Int J CARS (2016) 11:1201–1209
DOI 10.1007/s11548-016-1409-0

ORIGINAL ARTICLE

System events: readily accessible features for surgical phase detection

Anand Malpani[1] · Colin Lea[1] · Chi Chiung Grace Chen[2] · Gregory D. Hager[1]

Received: 11 February 2016 / Accepted: 31 March 2016 / Published online: 13 May 2016

Abstract
Purpose Surgical phase recognition using sensor data is challenging due to high variation in patient anatomy and surgeon-specific operating styles. Segmenting surgical procedures into constituent phases is of significant utility for resident training, education, self-review, and context-aware operating room technologies. Phase annotation is a highly labor-intensive task and would benefit greatly from automated solutions.
Methods We propose a novel approach using system events —for example, activation of cautery tools—that are easily captured in most surgical procedures. Our method involves extracting event-based features over 90-s intervals and assigning a phase label to each interval. We explore three classification techniques: support vector machines, random forests, and temporal convolution neural networks. Each of these models independently predicts a label for each time interval. We also examine segmental inference using an approach based on the semi-Markov conditional random field, which jointly performs phase segmentation and classification. Our method is evaluated on a data set of 24 robot-assisted hysterectomy procedures.
Results Our framework is able to detect surgical phases with an accuracy of 74 % using event-based features over a set of five different phases—ligation, dissection, colpotomy, cuff closure, and background. Precision and recall values for the cuff closure (Precision: 83 %, Recall: 98 %) and dissection (Precision: 75 %, Recall: 88 %) classes were higher than other classes. The normalized Levenshtein distance between predicted and ground truth phase sequence was 25 %.
Conclusions Our findings demonstrate that system events features are useful for automatically detecting surgical phase. Events contain phase information that cannot be obtained from motion data and that would require advanced computer vision algorithms to extract from a video. Many of these events are not specific to robotic surgery and can easily be recorded in non-robotic surgical modalities. In future work, we plan to combine information from system events, tool motion, and videos to automate phase detection in surgical procedures.

Keywords Surgical phase detection · System events · Sensor data · Surgical workflow analysis · Robot-assisted surgery · Surgical task flow · Surgical process modeling

Introduction

Birkmeyer et al. [1] have shown that postoperative outcomes are associated with technical skills of the operating surgeon and that peer review may be useful to assess surgical skills. Such peer review is impractical at scale due to time and resource constraints. However, this may become tractable if new tools are developed to efficiently index all surgical phases within each procedure.

We posit computational models that automatically analyze surgical procedures and extract critical phases will benefit both manual and automated video review. Computational models could also help focus surgical training by detecting and annotating common errors that occur in each step of a surgery. In addition, phase cataloging may be important for

✉ Anand Malpani
amalpan1@jhu.edu

[1] Department of Computer Science, The Johns Hopkins University, 3400 N. Charles St., Malone Hall Room 340, Baltimore, MD 21218, USA

[2] Department of Gynecology and Obstetrics, The Johns Hopkins University, Johns Hopkins Bayview Medical Center, 301 Mason Lord Drive, Suite 3200, Baltimore, MD 21224, USA

self-review and context-aware operating room technologies. For example, trainees could be shown a set of relevant surgical phase videos from the catalog based on a structured query. Surgeons could be provided statistics on the phases from their previous operating room performances along with patient outcomes. Useful information related to the current phase of the surgery could be displayed to the operating room members to enhance workflow efficiency.

In this paper, we describe work toward automated surgical phase detection in efforts to make these tools a possibility. The method we present relies on readily available event data such as a binary signal indicating whether an energy instrument is active. Although our data were acquired from a da Vinci surgical robot, we show that we achieve similar performance using only events that are easily acquired from most surgical platforms for laparoscopic, endoscopic, and open surgeries. The event-based signals are simpler than video or kinematic data, but, as we show later, can be highly discriminative of surgical phase.

Few papers have focused on using event-based data for phase recognition. The structured review presented in [2] shows that there has been a significant effort since 2002 to develop methods for surgical process modeling, but only a small fraction of this work has addressed surgical phase segmentation. Methods using techniques such as dynamic time warping [3,4], canonical correlation analysis [5], hidden Markov models [6], random forests [7], support vector machines, and conditional random fields [8] have been used on sensor data recorded during laparoscopic cholecystectomy procedures in order to perform surgical phase modeling. However, the sensor data used in this work—carbon dioxide pressure, weight of the irrigation and suction bag, inclination of the surgical table—require additional, and sometimes sophisticated, instrumentation of the operating room prior to the surgery. The method presented by Neumuth et al. in [9] for surgical phase detection by jointly representing each low-level action using the action class, instrument, and anatomy has been recently applied by Forestier et al. [10] to detect phases of surgery using manually labeled low-level activity information. Similarly, Katic et al. [11] proposed a rule-based surgical workflow analysis using manual low-level activity labels for phase detection. The low-level activity data that these approaches rely upon require explicit manual labeling, thereby limiting their scalability.

Previous approaches using tool motion data, video data, and combination of both have been developed to perform surgical process modeling. However, most of this work has operated at a different level of abstraction than phases. Twinanda et al. [12] performed whole procedure classification using endoscopic video data. Other work has focused on detection of low-level activities at the maneuver/subtask and gesture/surgeme level using machine learning approaches such as hidden Markov models [13–15], linear dynamical systems [16,17], conditional random fields [18,19], and many more. However, to the best of our knowledge, none of these methods have been successfully applied at the surgical phase granularity using live surgery data.

In the remainder of this paper, we present a framework for surgical phase detection using features obtained from system events collected from the da Vinci Surgical system (dVSS; Intuitive Surgical, Inc., Sunnyvale, CA), and we demonstrate its effectiveness at performing surgical phase recognition in robot-assisted hysterectomy.

Methods

Our phase detection framework consists of: aggregating system events over short time intervals (section "Feature extraction"), computing the surgical phase probability for each interval (section "Phase scoring"), and jointly segmenting and classifying all surgical phases (section "Joint phase segmentation and classification").

Feature extraction

We define a set of features, highlighted in Table 1, that summarize tool and event information within each 90-s interval. These features are motivated by the notion that many surgical phases must be completed using a specific set of tools. For example, a *Cuff Closure* should ideally be performed using a large needle driver.

We categorize tools into three types: *monopolar energy*, *bipolar energy*, and *normal*. The first two refer to cautery tools and the last refers to non-energized tools such as a needle driver. Note while some tools are intended for cautery actions, there are times when a surgeon will use them for other tasks like grasping.

For cautery tasks, the surgeon uses one form of energy over the other based on the step of the procedure and the surrounding anatomy. For example, a surgeon applies "bipolar" energy to coagulate a structure that is small enough to be grasped between its two grippers. This tool isolates most of the electrosurgical current passed to the grasped tissue or blood vessel. In contrast, a *monopolar* tool is used when dissecting a larger area where there are no significant anatomic structures or vasculature.

We use additional events recorded by the da Vinci including tool identity, tool changes, movement of the endoscope, repositioning ("clutching") the manipulators in the surgical console, and a head-in indicator indentifying whether a surgeon is working at the console. For evaluation, we compute results using events common among most surgical systems as well as ones also available for the da Vinci.

There are three types of features corresponding to the duration of an event during each 90-s interval, how many times it was activated, and whether or not it was in use within that

Table 1 System events-based features and their descriptions

Name	Description
Fraction of segment length for which	
MonopolarCutTime	Monopolar cut energy was active
MonopolarCoagTime	Monopolar coagulation energy was active
BipolarTime	Bipolar energy was active
TotalTime	Any of the energy types was active
CameraTime	Camera was moved
ClutchTime	Clutch was pressed
HeadInTime	Surgeon was looking into the console
Number of times	
MonopolarCutCount	Monopolar cut energy was activated
MonopolarCoagCount	Monopolar coagulation energy was activated
BipolarCount	Monopolar cut energy was activated
TotalCount	Any of the energy types was activated
CameraCount	Camera was moved
ClutchCount	Clutch was pressed
Binary flag indicating	
IsMonopolarTool	A monopolar instrument was in use
IsBipolarTool	A bipolar instrument was in use
IsNormalTool	A non-energy instrument was in use

period (as listed in Table 1). We compute a feature vector $\mathbf{f}_t$ for each time interval from 1 to T composed of each item listed in Table 1. When using all da Vinci events, each vector is of length 16.

Figure 1 shows a subset of the above features for a sample procedure from our data set.

Phase scoring

A score is computed for each interval which corresponds to the likelihood that the interval belongs to each class. Let $s_t \in \mathbb{R}^C$ be a vector at time t where C be the number of surgical phase classes. We compare three score models. The first is a linear model applied to features at each time step, the second assumes a nonlinear model applied to each time step, and the third assumes a nonlinear model applied to sequences of time steps.

Linear frame-wise model The first model assumes there is a linear vector $w_c \in \mathbb{R}^{16}$ that discriminates phase c from the rest of the data. Let the score $s_t^c = w_c^T f_t$. If phase label $y_t = c$ then the correct score, $s_t^{y_t}$ should be higher than the score for any other class such that $s_t^{y_t} > s_t^c$ for all c where $c \neq y_t$. We learn weights w with a one-versus-all support vector machine (SVM).

Nonlinear frame-wise model Each phase may be best classified using a nonlinear mapping of the given features in each interval. We follow the work of Stauder et al. [7] who model surgical phase using a random forest classifier. A random forest is an ensemble learning method that randomly learns which features are most indicative of each class. At each node in the tree, a subset of the features from the training data are selected and tested for their Gini's index as described in [20]. In our data, we observe different subsets of features are important in characterizing different active surgical phase; thus, the random forest is well suited to our problem. The score for the cth class is given by the posterior probability $s_t^c = P(c|f_t)$ as computed by this model.

Nonlinear temporal model The previous two models assume the label at each time step is only a function of the data at the current time step. However, in many phases the features may change substantially between the start and the end of a phase. For example, a surgeon may use a monopolar tool at the start of a dissection and a bipolar tool at the end.

We apply the temporal convolutional neural network (tCNN) of [21] to capture long-range dependencies across intervals. A set of temporal filters $W_I \in \mathbf{R}^{d \times F}$ model the features across a sequence of d intervals where F is the number of features in each interval. Let there be a total of I tempo-

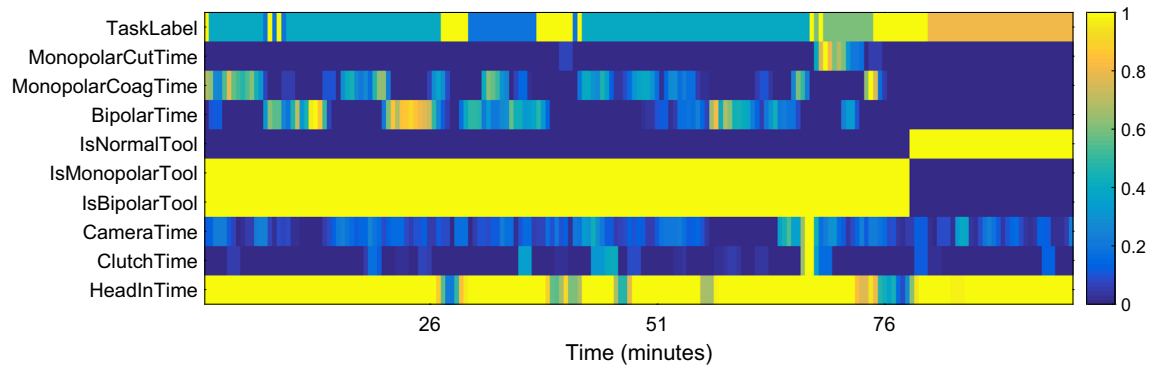

Fig. 1 System events-based features for a sample hysterectomy procedure from our data set. (*Note* feature values have been scaled to [0,1] for better contrast)

ral filters. Each filter models how features change over the course of a phase. The data for each class can be modeled as a function of these weights where variable α_i^c weighs how important each filter W_i is for class c. The score is computed as $s_t^c = \sum_{i=1}^{I} \alpha_i^c W_i * f_{t:t+d}$ where $f_{t:t+d}$ denotes the set of features from times t to $t+d$. Symbol $*$ refers to a temporal convolution where the features for each event are convolved over time with the filter.

Joint phase segmentation and classification

In frame-wise prediction, the class for each time step is $y_t = \arg\max_y s_t^y$ where y_t is the best scoring phase. While frame-wise accuracy is reasonable, some actions get oversegmented due to high variance in the data. We use a segmental inference method based on the semi-Markov conditional random fields to prevent this issue [22].

Let tuple $p_j = (y_j, t_j, d_j)$ be the jth action segment where y_j is the action label, t_j is the start interval, and d_j is the segment duration. There is a sequence of M segments $P = \{p_1, p_2, \ldots, p_M\}$ for $0 < M \leq T$ such that the start of segment j coincides with the end of the previous segment $t_j = t_{j-1} + d_{j-1}$ and the durations add up to the total number of intervals $\sum_{i=1}^{M} d_i = T$.

Given scores $\mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n)$, we find the segments P that maximize the cost $E(\mathbf{S}, P)$ of the whole sequence:

$$E(\mathbf{S}, P) = \sum_{j=1}^{m} g(\mathbf{S}, y_j, t_j, d_j) \tag{1}$$

The segment function $g(\cdot)$ is defined as a sum of the scores within that segment with the constraint that segment j and segment $j+1$ do not belong to the same phase:

$$g(S, y_j, t_j, d_j) = \begin{cases} \sum_{t=t_j}^{t_j+d_j-1} s_t^{y_j}, & \text{if } y_j \neq y_{j-1} \\ -\infty, & \text{otherwise} \end{cases} \tag{2}$$

This model can be viewed in the probabilistic setting as a conditional random field using $Pr(P|S) \propto \exp(-E(S, P))$.

We solve the following discrete constrained optimization problem to find all phases, their start times, and durations:

$$\begin{aligned} P = & \underset{P=\{p_1,\ldots,p_m\}}{\arg\max} \; E(\mathbf{S}, P) \\ \text{s.t.} & \sum_{i=1}^{m} d_i = T \quad \text{and} \quad 0 < m \leq T \end{aligned} \tag{3}$$

In the naive case, this problem has computational complexity $O(T^2C^2)$. We use the method proposed in [21] that is of the order $O(KTC^2)$ where K is an upper bound on the number of segments. K is typically much smaller than T.

Experiments

Hysterectomy data set

We collected data from a da Vinci surgical robot for robot-assisted hysterectomy (RAH) procedures during an ongoing institutional review board (IRB)-approved study [23]. We interfaced with the robot using the da Vinci research API [24] to collect time synchronized (1) endoscopic video, (2) tool motion data, and (3) system (console) events. The data set consists of 24 full RAH surgeries. This excludes those recordings that had missing video or system event data.

Hysterectomies are highly variable in duration and phase flow. This is unlike procedures like cholecystectomies which have been studied in many previous phase detection papers. Our data set contains surgeries that range from 47 min to 3 h and 47 min in length and contain between 8 and 18 phase instances. Six faculty surgeons performed the procedures with the assistance of more than 20 surgical residents. At least two surgeons participated in each procedure.

Phase labels

A set of surgical phases were defined after consulting with our collaborating gynecologist. These phases are listed in Table 2. Our event-based features cannot distinguish between anatomical structures so similar phases were grouped into a higher-level labels. In addition to the four surgical phase labels from Table 2, remaining portions of the surgery were labeled a background class named *No Label*. In total, our system classifies five phase labels: ligation, dissection, colpotomy, cuff closure, and no label.

Table 2 Phases during a robot-assisted hysterectomy procedure along with their duration distribution across the 24 surgeries (VCC: vaginal cuff closure)

Original phase label	Derived label	Prior
Ligation of left/right IP ligament Ligation of left/right round ligament Ligation of left/right utero-ovarian ligament	Ligation	0.067
Isolation of uterus Dissection of auxiliary structures	Dissection	0.460
Colpotomy (cutting the cervix)	Colpotomy	0.061
VCC using interrupted suturing VCC using V-lock suture VCC using running suturing VCC using figure-of-eight suturing	Cuff closure	0.161
Background	No label	0.251

A vocabulary consisting of the start point, end point, and description for each phase was created in consultation with an expert surgeon. A single individual (without a medical background) followed these instructions and labeled each procedure by manually annotating the start, stop, and phase type of each such instance. Another individual independently verified these phase labels.

Feature extraction

In total, the 24 RAH procedures contain approximately 50 h of data. Features are aggregated in overlapping intervals of 90 s resulting in 5781 intervals across all surgeries. In the discussion, we show sensitivity analysis on interval lengths from 60 to 180 s. Note it is possible for a single interval to contain more than one distinct phase label. As such, the label that is true for the longest is chosen as that interval's ground truth phase label.

In principle, we could compute a feature for every time step; however, the data tend to stay constant over long periods of time. As such, we only compute features every 30 s. This makes training our models much more reasonable. We explore different rates in the discussion.

Modeling tools implementation

All data were normalized using zero-mean and unit-variance scaling using statistics from the training data. Cross-validation was performed to find the hyperparameters in each model. The random forest uses 100 trees using out-of-bag estimation error over the range of $N = [10, 500]$. The minimum number of leaf nodes in each tree is set to 5. The temporal CNN was implemented using Keras,[1] an efficient library for developing deep learning models. We set the filter duration to be 20 intervals based on cross-validation. For segmental inference, we set the upper bound on the number of phases in a procedure sample to be 15.

Metrics

Results are evaluated using overall accuracy, per-class precision/recall, and a segmental Levenshtein distance. Accuracy, precision, and recall are computed using their standard formulae. The Levenshtein distance metric (LD) [25] emphasizes the difference in errors like false-positives between frame-wise and segmental inference. It computes the difference between two string sequences by computing the minimum number of edits (insertions, deletions and substitutions) that need to be performed to change one sequence into the other. Each set of predictions is split into its constituent segments. For example, "AAABBCCCC" becomes "ABC." The number of segments in each prediction and ground truth labeling may vary; thus, LD is normalized by the maximum number of segments in each prediction and ground truth labeling. Note smaller values for LD indicate better performance.

[1] Keras: Deep Learning library: http://keras.io.

Skewed phase distribution

Some surgical phases are much longer in duration than others. Table 2 shows the ground truth phase distribution is highly skewed toward *Dissection* and *No Label* class. To account for this, we subsampled the training data for the SVM and RF classifiers to create a balanced training data set. We created 100 iterations for training set in each of the validation folds. The final score $\mathbf{s}_t$ for a test sample was the average of the score over the 100 iterations. However, as the test set was expected to be skewed, the training data class distribution was set as the class weight for the SVM and RF models.

The most important phase labels from a surgical standpoint—*Ligation* and *Colpotomy*—are sometimes very short in duration. Using a step size of 60 s, most instances of these phases are contained by a single time step. In the discussion, we show performance using different sampling periods (10, 30, 45, 60 s).

Sensitivity analyses: interval length and feature set

In addition to the validation of the three models using the metrics listed above, we performed two sets of experiments to analyze the effect on phase prediction performance of our framework:

Interval length This is the time period over which the signals are aggregated. For an interval length of 120 s, if the bipolar energy tool was activated 10 times during the period $(t, t + 120)$, then its count feature at time t would be 10. We evaluated performance for interval lengths ranging from 60 to 180 s in increments of 30 s.

Feature set Although our data were recorded using a da Vinci system, a subset of the features, like those derived from energy activations and tool identification, can be captured easily and at a low cost using button sensors and RFID tags. These signals are generic across laparoscopic, endoscopic, and open surgical procedures. We evaluated our framework's prediction performance using a nine-dimensional subset vector (*EtECtTi*) containing three time-based energy features, three count-based energy features, and three tool information flags.

Results

Performance is computed using leave-one-surgery-out cross-validation over all 24 trials. We address several questions: (1)

Table 3 Phase prediction accuracy for various step sizes

Method	Time steps (s)			
	10	30	45	60
SVM	66.8	**67.1**	66.0	64.4
RF	71.5	71.7	**71.9**	70.4
tCNN	73.4	**74.3**	71.7	69.1
SVM(*Seg*)	70.2	**70.4**	70.1	67.1
RF(*Seg*)	**74.4**	74.3	**74.4**	72.0
tCNN(*Seg*)	74.5	**76.0**	73.6	70.3

Bold refers to time step with highest accuracy for each method
(Seg) refers to segmental inference-based phase predictions

What is the overall accuracy and precision/recall for each surgical phase? (2) What is the impact of segmental inference? (3) How do the interval length and time between intervals impact accuracy? and (4) Do signals specific to the da Vinci enhance performance versus signals available and generic to most other forms of surgery?

Overall frame-wise prediction accuracy is displayed in Table 3. Results using frame-wise inference are listed on top and using segmental inference are on bottom. In general, RF and tCNN perform better than SVM; however, these differences are only 4–5 %. Accuracy of the segmental predictions is higher than the corresponding frame-wise predictions by about 3 %. The phase label predictions from the three approaches along with the ground truth phase sequence from one of the data set procedures are shown in Fig. 2. Additionally, the feature importance based on mean-squared error at each node from RF showed that all the features were similar in importance.

Table 3 also shows that there is a minor increase in accuracy as the step size decreases from 60 to 10 s. The results stabilize around 30 s. This may be because phases with short duration, such as *Ligation*, yield a small number of samples. The improvement is largest for the temporal CNN which models how the features change over time.

Tables 4 and 5 show per-class precision and recall. Precision is higher for *Dissection* and *Cuff Closure*, moderate for *Colpotomy* and *No Label*, and low for *Ligation*. Segmental inference tends to improve precision in all except three cases (marked with a *). *Cuff Closure* phase has near perfect recall and *Dissection* has recall of 85 %. Recall for *Ligation* was poor in most cases.

Table 6 compares performance using the LD metric. The results are similar to observations in the overall accuracy. RF and tCNN perform similarly and are both better than SVM. The segmental inference performance across the three approaches improves the LD metric as well. As the step size decreases, the LD performance tends to decrease.

Table 7 shows effect on accuracy in phase prediction as part of the first sensitivity analysis (section "Sensitivity analyses: interval length and feature set") using features computed with interval lengths varying from 60 to 180 s. The performance is similar among all values; however, results at 60 s are marginally worse. This matches our intuition to choose 90-s intervals for the main results based on the typical phase lengths for hysterectomy procedures.

Table 8 compares results using all signals recorded by the da Vinci versus the subset *EtECtTi* of signals common to most surgical systems (section "Sensitivity analyses: interval length and feature set"). Our results show the performance using these generic features is only a small amount worse than using all features.

Discussion and future work

Our data set is highly realistic and contains natural variations in procedure flow pertaining to patient anatomy, type of hysterectomy (total, radical, subtotal), and surgeon style. Despite these challenges, the performance of our framework

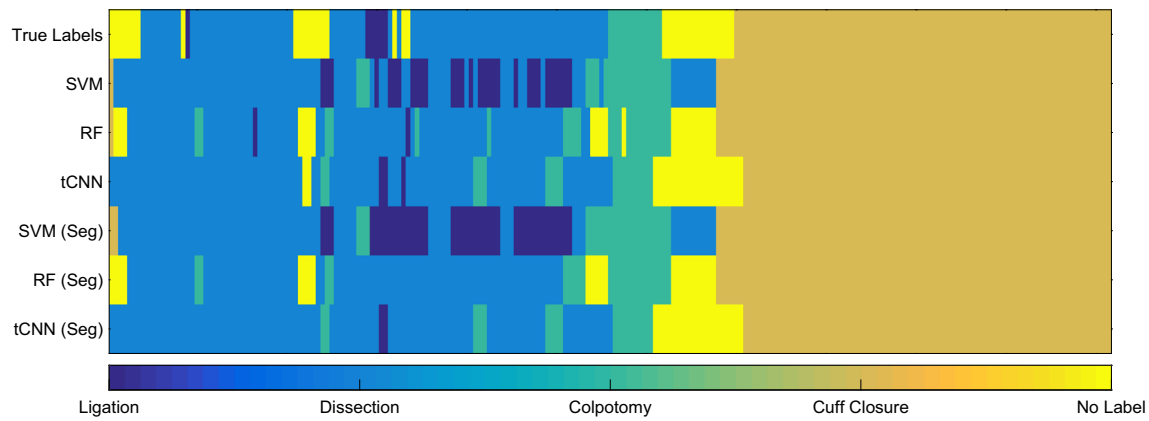

Fig. 2 Phase prediction for a hysterectomy procedure from our data set using system events-based features. *(Seg)* refers to segmental inference-based predictions

Table 4 Per-phase precision with a 30 s step size

Phase	SVM	RF	tCNN	SVM (*Seg*)	RF (*Seg*)	tCNN (*Seg*)
Ligation	24.1	36.0	37.7	14.3*	**44.6**	40.9
Dissection	72.7	73.7	78.2	75.0	72.9*	**78.3**
Colpotomy	38.9	60.1	57.7	41.3	63.8	**69.1**
Cuff closure	80.8	83.0	85.3	80.6*	83.1	**85.4**
No Label	55.8	62.6	68.8	61.6	**74.1**	70.6

Bold text refers to the highest precision values
(Seg) refers to segmental inference-based phase predictions
* Segmental inference lowered the precision value

Table 5 Per-phase recall with a 30 s step size

Phase	SVM	RF	tCNN	SVM (*Seg*)	RF (*Seg*)	tCNN (*Seg*)
Ligation	14.9	15.9	27.5	3.6	9.5	**24.4**
Dissection	78.2	84.0	82.2	82.8	**91.3**	85.6
Colpotomy	39.5	37.3	55.9	44.4	32.7	**60.2**
Cuff closure	97.0	98.0	95.3	**98.8**	**98.8**	95.0
No Label	44.6	52.4	61.4	50.2	50.7	**61.4**

Bold text refers to the highest recall values
(Seg) refers to segmental inference-based phase predictions

Table 6 Overall Levenshtein distance in phase prediction for the different time steps

Method	Time steps (s)			
	10	30	45	60
SVM	32.1	32.0	33.0	**33.8**
RF	27.2	27.0	26.6	**27.7**
tCNN	26.2	25.0	27.1	**29.0**
SVM(*Seg*)	29.9	30.1	29.9	**31.9**
RF(*Seg*)	24.8	25.0	25.3	**27.0**
tCNN(*Seg*)	25.2	23.9	26.2	**28.4**

Bold refers to the highest edit distance values for each method
(*Seg*) refers to segmental inference-based phase predictions. Smaller values for LD indicate better performance

Table 7 Phase prediction accuracy using different interval lengths for aggregating the features

Method	Interval length (s)				
	60	90	120	150	180
SVM (*Seg*)	69.8	70.4	70.2	70.7	70.6
RF (*Seg*)	72.4	74.3	74.5	74.1	74.6
tCNN (*Seg*)	76.0	76.0	77.0	76.3	76.1

Time step size was 30 s using the 16-dimensional feature set

was comparable to the overall accuracy of other reported results [7,8]. Precision and recall across phases are similar to those reported in [7]. That work also finds precision and recall of the dominant class tends to be much higher than other classes.

Table 8 Phase prediction accuracy using signals specific to the da Vinci (*all*) versus signals generic to many surgical systems (*EtECtTi*)

Feature set	SVM(*Seg*)	RF(*Seg*)	tCNN(*Seg*)
all	70.4	74.3	76.0
EtECtTi	61.6	71.0	72.5

EtECtTi is a nine-dimensional vector containing the three time-based energy features, three count-based energy features, and three tool information flags

Despite investigating several models with various distinct assumptions, we found all approaches achieved relatively similar performance. The first (SVM) assumed a simple linear model, the second (random forest) learned the most important subsets of features for each phase, and the third (temporal CNN) non-linearly modeled the temporal evolution of features. Based on these results and our experience working with these data, we surmise the biggest issue is not with the activity recognition models but with the way the problem is posed. The extreme temporal variability has a large negative impact on prediction. Some of the phases are many times longer than others. This results in many short phases being merged into neighboring larger ones. This was an issue with the tCNN because temporal filters tended to smooth out feature responses across short phases. It was especially apparent when using segmental inference.

The presented framework and its validation were based on events data captured from a robot-assisted surgery platform. However, we performed the same validation experiments by leaving out some of the robot-specific events such as camera motion, clutching, and the console head sensor. This analy-

sis showed that the performance of the different models in predicting the phase label did not decrease by a large amount using the smaller set of features generic to other forms of surgery (Table 8). Thus, our method can be applied and tested with non-robotic surgical systems. The previous work [7] has successfully captured these signals in the laparoscopic cholecystectomy procedure setting. This would enable large-scale studies that require surgical phase analysis in the domain of traditional laparoscopic as well as open surgery, in addition to robot-assisted procedures.

Information for surgical phase detection is distributed across different forms of data video, tool motion and system events. Each data type has its own advantages and disadvantages. While video contains the most context it is challenging to detect the action being performed, anatomy being operated upon, and the instruments in use. Tool motion data capture a surgeon's direct movements but lack contextual information such as what anatomy the surgeon is operating on. Events signals such as button presses and releases are the simplest and cheapest to acquire but do not capture anatomy or nuance in a surgeon's motions. Our presented work supports the hypothesis that phase information is contained in the system event signals. This information is not available through tool motion data and hard to extract from video data. Thus, future work should look at combining multiple modalities to capture complementary information about surgical phases.

There are many questions that require further investigation. For example, can our proposed approach apply to other surgical procedure data? How does workflow vary between different surgeons? Do certain workflows correlate with improved outcomes? How do patient anatomy or prior conditions affect the workflow? While this work highlights some of the tools necessary for addressing these questions, our analysis is limited by the size of our data set. To answer these questions, we must scale up the data set so there are a sufficient number of samples belonging to different sets of parameters like operating surgeon, patient's anatomy, for statistically significant analysis and results. Future research must consider this when generating new data sets.

Conclusion

Surgical phase detection, at scale, has many useful applications for surgical education, training, and assessment. Analysis of surgical phases and their impact on patient outcomes can provide important insights about critical steps in a surgery. We have presented a scalable solution for phase detection using system events captured during live surgical procedures. Our findings demonstrate that system events contain surgical phase information, and thus, they may be combined with tool motion and/or video data to automate surgical phase recognition with a better performance.

Acknowledgments We would like to thank Intuitive Surgical Inc. for providing us the da Vinci research API that enabled the data collection from the hysterectomy procedures. The user study to collect these data operated smoothly, thanks to the Johns Hopkins clinical engineering staff, IRB committee members, and the operating room nursing staff. A significant portion of data preprocessing was performed by S. Arora and her contribution. We would also like to acknowledge S. S. Vedula, N. Ahmidi, J. Jones, Y. Gao, and S. Khudanpur for their useful feedback.

Compliance with ethical standards

Funding Anand Malpani is currently funded through the Link Foundation-Modeling, Simulation and Training Fellowship; Colin Lea is funded through an Intuitive Surgical Technology Research Grant; the user study for collecting the original data set was supported through internal JHU funds.

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki Declaration and its later amendments or comparable ethical standards.

Informed consent Informed consent was obtained from all individual participants included in the study.

References

1. Birkmeyer JD, Finks JF, O'Reilly A, Oerline M, Carlin AM, Nunn AR, Dimick J, Banerjee M, Birkmeyer NJ (2013) Surgical skill and complication rates after bariatric surgery. N Engl J Med 369(15):1434–1442. doi:10.1056/NEJMsa1300625
2. Lalys F, Jannin P (2014) Surgical process modelling: a review. Int J Comput Assist Radiol Surg 9(3):495–511. doi:10.1007/s11548-013-0940-5
3. Ahmadi SA, Sielhorst T, Stauder R, Horn M, Feussner H, Navab N (2006) Recovery of surgical workflow without explicit models. In: Larsen R, Nielsen M, Sporring J (eds) Medical image computing and computer-assisted intervention—MICCAI 2006. Lecture notes in computer science, vol 4190. Springer, Berlin, Heidelberg, pp 420–428
4. Padoy N, Blum T, Essa I, Feussner H, Berger MO, Navab N (2007) A boosted segmentation method for surgical workflow analysis. In: Ayache N, Ourselin S, Maeder A (eds) Medical image computing and computer-assisted intervention—MICCAI 2007. Lecture notes in computer science, vol 4791. Springer, Berlin, Heidelberg, pp 102–109
5. Blum T, Feussner H, Navab N (2010) Modeling and segmentation of surgical workflow from laparoscopic video. In: Jiang T, Navab N, Pluim JPW, Viergever MA (eds) Medical image computing and computer-assisted intervention—MICCAI 2010. Lecture notes in computer science, vol 6363. Springer, Berlin, Heidelberg, pp 400–407
6. Padoy N, Blum T, Ahmadi SA, Feussner H, Berger MO, Navab N (2012) Statistical modeling and recognition of surgical workflow. Med Image Anal 16(3):632–641. doi:10.1016/j.media.2010.10.001
7. 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: Stoyanov D, Collins DL, Sakuma

I, Abolmaesumi P, Jannin P (eds) Information processing in computer-assisted interventions, no. 8498 in lecture notes in computer science, Springer International Publishing, pp 148–157
8. DiPietro R, Stauder R, Kayis E, Schneider A, Kranzfelder M, Feussner H, Hager GD, Navab N (2015) Automated surgical-phase recognition using rapidly-deployable sensors. In: Modeling and monitoring of computer assisted interventions (M2CAI)
9. Neumuth T, Straub G, Meixensberger J, Lemke HU, Burgert O (2006) Acquisition of process descriptions from surgical interventions. In: Bressan S, Kung J, Wagner R (eds) Database and expert systems applications, no. 4080 in lecture notes in computer science, Springer, Berlin, pp 602–611. doi:10.1007/11827405_59
10. Forestier G, Riffaud L, Jannin P (2015) Automatic phase prediction from low-level surgical activities. Int J Comput Assist Radiol Surg 10(6):833–841. doi:10.1007/s11548-015-1195-0
11. Katic D, Wekerle AL, Gartner F, Kenngott H, Muller-Stich BP, Dillmann R, Speidel S (2014) Knowledge-driven formalization of laparoscopic surgeries for rule-based intraoperative context-aware assistance. In: Stoyanov D, Collins DL, Sakuma I, Abolmaesumi P, Jannin P (eds) Information processing in computer-assisted intervention. Lecture notes in computer science, vol 8498. Springer, Switzerland, pp 158–167
12. Twinanda AP, Marescaux J, Mathelin Md, Padoy N (2015) Classification approach for automatic laparoscopic video database organization. Int J Comput Assist Radiol Surg. doi:10.1007/s11548-015-1183-4
13. Rosen J, Brown J, Chang L, Sinanan M, 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. doi:10.1109/TBME.2005.869771
14. Reiley CE, Lin HC, Varadarajan B, Vagvolgyi B, Khudanpur S, Yuh DD, Hager GD (2008) Automatic recognition of surgical motions using statistical modeling for capturing variability. Stud Health Technol Inform 132:396–401
15. Varadarajan B (2011) Learning and inference algorithms for dynamical system models of dextrous motion, Dissertation, The Johns Hopkins University
16. Haro BB, Zappella L, Vidal R (2012) Surgical gesture classification from video data. In: Ayache N, Delingette H, Golland P, Mori K (eds) Medical image computing and computer-assisted intervention—MICCAI 2012. Springer, Berlin, pp 34–41
17. Zappella L, Bejar B, Hager G, Vidal R (2013) Surgical gesture classification from video and kinematic data. Med Image Anal 17(7):732–745. doi:10.1016/j.media.2013.04.007
18. Tao L, Zappella L, Hager GD, Vidal R (2013) Surgical gesture segmentation and recognition. In: Mori K, Sakuma I, Sato Y, Barillot C, Navab N (eds) Medical image computing and computer-assisted intervention—MICCAI 2013. Lecture notes in computer science, vol 8151. Springer, Berlin, pp 339–346
19. Lea C, Vidal R, Hager GD (2016) Learning convolutional action primitives for fine-grained action recognition. IEEE international conference on robotics and automation, Stockholm (accepted)
20. Breiman L (2001) Random forests. Mach Learn 46(1):5–32. doi:10.1023/A:1010933404324
21. Lea C, Reiter A, Vidal R, Hager GD (2016) Efficient segmental inference for spatiotemporal modeling of fine-grained actions. arXiv:1602.02995 [cs]
22. Sarawagi S, Cohen WW (2005) Semi-Markov conditional random fields for information extraction. In: Advances in neural information processing systems 17. MIT Press, Cambridge, pp 1185–1192. http://papers.nips.cc/paper/2648-semi-markov-conditional-random-fields-for-information-extraction.pdf
23. Chen CCG, Tanner E, Malpani A, Vedula SS, Fader A, Scheib S, Hager GD (2015) Warm-up before robotic hysterectomy does not improve trainee operative performance: a randomized trial. In: American urogynecologic society annual meeting, pp 396–401
24. DiMaio SP, Hasser C (2008), The da Vinci research interface, http://www.midasjournal.org/browse/publication/622
25. Navarro G (2001) A guided tour to approximate string matching. ACM Comput Surv 33(1):31–88. doi:10.1145/375360.375365