their corresponding phases. Then as temporal smoothing, they average the classification output over the last 15 frames. They achieved a mean Jaccard of 71.9 during the challenge, yet they also improved their approach afterwards and can reach a Jaccard of 81.6 by using a HMM in the second step instead of the simple averaging.

The winning submission by Jin et al. uses LSTM units to create a recurrent CNN capable of capturing the temporal relation of phases without major post-processing [68]. The first layers of the network are based on a pre-trained ResNet-50 [60]. Again skipping the final layer, the obtained features were fed into an LSTM block to consider temporal information. As only post-processing, medically impossible phase transitions (e.g. trocar placement after dissection or gallbladder retraction before gallbladder packaging) were filtered out afterwards. With this approach they managed to reach a mean Jaccard score of 78.2.

2.4 Discussion

Several possible technologies for surgical workflow detection were briefly introduced in this chapter. By highlighting their respective strengths and weaknesses, it became apparent, that no universal “best” method exists. Due to their comparable performance, all techniques can reasonably be applied to this task. However, the diverse capabilities and secondary features of the different approaches can provide valuable advantages or additional insight to specific challenges. Most of the presented methods can work on incomplete data and are therefore in principle able to provide online classifications during a running surgery. Additionally, most procedures can classify a given data sample in real time, while others require more calculation time. Yet given the loose time constraints for practical workflow applications in ORs of up to a minute, all methods are capable of providing a result within acceptable response times. The decision, which method to apply to any given task, therefore has to be made based on availability of data and required secondary objectives.

DTW requires relatively high quality, low-dimensional data, although the fact that phases are not detected through classification, but by warping the examined surgery path to a known, average sequence, interpolation between phase transitions can be done with higher confidence, and one can gain insights into the processes at a higher resolution than the originally recorded phases. Of all methods presented here, only HMMs actually produce an explicit model. This to some degree requires domain knowledge to set up labels and transitions in a meaningful way, the resulting model can however immediately be used for further applications. Random forests do not provide any extractable model on the other hand, although they can handle even noisy data very well. On account of their selection of suitable splitting functions, it is very easy to calculate the most impactful elements of the provided data, which can support a deeper understanding of the underlying processes or suggest areas for further exploration. CNNs finally provide a strong advantage when handling video streams, as they currently clearly outperform other methods in this

area, even without the need for manual parameter optimization, despite their substantial training and classification times.

While phase detection is theoretically possible based solely on instrument information, this information is difficult to obtain automatically and in sufficient quality within the OR. For safety reasons only unobtrusive sensors can be employed, which are prone to external noise or interference from other instruments. Such signal noise can pose a problem to all approaches, which led to the random forests even effectively ignoring many instrument signals. Although it was still possible to get good results with only the remaining sensors, a major finding of these experiments was the concept of using simple, physical sensors where possible, as one might collect unexpected but valuable side-channel information.

A data source alternative to simple sensors attached to instruments is the surgical video. This approach is inherently advantageous, as no additional preparations need to be done, since the camera stream is a necessary core element in laparoscopic surgery and therefore by definition always available. Analysis of this video stream is not trivial, though, but recent advances in the field of computer vision, object recognition, and scene understanding can be applied to intra-operative images as well. The successful application of deep learning methods to simple tool detection and immediate phase detection have been shown here, together with a short summary of the M2CAI 2016 challenge results. Many further aspects however, such as the detection of anatomical structures and a more fine-grained instrument detection still provide interesting future research areas.

3 Event Impact Factors

The operating theatre is a very complex environment. Different afflictions require different interventions, each with their own planned workflow. Several people are involved, performing these interventions slightly differently based on their training, experience, or team composition. Various instruments, tools, and devices are being applied, which will also change based on availability of resources or new technological developments. Additionally to all performed actions in the OR, outside events can influence the situation. While all these variabilities can make identification of key elements highly challenging, each gesture can signal important changes, such as phase transitions or emergencies. The signaling effect of an individual gesture can vary greatly with its context, however. As an example, the act of the scrub nurse offering a new pair of surgical gloves to the surgeon is a completely normal step within the standardized preparation procedure during the very beginning of a surgery, before the first incision is made. Although if this event happens at a later stage, during a more critical surgical phase, the same gesture can reveal a severe problem and indicate a high risk of infection for both the surgeon and the patient if the surgeon's hand was accidentally cut. Therefore, it should be obvious that leaving out such contextual information only risks masking important cues.

As such events outside the regular surgery pattern can directly and indirectly through increased stress levels affect the outcome of the intervention and even the safety of the patient, it is critically important to be able to recognize and correctly evaluate such situations. Not surprisingly, several articles have been published to date, discussing the assessment of distractions [61, 146] and other stressful events [9, 10, 147] in the operating room.

Identification and analysis of such events requires considerable manual effort, yet does not guarantee to produce reliable results. At best, one can hope to produce a look-up-table for previously observed, specific situations, while in the worst-case crucial but non-obvious elements might be ignored. Due to large number of people involved, surgical phases under study, and instruments in use, an exhaustive prediction of all possible combinations is practically not feasible either. A method is required, through which as many contextual cues as possible can be expressed in numerical terms. These terms can then mathematically be combined in an approach to calculate their relative importance.

One application where such detailed analysis is necessary is the evaluation of novel medical devices. Manufacturers of surgical devices reasonably want to test their new products in real ORs to collect valuable feedback from the operating team. As the OR is a very expensive environment, the goal is often to extract as much information from as few surgeries as possible. Identifying relevant feedback from a small sample group can be problematic however, since outside factors, such as changing team composition and varying affinity for technology, greatly vary and influence the collected data. Based on the work already published in [124], this chapter will introduce methods to handle the mathematically related group decision-making problem and adapt it to ranking the different events occurring inside the OR, so we can compare the proverbial "apples and oranges".

3.1 Method

In an attempt to model the relationships within the OR, [16, 90] introduce the idea of viewing everything in the OR from different perspectives. Especially the OR domain model by Bigdelou et al. uses three different views, for the acting human role, the used device or function, and the temporal workflow phase. Then mapping tables are defined between each pair of views and manually filled with observed interactions (e.g. a nurse handling surgical gloves in the preparation phase of an intervention would be counted for the mapping pairs "nurse – gloves", for "gloves – preparation", and for "preparation – nurse" each).

The method introduced here is based on this idea, although an arbitrary, but larger number of simple “view” functions will be employed to numerically evaluate each event. Each resulting value is then normalized and combined to a single factor, using the methods of group decision-making introduced below.

3.1.1 The Group Decision-Making (GDM) Problem

In the area of operational research, the term group decision making (GDM) or multiperson decision-making (MPDM) describes the problem of reaching a common decision among a group of people. From a list of alternatives, the “best” option needs to be found for the group as a whole. The term “best” is intentionally not defined here, as involved people have their own, personal opinion on what this should mean. Therefore, it is assumed that every group member ranks all possible options differently according to their preferences. When buying a car for example, people factor initial costs, mileage, brand name, test reports, color, available extras etc. with different weights. The goal of GDM is to quantify these individual rankings, normalize them to a common scale, and combine them to reach a consensus that best matches all personal priorities.

This field has been under research for several decades by now [11, 139]. A major application of GDM are consensus systems for communities [6, 7, 160], although the concept of supporting decisions among several people has also been applied to the medical sector [129, 174]. The approach used here has been adapted mostly from the works of Herrera et al. [63] and Herrera-Viedma et al. [62]. An important parameter of the presented method is the choice of similarity measures and combining operators. As we will focus our efforts on the adaptation to a different solution, the most basic operators have been utilized here. Detailed descriptions and comparisons of other options are given in [32, 106].

A GDM in this subject is defined as a decision situation, for which a group of people (often referred to as experts) provides information, which provide the base to choose one or several options among a set of possible solution alternatives to a given question [63, 139]. Usually this preference information can be given in any of these three types of preference structures: preference ordering, utility function, or preference relation. As the simplest option, in the preference ordering all possible alternatives are arranged in a list from best to worst. No additional information is given, so it is not possible to differentiate varying gaps of confidence between neighboring positions or define ties of equal rank. Utility functions assess each alternative based on an external value, such as physical qualities or monetary value. As such, they may not always be available, yet they can provide more finely granulated insight and tend to be more objective. Preference relations finally provide the most detailed information, as all alternatives are directly compared pairwise. While this method is the most elaborate to collect, it even allows for intransitive relations for some alternatives.

Different experts can use different preference structures on the same topic, so all collected information has to be unified, before it can be combined for the selection process. Since preference relations can store the most information, a viable approach is to convert both the preference orderings and the utility functions to this format [63]. The details of these methods are explained in the coming sections, when they will be applied to the ranking of surgical events. Then the different individual preference relations can be aggregated to a single collective preference relation exploiting the multiplicative nature of the relations. Additionally, it is possible to influence the outcome by employing fuzzy logic operators during this step, such as “most” or “as many as possible”, to apply different relative weights to each value. The second step in the selection process is the exploitation, where the collective preference relation is converted to various choice degrees, again supporting fuzzy operators. Two possible choice degrees are the multiplicative quantifier guided dominance degree (MQGDD) and the multiplicative quantifier guided non-dominance degree (MQGNDD). Given the provided expert feedback and chosen fuzzy majority quantifiers, the MQGDD calculates how much a given alternative dominates all other alternatives, while the MQGNDD calculates for

each alternative the degree, that it is not dominated by others. The best alternatives to choose for the group as a whole are the ones with highest values in both choice degrees.

3.1.2 Adjustments and Application to Surgical Data Science

When applying the ideas described above to the ranking of surgical events, the mathematics will be used to evaluate practically every gesture throughout an intervention by multiple aspects and combine them to a combined factor. This allows us to number the relative importance of each activity and compare them based on this factor.

As described in more detail below, it is important to introduce new terminology for the calculation of impact factors to prevent confusion with similar sounding, yet different concepts of GDM. It is critical to note here, that the goal of this method is to combine several weak functions to a stronger score, and not to make a consensual medical decision of any kind within the OR.

3.1.2.1 Surgical Events

The surgical event *e* can be defined according to the requirements of the conducted experiment. A rudimentary approach is to consider every gesture performed by every person inside the OR. As this can be difficult to record and handle, it may be reasonable to limit the definition to only include specific surgical areas, phases, or persons. A suitable approach for evaluation of a prototypical medical device can also be the limitation to only take feedback and comments for the tested device into account [16]. Investigating distractions inside the OR for type, source, and possible impact on patient security is also a topic of major research interest [61, 146]. Each instance of an identified distraction can be taken as event to further help analyzing the impact of these distractions. A comparison and detailed explanations of how the elements of other works can be matched to surgical events are given in [124].

Every surgery can be seen as a set of events.

$$E = \{e_1, \dots, e_n\}$$

Each event[5] itself is a tuple of various components, based on different views.

$$e = (c_1, \dots, c_l)$$

Therefore, the components of an event can represent diverse aspects of the OR, for example the involved person (e.g. the surgeon or a nurse), a relative or absolute time component (e.g. during the dissection phase), the utilized instrument or device feature (e.g. ultrasound imaging or laparoscopic scissors), or the affected anatomical structure (e.g. gallbladder). Other components are imaginable, depending on the performed experiment.

3.1.2.2 Component Characteristic Functions

Components of events are usually rather generic, text-based annotations, which cannot be directly used for further calculations. Hence, functions need to be defined, in order to assign numerical values to each event based on the various components. Each function usually only evaluates a single characteristic of each component (e.g. the age or the years of experience of a person for such a component), although multiple components can be taken into account. Conversely, every component should be taken into account by some function, and multiple functions can and should be defined per component, evaluating different characteristics.

[5] From a purely mathematical perspective, the events defined here correspond to the alternatives in GDM, as numerical values will be assigned to them for further calculations.

We will define three different types of component characteristic functions (CCF) with varying degrees of discrimination power and required collection effort[6]: ordering, rating, and pairwise comparison.

The CCF-Ordering (CCF-O) defines a strict total ordering over all events based on a specific characteristic. In this case, an assigned value of 1 defines the "best" result. For the remaining events, values increase by 1 each, in decreasing order according to the examined characteristic. The performed experiment should define the meaning of "best", which can denote e.g. fastest, cheapest, safest, or other relevant meanings. This type of CCF is best suited for quickly obtainable feedback (for example through surveys with multiple domain experts), as a large range of possible options can be arranged quickly. On the other hand, this structure provides the least information, as it is by definition not possible to denote equal importance or relative distances in importance between different events. This constraint, although, has no mathematical background, as the further calculations explained below can also be done for multiple possibilities with the same order, assuming the remaining variables are handled carefully, in order not to lose their expected relations. Such a (non-strict) total ordering is still more limited than other CCF options, so if equality between events is expected or even desirable, a better suited rating (see below) can likely be defined.

A function of type CCF-Rating (CCF-R) assigns a utility value to each event. This can convey more information than a CCF-O, is both ties and varying distances can easily be defined between events. CCF-Rs may not always be feasible, although if possible, they usually are the most straightforward and objective to collect. Physical measurements are common ways to obtain CCF-Rs, e.g. age, time, costs, pressure, power consumption, and others. Higher values must indicate more relevance. The values need not be normalized or bounded, yet they must never be exactly 0 to facilitate later conversion; practically a suitable ε value should always be possible instead of 0.

Finally, a CCF-Pairwise comparison (CCF-P) is the most elaborate and discriminative form to evaluate event characteristics, by comparing all possible pairs of events individually. This theoretically even allows for intransitive relations, where superiority of an event e_a over e_b, and of e_b over e_c, does not necessarily imply the superiority of e_a over e_c. A major advantage of this property is the ability to handle missing data. In such cases, every compared pair of events, where at least for one event an examined characteristic is missing or otherwise cannot be evaluated, is rated with the neutral value of 1. Then an event e_a with missing data can have comparison values of 1 with other events e_b and e_c each, while these compare to each other with any possible value. This approach does not violate any constraints and requires no obscure placeholder values. CCF-Ps are stored as a unitriangular matrix $P \subset \mathbb{R}^{n\times n}$, where a historically defined [140] value range of $m_{ab} \in [1;9]$ defines varying degrees of dominance of event e_a over e_b, while the multiplicative reciprocals $m_{ab} \in \left[\frac{1}{1};\frac{1}{9}\right]$ define the opposite. A value of $m_{ab} = 1$ signifies equal importance between these events.

3.1.2.3 Component Characteristic Matrix

For further calculations, all these types of CCF need to be converted to a common structure. To this end, we define the component characteristic matrix (CCM). While a CCF can be defined independently from recorded events, the CCM always requires a known set of events, as each entry of the CCM depicts a comparison between two events. A CCM is a diagonally symmetrical matrix $M \subset \mathbb{R}^{n\times n}$ with multiplicative reciprocal elements, so that $m_{ij} \cdot m_{ji} = 1$. As before, a value $m_{ij} > 1$ indicates that event e_i has higher relative importance than event e_j, according to the examined characteristic. A value $m_{ij} < 1$ signifies the opposite, and $m_{ij} = 1$ shows, that both events are equal based on this characteristic. Similar to the definition of CCF-P, the values of a CCM are bound to $\left[\frac{1}{9};9\right]$.

[6] CCFs correspond to the preference structures an expert would give in GDM; the three variations here are taken from [63] and adjusted to our approach.

In order to best capture the information carried by the ordering of a CCF-O, every value m_{ij} of the corresponding CCM should depend on the difference of ordering of both compared events $o(i) - o(j)$, so that larger differences between the ordering of two events are represented analogously by larger differences in the resulting importance value. In order to preserve this relationship between orderings, yet still match them onto the allowed value range for CCMs, we first need to invert and normalize the ordering values to substitute values:

$$s_i = \frac{n - o(i)}{n - 1}$$

These substitute values range from 0 for the event ordered in last place, to 1 for the highest ranked event. The obtained difference scale $s_i - s_j$ now has to be fitted to the constraints for the values of a CCM, such as value range and multiplicative reciprocity. As shown in [63], among the functions fulfilling these constraints are exponential functions, so the simplest transfer function from CCF-O to CCM is given by:

$$m_{ij} = f\big(o(i), o(j)\big) = 9^{s_i - s_j}$$

These formulae still produce valid results, even if multiple events are assigned to the same order, contrary to the original assumption of a strict total ordering. In the case of equally ordered events, one must pay attention to replace the number n of all events with the number n' of all unique order positions in the formula above.

The utility values of a CCF-R also need to be converted, in order to form a CCM. The definition of a CCF-R is not bounded, so a prior normalization of the related utility values may not always be possible. Since the values of the CCM are always based on the compared impact between two events, A reasonable approach for the conversion is to set the two relevant utility values in relation to each other. The ratio between these values is an apparent solution for this, and as proven in [63], belongs to a family of functions suitable for this situation. Therefore, possible conversion functions from CCF-R to CCM are given below, with the trivial value of $z = 1$ being a valid solution:

$$m_{ij} = h\left(u(e_i), u\big(e_j\big)\right) = \left(\frac{u(e_i)}{u\big(e_j\big)}\right)^z, \qquad z > 0$$

It is apparent, that this ratio can easily exceed the defined value range of CCMs, even if the utility values were normalized before. Hence after calculating the preliminary values of the CCM, these need to be transformed to fit the defined range of $\left[\frac{1}{9}; 9\right]$, while maintaining the reciprocity of the values and the consistency of the original rating. Let the values of the CCM after conversion be within $\left[\frac{1}{a}; a\right]$. As shown in [62], all values $m_{ij} \in CCM$ can now be normalized through the following function:

$$norm(x) = x^{\frac{1}{\log_9 a}}$$

Lastly, the CCF-P can be converted to a CCM in a straightforward way. Since the CCF-P is already defined as triangular matrix in the proper value range, only the remaining 0 values of the matrix need to be filled, following the multiplicative reciprocal property:

$$m_{ij} = \frac{1}{m_{ji}}$$

3.1.2.4 Event Impact Factor Calculation

After obtaining all separate CCMs $\{M^1, \dots, M^r\}$, they must be combined to a single structure for the final calculation[7]. To this end, the different CCMs can be merged to a collective component characteristic matrix (CCCM) M^c. Suitable functions for combining each matrix value in this step are the arithmetic or geometric mean. It is possible, however, to use the more generalized form of ordered weighted arithmetic (OWA) or geometric (OWG) operators with custom weighting vectors [63, 106]. In this work, we will apply the geometric mean as special case of an OWG:

$$m_{ij}^c = \phi^G\left(m_{ij}^1, \dots, m_{ij}^r\right) = \prod_{k=1}^{r}\left(m_{ij}^k\right)^{\frac{1}{r}}$$

The newly created CCCM represents the combined comparisons between all pairs of events, taking all considered characteristics into account. Now all comparison values for each event can be combined, in order to achieve a single rating, depicting the importance of each event over all others. Similarly to the CCCM, different operators can be applied here. Here we will again use the geometric mean.

$$I_i = \frac{1}{2} \cdot \left(1 + \log_9 \phi^G\left(m_{ij}^c, j = 1, \dots, n\right)\right)$$

The vector I contains the event impact factors (EIF) for all examined surgical events[8], where higher values indicate more importance or a larger surgical impact. This value allows measuring and comparing the impact each event had on the total surgery. It is clear, that the choice of characteristics and the related CCFs considerably influence the outcome of the EIF calculation. If all CCFs are based e.g. on costs and time, then the EIF indicates combinations of these, without taking patient safety or outcome into account. Great care has to be taken when defining an experiment, so that all intended aspects are represented properly in the employed CCF.

3.2 Data and Experiments

For compatibility and reusability of the data, this method was evaluated on newly recorded surgeries comparable to those in 2.2. Specifically the surgery type for these experiments was chosen to be the same with laparoscopic cholecystectomies. The laparoscopic video of each surgery was also recorded and synchronized with the other recorded signals. The video data was not used directly to generate measurements or CCFs, but instead was used for manual segmentation into workflow phases, annotation of used instruments, and ground truth annotation with an accuracy of 1 second each (see also 3.2.3).

3.2.1 Pupil and Heart Rate Measurements

The eye-tracking headset “Pupil” [71] was used to track the motions of a single eye of the head surgeon during surgery. While the headset is capable of recording the surrounding environment and calculating the gaze point after calibration through a second camera, this functionality was not used in this work. Using the

[7] The first part of this calculation, creating the CCCM, matches the aggregation step in classical GDM, while the latter part, the computation of the EIF, resembles the exploitation step.

[8] The EIF vector corresponds to the MQGDD as calculated in GDM. Another similar vector, called MQGNDD, is usually calculated and combined with the MQGDD in GDM to retrieve the top alternatives. However, here we are not interested in only the top results, but a numerical value for all events, therefore it is not necessary to calculate the second vector.

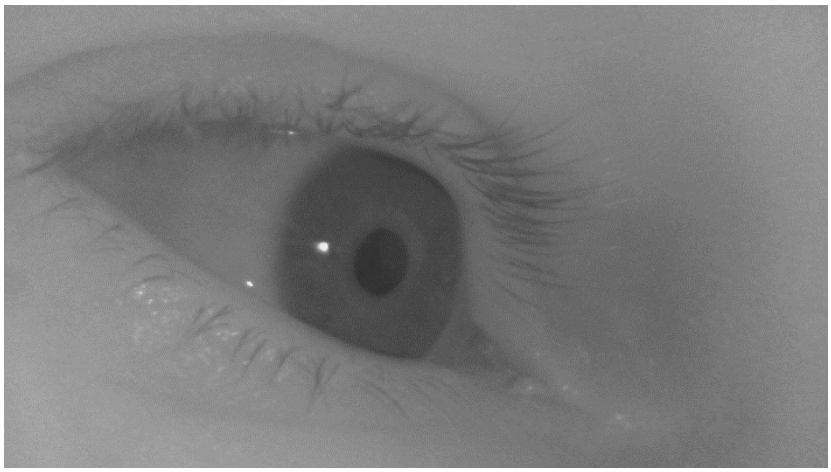

Figure 16: View from the eye tracking headset into the eye. The pupil movement as well as its size in pixel coordinates are extracted from the video stream and used for further calculations.

provided SDK functions the relative 2D position of the pupil as well as its diameter in pixel space were recorded with a frequency of 30 Hz (Figure 16). After the experiments, these data were processed to calculate derived values. One derived value was a simple median calculation of the pupil size over a sliding window of previously recorded frames. Another value was the movement distance between frames, summed up over the same sliding window of past frames. For the experiments in this work, a sliding window of 120 seconds was chosen, to filter out noisy signals and stay in the same temporal order of magnitude as other included signals.

In order to record heart and breathing rates, the wireless Zephyr BioHarness data acquisition system was used (BIOPAC Systems, Inc., 42 Aero Camio, Goleta, CA, USA). Due to available hardware, up to three members of the surgical team could be equipped with a sensor, which transferred the recorded data via Bluetooth to the recording PC. Other than heart and breathing rate, the harness also measured and recorded posture as angle (in degrees) between the torso and vertically up, where negative values denote leaning forward. All measurements were sampled with a frequency of 1 Hz, and a sliding median over the previous frames was calculated. The same window size of 120 seconds was chosen as above.

The data of the heart rate monitors as well as the eye-tracking headset were recorded on the same laptop computer, so that the recorded timestamps for both systems are identical (up to data transmission and processing delays). The system clock of this recording PC was regularly synchronized via wireless network connection with the recording server of the OR, which received and stored the signal data described in the last section. This ensures synchronous timestamps for all recorded data up to our intended accuracy of 1 second.

3.2.2 Definition of CCFs

The CCFs defined for this experiment all aim to highlight possible risks inside the OR, therefore leading to an EIF vector that focuses on patient safety. These CCFs can be grouped by the frequency, with which their values are expected to change.

The four fastest-changing CCFs are based on the sliding window calculations of the heart rate and eye tracking data described above. The first two functions are rating functions, using the median heart rate of the main surgeon and the assistant surgeon respectively as utility value (named CCF-R $_{\text{HR surgeon}}$ and CCF-R $_{\text{HR assistant}}$ each). The other two functions also perform ratings, based on the median pupil size and the eye movement of the acting person, as depicted above, for CCF-R $_{\text{pupil size}}$ and CCF-R $_{\text{pupil movement}}$. While the sliding window of these functions is 120 seconds, they can be calculated at any point in time, so their values can change each second. The functions utilizing the pupil data can for obvious reasons only be evaluated on events linked to the surgeon wearing the eye-tracking headset. Therefore, a second evaluation layer is

added for these functions to handle the missing data in other events. As first step, the CCF-Rs are evaluated on all suitable events only and converted to CCMs. Then a CCF-P is defined for each previous CCF-R as proxy function and used during the actual congregation. When either CCF-P is evaluated for events, which both provide pupil data, the corresponding value of the originally derived CCMs is taken. For comparisons, which include events without pupil data, always the neutral value of 1 is used. This is possible due to the intransitive nature of CCF-Ps explained in 3.1.2.2.

A single function has been defined on the characteristic of the used instruments. As some instruments are used only briefly, while others can be used for several minutes continuously, the value of this function also changes approximately every few minutes. Every instrument used during the observed surgery type was graded by a medical expert with regards to the likelihood of causing unintentional injuries, subdivided into three grades of "severe", "medium", and "low". Since multiple instruments are expected to receive equal labels, a rating function seems the likely choice, with utility values of 1 through 3 in increasing severity. Rating functions however take the difference of the utility values into account, which provides no actual meaning for such constructed values. Therefore, this situation provides the rare opportunity to define a CCF-O using a non-strict total ordering. The CCF-O $_{\text{instrument}}$ assigns the rank 1 to instruments labeled "severe", a rank of 2 to instruments with a "medium" risk of injury, and rank 3 to the remaining instruments of "low" risk. Additionally, the rank of "no risk" with value 4 is introduced for situations, where no instrument is in use. Since this does not represent a strict ordering anymore, the substitute values defined in 3.1.2.3 must not be calculated using the total number of events n, but the number of unique ordering ranks, in this case $n'_{inst.} = 4$. The available instruments as well as their respective risk assessment are given in Table 2.

Table 2: All available instruments and their relative risk levels. A level of 1 denotes a high risk of accidental injury to the patient, while a level of 3 indicates a very low risk of injury. The value 4 (no risk at all) is reserved for cases, in which no instruments are present.

Instrument	Trocar	Liver rod	Alligator forceps	Biopsy forceps (PE)	Clip applicator
Risk level	1	2	2	2	2
Instrument	Scissors	Irrigation rod	Electrified suction rod	Retraction bag	None
Risk level	1	3	2	3	4

Two further functions were based on the workflow phase, in which the event occurred. The first function CCF-R $_{\text{phase duration}}$ is a simple rating function, using the average length of each workflow phase (in seconds) as utility value for the characteristic. The second CCF-O $_{\text{phase risk}}$ is once more based on a simple survey of a medical expert, who was asked to arrange all workflow phases again by the risk of involuntary injuries to the patient. As the workflow phases are more differentiated, equal ranks are less likely, so this function specifies a strict total order. Table 3 repeats all distinct workflow phases and gives their respective average durations and the risk valuation by the expert.

Table 3: All surgical phases for this intervention type, in their typical temporal order. The risk rank assigns a strict ordering, where the phase of rank 1 has the highest risk for patient injuries, while the phase with rank 8 has the lowest risk.

Phase	Risk rank	Average duration (s)
Trocar placement	2	155
Preparation	8	28
Calot's triangle	1	563
Clipping & Cutting	3	404
Gallbladder dissection	5	569
Hemostasis	4	449
Gallbladder packaging	6	84
Gallbladder retraction	7	372

Finally, a few functions were created on characteristics, which do not change during surgery, but can be helpful to compare events across different patients and surgical teams. One ordering function evaluates the staff members' experience level (divided into 5 levels from novice to expert), while two other functions rate events based on the utility functions calculating the patient's body-mass-index (BMI) in $\frac{kg}{m^2}$ and age in years, respectively. Table 4 shows these key values for all four patients in this study, as well as the team composition during each intervention.

Table 4: Patient characteristics for all recorded surgeries.

Surgery	Patient's BMI (kg/m²)	Patient's age (years)
#1	31,2	40
#2	25	56
#3	19	41

3.2.3 Experimental setup

For this experiment, 3 surgeries have been recorded as described above. Based on the recorded laparoscopic video, each surgery was manually segmented into surgical steps. This denotes a finer granularity than the separation into phases, as done for the work in chapter 2 (see also [83]). Each step was defined around a single intention, such as "clipping the artery" or "dissecting tissue", and therefore changed mainly with the usage of different tools. These steps had a more flexible organization with more switches and repetitions, as they do not have a relatively accurate structure like workflow phases. Each step was declared an event, and the EIF for all events per surgery were calculated using the previously described CCFs.

Independently from the EIF calculation, a medical expert reviewed the surgical videos and identified as many "critical" steps for each surgery as they felt necessary. The expert was free to decide what they deemed as "critical", although the general focus was on patient outcome and safety. The expert was a junior surgeon, who was not performing any of the recorded surgeries personally, but knows the procedure and the surgeons involved very well.

After the EIF calculation was finalized, its results were discussed with the medical expert again to evaluate the results. The expert was asked to comment on the EIF ranking, specifically with regard to the questions if the EIF ranking is reasonable and of any medical value, and can the events with highest EIF plausibly be considered critical situations.

3.3 Results

For the three recorded surgeries, the main and assistant surgeons were monitored with the heartrate belt, and during two surgeries the pupils of the assistant surgeon were tracked. Two different main and assistant surgeons each performed the interventions in mixed composition. During two recordings, the main surgeon left the OR early to leave the simpler last steps purely to the assistant, so incomplete datasets had to be handled. The procedures took between 19 and 66 minutes, as they were of very contrasting difficulties. All patients were female and aged between 40 and 56.

Table 5: The five highest ranked events per surgery and their respective EIF.

Surgery 1	**EIF**	**Surgery 2**	**EIF**	**Surgery 3**	**EIF**
Dissecting Calot's triangle	0,589	Setting Trocar 4	0,609	Clipping cystic duct	0,587
Cutting cystic duct	0,587	Setting Trocar 3	0,608	Dissecting Calot's triangle (II)	0,573
Final dissection of gallbladder	0,560	Setting Trocar 2	0,595	Cutting cystic duct	0,570
Final hemostasis	0,551	Dissecting Calot's triangle	0,556	Hemostasis	0,569
Gallbladder dissection	0,549	Cutting cystic duct	0,544	Dissecting Calot's triangle (I)	0,554

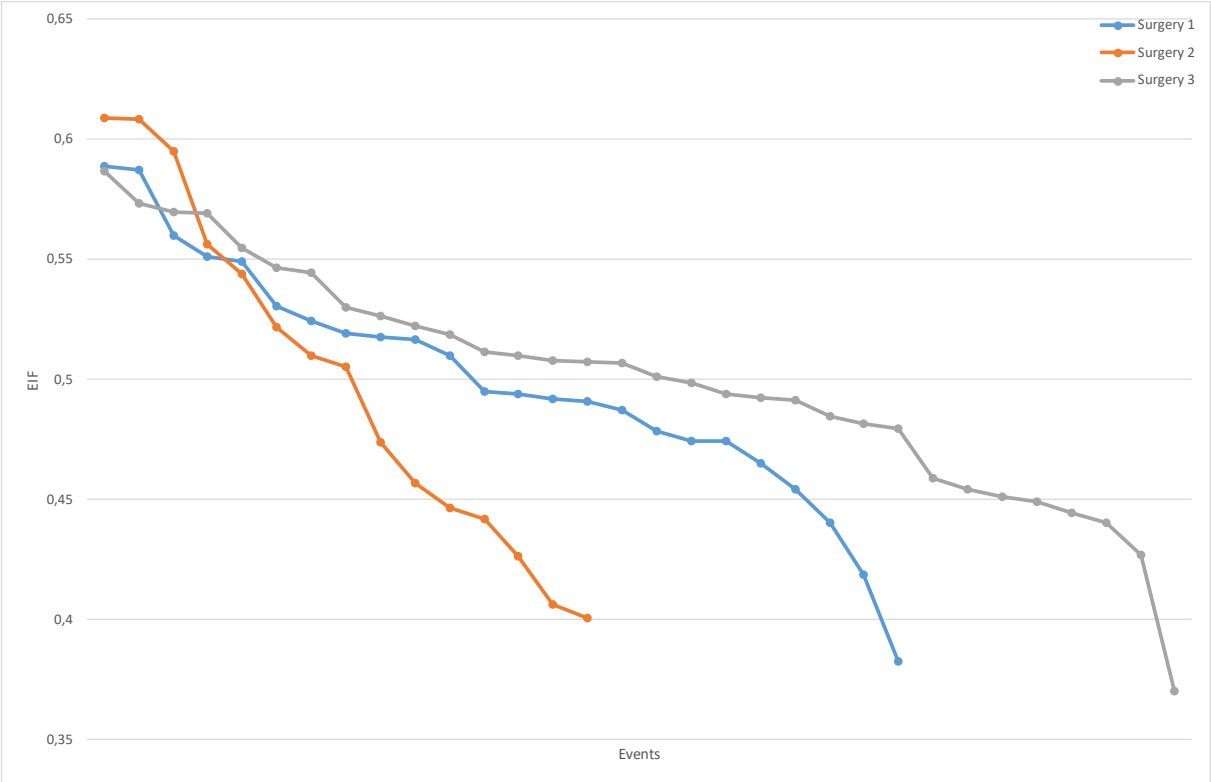

Figure 17: Events of all surgeries, sorted by their EIF in descending order.

3.3.1 EIF Calculation for Surgical Events

All 3 surgeries were manually segmented into a total of 71 surgical steps with an average length and standard deviation of $116s \pm 172s$ per step, which were used as events for further calculations. For each step, all suitable CCFs were evaluated (i.e. the CCFs based on pupil data could not be evaluated on one surgery). The values were converted to CCMs by comparison with all other events of the same surgery, and in case of CCF-Rs also normalized to fill the expected value range. The CCMs were then unified to EIF vectors according to the methods shown in 3.1.2. The resulting EIF of the top events per surgery are shown in Table 5. The lowest value of the EIF varied between 0.37 and 0.4 over all surgeries, while the highest value was in the range of 0.59 and 0.61. The largest difference in EIF within the same recording was in surgery 3, where the EIF stretched from 0.37 to 0.59. A plot of the EIF value distribution, sorted in descending order per surgery, is given in Figure 17. The most impactful events had an EIF of 0.55 or above.

3.3.2 Clinical Interpretation of Highest-Ranking Events

The tissue dissection during the preparation of Calot's triangle can be seen consistently within the 5 highest factored events of all surgeries, even if the dissection is split up as in one case. The last cutting or dissection steps during the "clipping and cutting" phase and the "gallbladder dissection" also tend to gain high impact factors.

As an interesting effect, during the shortest and likely easiest surgery, the placement of the three trocars is ranked highest, while these events are spread in the middle value range for the other interventions. This is likely caused by two cumulative reasons. First is the fact that trocar placement is among the few activities which leave relatively large injuries during minimally-invasive surgery, probably providing a substantial static basis for impact factor calculation. Secondly, since all other steps in this specific surgery could be finished quickly without any interruptions, the relative impact of these events was estimated to be relatively low, which indirectly supported the trocar placement steps.

Figure 18: EIF of observed events in their temporal order. Events for shorter surgeries (esp. surgery 2) have been spread out to match events of the same workflow phase over all surgeries. A few key events have been labeled for better orientation. Green circles highlight events possibly suitable for automatic triggering of further actions (see below).

A medical expert confirmed those findings and interpretations to be valid, as the cutting actions and the preparation of Calot's triangle are generally the most critical parts of a surgery, especially with regards to patient safety. The special case of the highly rated trocar placement steps was also estimated to relate to the otherwise highly unproblematic progress of the intervention, as trocar placement has a rather fixed difficulty compared to other aspects of this type of surgery.

3.3.3 Reliability of EIF Ranking across Multiple Surgeries

Figure 18 shows the EIF of all steps of each surgery in their regular temporal order. Since the procedures were of different length, they have a different number of steps associated with them. For clarity, the events in the figure were spaced equally within overlapping workflow phases. Therefore, the shorter surgery 2 has fewer data points, yet corresponding steps of all surgeries match as well as possible. With this rough alignment, clear patterns become evident. The preparation step (after trocar placement) is always rated with low factors, since the surgical site is only checked, and often no instruments are present. Then a region of high impact events occurs, as the next steps involve the dissection and cutting of tissue, which poses the highest risk for the patient. The events during the second half of the intervention are ranked lower again, with a few distinct spikes for the final hemostasis, during which the surgeon checks for bleedings and injuries and, if necessary, seals them through coagulation. Based on the coarse events and CCFs used in this experiment, it is already possible to read the general progression of the surgery from the calculated impact factors.

Some distinctive jumps in EIF values can be seen for some situations, as highlighted by green circles in Figure 18. These jumps could be detected automatically and be used for some applications, such as automated reporting. The IHE defined several integration profiles, of which some could benefit from this kind of information. While suitable profiles currently only exist in the domain of Radiology [65], the integration profiles of "Key Image Note", "Evidence Documents", or "Standardized Operational Log of Events" seem most fitting when extended to the domain of Surgery or adopted by similar, surgery-focused integration

profiles. In these profiles, images or other datapoints of particular interest are highlighted for later reference (similar to bookmarks). Key events identified in this manner through jumps in EIF can be of high importance regardless of regular workflow, even and especially events happening outside their usual context.

3.4 Discussion

In this chapter, concepts from group decision making within the field of operational research have been adopted to a completely novel method of surgical data science, allowing to quantitatively evaluate the impact of single events on the surgery. This methodology is to the best of our knowledge the first to solve such a problem in the context of surgical workflow analysis.

The core idea of this method is to carefully define events within the OR, which fit the intended experiments. Then all occurrences of these events are recorded and compared to each other through an arbitrarily large number of helper functions. All these comparisons are compiled into the EIF vector, assigning a single relevance factor to each event. Due to the variety of available function types and their corresponding transformations, a wide variety of characteristics can be evaluated per event, and even missing data can be handled naturally in the last combination step.

As this is the first adaptation of these methods to the surgical domain, unsurprisingly many open questions remain. Examining the influence of individual CCFs on the overall calculation is a likely next step. This may help better understand how the final EIF depends on the individual CCFs and may aid in identifying suitable characteristics to exploit for future experiments. Also, the effect of different parameters, both in the definition of CCFs and during the compilation of the EIF, offers room for optimization. During the experiments in this thesis, all parameters have been chosen to be the simplest values, which are acceptable in their respective formulae (e.g. 0 or 1). This includes the usage of aggregation operators. In both possible situations, the geometric mean has been used in this work, while other publications in the original area of GDM employ many different functions at this step of the calculation, up to fuzzy logic quantifiers. These are expected to emphasize some expert opinions or filter out others, depending on the magnitude or popularity of the opinion. Yet as the meaning behind the functions is radically changed in EIF calculation, the consequences of including such advanced quantifiers is difficult to predict.

Finally, other applications of EIF calculation in the surgical domain should be explored. This requires suitable definitions of events, their characteristics, and appropriate measurements and evaluation functions. Such future inputs should take a general applicability into account. While the surgeons and OR staff affected by the data collection for these experiments were themselves interested in the results and therefore very obliging and cooperative, some privacy concerns were encountered on occasion. The monitoring of surgeons' vital signals introduces a new level of surveillance, which could justifiably be mistrusted, yet the possible alternatives or benefits of this are better discussed separately.

4 Unified Surgical Display

In the last chapters, several approaches were presented to automatically detect what is happening inside the OR, and how important each of these steps are. Although these methods lay important groundwork, no application field has been thoroughly analyzed yet to understand how to utilize this knowledge. Some peripheral ideas have already been given, such as the evaluation of dexterity of young surgeons or a usage analysis for medical devices, which can be implemented by themselves as isolated systems. In contrast to this, a general solution to make workflow and context awareness available and useable throughout all of the OR and OR management will be described in this chapter.

A modern OR provides a plethora of advanced medical devices to the surgical team. These devices are usually either not connected to a data network, or only to a closed, proprietary network, so outside access to the data is by default not possible. Some approaches have already been made to evaluate the feasibility of streaming the video data to external displays [143, 159], or even to collect and store any kind of recorded data in a structured manner [135]. A national collaboration effort of the German government and several involved research centers [110] recently also came to the conclusion, that improved networking and interoperability is necessary for the future development of the surgical environment.

Most intraoperative devices are usually equipped with their own, separate controls (like keyboards), adjustable monitors, and a device-specific, highly specialized user interface. Yet due to sterility constraints, most of these devices cannot be placed in close proximity to the patient and surgeon. Consequently, in a surgery where multiple devices should be used, their placement is often arbitrarily decided based on available space, accessibility to other areas of the OR, and a safe margin to the actual operating table. Some manufacturers try to mitigate this issue through stand-, wall-, or ceiling-mounted monitors, yet these still tend to be rather large and can still obstruct the view of the surgeon or other staff members. This results in the situation that the surgeon has to switch their focus often far from the surgical site to access required information, despite the well-known negative impact this can have on the surgeon's performance [56]. Early experiments in better placement of medical data displays had promising results [151], even though only prerecorded, static imaging data was used.

A different method to handle the problem of display placement is the approach to avoid a group of classical displays completely. Some groups are working on providing audible cues instead of the common visual ones [1, 57], while many researchers already work with the technique of context-aware displays, very often in conjunction with augmented reality overlays [75, 78, 114]. A few groups also so far developed attempts at smart, unified surgical displays. An early work by Meyer et al. [107] collected various data from a diverse set of medical devices from multiple vendors, and ranked them by the number of staff members in the OR, which depend on either data stream. Then a central display shows relevant data in a divided interface. Information, which is relevant for at least two staff members throughout the whole surgery, is shown in a static pane, while information, which is only required during certain phases of the full process, is presented in a dynamically advancing pane. The prioritization of each possible data stream as well as the layout of the interface were done manually and prior to deployment in the OR. The state of the display changes automatically based on detected cues from manual data reporting or patient tracking data. No workflow modeling or automatic detection is done in this approach. Another, more theoretical approach by Schreiber et al. [145] already incorporates input from a workflow detection engine to adjust the displayed data to the surgical context, although the prioritization of different data streams is still taken from manual labels in an external database. This system allows for different layouts of a single screen, or a combined layout distributed over multiple screens of arbitrary sizes, yet the specific layout for each setup still has to be predefined manually by arranging several categorized areas (e.g. "essential", "alert", "navigation" etc.) among the combined screen space.

In a typical OR, the surgeon also does not have direct access to the controls of the available medical devices. Therefore, they have to resort to indirect methods, such as using long pointer sticks, explaining their intentions and the required device interactions to unsterile assistants, or leaving the sterile field themselves. Neither of these options are popular among surgeons as they disrupt their concentration, so all of them are only rarely used practically. Thus, many advanced features of the medical devices remain unused and many settings unchanged from the default, simply for efficiency and convenience reasons. This is a well-known problem, so many articles have been published, trying to provide alternative input methods to medical devices. Gesture recognition is a method used very often in research [15, 55, 59, 100], yet since surgeons usually require both hands on instruments at most times, the practical adoption of gesturing controls is still very low. Speech recognition systems able to interpret simple voice commands are already commercially available, yet usually still very limited. Also, their reaction times are usually much slower than that of human assistants, as besides the technical difficulties, these systems have no knowledge of the surgical context and cannot anticipate upcoming requests. Recent research [108] takes advantage of advanced, modern speech understanding systems, so the range of possible commands increases. Other works already try to incorporate surgical state into context-aware control mechanisms for robotic assistants [38] or connected OR devices in general [136].

As already said above, workflow intelligence can be included in many medical devices as separate, standalone system, and some device manufacturers have started to do this, although so far in a highly limited, often purely passive manner by grouping interface elements according to medical phases. As of yet, no generic, manufacturer-agnostic networking between surgical devices exists beyond the standardized access to the hospital's Picture Archiving and Communication System (PACS). This originates partly from legal constraints and protection of proprietary systems, even though cross-system networking could offer a wide variety of data for workflow recognition, which in turn could provide situational context information beneficial for most medical devices. This can also be seen as a first step towards a fully implemented TIMMS as described in [90]. An early, preliminary implementation of the work presented in this chapter has already been shown in a workshop [152].

A unified surgical display can act as central information hub, connecting all devices, and imparting both workflow knowledge and a central interface to them. Therefore, in order to link the devices in an OR in a meaningful way, this chapter will introduce "one display to connect them all and in the workflow bind them."

4.1 Operating Room Setup

A cooperation as described above has many technical, organizational, and legal requirements to its networking. The technical aspects include providing sufficient bandwidth and latency for different medical scenarios, while preventing or resisting interference with other signals. The legal constraints should cover the protection of the patient's data both from eavesdropping and outside manipulation, as well as allow for traceable and verifiable communication protocols between devices. Finally, the organizational properties involve the intelligent collection, filtering, distribution, and display of data and other, related signals.

Most technical challenges in the fields of networking, encryption, and authentication can be considered solved with regards to this application. Highly efficient data transfer options exist, both wired (such as IEEE 802.3 Ethernet) and wireless (like different variants of IEEE 802.11 Wi-Fi, or the expected "5G" mobile broadband networking), which usually even include options for robust communication and data recovery. Additionally, a wide variety of cryptographic methods enables authentication and protects data integrity via encryption and digital signatures.

A significant legal obstacle is the availability of open control interfaces and the responsibility for resulting actions. In many jurisdictions, the original manufacturer currently remains responsible for malfunctions of

their devices, even if the control signals were to come from an independent, external source. This situation naturally deters device manufacturers from offering such external interfaces at present. While projects like the German OR.NET [110] converge on this issue, it is mainly a task for political representatives to work on relevant laws and regulations.

This section is focused solely on the organizational aspects mentioned above. Here the unified display itself will be presented together with a possible topological network setup, as well as requirements to collecting data from different sources and feeding control signals back to connected devices.

4.1.1 Unified Display Hardware

The unified display itself can be a regular, modern tablet PC. The needed networking capabilities are therefore in most cases already fulfilled. Sufficient battery run time can ease the setup per patient, although power supply can be provided along mounting points. To comply with the sterility requirements inside the OR, the whole display can easily be covered with sterile draping. Most capacitive and resistive touch interfaces continue to work even under several layers of sterile foil and surgical gloves, so no additional input devices (like mice, keyboards, or styli) will be necessary. The display can be mounted via adjustable arm directly on the patient's bed, as close to the surgeon's field of view on the surgical site as possible. This allows the surgeon to focus mainly on the surgical tasks, without the need to move and shift focus to external monitors when looking up relevant information. The core idea behind this concept is the close proximity to the main area of interaction, comparable to how navigational systems in a car cockpit are positioned near to the driver's main view.

The tablet runs a specialized, custom software, which focuses on natural user experience and high usability [171]. This goal should be achieved mainly by reduction to the relevant minimum through context-aware selection and filtering of information and possible interface elements, as described in detail in the sections 4.2 and 4.3.

4.1.2 Networking Requirements

Collaboration between different devices in the OR, especially with a central interface, naturally requires a communication network between all intraoperatively used devices and sensors. The specifics of the actual network topology for this (e.g. ring or star shaped) are not as relevant, as long as parallel, bidirectional communication among an arbitrary number of clients is possible. A star-shaped network with the unified display as central hub is theoretically sufficient, yet highly impractical. It is very likely that other devices also need to communicate with each other or external systems, like an image viewer, which needs to be able to access the hospital PACS, or the workflow detection (see chapter 2) and EIF calculation (chapter 3), which will likely run on separate servers.

The various possible aspects of a unified display also have very different timing constraints. As phase and event transitions usually take several seconds, workflow analysis and EIF calculation can be done already with relatively slow communication, as their results do not need to be updated that frequently. Switching of general device states (e.g. lights, or the settings of imaging devices) also do not have strict real-time requirements, although for reasons of proper usability they should come in effect within approximately a second. However, since commands typically can be represented in comparably small data packages, fast transmissions and negligible delays should be expected. Direct and continuous control of devices (such as the HF generator for electrical coagulation, or surgical robots) on the other hand must happen in real-time. Input lag and latencies in the control of such devices can lead to a negative feedback loop and dangerous oversteering, which can cause severe injuries. Finally, the streaming of image sources to the central display makes high demands on bandwidth and video encoders. Depending on the modality, the imaging data should be able to be streamed to the display without noticeable delay, especially if the imaging device is used for navigational purposes and part of a hand-eye-coordinated feedback loop.

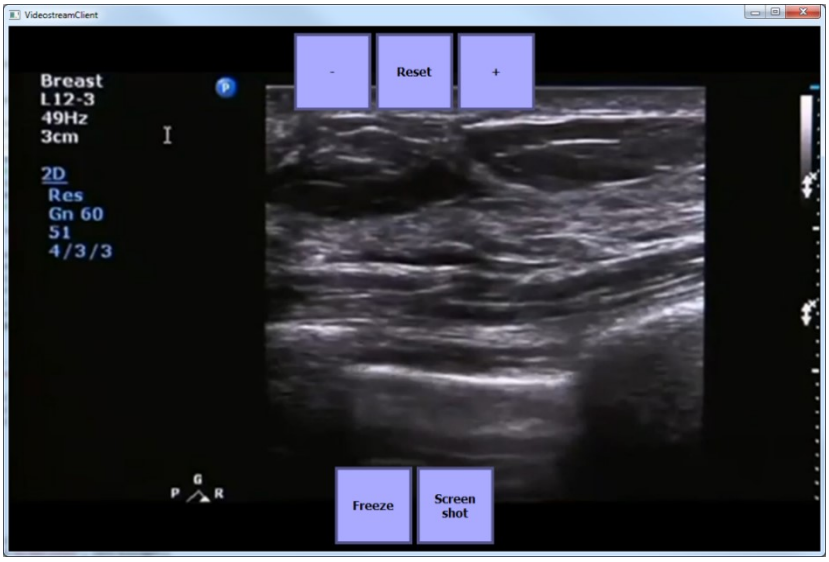

Figure 19: Overlays can enhance the usability of legacy devices, as additional functions (both in software and through other devices) can directly be called.

As mentioned in the last section, the unified display itself is technically a regular tablet PC, and as such already equipped with networking adapters (in most cases wirelessly, but wired connections are also often possible). Additional control servers, such as for the calculation of workflow information or for logging purposes, also should provide built-in, regular networking connectors. Similarly, future medical devices are likely to offer networking interfaces, assuming that proper standards and regulations have been established. All devices of a current integrated OR, where available, are already connected through proprietary communication buses. This network could theoretically be utilized though a translating device, which is able to connect to both the regular OR network and the bus system of the integrated OR. As already mentioned above, the main obstacle to this are not technical, but legal issues. Lastly, legacy devices usually do not have available data connections. Nonetheless, in every case these devices have some form of keyboard-like input and a monitor as output, which are often connected internally through standard plugs. Therefore, it is possible to capture the video output, encode it, and feed it into the network through appropriate additional hardware.

4.1.3 Data Aggregation

All connected devices in the OR network should follow the basic publisher-subscriber design pattern [50]. Imaging devices, sensors, and other instruments, which can provide data, implement the publisher interface, while devices like the unified display, which can receive and process data, subscribe to these data sources. Data should only be sent, if at least one subscriber is listening to updates, to prevent unnecessary congestion of the network and allow for maximal transfer rates. It is likely that some sensors will be active throughout the full surgery, as their data can be used to detect workflow changes or surgical events. Other data, especially with a high bandwidth load, are only observed as needed, e.g. when the video stream of an imaging device is shown only during a specific phase. In order to allow devices of multiple vendors to communicate with each other, a common standard is required. The working group “DICOM in Surgery” [89] can provide such a standardization as shared communication base.

Modern devices can directly provide suitable measurements and vital signs to the network, possibly through a translation tool, which converts between proprietary protocols and the shared network. On the other hand, frame grabbers can capture the image output of legacy devices (as mentioned above, see also Figure 19) and provide the image stream directly to the network. During this step, additional post-processing can be performed on the recorded signal. Numerical measurements, which are printed to the screen as text, can be extracted via text recognition. Furthermore, the device can be extended by added

image processing functionality, such as contrast enhancement, color correction, object recognition, or if possible supplementary calibration and measurements.

4.1.4 Control Channels

The unified display offers a prime opportunity for interactive controls in the OR due to its location close to the surgical site and easy accessibility to the surgeon. To facilitate this, devices can provide interfaces and an abstract control definition, which will be described in further detail in section 4.34.2.2. This definition essentially only lists basic interaction concepts like switches or numerical range inputs, without any layouting information, such as positions or sizes. These specifications can then be used to generate user interfaces suitable to the display size and current situation as described below. Additionally, to these interaction elements, also controls for detailed device parameters can be listed. Control signals and appropriate parameters for these commands can be received by the devices through secured interfaces (comparable to the concept of web services) available to the OR internal network. The unified display can then collect such definitions in order to send signals back to the devices, both from UI interaction from the surgical team, as well as predefined commands, which can be triggered automatically by workflow transitions or other detected events. Legacy devices, which cannot offer such interfaces themselves, can be included into the network through translational tools again. Typically, legacy devices provide some sort of keyboard and mouse input, which can be emulated by software and sent to the device as gestures, simulating the input of a regular human operator [100].

4.2 Information Selection

To prevent mistakes, the surgeon should be able to focus on the patient as much as possible. Additional data is often necessary and important for certain steps of a procedure, but they are not needed continuously throughout the full operation. In most cases, specific data (like imaging sources or preoperative scans) are only relevant during few steps of the surgery.

Current information systems provide large monitor walls, mounted on the walls or ceiling, with all available data displayed at the same time, arranged next to each other. While the arrangement and selection of shown elements is generally customizable, the manual effort to do this is typically not made, especially by the surgeons who are supposed to benefit from it afterwards. Therefore, the task of locating and extracting the wanted information from this massive collection of data poses a substantial cognitive load to the surgeon. This becomes more crucial if the surgeon requires atypical information, of which they do not know the position and need to search for it on the screen first.

Through the contextual knowledge of workflow detection and the importance weighting of impact factor calculation the unified display can estimate the most important elements to display at any given time during the surgery. For safety reasons, the surgeon should obviously always have the possibility to switch to any available view at all times, although even this fallback option can benefit from the context knowledge by presenting other options in order of their estimated impact on the current situation. This general idea can not only be applied to displayed data, but as well to interaction elements, as not all commands are equally relevant throughout the surgery either. Any type of interaction with members of the staff as in [16] are generally suitable for this kind of presentation.

4.2.1 Display of Most Relevant Data Source

To enable a context specific display of only appropriate information, the unified display has to collect a list of all available sources in the OR network first. All data sources and their modalities should be identified through unique names or ID numbers. Then a ranking can be applied based on the EIF calculation presented in chapter 3.

Among the characteristics included for each event (see 3.1.2.1), the ones regarding tools and devices are key for this approach. First, a virtual event is created for each device and, where applicable, each supported mode. The related devices and their possible settings are considered as tool components. The currently detected workflow phase is taken for the timing component, while the main person using the display (e.g. the head surgeon, see also 4.3) is used for the human role aspect. Other views are filled as needed with either constant values, or other reasonable values extracted from the current surgical situation where possible. During the procedure, only those CCFs are evaluated, which assign a value based to some degree on the collected imaging sources. Therefore CCFs, which only consider the device, will always provide a constant factor. When staff or patient data, which does not change within one surgery, is also taken into account, these functions will provide a constant factor during the current surgery, but this factor changes between different patients. A dynamic factor during the surgery is delivered by CCFs, which also take changing aspects such as the workflow phase into account. Lastly, these CCFs are combined according to the regular EIF algorithm. The final EIF ranking of the virtual events resembles the ranking, with which each related image source should be displayed at any given time.

4.2.2 Context-Specific Interactive Control Elements

As mentioned in 4.1.4, the unified display discovers network attached devices and collects the definition of their available commands and input elements. These elements are defined in an abstract way as either of the following:

- Triggers, which send a single, parameterless signal to the device; comparable to regular buttons.
- Switches, which have two states (usually “on” and “off”) and send a single signal with state information to the device when changed; can be represented either as switchable button, checkbox, switch, or single selection of two options.
- Value ranges, which can take any numerical value between two limits, and send a signal with according value parameter to the device for every change in the value; can be visualized as slider, dial, numeric value input, or single selection list of consecutive integer values if the value range permits.

Additionally, a passive value display can be declared, e.g. to display the actual value next to a slider defining a nominal value. All elements contain an identifier and a textual label for presentation. Several Control elements can be assembled together into logical groups (e.g. table control, or ultrasound Doppler settings) and possibly organized within workflow phases or target roles. No layouting data of any kind, like positions of sizes, are stored, neither in absolute screen space, nor in relation to each other. The final rendering of the chosen elements is done with regards to the available screen size and described in the next section.

The ranking of possible elements is done analogously to the selection of data sources shown in 4.2.1, yet with the focus of CCFs on general devices providing input channels. If such a level of detail is available through the CCFs, the ranking of UI elements can be done down to each logical control group individually, otherwise the devices can be ranked as whole, and the shown controls for high ranking devices are selected according to the assigned workflow phases. For this aspect of the unified display it is more important to calculate a correct ranking compared to a single best selection, as several controls or control groups can be displayed at the same time.

4.3 Dynamic User Interface Generation

The results of the ranking and selection process of sources and UI elements depend fully on the chosen CCFs and the parameters provided to them. Therefore, it is a reasonable approach to not only change the parameters between devices and over time, but also take advantage of changing parameters in other views

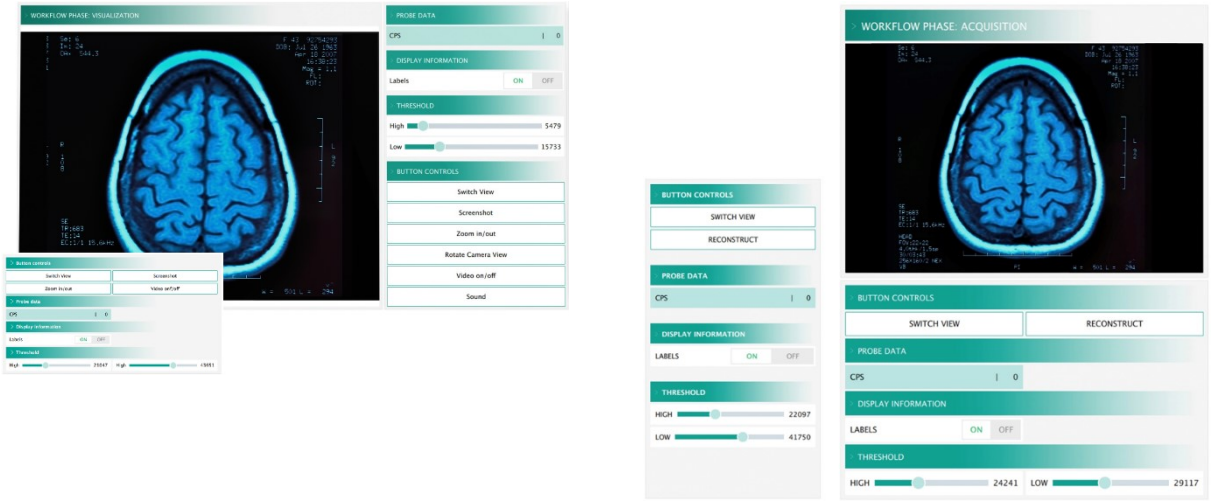

Figure 20: Different generated views of the same UI elements. Differences are in the orientation (landscape on the left, portrait on the right) and the consumed screen estate.

on the OR environment, such as the staff role. This allows for specialized displays for all members of the surgical team.

The general method dynamically generates specialized user interfaces for the unified display based on the abstract control information provided by connected devices, after all elements have been ranked. Most control elements can be represented by a variety of interaction elements (see above), so the choice for a specific visualization should follow the best usability option for a given device and input mode. The sizing of all elements as well as the layout in rows and columns depend on the available screen size, display orientation, and the number of selected controls to show (Figure 20). Larger screens generally allow for more elements, while keeping the physical size of each element within reasonable bounds, suitable for touch interaction. Image sources, on the other hand, must have a high priority, as these sources cannot be easily scaled down without loss of information or precision. Consequently, when shown, image sources move other elements to the rim of the screen and thus limit the number of interaction elements further. All elements within a logical group, as well as the groups themselves can be arranged mainly in rows or columns, depending on the screen orientation.

4.3.1 Large Display for Main Surgeon

The most screen real estate on displays intended for the main surgeon should be dedicated to available imaging sources. Regular displays for these data already exist in ORs, yet the key advantage of a unified display for the main surgeon, even if only the same data is presented, lies in the close proximity of the provided data to the actual surgical site. As added benefit, some space for UI elements can be provided to the side of the imaging data, yet it should be limited to very few, important options, to prevent cluttering the screen and distracting the surgeon. Also, since surgeons should be able to fully focus on the patient and tend to continuously use both hands to control instruments, placing regularly used commands (e.g. HF coagulation and cutting) on the surgical display would rather hinder than support surgeons. Control elements mainly used for changing settings related to the shown image source are best suited in this case. Parameter changes for image modalities usually require both experience and technical understanding of the underlying imaging technique. Furthermore, it can be difficult to explain visualization preferences to external assistants, so the ability to change them personally and according to the patient situation can accelerate or sometimes even enable the customization process.

4.3.2 Auxiliary Displays for Assistants and Nurses

Large enough displays can be provided also to assistant surgeons, so that they can better relate the shown image sources to the performed actions during their training. Assistant surgeons can be provided with

Figure 21: UI generated for a small, handheld device. OR assistants usually do not require direct access to imaging data, access to functions relevant to their tasks, such as documentation is sufficient.

different and more commonly used control commands than the head surgeon, as they are more likely to have the manual capacity to operate the display controls during the procedure. This will promote assistants to take more control of the parameters of used devices, which is not being done at all in most cases currently (Figure 21).

Smaller screens are in most cases sufficient for nurses. At most, they use live video streams to follow the progress of the surgery and prepare future requests, no medical decisions are made by them. Additional overlays can display workflow information and predictions about upcoming tasks to support nurses in training. General state changes within the operating theatre, such as changes in lighting, are currently done by the non-sterile nurse, although they are also responsible for documenting the procedure and refilling heavily used supplies, often for multiple ORs. To prevent waiting times and conflicts, such controls can be given on a display to the sterile scrub nurse.

4.4 Discussion

The unified surgical display presented in this chapter is an approach to combine the methods of surgical workflow analysis and event impact factor calculation, in order to change the way information is conveyed inside the OR, and attempting to implement a TIMMS [92]. The basis for this is a data transfer network between all devices in the OR. As mentioned before, this requires changes in the current laws governing medical devices, to allow for such connections without burdening some few manufacturers with all related legal risks. Then a communication protocol can be defined for device discovery, which immediately enables the data collection required for surgical workflow detection. Both the detected workflow knowledge and the available sensor data can be utilized to calculate event impact factors in real-time, through which displayed data and control elements can be chosen. All this facilitates a central, surgical interface, tailored to each intervention, each patient, and each member of the surgical staff.

The advantages of such a unified display are manifold. A minor improvement to the situation in the operating theatre is the enhanced usability of presented data, due to the better placement, closer to the actual surgical situs. As this only requires passive data collection, this can already be employed in a modern OR, given that all active devices at least offer a regular monitor output. When feedback channels are possible, directly providing control elements to the surgical team through the unified display can also increase the usage of secondary device features. These non-critical functions and parameters are often ignored, as their additional benefit does not yet outweigh the costs of using them through assistants or separate control interfaces. When they can be operated directly by the head or assistant surgeon, this interaction cost drops, and the available functions are more likely to be exploited. Finally, such a central display enables completely new interventions with heavy usage of different devices. These would not be feasible so far, as simply the space in an OR prevents arranging too many appliances around the patient and the team, especially in a way, which would allow the surgeons to reach their control panels and manipulate

the tools in a sterile way. With the controls and data display combined into the unified display, the majority of devices can be placed further away, with only their active elements (such as sensors or robotic effectors) requiring access to the patient.

The unified surgical display is intended to build the infrastructure of smart operating rooms in the long term. Through the common network, protocols, and central interface, it allows even for small and independent developers to add value to the OR. As certification is already a required part of joining the network communication, the safety and integrity of the surgical environment can be perpetuated, and in the event of mistaken commands, these can be traced back to the originating element. Manufacturers will also be able to design light, “headless” devices, only focused on their medical functions, as all inputs and outputs will be handled through the unified display. Even pure software applications can capture available data, offer their functions to the network, and display their results centrally. This allows hospitals to maintain more flexible, modular operating rooms, which can also offer higher functionality and more specialization, down to patient-specific and individual procedures, even enabling personalized single-use tools.

5 Conclusions

The future of surgery is a field inspiring many research groups, and this thesis hopefully contributed to its advancement. Chapter 2 presented and compared several methods to automatically detect surgical workflow. While some immediate applications can be derived from the recognized workflow information, its main goal is to provide an infrastructure for context awareness inside the OR, so that multiple tasks on different devices can be automated. In chapter 3, a first approach was described to rank events in the OR according to their relative importance. This provides meaningful additional situational knowledge besides the purely temporal data obtained from workflow analysis. Finally, chapter 4 combines these aspects into a novel, central data and control hub, the unified surgical display. By accessing all available data in the OR and selecting and presenting only the contextually most relevant information, this can already be seen as a next step towards implementing a TIMMS [92] in the operating theatre.

Surgical data science is just developing as a research area, and still lacks adoption by numerous device manufacturers in the medical domain. Most data used in research so far is either collected by manually taking notes, through additional sensors, or by accessing specific research data ports, unavailable on devices in regular medical use. An important step towards the next technological maturity level [90] of the digital operating room is the standardization and opening of intra-operative device communication to allow for general and vendor-independent interoperability. The surgery domain of the IHE initiative [29] is working on integration profiles to solve this issue, and the broad support by numerous industrial partners can be seen as indicator for its successful future adaptation.

Following the expected further evolution of surgery [102], a future surgeon is able to take considerably more information into account for each individual case than before due to the better integration of devices and databases and automatic compilation by smart, digital assistance systems. Contrary to some expectations, this does not only benefit the prestigious and highly technological model-ORs of few, selected hospitals. As many achievements in this domain are mainly software solutions, these methods can equally well support extreme and remote medical situations, such as in developing countries or on space stations. A health worker with a solid medical basis, but without specialized knowledge will still be able to perform some rudimentary interventions, as software assistance on top of a sufficient database can provide step-by-step instructions. While a truly automated surgeon still lies more within the realm of science-fiction, surgery in countless adaptations can profit from applied surgical data science, such as presented in this thesis.

6 List of Abbreviations

ANN	Artificial Neural Network
BMI	Body Mass Index
BoW	Bag-of-words
CCCM	Collective Component Characteristic Matrix
CCF	Component Characteristic Function
CCF-O	CCF-Ordering
CCF-P	CCF-Pairwise comparison
CCF-R	CCF-Rating
CCM	Component Characteristic Matrix
CNN	Convolutional Neural Networks
CPU	Central Processing Unit
CRF	Conditional Random Field
DICOM	Digital Imaging and Communications in Medicine
DOR	Digital Operating Room
DPM	Digital Patient Model
DTW	Dynamic Time Warping
EIF	Event Impact Factors
GDM	Group Decision Making
GP-GPU	General Purpose GPU
GPU	Graphics Processing Unit
HMM	Hidden Markov Model
HOG	Histogram of Oriented Gradients
HSV	Hue, Saturation, Value color space
IEEE	Institute of Electrical and Electronics Engineers
IHE	Integrating the Healthcare Enterprise initiative
LSTM	Long Short-Term Memory
M2CAI	Modeling and Monitoring of Computer Assisted Interventions
MPDM	Multiperson Decision Making
MQGDD	Multiplicative Quantifier-Guided Dominance Degree
MQGNDD	Multiplicative Quantifier-Guided Non-Dominance Degree
OR	Operating Room

OWA	Ordered Weighted Arithmetic operator
OWG	Ordered Weighted Geometric operator
PACS	Picture Archiving and Communication System
RFID	Radio-Frequency Identification
RGB	Red, Green, Blue color space
RPN	Region Proposal Network
SDK	Software Development Kit
SDS	Surgical Data Science
SIFT	Scale-Invariant Feature Transform
SNR	Signal-to-Noise Ratio
SPM	Surgical Process Model
SVM	Support Vector Machine
TIMMS	Therapy Imaging and Model Management System
UI	User Interface

7 References

1. Ahmad A, Adie SG, Wang M, Boppart S a (2010) Sonification of optical coherence tomography data and images. Opt Express 18:9934. doi: 10.1364/OE.18.009934

2. Ahmadi S-A, Padoy N, Rybachuk K, Feußner H, Heining SM, Navab N (2009) Motif discovery in OR sensor data with application to surgical workflow analysis and activity detection. M2CAI

3. Ahmadi S-A, Sielhorst T, Stauder R, Horn M, Feussner H, Navab N (2006) Recovery of surgical workflow without explicit models. Med Image Comput Comput Assist Interv 9:420–428. doi: 10.1007/11866565_52

4. Allan M, Ourselin S, Thompson S, Hawkes DJ, Kelly J, Stoyanov D (2013) Toward detection and localization of instruments in minimally invasive surgery. IEEE Trans Biomed Eng 60:1050–8. doi: 10.1109/TBME.2012.2229278

5. Allan M, Thompson S, Clarkson MJ, Ourselin S, Hawkes DJ, Kelly J, Stoyanov D (2014) 2D-3D Pose Tracking of Rigid Instruments in Minimally Invasive Surgery. Inf Process Comput Interv 8498:1–10. doi: 10.1007/978-3-319-07521-1_1

6. Alonso S, Herrera-Viedma E, Chiclana F, Herrera F (2010) A web based consensus support system for group decision making problems and incomplete preferences. Inf Sci (Ny) 180:4477–4495. doi: 10.1016/j.ins.2010.08.005

7. Alonso S, Pérez IJ, Cabrerizo FJ, Herrera-Viedma E (2013) A linguistic consensus model for Web 2.0 communities. Appl Soft Comput 13:149–157. doi: 10.1016/j.asoc.2012.08.009

8. Anguera X, Macrae R, Oliver N (2010) Partial sequence matching using an Unbounded Dynamic Time Warping algorithm. 2010 IEEE Int Conf Acoust Speech Signal Process 3582–3585. doi: 10.1109/ICASSP.2010.5495917

9. Arora S, Hull L, Sevdalis N, Tierney T, Nestel D, Woloshynowych M, Darzi A, Kneebone R (2010) Factors compromising safety in surgery: stressful events in the operating room. Am J Surg 199:60–65. doi: 10.1016/j.amjsurg.2009.07.036

10. Arora S, Sevdalis N, Nestel D, Woloshynowych M, Darzi A, Kneebone R (2010) The impact of stress on surgical performance: A systematic review of the literature. Surgery 147:318-330.e6. doi: 10.1016/j.surg.2009.10.007

11. Arrow KJ (1963) Social Choice and Individual Values

12. Bardram JE, Doryab A, Jensen RM, Lange PM, Nielsen KLG, Petersen ST (2011) Phase recognition during surgical procedures using embedded and body-worn sensors. In: 2011 IEEE International Conference on Pervasive Computing and Communications (PerCom). IEEE, pp 45–53

13. Berliner L, Lemke HU (2015) The Digital Patient Model and Model Guided Therapy. In: Berliner L, Lemke HU (eds). Springer International Publishing, Cham, pp 9–19

14. Bhatia B, Oates T, Xiao Y, Hu P (2007) Real-time identification of operating room state from video. 19Th Int Conf Innov Appl Artif Intell 1761–1766. doi: 978-1-57735-323-2

15. Bigdelou A, Stauder R, Benz T, Okur A, Blum T, Ghotbi R, Navab N (2012) HCI Design in the OR: A Gesturing Case-study. In: MICCAI Workshop on Modeling and Monitoring of Computer Assisted Interventions (M2CAI). Nice, France

16. Bigdelou A, Sterner T, Wiesner S, Wendler T, Matthes F, Navab N (2011) OR Specific Domain Model for Usability Evaluations of Intra-operative Systems. In: Taylor RH, Yang G-Z (eds) Information Processing in Computer-Assisted Interventions. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 25–35

17. Bihlmaier A, Schreiter L, Raczkowsky J, Wörn H (2016) Hierarchical Task Networks as Domain-Specific Language for Planning Surgical Interventions. In: Menegatti E, Michael N, Berns K, Yamaguchi H (eds). Springer International Publishing, Cham, pp 1095–1105

18. Blum T, Feußner H, Navab N (2010) Modeling and segmentation of surgical workflow from laparoscopic video. Med Image Comput Comput Assist Interv 13:400–7

19. Blum T, Navab N, Feußner H (2010) Methods for Automatic Statistical Modeling of Surgical Workflow. Proc Meas Behav 2010:64–65

20. Blum T, Padoy N, Feußner H, Navab N (2008) Modeling and online recognition of surgical phases using Hidden Markov Models. Med Image Comput Comput Assist Interv 11:627–35

21. Blum T, Padoy N, Feußner H, Navab N (2008) Workflow mining for visualization and analysis of surgeries. Int J Comput Assist Radiol Surg 3:379–386. doi: 10.1007/s11548-008-0239-0

22. Bodenstedt S, Wagner M, Katić D, Mietkowski P, Mayer B, Kenngott H, Müller-Stich B, Dillmann R, Speidel S (2017) Unsupervised temporal context learning using convolutional neural networks for laparoscopic workflow analysis. arXiv:170203684 [csCV]

23. Bouarfa L, Akman O, Schneider A, Jonker PP, Dankelman J (2011) In-vivo real-time tracking of surgical instruments in endoscopic video. Minim Invasive Ther Allied Technol 21:129–134. doi: 10.3109/13645706.2011.580764

24. Bouarfa L, Dankelman J (2012) Workflow mining and outlier detection from clinical activity logs. J Biomed Inform 45:1185–90. doi: 10.1016/j.jbi.2012.08.003

25. Bouarfa L, Jonker PP, Dankelman J (2011) Discovery of high-level tasks in the operating room. J Biomed Inform 44:455–462. doi: 10.1016/j.jbi.2010.01.004

26. Bouarfa L, Schneider A, Feußner H, Jonker PP, Dankelman J (2010) Preoperative patient data classification for intraoperative complexity prediction. J Inf Technol Biomed 1–8

27. Breiman L (2001) Random forests. Mach Learn 45:5–32. doi: 10.1023/A:1010933404324

28. Burgert O, Fink E, Wiemuth M, Thies C (2014) A model-guided peri-operative information systems approach. In: 2014 Cairo International Biomedical Engineering Conference (CIBEC). IEEE, pp 95–98

29. Burgert O, Liebmann P, Treichel T, Lemke HU (2011) Towards a new IHE-Domain "Surgery." Int J CARS 6:156–158

30. Burges CJC (1998) A tutorial on support vector machines for pattern recognition. Data Min Knowl Discov 2:121–167. doi: 10.1023/A:1009715923555

31. Cadène R, Robert T, Thome N, Cord M (2016) M2CAI Workflow Challenge: Convolutional Neural Networks with Time Smoothing and Hidden Markov Model for Video Frames Classification. arXiv:161005541

32. Chiclana F, Tapia García JM, del Moral MJ, Herrera-Viedma E (2012) A statistical comparative study of different similarity measures of consensus in group decision making. Inf Sci (Ny) 1–19

33. Cleary K, Chung HY, Mun SK (2004) OR2020 workshop overview: operating room of the future. Int Congr Ser 1268:847–852. doi: 10.1016/j.ics.2004.03.287

34. Cleary K, Kinsella A, Mun SK (2005) OR 2020 workshop report: Operating room of the future. Int Congr Ser 1281:832–838. doi: 10.1016/j.ics.2005.03.279

35. Criminisi A (2011) Decision Forests: A Unified Framework for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning. Found Trends® Comput Graph Vis 7:81–227. doi: 10.1561/0600000035

36. Criminisi A, Shotton J, Robertson D, Konukoglu E (2011) Regression Forests for Efficient Anatomy Detection and Localization in CT Studies. Med Comput Vision 106–117

37. Dalal N, Triggs B Histograms of Oriented Gradients for Human Detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). IEEE, pp 886–893

38. Daniele Comparetti M, Beretta E, Kunze M, De Momi E, Raczkowsky J, Ferrigno G (2014) Event-based device-behavior switching in surgical human-robot interaction. In: 2014 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp 1877–1882

39. Dergachyova O, Bouget D, Huaulmé A, Morandi X, Jannin P (2016) Automatic data-driven real-time segmentation and recognition of surgical workflow. Int J Comput Assist Radiol Surg 11:1081–1089. doi: 10.1007/s11548-016-1371-x

40. Despinoy F, Bouget D, Forestier G, Penet C, Zemiti N, Poignet P, Jannin P (2016) Unsupervised Trajectory Segmentation for Surgical Gesture Recognition in Robotic Training. IEEE Trans Biomed Eng 63:1280–1291. doi: 10.1109/TBME.2015.2493100

41. DiPietro R, Stauder R, Kayis E, Schneider A, Kranzfelder M, Feußner H, Hager GD, Navab N (2016) Automated Surgical-Phase Recognition Using Rapidly-Deployable Sensors. M2CAI Work MICCAI, Munich

42. Feußner H, Park A (2017) Surgery 4.0: the natural culmination of the industrial revolution? Innov Surg Sci 2:4–7. doi: 10.1515/iss-2017-0036

43. Forestier G, Lalys F, Riffaud L, Louis Collins D, Meixensberger J, Wassef SN, Neumuth T, Goulet B, Jannin P (2013) Multi-site study of surgical practice in neurosurgery based on surgical process models. J Biomed Inform 46:822–829. doi: 10.1016/j.jbi.2013.06.006

44. Forestier G, Lalys F, Riffaud L, Trelhu B, Jannin P (2012) Classification of surgical processes using dynamic time warping. J Biomed Inform 45:255–264. doi: 10.1016/j.jbi.2011.11.002

45. Forestier G, Petitjean F, Riffaud L, Jannin P (2017) Automatic matching of surgeries to predict surgeons' next actions. Artif Intell Med. doi: 10.1016/j.artmed.2017.03.007

46. Forestier G, Riffaud L, Jannin P (2015) Automatic phase prediction from low-level surgical activities. Int J Comput Assist Radiol Surg 10:833–41. doi: 10.1007/s11548-015-1195-0

47. Franke S, Meixensberger J, Neumuth T (2013) Intervention time prediction from surgical low-level tasks. J Biomed Inform 46:152–159. doi: 10.1016/j.jbi.2012.10.002

48. Franke S, Meixensberger J, Neumuth T (2015) Multi-perspective workflow modeling for online surgical situation models. J Biomed Inform 54:158–166. doi: 10.1016/j.jbi.2015.02.005

49. Franke S, Neumuth T (2013) A Framework for Multi-Model Surgical Workflow Management. Biomed Eng / Biomed Tech 58:24–25. doi: 10.1515/bmt-2013-4316

50. Gamma E, Helm R, Johnson R, Vlissides J (1994) Design Patterns: Elements of Reusable Object-Oriented Software

51. Geurts P, Ernst D, Wehenkel L (2006) Extremely randomized trees. Mach Learn 63:3–42. doi: 10.1007/s10994-006-6226-1

52. Gibaud B, Penet C, Jannin P (2014) OntoSPM: a core ontology of surgical procedure models. Surgetica 2–4

53. Glaser B, Dänzer S, Neumuth T (2015) Intra-operative surgical instrument usage detection on a multi-sensor table. Int J Comput Assist Radiol Surg 10:351–362. doi: 10.1007/s11548-014-1066-0

54. Glaser B, Schellenberg T, Franke S, Dänzer S, Neumuth T (2015) Surgical instrument similarity metrics and tray analysis for multi-sensor instrument identification. In: Webster RJ, Yaniv ZR (eds). p 941526

55. Grätzel C, Fong T, Grange S, Baur C (2004) A non-contact mouse for surgeon-computer interaction. Technol Health Care 12:245–57

56. Hanna GB, Shimi SM, Cuschieri A (1998) Task performance in endoscopic surgery is influenced by location of the image display. Ann Surg 227:481–4

57. Hansen C, Black D, Lange C, Rieber F, Lamadé W, Donati M, Oldhafer KJ, Hahn HK (2012) Auditory support for resection guidance in navigated liver surgery. Int J Med Robot 1–12. doi: 10.1002/rcs.1466

58. Haro BB, Zappella L, Vidal R (2012) Surgical Gesture Classification from Video Data. In: MICCAI. pp 1–8

59. Hartmann F, Schlaefer A (2013) Feasibility of touch-less control of operating room lights. Int J Comput Assist Radiol Surg 8:259–268. doi: 10.1007/s11548-012-0778-2

60. He K, Zhang X, Ren S, Sun J (2015) Deep Residual Learning for Image Recognition. arXiv:151203385

61. Healey a N, Sevdalis N, Vincent C a (2006) Measuring intra-operative interference from distraction and interruption observed in the operating theatre. Ergonomics 49:589. doi: 10.1080/00140130600568899

62. Herrera-Viedma E, Herrera F, Chiclana F, Luque M (2004) Some issues on consistency of fuzzy preference relations. Eur J Oper Res 154:98–109. doi: 10.1016/S0377-2217(02)00725-7

63. Herrera F, Herrera-Viedma E, Chiclana F (2001) Multiperson decision-making based on multiplicative preference relations. Eur J Oper Res 129:372–385. doi: 10.1016/S0377-2217(99)00197-6

64. Hochreiter S, Schmidhuber J (1997) Long Short-Term Memory. Neural Comput 9:1735–1780. doi: 10.1162/neco.1997.9.8.1735

65. IHE International I (2018) IHE Radiology (RAD) Technical Framework Integration Profiles. https://www.ihe.net/resources/technical_frameworks/

66. Jannin P, Morandi X (2007) Surgical models for computer-assisted neurosurgery. Neuroimage 37:783–791

67. Jia Deng, Wei Dong, Socher R, Li-Jia Li, Kai Li, Li Fei-Fei (2009) ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, pp 248–255

68. Jin Y, Dou Q, Chen H, Yu L, Heng P-A (2016) EndoRCN: Recurrent Convolutional Networks for Recognition of Surgical Workflow in Cholecystectomy Procedure Video. http://camma.u-strasbg.fr/m2cai2016/reports/Jin-Workflow.pdf. Accessed 7 Oct 2017

69. Joskowicz L (2017) Computer-aided surgery meets predictive, preventive, and personalized medicine. EPMA J 8:1–4. doi: 10.1007/s13167-017-0084-8

70. Kassidas A, MacGregor JF, Taylor P a (1998) Synchronization of batch trajectories using dynamic time warping. AIChE J 44:864–875. doi: 10.1002/aic.690440412

71. Kassner M, Patera W, Bulling A (2014) Pupil: an open source platform for pervasive eye tracking and mobile gaze-based interaction. In: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing Adjunct Publication - UbiComp '14 Adjunct. ACM Press, New York, New York, USA, pp 1151–1160

72. Katić D, Julliard C, Wekerle AL, Kenngott H, Müller-Stich BP, Dillmann R, Speidel S, Jannin P, Gibaud B (2015) LapOntoSPM: an ontology for laparoscopic surgeries and its application to surgical phase recognition. Int J Comput Assist Radiol Surg 10:1427–1434. doi: 10.1007/s11548-015-1222-1

73. Katić D, Maleshkova M, Engelhardt S, Wolf I, März K, Maier-Hein L, Nolden M, Wagner M, Kenngott H, Müller-Stich BP, Dillmann R, Speidel S (2017) What does it all mean? Capturing Semantics of Surgical Data and Algorithms with Ontologies. arXiv:170507747 1–4

74. Katić D, Schuck J, Wekerle A-L, Kenngott H, Müller-Stich BP, Dillmann R, Speidel S (2016) Bridging the gap between formal and experience-based knowledge for context-aware laparoscopy. Int J Comput Assist Radiol Surg 11:881–888. doi: 10.1007/s11548-016-1379-2

75. Katić D, Spengler P, Bodenstedt S, Castrillon-Oberndorfer G, Seeberger R, Hoffmann J, Dillmann R, Speidel S (2015) A system for context-aware intraoperative augmented reality in dental implant surgery. Int J Comput Assist Radiol Surg 10:101–108. doi: 10.1007/s11548-014-1005-0

76. Katić D, Wekerle A-L, Gärtner F, Kenngott H, Müller-Stich BP, Dillmann R, Speidel S (2013) Ontology-based prediction of surgical events in laparoscopic surgery. In: Holmes DR, Yaniv ZR (eds). p 86711A

77. Katić D, Wekerle A-L, Gärtner F, Kenngott H, Müller-Stich BP, Dillmann R, Speidel S (2014) Knowledge-Driven Formalization of Laparoscopic Surgeries for Rule-Based Intraoperative Context-Aware Assistance. In: 5th International Conference, IPCAI. Fukuoka, Japan, pp 158–167

78. Katić D, Wekerle A-L, Görtler J, Spengler P, Bodenstedt S, Röhl S, Suwelack S, Kenngott HG, Wagner M, Müller-Stich BP, Dillmann R, Speidel S (2013) Context-aware Augmented Reality in laparoscopic surgery. Comput Med Imaging Graph 37:174–182. doi: 10.1016/j.compmedimag.2013.03.003

79. Kenngott HG, Apitz M, Wagner M, Preukschas AA, Speidel S, Müller-Stich BP (2017) Paradigm shift: cognitive surgery. Innov Surg Sci 2. doi: 10.1515/iss-2017-0012

80. Kranzfelder M, Schneider A, Fiolka A, Koller S, Reiser S, Vogel T, Wilhelm D, Feussner H (2014) Reliability of sensor-based real-time workflow recognition in laparoscopic cholecystectomy. Int J Comput Assist Radiol Surg 9. doi: 10.1007/s11548-014-0986-z

81. Kranzfelder M, Schneider A, Fiolka A, Schwan E, Gillen S, Wilhelm D, Schirren R, Reiser S, Jensen B, Feußner H (2013) Real-time instrument detection in minimally invasive surgery using radiofrequency identification technology. J Surg Res 1–7. doi: 10.1016/j.jss.2013.06.022

82. Lafferty J, McCallum A, Pereira FCN (2001) Conditional random fields: Probabilistic models for segmenting and labeling sequence data. ICML '01 Proc Eighteenth Int Conf Mach Learn 8:282–289. doi: 10.1038/nprot.2006.61

83. Lalys F, Jannin P (2014) Surgical process modelling: a review. Int J Comput Assist Radiol Surg 9:495–511. doi: 10.1007/s11548-013-0940-5

84. Lalys F, Riffaud L, Bouget D, Jannin P (2012) A Framework for the Recognition of High-Level Surgical Tasks From Video Images for Cataract Surgeries. IEEE Trans Biomed Eng 59:966–976. doi: 10.1109/TBME.2011.2181168

85. Lalys F, Riffaud L, Morandi X, Jannin P (2011) Surgical Phases Detection from Microscope Videos by Combining SVM and HMM. In: Menze B, Langs G, Tu Z, Criminisi A (eds) Medical Computer Vision. Recognition Techniques and Applications in Medical Imaging. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 54–62

86. Lea C, Hager GD, Vidal R (2015) An Improved Model for Segmentation and Recognition of Fine-Grained Activities with Application to Surgical Training Tasks. In: 2015 IEEE Winter Conference on Applications of Computer Vision. IEEE, pp 1123–1129

87. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86:2278–2324. doi: 10.1109/5.726791

88. Lemke HU, Berliner L (2007) Specification and design of a Therapy Imaging and Model Management System (TIMMS). In: Horii SC, Andriole KP (eds). p 651602

89. Lemke HU, Berliner L (2011) PACS for surgery and interventional radiology: Features of a Therapy Imaging and Model Management System (TIMMS). Eur J Radiol 78:239–242. doi: 10.1016/j.ejrad.2010.05.030

90. Lemke HU, Berliner L (2013) The digital operating room: towards intelligent infrastructures and processes. In: Law MY, Boonn WW (eds) Medical Imaging 2013: Advanced Pacs-Based Imaging Informatics and Therapeutic Applications. p 86740R

91. Lemke HU, Trantakis C, Köchy K, Müller A, Strauss G, Meixensberger J (2004) Workflow analysis for mechatronic and imaging assistance in head surgery. Int Congr Ser 1268:830–835. doi: 10.1016/j.ics.2004.03.359

92. Lemke HU, Vannier MW (2006) The operating room and the need for an IT infrastructure and standards. Int J Comput Assist Radiol Surg 1:117–121. doi: 10.1007/s11548-006-0051-7

93. Li X, Zhang Y, Li M, Chen S, Austin FR, Marsic I, Burd RS (2016) Online process phase detection using multimodal deep learning. In: 2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON). IEEE, pp 1–7

94. Li X, Zhang Y, Zhang J, Chen Y, Chen S, Gu Y, Zhou M, Farneth RA, Marsic I, Burd RS (2017) Progress Estimation and Phase Detection for Sequential Processes. Cornell Univ Libr

95. Liebmann P, Meixensberger J, Wiedemann P, Neumuth T (2013) The impact of missing sensor information on surgical workflow management. Int J Comput Assist Radiol Surg 8:867–875. doi: 10.1007/s11548-013-0824-8

96. Liebmann P, Neumuth T (2010) Model driven design of workflow schemata for the operating room of the future. Inform 2010 Serv Sci - Neue Perspekt für die Inform 175:415–419

97. Lillie MC (1998) Cranial surgery dates back to Mesolithic. Nature 391:854. doi: 10.1038/36023

98. Lin HC, Hager GD (2009) User-Independent Models of Manipulation Using Video Contextual Cues. In: M2CAI-Workshop

99. Lowe DG (2004) Distinctive Image Features from Scale-Invariant Keypoints. Int J Comput Vis 60:91–110. doi: 10.1023/B:VISI.0000029664.99615.94

100. Ma M, Fallavollita P, Habert S, Weidert S, Navab N (2016) Device- and system-independent personal touchless user interface for operating rooms. Int J Comput Assist Radiol Surg 11:853–861. doi: 10.1007/s11548-016-1375-6

101. Maier-Hein L, Vedula S, Speidel S, Navab N, Kikinis R, Park A, Eisenmann M, Feussner H, Forestier G, Giannarou S, Hashizume M, Katić D, Kenngott H, Kranzfelder M, Malpani A, März K, Neumuth T, Padoy N, Pugh C, Schoch N, Stoyanov D, Taylor R, Wagner M, Hager GD, Jannin P (2017) Surgical Data Science: Enabling Next-Generation Surgery. arXiv:170106482

102. Maier-Hein L, Vedula SS, Speidel S, Navab N, Kikinis R, Park A, Eisenmann M, Feussner H, Forestier G, Giannarou S, Hashizume M, Katić D, Kenngott H, Kranzfelder M, Malpani A, März K, Neumuth T, Padoy N, Pugh C, Schoch N, Stoyanov D, Taylor R, Wagner M, Hager GD, Jannin P (2017) Surgical data science for next-generation interventions. Nat Biomed Eng 1:691–696. doi: 10.1038/s41551-017-0132-7

103. Maktabi M, Neumuth T (2016) Online Medical Device Use Prediction: Assessment of Accuracy. Stud Health Technol Inform 228:557–561. doi: 10.3233/978-1-61499-678-1-557

104. Malpani A, Lea C, Chen CCG, Hager GD (2016) System events: readily accessible features for surgical phase detection. Int J Comput Assist Radiol Surg 11:1201–1209. doi: 10.1007/s11548-016-1409-0

105. Meißner C, Meixensberger J, Pretschner A, Neumuth T (2014) Sensor-based surgical activity recognition in unconstrained environments. Minim Invasive Ther Allied Technol 23:198–205. doi: 10.3109/13645706.2013.878363

106. Merigó JM, Casanovas M (2010) Geometric operators in decision making with minimization of regret. Int J Comput Syst Sci Eng 514–521

107. Meyer M a., Levine WC, Egan MT, Cohen BJ, Spitz G, Garcia P, Chueh H, Sandberg WS (2007) A computerized perioperative data integration and display system. Int J Comput Assist Radiol Surg 2:191–202. doi: 10.1007/s11548-007-0126-0

108. Miehle J, Ostler D, Gerstenlauer N, Minker W (2017) The next step: intelligent digital assistance for clinical operating rooms. Innov Surg Sci 2. doi: 10.1515/iss-2017-0034

109. Minsky M, Papert S (1969) Perceptrons: An Introduction to Computational Geometry. MIT Press, Cambridge MA

110. Moser H, Dehm J, Gessner C, Golatowski F, Heidenreich G, Onken M (2015) Weißbuch: Sichere Dynamische Vernetzung in Operationssaal und Klinik. VDE Verband der Elektrotechnik, Elektronik, Informationstechnik e.V., Frankfurt

111. Nakawala H, Momi E De, Pescatori LE, Morelli A, Ferrigno G (2017) Inductive learning of the surgical workflow model through video annotations. In: IEEE 30th International Symposium on Computer Based Medical Systems, Thessaloniki, Greece

112. Nara A, Allen C, Izumi K (2017) Surgical Phase Recognition using Movement Data from Video Imagery and Location Sensor Data. In: Advances in Geographic Information Science. pp 229–237

113. Nara A, Izumi K, Iseki H, Suzuki T, Nambu K, Sakurai Y (2009) Surgical workflow analysis based on staff's trajectory patterns. In: M2CAI workshop, MICCAI, London

114. Navab N, Traub J, Sielhorst T, Feuerstein M, Bichlmeier C (2007) Action- and Workflow-Driven Augmented Reality for Computer-Aided Medical Procedures. IEEE Comput Graph Appl 27:10–14. doi: 10.1109/MCG.2007.117

115. Neumann J, Neumuth T (2015) Standardized semantic workflow modeling in the surgical domain. In:

2015 17th International Conference on E-health Networking, Application & Services (HealthCom). IEEE, pp 11–16

116. Neumann J, Neumuth T (2015) Towards a framework for standardized semantic workflow modeling and management in the surgical domain. Curr Dir Biomed Eng 1:172–175. doi: 10.1515/cdbme-2015-0043

117. Neumann J, Rockstroh M, Franke S, Neumuth T (2016) BPMN SIX – A BPMN 2.0 Surgical Intervention Extension Concept and Design of a BPMN Extension for Intraoperative Workflow Modeling and Execution in the Integrated Operating Room. In: 7th Workshop on Modeling and Monitoring of Computer Assisted Interventions (M2CAI), Athens, Greece

118. Neumuth T, Jannin P, Schlomberg J, Meixensberger J, Wiedemann P, Burgert O (2011) Analysis of surgical intervention populations using generic surgical process models. Int J Comput Assist Radiol Surg 6:59–71. doi: 10.1007/s11548-010-0475-y

119. Neumuth T, Liebmann P, Wiedemann P, Meixensberger J (2012) Surgical Workflow Management Schemata for Cataract Procedures. Methods Inf Med 51:371–382. doi: 10.3414/ME11-01-0093

120. Neumuth T, Loebe F, Jannin P (2012) Similarity metrics for surgical process models. Artif Intell Med 54:15–27. doi: 10.1016/j.artmed.2011.10.001

121. Neumuth T, Mansmann S, Scholl MH, Burgert O (2008) Data Warehousing Technology for Surgical Workflow Analysis. IEEE

122. Neumuth T, Strauß G, Meixensberger J, Lemke HU, Burgert O (2006) Acquisition of process descriptions from surgical interventions. Database Expert 602–611

123. Nguyen A, Yosinski J, Clune J (2015) Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 427–436

124. Okur A, Stauder R, Feussner H, Navab N (2017) Quantitative Characterization of Components of Computer Assisted Interventions. arXiv:170200582 [csOH]

125. Padoy N, Blum T, Ahmadi S-A, Feußner H, Berger M-O, Navab N (2010) Statistical modeling and recognition of surgical workflow. Med Image Anal 1–22. doi: 10.1016/j.media.2010.10.001

126. Padoy N, Blum T, Essa I, Feußner H, Berger M-O, Navab N (2007) A boosted segmentation method for surgical workflow analysis. Int Conf Med Image Comput Comput Interv 10:102–9

127. Padoy N, Blum T, Feußner H, Berger M-O, Navab N (2008) On-line recognition of surgical activity for monitoring in the operating room. In: Proceedings of the 20th national conference on Innovative applications of artificial intelligence. pp 1718–1724

128. Padoy N, Mateus D, Weinland D, Berger M-O, Navab N (2009) Workflow monitoring based on 3D motion features. In: 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops. IEEE, pp 585–592

129. Perez IJ, Cabrerizo FJ, Herrera-Viedma E (2010) A Mobile Decision Support System for Dynamic Group Decision-Making Problems. IEEE Trans Syst Man, Cybern - Part A Syst Humans 40:1244–1256. doi: 10.1109/TSMCA.2010.2046732

130. Pernek I, Ferscha A (2017) A survey of context recognition in surgery. Med Biol Eng Comput 1–16. doi: 10.1007/s11517-017-1670-6

131. Petrone P, Niola M, Di Lorenzo P, Paternoster M, Graziano V, Quaremba G, Buccelli C (2015) Early Medical Skull Surgery for Treatment of Post-Traumatic Osteomyelitis 5,000 Years Ago. PLoS One 10:e0124790. doi: 10.1371/journal.pone.0124790

132. Quellec G, Lamard M, Cochener B, Cazuguel G (2014) Real-Time Segmentation and Recognition of Surgical Tasks in Cataract Surgery Videos. IEEE Trans Med Imaging 33:2352–2360. doi: 10.1109/TMI.2014.2340473

133. Rabiner LR (1989) A tutorial on hidden Markov models and selected applications in speech recognition.

Proc IEEE 77:257–286. doi: 10.1109/5.18626

134. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In: Advances in Neural Information Processing Systems. pp 91–99

135. Rockstroh M, Franke S, Neumuth T (2014) Requirements for the structured recording of surgical device data in the digital operating room. Int J Comput Assist Radiol Surg 9:49–57. doi: 10.1007/s11548-013-0909-4

136. Rockstroh M, Franke S, Neumuth T (2015) Closed-loop approach for situation awareness of medical devices and operating room infrastructure. Curr Dir Biomed Eng 1:176–179. doi: 10.1515/cdbme-2015-0044

137. Rosenblatt F (1958) The perceptron: A probabilistic model for information storage and organization in the brain. Psychol Rev 65:386–408. doi: 10.1037/h0042519

138. Rupprecht C, Lea C, Tombari F, Navab N, Hager GD (2016) Sensor substitution for video-based action recognition. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp 5230–5237

139. Saaty TL (1990) How to make a decision: The analytic hierarchy process. Eur J Oper Res 48:9–26. doi: 10.1016/0377-2217(90)90057-I

140. Saaty TL (2008) Decision making with the analytic hierarchy process. Int J Serv Sci 1:83. doi: 10.1504/IJSSCI.2008.017590

141. Sahu M, Mukhopadhyay A, Szengel A, Zachow S (2016) Tool and Phase recognition using contextual CNN features. http://camma.u-strasbg.fr/m2cai2016/reports/Sahu-ToolandWorkflow.pdf. Accessed 7 Oct 2017

142. Sakoe H, Chiba S (1978) Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans Acoust 26:43–49. doi: 10.1109/TASSP.1978.1163055

143. Schneider A, Wilhelm D, Doll D, Rauschenbach U, Finkenzeller M, Wirnhier H, Illgner K, Feussner H (2007) Wireless live streaming video of surgical operations: an evaluation of communication quality. J Telemed Telecare 13:391–396. doi: 10.1258/135763307783064386

144. Schneider A, Wilhelm D, Fiaschi Schneider M, Schuster T, Kriner M, Leuxner C, Can S, Fiolka A, Spanfellner B, Sitou W, Feußner H (2011) Laparoscopic Cholecystectomy-a Standardized Routine Laparoscopic Procedure: Is it Possible to Predict the Duration of an Operation? J Healthc Eng 2:287–298

145. Schreiber E, Franke S, Bieck R, Neumuth T (2016) A concept for consistent and prioritized presentation of surgical information. 16–19

146. Sevdalis N, Undre S, McDermott J, Giddie J, Diner L, Smith G (2014) Impact of intraoperative distractions on patient safety: a prospective descriptive study using validated instruments. World J Surg 38:751–8. doi: 10.1007/s00268-013-2315-z

147. Sexton JB, Thomas EJ, Helmreich RL (2000) Error, stress, and teamwork in medicine and aviation: cross sectional surveys. BMJ 320:745–749. doi: 10.1136/bmj.320.7237.745

148. Shannon CE (1948) A Mathematical Theory of Communication. Bell Syst Technol 27:379–423

149. Sharp T (2008) Implementing Decision Trees and Forests on a GPU. In: Computer Vision – ECCV 2008. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 595–608

150. Simonyan K, Zisserman A (2014) Very Deep Convolutional Networks for Large-Scale Image Recognition

151. Soehngen E, Rahmah NN, Kakizawa Y, Horiuchi T, Fujii Y, Kiuchi T, Hongo K (2012) Operation-Microscope-Mounted Touch Display Tablet Computer for Intraoperative Imaging Visualization. World Neurosurg 77:381–383. doi: 10.1016/j.wneu.2011.06.017

152. Stauder R, Belagiannis V, Schwarz LA, Bigdelou A, Söhngen E, Ilic S, Navab N (2012) A User-Centered and Workflow-Aware Unified Display for the Operating Room. In: MICCAI Workshop on Modeling and Monitoring of Computer Assisted Interventions (M2CAI). Nice, FR

153. Stauder R, Hahn T, Scherzer D, Navab N (2017) Scene and Structure Detection in Laparoscopic Videos Using Different Convolutional Neural Networks. to Appear arXiv

154. Stauder R, Kayis E, Navab N (2017) Learning-based Surgical Workflow Detection from Intra-Operative Signals. arXiv:170600587 [csLG]

155. Stauder R, Okur A, Peter L, Schneider A, Kranzfelder M, Feußner H, Navab N (2014) Random Forests for Phase Detection in Surgical Workflow Analysis. In: 5th International Conference on Information Processing in Computer-Assisted Interventions (IPCAI)

156. Stauder R, Ostler D, Kranzfelder M, Koller S, Feußner H, Navab N (2016) The TUM LapChole dataset for the M2CAI 2016 workflow challenge. arXiv:161009278 [csCV]

157. Stauder R, Ostler D, Vogel T, Wilhelm D, Koller S, Kranzfelder M, Navab N (2017) Surgical data processing for smart intraoperative assistance systems. Innov Surg Sci 2:145–152. doi: 10.1515/iss-2017-0035

158. Sutton C, McCallum A (2010) An introduction to conditional random fields. arXiv Prepr arXiv10114088

159. Suzuki T, Yoshimitsu K, Muragaki Y, Iseki H (2013) Intelligent Operating Theater: Technical Details for Information Broadcasting and Incident Detection System. J Med Biol Eng 33:69–78. doi: 10.5405/jmbe.982

160. Tapia García JM, Del Moral MJ, Martínez MA, Herrera-Viedma E (2012) A Consensus Model for Group Decision-Making Problems with Interval Fuzzy Preference Relations. Int J Inf Technol Decis Mak 11:709–725. doi: 10.1142/S0219622012500174

161. Tormene P, Giorgino T, Quaglini S, Stefanelli M (2009) Matching incomplete time series with dynamic time warping: an algorithm and an application to post-stroke rehabilitation. Artif Intell Med 45:11–34. doi: 10.1016/j.artmed.2008.11.007

162. Toti G, Garbey M, Sherman V, Bass BL, Dunkin BJ (2015) A Smart Trocar for Automatic Tool Recognition in Laparoscopic Surgery. Surg Innov 22:77–82. doi: 10.1177/1553350614531659

163. Tran DT, Sakurai R, Yamazoe H, Lee J-H (2017) Phase Segmentation Methods for an Automatic Surgical Workflow Analysis. Int J Biomed Imaging 2017:1–17. doi: 10.1155/2017/1985796

164. Tran DT, Sakurai R, Yamazoe H, Lee J (2016) Improving phases segmentation in surgical workflow using topic model for visual motion words. In: 2016 IEEE/SICE International Symposium on System Integration (SII). IEEE, pp 502–507

165. Twinanda AP, Alkan EO, Gangi A, de Mathelin M, Padoy N (2015) Data-driven spatio-temporal RGBD feature encoding for action recognition in operating rooms. Int J Comput Assist Radiol Surg 10:737–747. doi: 10.1007/s11548-015-1186-1

166. Twinanda AP, Mutter D, Marescaux J, de Mathelin M, Padoy N (2016) Single- and Multi-Task Architectures for Surgical Workflow Challenge at M2CAI 2016. arXiv:161008844

167. Twinanda AP, Shehata S, Mutter D, Marescaux J, de Mathelin M, Padoy N (2016) EndoNet: A Deep Architecture for Recognition Tasks on Laparoscopic Videos

168. Unger M, Chalopin C, Neumuth T (2014) Vision-based online recognition of surgical activities. Int J Comput Assist Radiol Surg 9:979–986. doi: 10.1007/s11548-014-0994-z

169. Vedula SS, Hager GD (2017) Surgical data science: the new knowledge domain. Innov Surg Sci 2. doi: 10.1515/iss-2017-0004

170. Volkov M, Hashimoto DA, Rosman G, Meireles OR, Rus D (2017) Machine learning and coresets for automated real-time video segmentation of laparoscopic and robot-assisted surgery. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp 754–759

171. Widgor D, Wixon D (2011) Brave NUI world: designing natural user interfaces for touch and gesture. Elsevier

172. Yamato J, Ohya J, Ishii K (1992) Recognizing human action in time-sequential images using hidden Markov model. In: Proceedings 1992 IEEE Computer Society Conference on Computer Vision and Pattern

Recognition. IEEE Comput. Soc. Press, pp 379–385

173. Zappella L, Béjar B, Hager G, Vidal R (2013) Surgical gesture classification from video and kinematic data. Med Image Anal 17:732–745. doi: 10.1016/j.media.2013.04.007

174. Zhou S-M, Chiclana F, John RI, Garibaldi JM (2011) Alpha-level aggregation: a practical approach to type-1 OWA operation for aggregating uncertain information with applications to breast cancer treatments. IEEE Trans Knowl Data Eng 23:1455–1468

175. Zia A, Zhang C, Xiong X, Jarc AM (2017) Temporal clustering of surgical activities in robot-assisted surgery. Int J Comput Assist Radiol Surg 12:1171–1178. doi: 10.1007/s11548-017-1600-y