Int J CARS
DOI 10.1007/s11548-008-0239-0

ORIGINAL ARTICLE

Workflow mining for visualization and analysis of surgeries

Tobias Blum · Nicolas Padoy · Hubertus Feußner · Nassir Navab

Received: 10 January 2008 / Accepted: 3 June 2008

Abstract
Objective Modeling the workflow of a surgery is a topic of growing interest. Workflow models can be used to analyze statistical properties of a surgery, for intuitive visualization, evaluation and other applications. In most cases, workflow models are created manually, which is a time consuming process that might suffer from a personal bias. In this work, an approach for automatic workflow mining is presented.
Materials and methods Ten process logs, each describing a single instance of a laparoscopic cholecystectomy, are used to build a Hidden Markov Model (HMM). Using a merging approach, models at different levels of detail are generated. These embody statistical information concerning aspects like duration of actions or tool usage during the surgery.
Results A Graphical User Interface (GUI) is presented, that uses a graph representation of the HMM to intuitively visualize surgical workflow. It allows changing the level of detail by expanding and merging nodes. The GUI can also be used to compare videos of surgeries which are synchronized to the model.
Conclusions The proposed method allows automatic generation and visualization of a statistical model describing the workflow of a surgery.

T. Blum (✉) · N. Padoy · N. Navab
Computer Aided Medical Procedures (CAMP),
Technische Universität München, Munich, Germany
e-mail: blum@cs.tum.edu

H. Feußner
Department of Surgery, Klinikum Rechts der Isar,
Technische Universität München, Munich, Germany

N. Padoy
LORIA-INRIA Lorraine, Nancy, France

Keywords Workflow mining · Surgical workflow analysis · Information visualization · Cholecystectomy · Hidden Markov models

Introduction

van der Aalst [1] distinguishes between two approaches towards workflow design. The traditional approach is typically used to improve a process or develop a new one. Here, experts design a new workflow that is implemented afterwards. In recent years, there has been an increasing interest in another approach called Workflow mining. It reverses the traditional method by collecting data at runtime and automatically deriving a model from this data. The Workflow Mining approach, also called Process Mining, relies on an existing workflow and is therefore complementary to the traditional one. Obtaining a model that reflects an existing workflow is of great value for understanding and redesigning it.

While it is possible to design such a model of an existing workflow manually, there are many advantages of automating this work. Manual design is a very time consuming process that requires expert knowledge and it may be affected by a personal bias. Often, manually designed models are normative in the sense that they describe what should be done rather than describing the actual process [1]. To avoid this, workflow mining uses a set of process logs, each describing one instance of the process, and automatically derives the model from these logs. Automatic modeling can certainly not replace expert knowledge, but analysis of workflow can benefit from such more objective methods.

Most work in this area focuses on higher level workflows, e.g. patient or information flows in the medical domain. In [2], simulated process logs of hospital-wide workflows, containing events like *blood test* or *surgery*, were used to build

Petri-Net like models. A merging technique is used to build a model based on 500 event traces.

Also for computer aided surgery, models that are built by retrospective analysis of surgical data are of great use [3]. So for example a model that describes an existing surgical workflow is extremely valuable when analyzing a surgery for pre-development planning of surgical assist systems. Neumuth et al. [4] analyzed important aspects of micro-surgical lumbar distectomies for the purpose of assessing the potential of a new system. They defined several aggregations tasks for aspects of interest and directly derived the information from the process logs.

For designing future Operating Rooms (ORs), which are specialized for certain surgeries, a model can also reveal important information concerning the surgery. Workflow mining could be used efficiently in post-development stages as well: for example to evaluate new surgical assist systems, by comparing a model of the old workflow to a model of the new one, or by comparing the model of the designed and planned workflow to a model of the real workflow, once the staff is daily using the system. When using statistical models, this can even be used to perform quantitative comparisons. So it is possible to analyze aspects like the time spent for different phases, the use of surgical tools and imaging modalities, or the different courses an intervention can take.

Furthermore, models can be used to visualize the workflow of a surgery. An important advantage of graphical representations is that they ease interdisciplinary communication, which is a crucial issue for developing surgical assist systems. Additionally, they can be used for educational purposes. Advantages and methods for visualizing the process log of a single instance of a surgery have been discussed by Neumuth et al. [5]. They used different perspectives on the process log, emphasizing either on temporal or logical relations between work steps.

Most workflow mining methods use Petri-Net like models, as these share many similarities to workflow models that are established in business sciences. In this work, a statistical approach, using Hidden Markov Models (HMMs) is taken to model the workflow inside the OR. Unlike Petri-Nets, HMMs do not allow for concurrent actions, but can be used to model properties like the average duration of actions, the variance of their duration or transitions probabilities between single actions. They are also very flexible and can be used with continuous data, like positions of the surgeon or tools, obtained by a tracking system. Another reason for choosing HMMs is that usually the process logs contain only a relative low number of different high-level events. Each of these events can be represented by one Petri-Net state. In our case, the surgical workflow is modeled at a finer level of detail and many different combinations of tools might be used. Using Petri-Nets, we would have to represent each of these combinations with one state, resulting in a huge model. Using HMMs we can represent different combinations of tools using only one state.

HMMs with a small number of states have been used previously to model surgical actions for evaluating and understanding surgical skills. In [6] a surgical simulator was used to obtain position data of two surgical tools during an exercise of touching a virtual sphere. Based on this data, a four-state HMM was trained and used to gain insights about hand movement patterns. In a similar setup [7] used position data from a simulator to train four-state HMMs in order to classify instrument trajectories of experts or novices. In [8] force- and torque-data from a simulator were used to build a 15-state Markov Model describing the action of tying a knot in a minimally invasive setup. Using models of experts and residents, a learning curve could be shown. In these models however, it is difficult to give semantic meaning to each single state.

Laparoscopic cholecystectomy has been modeled previously in [9] using timed automata for the sake of developing a surgical assist robot system. They modeled the behavior of a surgeon and a scrub nurse, but currently this model is not linked to process logs and does not allow for any statistical analysis. We have presented a completely automated approach for generating a model of a laparoscopic cholecystectomy using dynamic time warping (DTW) in [10]. This method allows analyzing the sequence and average duration of actions and their occurrence along the timeline. The main drawback of this approach is that DTW can only handle variations over time and has problems when an intervention can take different courses. Its graphical representation is linear and contains less information than an HMM, which is therefore more intuitive. For segmentation purposes, a HMM model of cholecystectomies was presented in [11]. It allows recovering identified surgical phases automatically, but contains few states. As each phase is represented by only one HMM state, the actions occurring inside one phase are loosely modeled and no insight is given about the ordering of their occurrences.

In this work, we present the use of a probabilistic merging approach to generate HMMs, describing the workflow of a laparoscopic cholecystectomy at different levels of detail. This is done based on process logs, representing tool usage during ten surgeries. It is investigated, how this model can be used to provide an intuitive graphical visualization of the workflow. A GUI is presented that can be used to display the model at different levels of detail and gives access to statistical properties.

Materials and methods

Cholecystectomy is a common but complex surgery that is performed laparoscopically in 95% of the cases. The objective of this surgery is to remove the gallbladder and it is usually performed because of symptomatic gallstone disease.

It starts with the positioning of four trocars which are used for insertion of the instruments into the body. The most important intermediate steps are the dissection, clipping and cutting of the bile duct and of the cystic artery. Next, the gallbladder is separated from the liver and removed using a retraction sac. It finishes with the removal of the trocars and the suturing of the induced holes.

In minimally-invasive surgeries, the instrument use strongly correlates with the underlying surgical workflow. To record the surgical actions during the procedure, instrument presence was acquired for $K = 17$ laparoscopic instruments and represented as a multivariate time series S, where

$S_{t,k} = 1$ *if and only if* instrument k is used at time t.

The instrument signals for an exemplary operation are displayed in Fig. 1. The vertical lines display the segmentation in 14 phases. These phases have been defined by surgeons according to clinical practice. The beginning and end of all phases could be identified without ambiguity in all ten surgeries. These multivariate time series can also be seen as process logs where each time an instrument is inserted or removed, a new action starts. So, several consecutive, identical instrument vectors will be referred to as action.

For this work ten surgeries $V = \{S^1, \ldots, S^{10}\}$ have been acquired and labeled with 14 phases. One surgeon did nine of the surgeries, where some parts have been performed by assistants. The tenth surgery has been done completely by another surgeon from the same school.

Hidden Markov models

A HMM is a statistical model that can be used to classify and generate time series. We will follow the notation of the classical HMM tutorial [12]. A HMM is described by the quintuplet $\lambda = (A, B, \pi, N, M)$, where N is the number of hidden states and A defines the probabilities of making a transition from one hidden state to another. M is the number of observation symbols. In our case the observation symbols consist of one symbol for each combination of tools that can be used. B defines a probability distribution over all observation symbols for each state. π is the initial state distribution accounting for the probability of being in one state at time $t = 0$. An exemplary HMM, describing one phase of the laparoscopic cholecystectomy, is shown in Fig. 2. The natural way to visualize a HMM is by using a graph, where nodes represent the hidden states and the edges represent the transition probabilities. The nodes are labeled according to the observation symbol probability. In this example, the average time spent in one state is visualized by the size of the state and the occurrence of a transition by the size of the edge. The phase visualized here usually consists of three clipping actions and the use of scissors at the end. But it also happens that clipping is performed two or four times and in some cases additional instruments are used. The remainder of this section addresses the problem of automatically generating a model as seen in Fig. 2 from a set of process logs or time series. Instead of directly building a model of the whole surgery, for computational issues, every phase is processed independently, and the resulting models are concatenated at the end.

Given the parameters λ, the probability of a HMM generating, or explaining, a certain observation sequence $P(S|\lambda)$ can be computed. When modeling the workflow of a surgery we seek for a model that explains the training examples well, i.e. find a set of parameters with a high probability $P(V|\lambda) = P(S^1|\lambda) * \cdots * P(S^{10}|\lambda)$. A common way to do this is to randomly initialize a HMM several times and use the expectation maximization (EM) method which iteratively changes A, B and π, converging to a local maximum of $P(V|\lambda)$. However, since the initialization is random, the

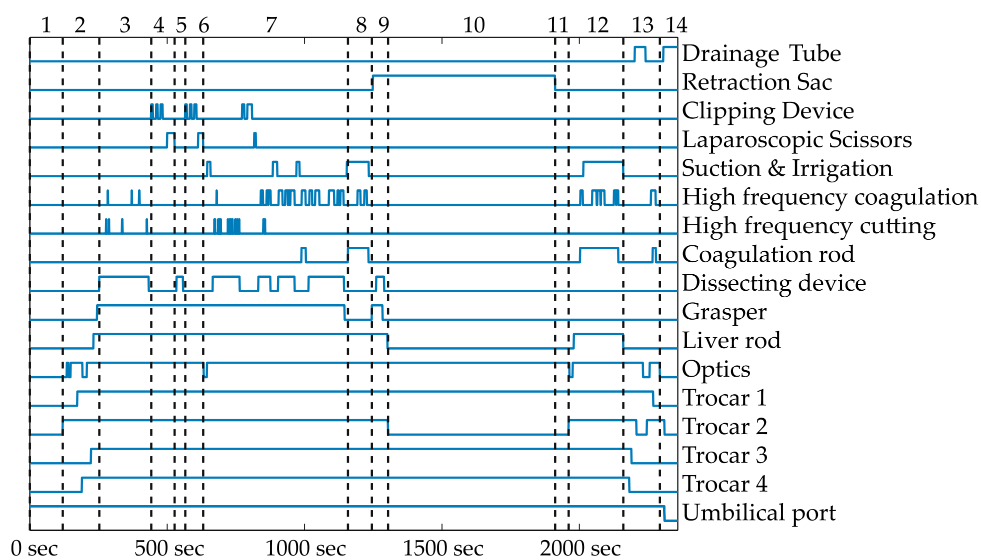

Fig. 1 Instrument use during one exemplary surgery. Phases are indicated by the *dotted lines*

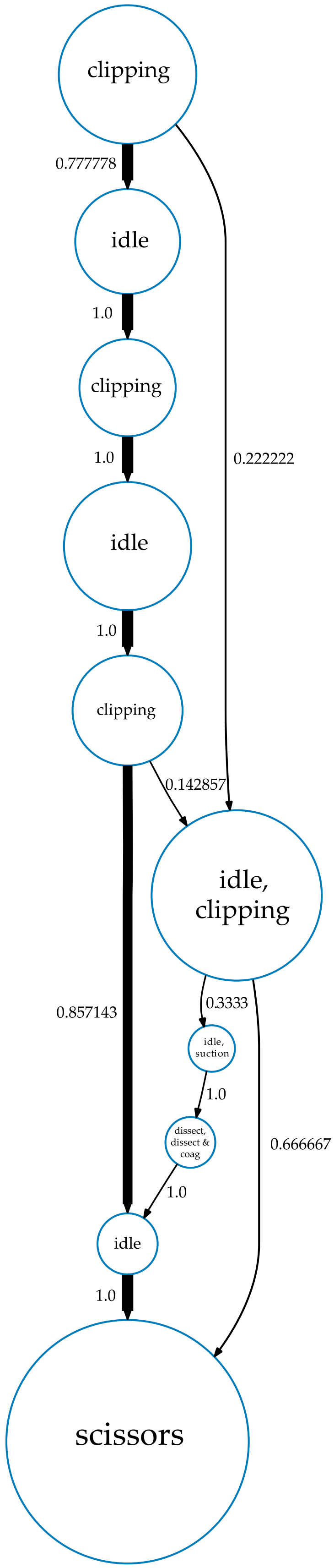

Fig. 2 Visualization of an HMM describing phase 4 of the surgery

obtained model is not easily reproducible and the method does not comply with the objective of generating a human-readable graph. This method is also not capable of changing the topology as it cannot change the number of states or add transitions. Another way to build an HMM would be to design the topology manually using approximate values for A, B and π and then use EM to estimate better values for them. This however is a laborious task that requires much expert knowledge. Instead, we use another approach that automatically derives both, the topology and also the other parameters from the training data.

Model merging

In order to automatically build the HMM topology, a HMM merging approach is used [13]. Here, an initial HMM λ_0 is constructed by adding parallel paths of states for each surgery. Each of these paths contains one state for each action and describes the actions that occurred in one process log. An illustrative example of this, using only two surgeries and simplified transition probabilities, is shown in Fig. 3. This initial model is of no use as it only explains the training data but does not describe the general workflow of this phase. To obtain a more compact model that describes the general workflow, states are merged iteratively. In each step t, a HMM with parameters λ_t is generated by merging two states from λ_{t-1}. The algorithm uses a best-first heuristic, always choosing the two states with maximum value for $P(V|\lambda_t)$. As can be seen in the example in Fig. 3, after several merging steps a more compact model is obtained that still explains both training sequences.

A straightforward implementation of this method cannot be computed in reasonable time, as its complexity is $O(N^5TL)$, where N is the number of initial states, T the duration of the surgeries and L the number of training surgeries. Using an approximation of $P(V|\lambda_t)$ based on the Viterbi path approximation (see [14] for a detailed explanation) and a graph implementation of the HMM structure, this can be reduced to $O(N^2TL)$. For the observation symbol probability, different distributions have been tested. The merging process showed to be largely unaffected by the choice of the probability distribution. To obtain a sequential model, that does not contain loops, the algorithm can be modified to only allow merges that do not lead to a topology with loops.

Some properties of the merging approach can be seen in Fig. 2. For the typical workflow of this phase, one state per action is maintained. If there are several actions that occur less frequently, they are likely to be merged as this will only slightly reduce $P(V|\lambda_t)$. So we obtain a model that focuses on the typical workflow, but also contains unusual events.

A HMM embodies much information that can be used to analyze the workflow. Using the transition probabilities, the

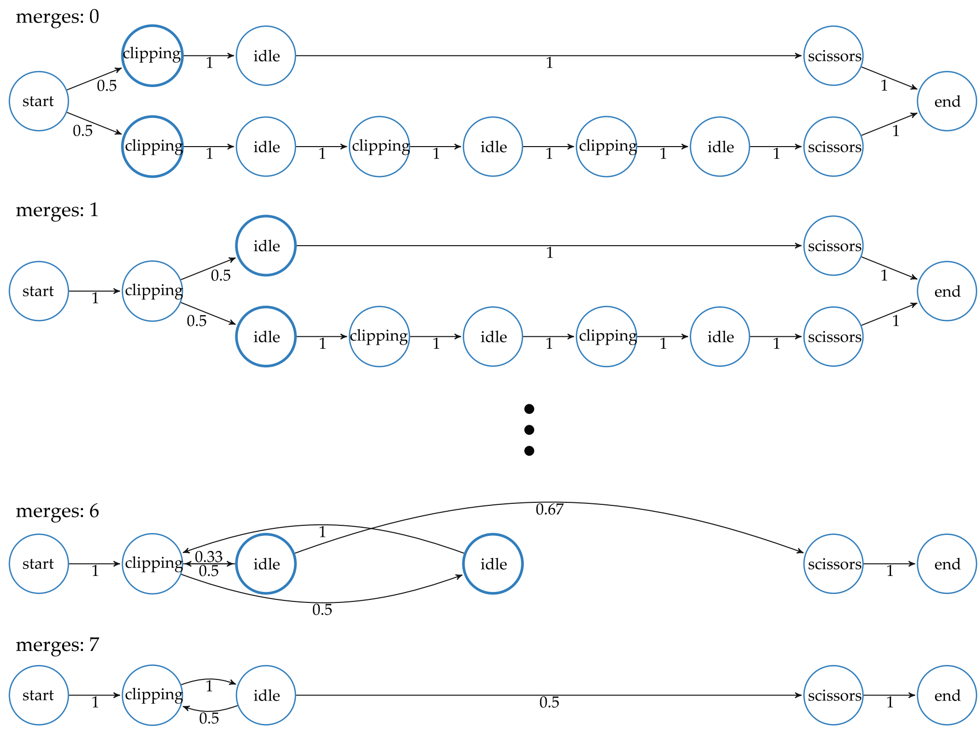

Fig. 3 Illustration of the model merging approach. The initial model is build from two exemplary sequences {clipping, idle, scissors} and {clipping, idle, clipping, idle, clipping, idle, scissors}. The two states that are merged in the next step are indicated by the *bold border*

average time spent in one state and the probability of making a transition to another state can be computed. It is also possible to compute the variance of the average duration or the average time an instrument is used. From the transitions probabilities or using sampling techniques, it is also possible to estimate properties like the average time between two states or the likelihood of taking a certain intervention course.

It should be noted that in every step a valid model is generated. So, the result of this merging algorithm is not a single HMM but a set of HMMs with decreasing number of states. The whole process can also be seen as a tree, where the states of the initial HMM are the leaves and each state that is generated by merging is represented as the parent of the two merged states. This will be used to allow changing the level of detail when visualizing the workflow, as explained below.

Results

To allow visual analysis of the workflow and to access the statistical properties of the HMM a GUI is used. An example of this GUI is shown in Fig. 4. As already mentioned, the natural way of visualizing a HMM is using a graph. For generating the spatial layout of the nodes and edges GraphViz [15], an open source graph drawing tool, is used. The edges are weighted according to the transition probability they represent, and GraphViz builds the layout of the graph to keep edges with a high weight short and straight. Doing this, the typical workflow is visualized along a straight line, while unusual actions are visualized to the left or right of the typical sequence of actions. The current version of the GUI allows to access the most important statistical properties of the HMM. The transition probabilities and average duration of a state are shown when moving the mouse onto the corresponding node or edge. The average time spent in one state across all ten training surgeries is visualized by the size of the node.

A general problem in information visualization is to display the right amount of information. A graph, containing only few nodes, for the most common actions, will be best suited to understand and analyze the most important aspects of the workflow. But it is not appropriate for detailed analysis or for examining uncommon events. Displaying every bit of information by using a graph with a huge number of states, allows a more detailed analysis of the workflow. But such a visualization will also include a lot of unimportant information and will be hard to interpret. To deal with this problem, we reduce the number of visible elements by clustering. When doing clustering, the graph is simplified by merging several nodes into one. By allowing splitting it again, it is possible to analyze parts of the graph in more detail. Most mentions of clustering in graph visualization are purely structure-based [16], using only the graph structure to perform clustering. We use a content-based approach by utilizing the information obtained during the model merging process. Nodes can be expanded or merged using the merging information stored in the tree structure that was described in the last section. This method is content-based as it does not rely on aspect like neighborhood information in the graph,

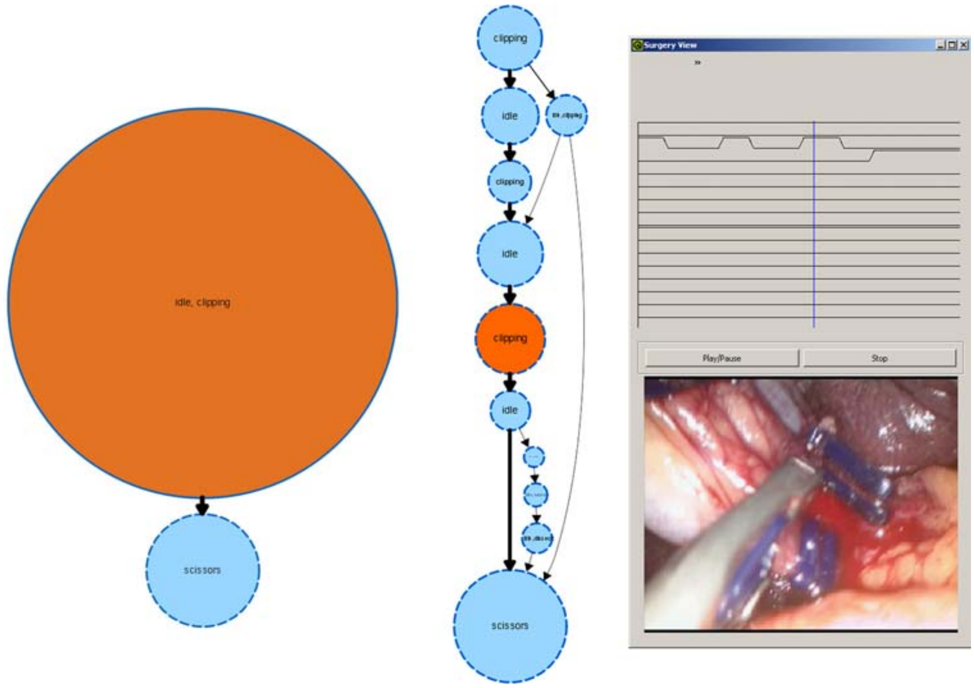

Fig. 4 In the GUI, the level of detail of the HMM can be changed simply by expanding or merging states. On the *left side* a coarse visualization is used, where only one state for clipping and idle and another for scissors is used. On the *right side*, the nodes have been expanded several times. The level of detail can also be changed during video replay. While the video is playing, the corresponding node in the model is highlighted

but on the merging process, which is driven by the data. A very compact visualization as seen in Fig. 4 on the left side can be expanded to a more detailed view as seen on the right side. While it would be possible to allow splitting states until the initial HMM λ_o is reached, the GUI limits the number of splits. Only splits are allowed that significantly raise the probability $P(V|\lambda)$. A split that does not raise this probability, does also not contribute to better explaining the workflow as it adds no significant amount of information. It must be noted that the order of merges during the model merging does not restrict the order in which the nodes can be expanded in the GUI. So it is possible to expand one part of the graph, while viewing another part at a low level of detail.

The GUI allows to simultaneously replay a video of the surgery and highlight the corresponding state of the HMM. This can also be seen in Fig. 4. On top of the video, the instrument vector or process log of this surgery is displayed. To synchronize the video of surgery i with the HMM, the Viterbi algorithm [12] is used to estimate the most likely sequence of states in the model given the observations of S^i. While the video is running, the level of detail of the HMM can still be changed.

In addition to visualizing the HMM as a graph, the statistical parameters can be analyzed directly. Table 1 shows some of the parameters of the right HMM in Fig. 4. In this table, the average time spent in the states, the probability of reaching a state and the instrument use in the states are shown. Again we can make use of the merging, and display the statistical properties of a more compact HMM. The parameters of the compact one shown on the left side of Fig. 4 are given in Table 2.

Table 1 Statistical properties embodied in the 11-state HMM shown on the right of Fig. 4

Node	Avg. duration (s)	Probability of reaching (%)	Probability of an instrument being used
First clipping	11.44	100.00	Clipping device = 100.00
First idle	12.71	77.78	No instrument = 100.00
Second clipping	9.00	77.78	Clipping device = 100.00
Second idle	13.25	88.89	No instrument = 100.00
Third clipping	14.13	88.89	Clipping device = 100.00
Third idle	7.13	88.89	No instrument = 100.00
Scissors	21.33	100.00	Scissors = 100.00
Exception 1	29.00	22.22	Clipping device = 44.83
			No instrument = 55.17
Exception 2	20.00	11.11	Clipping device = 65.00
			No instrument = 35.00
Exception 3	27.00	11.11	Suction and irrigation = 75.00
			No instrument = 25.00
			Dissecting device =76.32
Exception 4	38.00	11.11	HF cutting = 5.26
			No instrument = 18.42

Note, that the size of the nodes in Fig. 4 represents the average duration multiplied by the probability of reaching the node

Discussion

Automatic generation of workflow models is of great value for analyzing the surgical workflow. We presented a method

Table 2 Statistical properties embodied in the 2-state HMM shown on the left of Fig. 4

Node	Avg. duration (s)	Probability of reaching node (%)	Probability of an instrument being used (%)
Clipping and idle	78.67	100.00	Clipping device = 47.38
			Suction and irrigation = 2.89
			Dissecting device = 4.13
			HF cutting = 0.28
			No instrument = 45.32
Scissors	21.33	100.00	Scissors = 100.00

that allows mining of a statistical model from a set of process logs. This method was demonstrated at the example of a laparoscopic cholecystectomy. In this work only binary information about instrument use during a surgery was used. As HMMs are very flexible and can also deal with other types of information, this approach can be adapted to use position or movement data, biomedical signals and other kind of sensors. We believe that in the future such workflow mining methods will become even more important, as the number of sensors in the OR is steadily increasing. In the future, process logs could be generated automatically for each surgery, containing information like tool presence or even tool position, obtained using RFID-technology, position and orientation of the patient table or information acquired from video images using computer vision techniques.

The data used for this work was acquired manually by videotaping the surgeries and labeling them afterwards, which makes it hard to record a large number of surgeries. To be able to get more data, we are currently developing a trocar equipped with a sensor that is capable of detecting insertion and removal of instruments. Mining large amounts of process logs would not only result in accurate statistical information about surgical procedures, but also allow to analyze common or complex problems or to compare surgeons, hospitals and different ways/schools to carry out a surgery. One limitation of the model merging method is that, despite using some approximations, it is still very slow. The computation has however to be performed only once. For processing a larger number of process logs it will be an issue how to speed it up. Some ideas to handle this problem have already been discussed in [14].

Always when dealing with information it must also be taken into account, how to present this information in an appropriate way. In addition to the generation of the model, we have presented a graph visualization where the level of detail can be changed by the user, based on a content-based clustering. Using the GUI, that can also replay synchronized surgical videos, it is very easy and intuitive to explore the workflow of a surgery.

This work provides a novel approach to analyze the surgical workflow. The model and its visualization were evaluated by our medical partner, who verified the consistency of the automatic generation. He is now very interested in using this method for objective analysis of workflow, educational purposes and benchmarking of experienced surgeons. Future work will therefore focus on carrying out a systematic validation of the method, by collecting data from different surgeons and analyzing their feedbacks on the resulting model. Such a workflow mining approach can certainly not replace manual modeling and analysis of workflow using expert knowledge. It is, however, a complementary method, which allows getting accurate statistical information and an unbiased view on surgical workflow.

References

1. van der Aalst WMP, van Dongen BF, Herbst J, Maruster L, Schimm G, Weijters AJMM (2003) Workflow mining: a survey of issues and approaches. Data Knowl Eng 47:237–267
2. Maruster L, van der Aalst WMP, Weijters AJMM, van der Bosch A, Daelemans W (2002) Automated discovery of Workflow Models from hospital data. In: Proceedings of the ECAI Workshop on Knowledge Discovery and Spatial Data. pp 183–190
3. Jannin P, Morandi X (2007) Surgical models for computer-assisted neurosurgery. NeuroImage 37:783–791
4. Neumuth T, Trantakis C, Eckhardt F, Dengl M, Meixensberger J, Burgert O (2007) Supporting the analysis of intervention courses with surgical process models on the example of 14 microsurgical lumbar discectomies. Int J Comput Assist Radiol Surg 2(Suppl 1):436–438
5. Neumuth T, Schumann S, Strauß G, Jannin P, Meixensberger J, Dietz A, Lemke HU, Burgert O (2006) Visualization options for surgical workflows. Int J Comput Assist Radiol Surg 1(suppl 1):438–440
6. Megali G, Sinigaglia S, Tonet O, Cavallo F, Dario P (2006) Understanding expertise in surgical gesture by means of Hidden Markov Models. International conference on biomedical robotics and biomechatronics (BioRob), Tuscany, Italy, pp 625–630
7. Leong JJH, Nicolaou M, Atallah L, Mylonas GP, Darzi AW, Yang GZ (2006) HMM assessment of quality of movement trajectory in laparoscopic surgery. In: Proceedings of medical image computing and computer-assisted intervention (MICCAI), Copenhagen, Denmark, pp 752–759

8. Rosen J, Brown JD, Chang L, Sinanan MN, Hannaford B (2006) Generalized approach for modeling minimally invasive surgery as a stochastic process using a discrete Markov model. IEEE Trans Biomed Eng 53(3):399–413
9. Ohnuma K, Masamune K, Yoshimitsu K, Sadahiro T, Vain J, Fukui Y, Miyawaki F (2006) Timed-automata-based model for laparoscopic surgery and intraoperative motion recognition of a surgeon as the interface connecting the signal scenario and the real operating rool. Int J Comput Assist Radiol Surg 1(suppl 1):442–445
10. Ahmadi A, Sielhorst T, Stauder R, Horn M, Feußner H, Navab N (2006) Recovery of surgical workflow without explicit models. In: Proceedings of medical image computing and computer-assisted intervention (MICCAI), Copenhagen, Denmark, pp 420–428
11. Padoy N, Horn M, Feußner H, Berger MO, Navab N (2007) Recovery of surgical workflow: a model-based approach. Int J Comput Assist Radiol Surg 2(suppl 1):481–482
12. Rabiner LR (1989) A tutorial on Hidden Markov Models and selected applications in speech recognition. Proc IEEE 77(2):257–286
13. Stolcke A, Omohundro S (1994) Inducing probabilistic grammars by Bayesian model merging. Grammatical inference and applications. In: Proceedings of the second international colloquium on grammatical inference, Alicante, Spain, pp 106–118
14. Stolcke A, Omohundro S (1994) Best-first model merging for hidden markov model induction. Technical report TR-94–403, ICSI, Berkeley, CA
15. Ellson J, Ganser E, Koutsofis L, North SC, Woodhull G (2001) GraphViz—open source graph drawing tools. Graph drawing: 9th international symposium, Vienna, Austria, pp 483–484
16. Herman I, Melançon G, Marshall MS (2000) Graph visualization and navigation in information visualization: a survey. IEEE Trans Visualiz Comput Graph 6(1):24–43