2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Temporal Segmentation of Surgical Sub-tasks through Deep Learning with Multiple Data Sources

Yidan Qin[1,2], Sahba Aghajani Pedram[1,3], Seyedshams Feyzabadi[1], Max Allan[1], A. Jonathan McLeod[1], Joel W. Burdick[2], Mahdi Azizian[1]

***Abstract*— Many tasks in robot-assisted surgeries (RAS) can be represented by finite-state machines (FSMs), where each state represents either an action (such as picking up a needle) or an observation (such as bleeding). A crucial step towards the automation of such surgical tasks is the temporal perception of the current surgical scene, which requires a real-time estimation of the states in the FSMs. The objective of this work is to estimate the current state of the surgical task based on the actions performed or events occurred as the task progresses. We propose Fusion-KVE, a unified surgical state estimation model that incorporates multiple data sources including the Kinematics, Vision, and system Events. Additionally, we examine the strengths and weaknesses of different state estimation models in segmenting states with different representative features or levels of granularity. We evaluate our model on the JHU-ISI Gesture and Skill Assessment Working Set (JIGSAWS), as well as a more complex dataset involving robotic intra-operative ultrasound (RIOUS) imaging, created using the da Vinci® Xi surgical system. Our model achieves a superior frame-wise state estimation accuracy up to 89.4%, which improves the state-of-the-art surgical state estimation models in both JIGSAWS suturing dataset and our RIOUS dataset.**

I. INTRODUCTION

In the field of surgical robotics research, the development of autonomous and semi-autonomous robotic surgical systems is among the most popular emerging topics [1]. Such systems allow RAS to go beyond teleoperation and assist the surgeons in many ways, including autonomous procedures, user interface (UI) integration, and providing advisory information [2], [3]. One prerequisite for these applications is the perception of the current state of the surgical task being performed. These states include the actions performed or the changes in the environment observed by the system. For instance, during suturing, the system needs to know if the needle is visible from the endoscopic view before providing more advanced applications such as advising the needle position or autonomous suturing. Additionally, the recognition of higher-level surgical states, or surgical phases, has a wide range of applications in post-operative analysis and surgical skill evaluation [4].

The recognition and segmentation of the robot's current action is one of the main pillars of the surgical state estimation process. Many models have been developed for the segmentation and recognition of fine-grained surgical actions that last for a few seconds, such as cutting [5]–[8], as well as surgical phases that last for up to 10 minutes, such as bladder dissection [9]–[11]. The recognition of fine-grained surgical states is particularly challenging due to their short duration and frequent state transitions. Most work in this field has focused on developing models using only one type of input data, such as kinematics or vision. Some studies have focused on learning based on robot kinematics, using models such as Hidden Markov Models [12]–[14] and Conditional Random Fields (CRF) [15]. Zappella et al. proposed methods of modeling surgical video clips for single-action classification [16]. The Transition State Clustering (TSC) and Gaussian Mixture Model methods provide unsupervised or weakly-supervised methods for surgical trajectory segmentation [17], [18]. More recently, deep learning methods have come to define the state-of-the-art, such as Temporal Convolutional Networks (TCN) [19], Time Delay Neural Network (TDNN) [7], and Long-Short Term Memory (LSTM) [6], [20]. Instead of using robot kinematics data, vision-based methods have been developed based on Convolutional Neural Networks (CNN). Vision-based models in RAS use the vision data that is readily available from the endoscopic view. Concatenating spatial features on the temporal axis with spatio-temporal CNNs (ST-CNN) has been explored in [21]. Jin et al. introduced the post-processing of predictions using prior knowledge inference [22]. TCN can also be applied to vision data for action segmentation, taking the encoding of a spatial CNN as input [19]. Ding et al. proposed a hybrid TCN-BiLSTM network [23]. The limitation shared by single-input action recognition models is the large discrepancy among states' representative vision and kinematics features, making them distinguishable through different types of input data.

Comparing to action recognition datasets such as ActivityNet [24], RAS data enjoys the luxury of having syn-

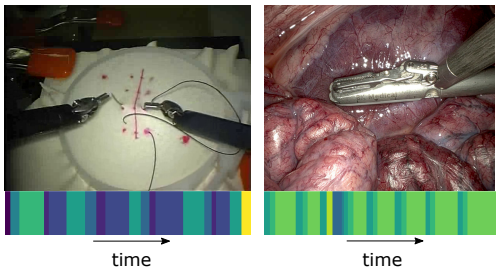

Fig. 1. Sample data from JIGSAWS (left) and RIOUS dataset (right). The bottom row shows a sample state sequence of each task, where each color denotes a state label.

[1]Intuitive Surgical Inc., 1020 Kifer Road, Sunnyvale, CA,94086, USA
[2]Department of Mechanical and Civil Engineering, California Institute of Technology, 1200 E California Blvd, Pasadena, CA, 91125, USA
[3]Department of Mechanical and Aerospace Engineering, University of California, Los Angeles, Los Angeles, CA, 90095, USA
Emails: Ida.Qin@intusurg.com, Mahdi.Azizian@intusurg.com

chronized vision, system events, and robot kinematics data. The attempts of incorporating multiple types of input data have been focusing on using derived values as additional variables to a single model. Lea et al. measured two scene-based features in JIGSAWS as additional variables to the robot kinematics data in their Latent Convolutional Skip-Chain CRF (LC-SC-CRF) model [5]. Zia et al. collected the robot kinematics and system events data from RAS to perform surgical phase recognition [10]. While these attempts have proven to improve the model accuracy, to the best of our knowledge, there is yet to be a unified method that incorporates multiple data sources directly for fine-grained surgical state estimation.

In addition to robot actions, the finite state machine (FSM) of a surgical task should also include the environmental changes observed by the robot. The non-action states were omitted in popular surgical action segmentation datasets such as JIGSAWS [25] and Cholec80 [26]; however are important for applications such as autonomous procedures. They are also challenging to recognize as some non-action states may not be well-reflected in a single-source dataset.

Contributions: In this paper, we propose a unified approach of fine-grained state estimation in RAS using multiple types of input data collected from the da Vinci® surgical system. The input data we use includes the endoscopic video, robot kinematics, and the system events of the surgical system. Our goal is to achieve the real-time fine-grained state estimation of the surgical task being performed. To re-emphasize, we refer to fine-grained states as states that last in the scale of seconds. Our main contributions include:

- Implement a unified state estimation model that incorporates vision-, kinematics-, and event-based state estimation results;
- Improve the frame-wise state estimation accuracy of state-of-the-art methods by up to 11% through the incorporation of multiple sources of data;
- Demonstrate the advantages of a multi-input state estimation model through the comparison of single-input models' performances in recognizing states with different representative features or levels of granularity in a complex and realistic surgical task.

We evaluated the performance of our model using JIGSAWS and a new RIOUS (robotic intra-operative ultrasound) dataset we developed. RIOUS consists of phantom and porcine experiments on a da Vinci® Xi surgical system (Fig. 1). Comparing to JIGSAWS, which is relatively simple as it only contains dry-lab tasks with no camera motion nor non-action annotations, RIOUS dataset better resembles real-world surgical tasks. This is because RIOUS dataset contains dry-lab, cadaveric and in-vivo experiments[1], as well as camera movements and annotations of both action and non-action states. We evaluated the accuracy of multiple state estimation models in the recognition of states with different representative features. Each model has its respective strengths and weaknesses, which supports the superior performance of our unified approach of state estimation.

[1]All in-vivo experiments were performed on porcine models under Institutional Animal Care and Use Committee (IACUC) approved protocol.

II. METHOD

Our proposed model (Fig. 2) consists of four single-source state estimation models based on vision, kinematics, and system events, respectively. The outputs are fed to a fusion model that makes a comprehensive inference. In this section, we discuss each individual model as well as the fusion model which effectively combines the outputs of each model.

A. Vision-based Method

The vision-based state estimation model is a CNN-TCN model [19] that takes the endoscopic camera stream as the input in the form of a series of video frames. The CNN architecture we deploy is VGG16 [27]. The spatial CNN component serves as a feature extractor and maps each $224 \times 224 \times 3$ RGB image to a vector $X^{vis} \in \mathbb{R}^N$ where N is the number of features. X^{vis} is then fed to the TCN component, which is an encoder-decoder network (Fig. 3). At time step t, the input vector is denoted by X_t^{vis} for $0 < t \leqslant T$. For the l^{th} 1-D convolutional layers ($l \in \{1, ..., L\}$), F_l filters of kernel size k are applied along the temporal axis that capture the temporal progress of the input data. T_l is the number of time steps in the l^{th} layer. In each layer, the filters are parameterized by a weight tensor $W^{(l)} \in \mathbb{R}^{F_l \times k \times F_{l-1}}$ and a bias vector $b^{(l)} \in \mathbb{R}^{F_l}$. The raw output activation vector for the l^{th} layer at time t, $E_t^{(l)}$, is calculated from a subsection of the normalized activation matrix from the previous layer $\hat{E}^{(l-1)} \in \mathbb{R}^{F_{l-1} \times T_{l-1}}$

$$E_t^{(l)} = f(W^{(l)} * \hat{E}_{t:t+k-1}^{(l-1)} + b_i^{(l)}) \tag{1}$$

where f is a Rectified Linear Unit (ReLU) [28]. A max pooling layer of stride 2 is applied after each convolutional layer in the encoder part such that $T_l = \frac{T_{l-1}}{2}$. The pooling layer is followed by a normalization layer, which normalizes the l^{th} activation vector at time t, $E_t^{(l)}$, using its highest value

$$\hat{E}_t^{(l)} = \frac{E_t^{(l)}}{max(E_t^{(l)}) + \epsilon} \tag{2}$$

where $\epsilon = 10^{-5}$ is a small number to ensure non-zero denominators, and $\hat{E}_t^{(l)}$ is the normalized output activation vector. In the decoder part, an upsampling layer that repeats each data point twice proceeds each temporal convolutional and normalization layers. The output vector $\hat{D}_t^{(l)}$ is calculated and normalized in the same manner as the encoder part. The state estimation at frame t is done by a time-distributed fully-connected layer with softmax to normalize the logits.

Implementation details: The training of the CNN feature extractor starts with the VGG16 network initialized with ImageNet pre-trained weights. We fine-tune the weights by training with one fully-connected layer on top of the VGG16 model for state estimation. The feature vector $X_t^{vis} \in \mathbb{R}^{N=1024}$. We use $L = 3$ with $F_l = \{32, 64, 96\}$, and $k = 6.1s$ for the JIGSAWS suturing dataset and $k = 3.4s$ for

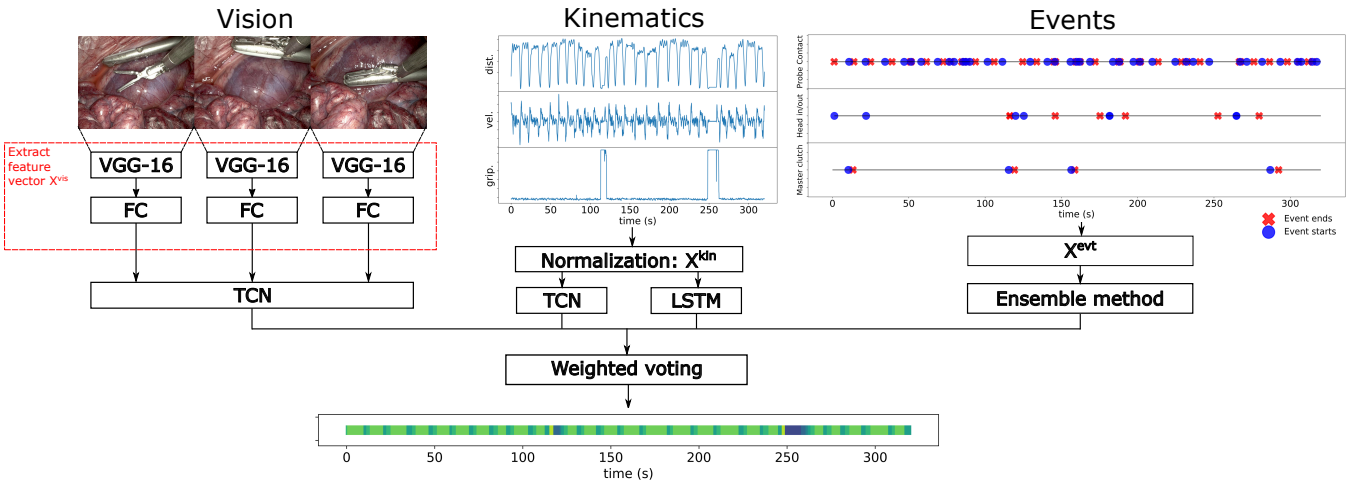

Fig. 2. Our model contains four single-input state estimation models receiving three types of input data (X: vision features extracted from). A fusion model that receives individual model outputs is used to make the comprehensive state estimation result.

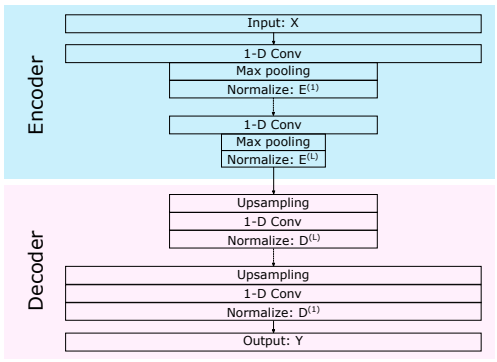

Fig. 3. The encoder-decoder TCN network that hierarchically models vision or kinematics data to states.

the RIOUS dataset. For training, we use the cross entropy loss with Adam optimization algorithm [29].

For our application of real-time state estimation, the model can only use the information from the current and preceding time steps; therefore for the RIOUS dataset, we assume a causal setting and pad the temporal input with $\frac{k}{2}$ zeros on the left side before the convolutional layer and crop $\frac{k}{2}$ data points on the right side afterwards.

B. Kinematics-based Methods

We incorporate both forward LSTM and TCN to better capture states with different duration. LSTM has no constraints on learning only from the nearby data on the temporal axis. Rather, it maintains a memory cell and learns when to read/write/reset the memory [30]. It has been shown that LSTM-based approaches exceed the state-of-the-art performance in longer-duration action recognition [6]. We incorporate both TCN, which applies temporal convolution to learn local temporal dependencies, and LSTM, which is able to capture longer-term data progress. Although the bi-directional LSTM model yields a higher accuracy [6], it is not applicable for the real-time state estimation task where no future data is available; therefore we use a forward LSTM with forget gates and peephole connections [30]. The loss function for the LSTM model is the cross entropy between the ground truth and the predicted labels, and the stochastic gradient descent (SGD) is used to minimize loss.

Implementation details: For the LSTM model, we perform a grid search over the initial learning rate (0.5 or 1.0), the number of hidden layers (1 or 2) in the model, the number of hidden units per layer (256, 512, 1024, or 2048), and the dropout probability (0 or 0.5). The optimized set of parameters is 1 hidden layers with 1024 hidden units and 0.5 dropout probability for JIGSAWS, and 512 hidden units for the RIOUS dataset. The optimized initial learning rate is 1.0. For the TCN model, we mostly follow the same protocol of the vision-based TCN model described earlier. We use $L = 2$ with $F_l = \{64, 96\}$. The feature vector for the kinematics data $X^{kin} \in \mathbb{R}^N$, where $N = 26$ for the JIGSAWS suturing dataset and $N = 19$ for the RIOUS dataset.

C. Event-based Method

At each timestamp t, The da Vinci® Xi surgical system registers multiple binary system events (details in section IIIA). We experimented with various classification algorithms, including Adaboost classifier, decision tree, Random Forest (RF), Ridge classifier, Support Vector Machine (SVM), and SGD [31]. The classification was performed directly for state estimation using the set of system events collected at each timestamp X_t^{evt} as features. We performed grid search over the parameters of each model and evaluated each model's performance using the Area Under the Receiver Operating Characteristic Curve (ROC AUC) score [32]. The evaluation process was iterated 200 times, with an early stopping criterion of score improvement under 10^{-6}. At each iteration, we recorded the best-performing model with replacement. The top three models that were selected most frequently are included, and the final state estimation result is the mean of each model's prediction.

The three top-performing models for our RIOUS dataset are RF (n_{trees}=500, min_samples_split=2), SVM (penalty=$L2$, kernel=linear, C=2, multi_class=crammer_singer), and RF (n_{trees}=400, min_samples_split=3).

D. Fusion of Multiple Models

The individual state estimation models have their respective strengths and weaknesses, since different states have inherent features that make them easier to be recognized by one type of data than the other(s). For instance, the 'transferring needle from left to right' state in the JIGSAWS suturing dataset can be distinctly characterized by the sequential opening and closing of the left and right needle drivers which is captured by the kinematics data.

We therefore use a weighted voting method that incorporates the prediction vectors in all models. At time t, let $\mathbf{Y}^{(t)} \in \mathbb{R}^{a\times b}$, where a is the number of models and b is the total number of possible states in a dataset. Row vector $\mathbf{Y}^{(t)}_{i,\cdot}$ is the output vector of the i^{th} model at time t and $\sum_{j=1}^{b} \mathbf{Y}^{t}_{i,j} = 1$. The overall probability for the system to be in the j^{th} state at time t - according to the models - is then

$$P^{(t)}_j = \sum_{i=1}^{a} \alpha_{i,j} \mathbf{Y}^{(t)}_{i,j} \tag{3}$$

where $\alpha_{i,j}$ is the weighting factor for the i^{th} model predicting the j^{th} state. α is calculated from the diagnostic odds ratio (OR) derived from the model's accuracy in recognizing each state in the training data: $\alpha_{i,j} = \frac{TP_{i,j}\cdot TN_{i,j}}{FP_{i,j}\cdot FN_{i,j}+\epsilon}$ where the (i,j)'s components of TP, TN, FP, FN are the number of true positives, true negatives, false positives, and false negatives of the i^{th} model on recognizing the j^{th} state, respectively. $\epsilon = 10^{-5}$ is a placeholder such that the denominator is not zero. α is normalized proportionally such that $\sum_{i=1}^{a} \alpha_{i,j} = 1$. The comprehensive estimate of state at time t $S^{(t)}$ is then made by

$$S^{(t)} = \underset{j}{\operatorname{argmax}}\, P^{(t)}_j. \tag{4}$$

III. Experimental Evaluations

We used two datasets to evaluate our models: JIGSAWS and RIOUS datasets (Table I).

A. Datasets

JIGSAWS: The JIGSAWS dataset consists of three types of finely-annotated RAS tasks [25], with synchronized video and kinematics data. These tasks are performed in a benchtop setting. We used the suturing dataset, which has 39 trials recorded at 30Hz, each around 1.5 minutes and contains close to 20 action instances. There are 9 possible actions (Fig. 4a). The kinematics variables we used include the end effector positions, velocities, and gripper angles of the patient-side manipulator (PSM). The raw kinematics data uses the rotation matrix to represent the end-effector's orientation. To reduce data dimensionality, we converted the rotation matrix (9 variables) to Euler angles (3 variables).

RIOUS: To explore the full potential of our unified model, we collected a robotic intra-operative ultrasound (RIOUS) dataset on a da Vinci® Xi surgical system at Intuitive Surgical Inc. (Sunnyvale, CA), in which we performed ultrasound scanning on both phantom and porcine kidneys. In RAS, using a drop-in ultrasound probe to scan the organs is a common technique practiced by surgeons to localize underlying anatomical structures including tumors and vasculature. The real-time state estimation of this task allows us to develop smart-assist technologies for surgeons as well as enabling supervised autonomous techniques to perform such tasks.

The RIOUS dataset contains 30 trials performed by 5 users with no RAS experience but familiar with the da Vinci® surgical system. Each trial is around 5 minutes and contains roughly 80 action instances. 26 trials are performed on a phantom kidney in dry-lab setting and 4 are performed on a porcine kidney in operating room setting. The data is annotated with eight states (Fig. 4b). Two out of the four arms were used, one holding an endoscope and the other holding a pair of Prograsp™ forceps. The ultrasound machine used is the bk5000 with a robotic drop-in probe from BK Medical Holding Company, Inc. Both video and kinematics entries were synchronized and down-sampled to 30Hz. The kinematics variables we used include the instrument's end-effector positions, velocities, gripper angles, and the endoscope positions. We used the same pre-processing method as the suturing kinematics data. We also collected six system events data from the da Vinci® surgical system, including camera follow, instrument follow, surgeon head in/out of the console, master clutch for the hand controller, and two ultrasound probe events. The ultrasound probe events detect if the probe is being held by the forceps and if the probe is in contact with the tissue, respectively. All events are represented as binary on/off time series.

B. Metrics

We use two evaluation metrics for our state estimation model: the frame-wise state estimation accuracy and the edit distance. The frame-wise accuracy is the percentage of correctly recognized frames, which is measured without accounting for the temporal consistency. This is because the model has only the knowledge of the current and preceding

TABLE I

Datasets State Descriptions and Duration

JIGSAWS Suturing Dataset

Action ID	Description	Duration (s)
G1	Reaching for the needle with right hand	2.2
G2	Positioning the tip of the needle	3.4
G3	Pushing needle through the tissue	9.0
G4	Transferring needle from left to right	4.5
G5	Moving to center with needle in grip	3.0
G6	Pulling suture with left hand	4.8
G7	Orienting needle	7.7
G8	Using right hand to help tighten suture	3.1
G9	Dropping suture and moving to end points	7.3

RIOUS Dataset

State ID	Description	Duration (s)
S1	Probe released, out of endoscopic view	17.3
S2	Probe released, in endoscopic view	10.6
S3	Reaching for probe	4.1
S4	Grasping probe	1.3
S5	Lifting probe up	2.2
S6	Carrying probe to tissue surface	2.3
S7	Sweeping	8.1
S8	Releasing probe	2.5

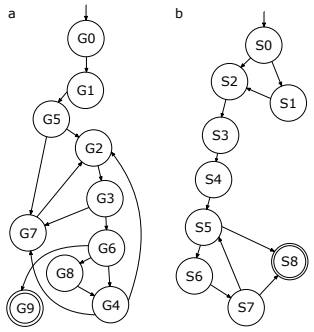

Fig. 4. FSMs of the JIGSAWS suturing task (a) and the RIOUS imaging task (b). The 0 states are the starting of tasks. The states with a double circle are the accepting (final) states. The actions in the JIGSAWS suturing task are represented with gestures (G) and the states in the RIOUS imaging task are represented with states (S).

data entries in the real-time state estimation setting. The edit distance, or Levenshtein distance [33], measures the number of insertion, deletion, and substitution needed to convert the inferred sequence of states in the segment level to the ground truth. We normalize the edit distance following [5], [6]. We evaluate both datasets using *Leave One User Out* as described in [34]. For the ultrasound imaging task, we assume a causal setting, in which the models only have knowledge of the current and preceding time steps. This is to mimic the real-time state estimation application of our model, in which the robot cannot foresee the future. For the JIGSAWS suturing task, we assume a non-causal setting for more direct comparisons with the reported accuracy of the state-of-the-art methods. The edit distance is therefore only used for JIGSAWS.

IV. Results and Discussions

Table II compares the performances of the state-of-the-art surgical state estimation models with an ablated version of our model (Fusion-KV), consisting of the kinematics- and vision-based models as well as the fusion model. Table III compares the performances of our full fusion model (Fusion-KVE) and Fusion-KV with their single-source components using the RIOUS dataset. In Fig. 5, we show an example of state estimation results of our fusion models and their components for a string of ultrasound imaging sequences. Fig. 6 shows the weight matrix α distributions used in our fusion models. A large $\alpha_{i,j}$ indicates that the i^{th} model performs well in estimating the j^{th} state during training.

In Table II, Fusion-KV achieves a frame-wise accuracy of 86.3% and edit distance score of 87.2 for the JIGSAWS suturing dataset, both improving the state-of-the-art surgical state estimation models. For the RIOUS dataset (Table III), Fusion-KVE achieves a frame-wise accuracy of 89.4%, with an improvement of 11% comparing to the best-performing single-input model. Fusion-KV also achieves a higher accuracy comparing to single-input models.

A closer observation of the inferred state sequences by various models and their weighting factors as shown in Fig. 5 and Fig. 6 reveals the key aspects of improvements of our method. Although kinematics-based state estimation models generally have a higher frame-wise accuracy comparing to vision-based models (Tables II and III), which are very sensitive to camera movements, each model has its respective strengths and weaknesses. For instance, at around 200s of the illustrated sequence in Fig. 5, both kinematics-based models show a consecutive block of errors where the models fail to recognize the 'probe released and in endoscopic view' state. Considering the relatively random robotic motions in this state, this is to be expected. The low weighting factors for both kinematics-based model in estimating this state, as shown in Fig. 6, also support this observation. On the other hand, the vision-based model correctly estimates this state, since the state is more visually distinguishable. When incorporating both vision- and kinematics-based methods, our fusion models perform weighted voting based on the training accuracy of each model. In this example, the weighting factor for the vision-based model is higher than the kinematics-based models; therefore, our fusion models are able to correctly estimate the current state of the surgical task. In other states where the robotic motions are more consistent but the vision data is less distinguishable, the kinematics-based models have higher weighting factors.

The incorporation of system events further improves the accuracy of our fusion model. Comparing Fusion-KV and Fusion-KVE, we observe fewer errors - many are corrected where α for the event-based model is high, such as states with shorter duration or frequent camera movements. At around 250s to 300s of the presented sequence, frequent state transitions can be observed. Fusion-KVE is able to estimate the states more accurately and shows fewer fluctuations comparing to other models. The event-based model is less sensitive to environmental noises, as the events are

TABLE II
Results on JIGSAWS suturing dataset

JIGSAWS Suturing

Method	Input data type	Accuracy (%)	Edit Dist.
ST-CNN [21]	Vis	74.7	66.6
TCN [19]	Kin	79.6	85.8
Forward LSTM [6]	Kin	80.5	75.3
TCN [19]	Vis	81.4	83.1
TDNN [7]	Kin	81.7	-
TricorNet [23]	Kin	82.9	86.8
Bidir. LSTM [6]	Kin	83.3	81.1
LC-SC-CRF [5]	Kin+Vis	83.5	76.8
Fusion-KV	**Kin+Vis**	**86.3**	**87.2**

TABLE III
Results on RIOUS dataset

RIOUS dataset

Method	Input data type	Accuracy (%)
ST-CNN [21]	Vis	46.3
TCN [19]	Vis	54.8
LC-SC-CRF [5]	Kin	71.5
Forward LSTM [6]	Kin	72.2
TDNN [7]	Kin	78.1
TCN [19]	Kin	78.4
Fusion-KV	Kin+Vis	82.7
Fusion-KVE	**Kin+Vis+Evt**	**89.4**

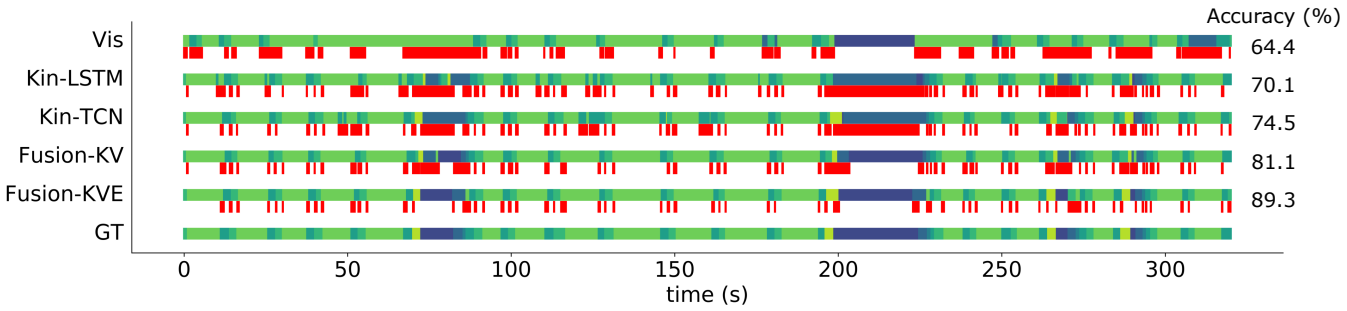

Fig. 5. Example ultrasound imaging state estimation results of the vision-based model (Vis) and the kinematics-based models (Kin-LSTM and Kin-TCN) used in our fusion models, along with Fusion-KV and Fusion-KVE, comparing to the ground truth (GT). The model used is trained with LOUO, and the example trial is performed by the unseen user. The top row of each block bar shows the state estimation results, and the frames marked in red in the bottom row are the discrepancies between the state estimation results and the ground truth.

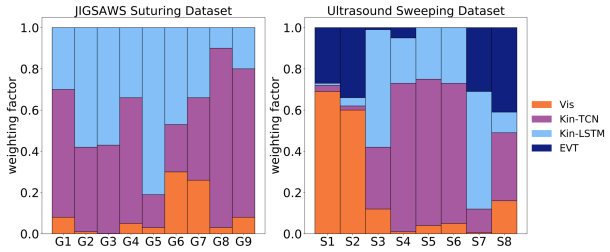

Fig. 6. Distributions of the normalized weighting factor matrix α for the JIGSAWS suturing task and the RIOUS imaging task. A larger weighting factor means that the model performs better at estimating the corresponding state.

collected directly from the surgical system. Additionally, when the state transition is frequent, models that solely explore the temporal dependencies of input data, such as TCN and LSTM, are less accurate. As the event-based model does not take the temporal correlations into consideration, incorporating such data source reduces the fluctuation in state estimation results, especially when the state transition is frequent or the duration of each state is short.

The average duration of each state in both JIGSAWS suturing dataset and the RIOUS dataset varies significantly, as shown in Table I. To better capture states with different lengths of duration, we implemented two kinematics-based state estimation models: TCN and forward LSTM. Fig. 6 supports our decision. When the average duration of a state is high, the LSTM-based model has a higher weighting factor. Similarly, the TCN-based model has a higher weighting factor for shorter-duration states.

As mentioned before, the RIOUS dataset is more complex compared to JIGSAWS and resembles real-world surgical tasks more closely. It is, therefore, more complicated and harder to be well-captured by a single-input state estimation model. Furthermore, our application of real-time state estimation limits the amount of data available to the model. Although running multiple state estimation models at the same time inevitably requires higher computing power, our fusion state estimation model is robust against complex and realistic surgical tasks such as ultrasound imaging and achieves a superior frame-wise accuracy.

V. Conclusions and Future Work

In this paper, we introduce a unified approach of fine-grained state estimation for various surgical tasks using multiple sources of input data from the da Vinci® Xi surgical system. Our models (including Fusion-KV and Fusion-KVE) improve the state-of-the-art performance for both the JIGSAWS suturing dataset and the RIOUS dataset. Fusion-KVE, which takes advantage of the system events (absent in the JIGSAWS dataset), further improves Fusion-KV. Our RIOUS dataset is more complex than JIGSAWS and resembles the real-world surgical tasks, with dry-lab, cadaveric and in-vivo experiments, as well as camera movements and annotations of both action and non-action states. Our unified model proves its robustness against complex and realistic surgical tasks by achieving a superior frame-wise accuracy even in a causal setting, where the model has knowledge of only the current and preceding time steps.

We show how different types of input data (vision, kinematics, and system events) have their respective strengths and weaknesses in the recognition of fine-grained states. The fine-grained state estimation of surgical tasks is challenging due to the duration of various states and frequent state transitions. We show that by incorporating multiple types of input data, we are able to extract richer information during training and more accurately estimate the states in a surgical setting. A possible next step of our work would be to use the weighting factor matrix for boosting methods to more efficiently train the unified state estimation model. Although modeled as an FSM, the fine-grained states within each surgical task are estimated independently, without influence from the previous state(s). Another potential next step would be to perform state prediction based on previously estimated state sequence. In the future, we also plan to apply this state estimation framework to applications such as smart-assist technologies and supervised autonomy for surgical subtasks.

Acknowledgment

This work was funded by Intuitive Surgical, Inc. We would like to thank Dr. Azad Shademan and Dr. Pourya Shirazian for their support of this research.

REFERENCES

[1] G. P. Moustris, S. C. Hiridis, K. M. Deliparaschos, and K. M. Konstantinidis, "Evolution of autonomous and semi-autonomous robotic surgical systems: a review of the literature," *The international journal of medical robotics and computer assisted surgery*, vol. 7, no. 4, pp. 375–392, 2011.

[2] P. Chalasani, A. Deguet, P. Kazanzides, and R. H. Taylor, "A computational framework for complementary situational awareness (csa) in surgical assistant robots," in *2018 Second IEEE International Conference on Robotic Computing (IRC)*. IEEE, 2018, pp. 9–16.

[3] S. P. DiMaio, C. J. Hasser, R. H. Taylor, D. Q. Larkin, P. Kazanzides, A. Deguet, B. P. Vagvolgyi, and J. Leven, "Interactive user interfaces for minimally invasive telesurgical systems," Feb. 15 2018, uS Patent App. 15/725,271.

[4] A. Zia, A. Hung, I. Essa, and A. Jarc, "Surgical activity recognition in robot-assisted radical prostatectomy using deep learning," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-López, and G. Fichtinger, Eds. Cham: Springer International Publishing, 2018, pp. 273–280.

[5] C. Lea, R. Vidal, and G. D. Hager, "Learning convolutional action primitives for fine-grained action recognition," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 1642–1649.

[6] R. DiPietro, C. Lea, A. Malpani, N. Ahmidi, S. S. Vedula, G. I. Lee, M. R. Lee, and G. D. Hager, "Recognizing surgical activities with recurrent neural networks," in *International conference on medical image computing and computer-assisted intervention*. Springer, 2016, pp. 551–558.

[7] G. Menegozzo, D. DallAlba, C. Zandonà, and P. Fiorini, "Surgical gesture recognition with time delay neural network based on kinematic data," in *2019 International Symposium on Medical Robotics (ISMR)*. IEEE, 2019, pp. 1–7.

[8] E. Mavroudi, D. Bhaskara, S. Sefati, H. Ali, and R. Vidal, "End-to-end fine-grained action segmentation and recognition using conditional random field models and discriminative sparse coding," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 1558–1567.

[9] T. Yu, D. Mutter, J. Marescaux, and N. Padoy, "Learning from a tiny dataset of manual annotations: a teacher/student approach for surgical phase recognition," *arXiv preprint arXiv:1812.00033*, 2018.

[10] A. Zia, C. Zhang, X. Xiong, and A. M. Jarc, "Temporal clustering of surgical activities in robot-assisted surgery," *International journal of computer assisted radiology and surgery*, vol. 12, no. 7, pp. 1171–1178, 2017.

[11] G. Yengera, D. Mutter, J. Marescaux, and N. Padoy, "Less is more: surgical phase recognition with less annotations through self-supervised pre-training of cnn-lstm networks," *arXiv preprint arXiv:1805.08569*, 2018.

[12] L. Tao, E. Elhamifar, S. Khudanpur, G. D. Hager, and R. Vidal, "Sparse hidden markov models for surgical gesture classification and skill evaluation," in *International conference on information processing in computer-assisted interventions*. Springer, 2012, pp. 167–177.

[13] J. Rosen, J. D. Brown, L. Chang, M. N. Sinanan, and B. Hannaford, "Generalized approach for modeling minimally invasive surgery as a stochastic process using a discrete markov model," *IEEE Transactions on Biomedical engineering*, vol. 53, no. 3, pp. 399–413, 2006.

[14] M. Volkov, D. A. Hashimoto, G. Rosman, O. R. Meireles, and D. Rus, "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, 2017, pp. 754–759.

[15] L. Tao, L. Zappella, G. D. Hager, and R. Vidal, "Surgical gesture segmentation and recognition," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2013, pp. 339–346.

[16] L. Zappella, B. Béjar, G. Hager, and R. Vidal, "Surgical gesture classification from video and kinematic data," *Medical image analysis*, vol. 17, no. 7, pp. 732–745, 2013.

[17] S. Krishnan, A. Garg, S. Patil, C. Lea, G. Hager, P. Abbeel, and K. Goldberg, "Transition state clustering: Unsupervised surgical trajectory segmentation for robot learning," in *Robotics Research*. Springer, 2018, pp. 91–110.

[18] B. van Amsterdam, H. Nakawala, E. De Momi, and D. Stoyanov, "Weakly supervised recognition of surgical gestures," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 9565–9571.

[19] C. Lea, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks: A unified approach to action segmentation," in *European Conference on Computer Vision*. Springer, 2016, pp. 47–54.

[20] R. DiPietro, N. Ahmidi, A. Malpani, M. Waldram, G. I. Lee, M. R. Lee, S. S. Vedula, and G. D. Hager, "Segmenting and classifying activities in robot-assisted surgery with recurrent neural networks," *International journal of computer assisted radiology and surgery*, pp. 1–16, 2019.

[21] C. Lea, A. Reiter, R. Vidal, and G. D. Hager, "Segmental spatiotemporal cnns for fine-grained action segmentation," in *European Conference on Computer Vision*. Springer, 2016, pp. 36–52.

[22] Y. Jin, Q. Dou, H. Chen, L. Yu, J. Qin, C.-W. Fu, and P.-A. Heng, "Sv-rcnet: workflow recognition from surgical videos using recurrent convolutional network," *IEEE transactions on medical imaging*, vol. 37, no. 5, pp. 1114–1126, 2017.

[23] L. Ding and C. Xu, "Tricornet: A hybrid temporal convolutional and recurrent network for video action segmentation," *arXiv preprint arXiv:1705.07818*, 2017.

[24] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 961–970.

[25] N. Ahmidi, L. Tao, S. Sefati, Y. Gao, C. Lea, B. B. Haro, L. Zappella, S. Khudanpur, R. Vidal, and G. D. Hager, "A dataset and benchmarks for segmentation and recognition of gestures in robotic surgery," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 9, pp. 2025–2041, 2017.

[26] A. P. Twinanda, S. Shehata, D. Mutter, J. Marescaux, M. De Mathelin, and N. Padoy, "Endonet: a deep architecture for recognition tasks on laparoscopic videos," *IEEE transactions on medical imaging*, vol. 36, no. 1, pp. 86–97, 2016.

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[28] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[30] F. A. Gers and J. Schmidhuber, "Recurrent nets that time and count," in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, vol. 3. IEEE, 2000, pp. 189–194.

[31] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[32] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.

[33] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet physics doklady*, vol. 10, no. 8, 1966, pp. 707–710.

[34] Y. Gao, S. S. Vedula, C. E. Reiley, N. Ahmidi, B. Varadarajan, H. C. Lin, L. Tao, L. Zappella, B. Béjar, D. D. Yuh, *et al.*, "Jhu-isi gesture and skill assessment working set (jigsaws): A surgical activity dataset for human motion modeling," in *MICCAI Workshop: M2CAI*, vol. 3, 2014, p. 3.