Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 55-38 (2022) 132–137

A Cognitive Architecture for Robot-Assisted Surgical Procedures

Elena Zini, Marco Minelli, Lorenzo Sabattini and Federica Ferraguti

Department of Sciences and Methods of Engineering, University of Modena and Reggio Emilia, Italy {name.surname}@unimore.it

Abstract: Robot-Assisted Minimally Invasive Surgery (RAMIS) procedures typically require the presence of two expert figures in the operating room (OR): the main surgeon, sitting at the console of a teleoperated surgical robot, and the assistant surgeon, who performs secondary operations directly on the patient by means of manual tools. In this paper we propose a novel strategy to allow the execution of a RAMIS procedure by a single surgeon, robotizing the role of the assistant surgeon. In addition to the teleoperation system used by the main surgeon, the architecture is augmented with an action recognition module, that recognize the operations the surgeon is performing, and a supervisory controller, which takes decisions according to the procedure state. The autonomous robotic arm serves as an assistant surgeon and it is equipped by a surgical tool to accomplish the required tasks. The proposed solution has been validated on a simplified physical set-up, with the aim of verifying and confirming its applicability and effectiveness.

Keywords: Robot surgery, telemanipulation, action recognition.

1. INTRODUCTION

Nowadays, the execution of Robot-Assisted Minimally Invasive Surgery (RAMIS) procedures requires several units of medical personnel working in the operating room. Just to report an example, a typical laparoscopic intervention needs the work of a main surgeon, of an assistant surgeon, of two nurses and of an anesthetist. The advent of robotic surgery and robots such as the da Vinci Surgical System (Intuitive Surgical, Inc., Sunnyvale, CA) has not decreased this number. Indeed, several assistants are required into the operating room for supporting the main surgeon that teleoperates the surgical robot. Among these people, an assistant surgeon is always requested for performing all surgical procedures that the main surgeon cannot perform with the robot he/she is teleoperating, e.g. aspiration of blood, removal of dead tissues and moving organs Chiu et al. (2008). Typically, the role of the assistant surgeon is taken by an expert surgeon, even if he/she is requested to perform critical tasks for 30% of the time of the surgical procedure. Considering the hourly cost of a surgeon, the current practice is very inefficient from an economic point of view. Furthermore, the current practice is very inefficient also from a social point of view. Indeed, both assistant and main surgeons need to rest for a fixed number of hours among interventions, reducing by a half the number of available surgeons leading to unnecessary long waiting lists.

Artificial Intelligence (AI) (Winston (1992)) is the theory and development of computer systems able to perform tasks normally requiring human intelligence, such as visual perception, speech recognition, decision making and translation between languages [1]. AI has been constantly rising in the last years and it represents a powerful tool when employed in applications where human errors need to be mitigated. In the surgical field, it finds application in a variety of areas, such as clinical decision support, patient monitoring, pre-operative surgical planning and many others. A very interesting domain in which the power of AI started to be exploited is the possibility of providing the robot with autonomy during surgeries. Indeed, the introduction of autonomy in robotic surgery would result in increasing efficiency and repeatability, reducing costs, and improving execution quality thanks, for example, to real-time bio-signal feedback and computer-aided guidance.

In this paper, we will focus on reaching a greater degree of autonomy through the introduction of an autonomous robotic surgeon's assistant that can make decisions on its own and properly cooperate with the main surgeon autonomously. To achieve this goal, it is necessary to develop an intelligent system that can recognize the actions the main surgeon performs during the surgical operation and consequently robotize the assistant's tasks accordingly. The benefits derived from such automation are the reduction of the number of surgeons required for a single surgical operation and the possibility to parallelize operations within hospitals. This leads to greater economic and social efficiency by reducing medical expenses and speeding up the long waiting lists. The authors in van Amsterdam et al. (2021) proved that so far deep-learning-based models are the go-to choice. Actually, unlike graphical models, they are able to capture even complex temporal dependencies during surgical motions. On the other hand, unsupervised and semi-supervised methods reach lower performances and related works in this area are still very limited. To report some examples, Funke et al. (2019) propose to use a 3D Convolutional Neural Network (CNN) to learn spatiotemporal features from consecutive

[1] https://www.oxfordreference.com/

10.1016/j.ifacol.2023.01.145

video frames. Different approaches based on the use of CNNs are developed also by Khatibi and Dezyani (2020). However, both cases propose a generic solution for the action recognition problem, without explicitly relating it to a robotic assistance.

The majority of current state-of-the-art approaches are restricted also to offline functionalities, making their use impossible in real-world scenarios, where high speed is one of the main requirements. In scenarios like the one considered in this paper, where a collaboration between surgeon and robot is required and the goal is to automatize the assistant tasks, video frames must be processed and recognized in real-time.

A cognitive robotic architecture for assisting an operator to perform a cooperative task has been proposed in De Rossi et al. (2019), using a CNN to recognize the actions, and a Model Predictive Controller-based motion control to deal with an unreliable confidence level of the action recognition. In this paper we propose an alternative approach to De Rossi et al. (2019), based on a different AI module and capable of providing more stable detections. Moreover, we neglect the effect of the confidence level on the action recognition since our strategy allows us to retrieve high level of accuracy in the action recognition, simplifying also the design of the supervisory controller, which here has been implemented as a simple Finite State Machine.

In this paper we exploit the model proposed in Singh et al. (2017) which aims at achieving a real-time spatio-temporal action localisation and prediction. It adopts a real-time Single Shot Multibox Detector (SSD) CNN to regress and classify detection boxes in each video frame potentially containing an action of interest. A SSD is capable of predicting the object's bounding box in a single shot, making it one of the fastest object detection algorithms available. Starting from the resulting detections, some post-processing is then applied to link up the spatial action bounding boxes over time to create the so-called action tubes. The SSD model has been evaluated on UCF101-24 dataset, a subset of the UCF101 dataset Gao et al. (2014), containing 24 human action categories of sports.

We started by establishing a series of simplified actions to form a surgical procedure of cutting, puncturing and suturing, and by creating a specific dataset of these actions emulating the surgical procedure. Moreover, we developed a control architecture for allowing cooperation between the surgeon and the robot.

The contribution of this paper are:

- A strategy to allow the execution of a RAMIS procedure by a single surgeon providing superior performance than previous works with a simpler architecture.
- An experimental validation of the proposed strategy.

The rest of this paper is organized as follows. Section 2 reports the strategies we followed to create the dataset, while Section 3 describes the system architecture. Section 4 reports the experimental validation. Finally, conclusions and future works are reported in Section 5.

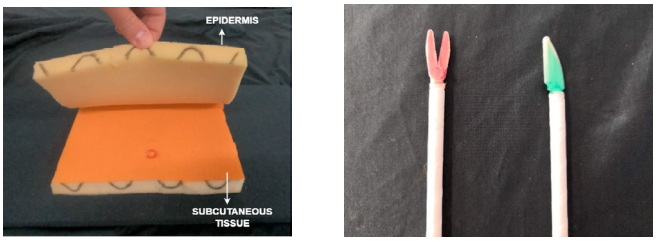

Fig. 1. The phantom and the 3D printed surgical tools.

Fig. 2. Artificial setup for the acquisition of the dataset.

2. DATASET CREATION

In order to evaluate the applicability of the proposed architecture regardless of the problems related to the execution of a real surgical operation, we created an ad-hoc artificial set-up which emulates a surgical procedure. We created an artificial phantom, shown in Fig. 1(a), to emulate the part of the body involved in the operation. In particular, we used two layers of foam rubber laid to emulate the skin. The first layer is movable and represents the epidermis, while the second one is fixed and represents the subcutaneous tissue. The emulated surgical tools used to carry out the tasks, shown in Fig. 1(b), are 3D printed and simulate a scalpel and a forceps. Fig. 2 shows the setup during the acquisition of the videos for the dataset creation.

The surgical operation we take into consideration consists in performing a subcutaneous injection. This procedure is performed by a main surgeon and an assistant surgeon performing the following tasks:

(1) Holding the upper tissue to make it stable;
(2) Cutting the tissue;
(3) Lifting the flap of tissue resulting from the cutting operation;
(4) Making a puncture into the subcutaneous tissue;
(5) Holding the upper tissue;
(6) Suturing the wound.

Tasks 1, 3, and 6 are allocated to the main surgeon, while tasks 2, 4, and 5 are performed by the assistant surgeon. Nevertheless, the tasks could be easily exchanged between the main surgeon and the assistant surgeon. We recorded six videos of the procedure using an Intel RealSense Depth Camera D415 with a resolution of 960×540 pixels and a frame rate of 60 fps. The annotations were performed using Microsoft Virtual object Tagging Tool [2], which allows to draw bounding boxes around regions of interest in visual data and save them in the preferred format (COCO [3] in our case). Bawa et al. (2021) was

[2] https://github.com/microsoft/VoTT
[3] https://cocodataset.org/#format-data

Table 1. Number of action samples in the training, validation and test phase.

Label	Train	Val	Test	Total
Idle	258	32	33	323
FlapHoldDown	1509	167	187	1863
FlapHoldUp	837	127	95	1059
Suture	1482	185	185	1852

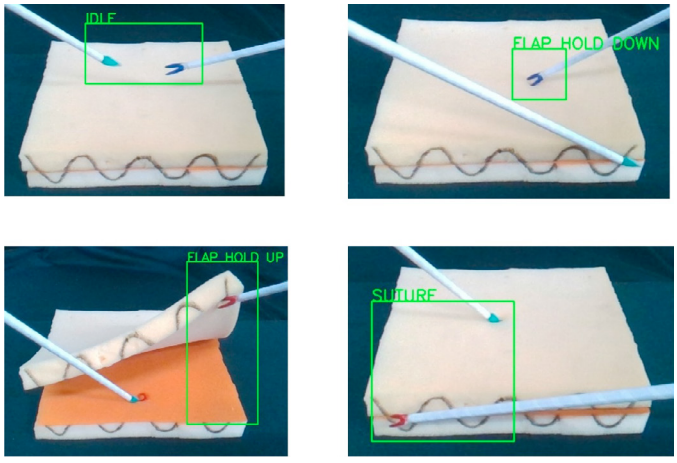

Fig. 3. Annotated bounding boxes of action categories.

considered as a guideline. The actions we annotate concern only those which are performed by the main surgeon since the main purpose of the system is to recognize the main surgeon's actions and not the assistant's ones, that will be subsequently robotized on the basis of recognition. Four action categories were chosen:

- **Idle**: the action in which both the surgical tools are positioned in the middle of the operating area waiting for a movement to be executed.
- **FlapHoldDown**: the action of holding the tissue involved in the surgical operation, i.e. holding the upper tissue during the incision.
- **FlapHoldUp**: the action of lifting the flap of tissue resulting from the incision. This action is needed to allow further operations that require having access to the subcutaneous tissue, such as inserting the needle for making the puncture.
- **Suture**: the action of sewing the tissue to close the wound.

Table 1 reports a list of the actions in the created dataset, with the number of samples in each of the training, validation and test phase. Finally, Fig. 3 reports a picture of each of the actions segmented in the created dataset and described so far. The main surgeon tool has two different colors, which distinguish accordingly to the action the surgeon is performing.

3. COGNITIVE ARCHITECTURE FOR ROBOT-ASSISTED SURGICAL PROCEDURES

The proposed control architecture, shown in Fig. 4 is composed by an *Action Recognition Module* and a robotic assistant managed by a *Supervisory Controller*. The action recognition module, based on a real-time Single Shot Multibox Detector (SSD) Convolutional Neural Network (Liu et al. (2016)), elaborates the frames of the video of the procedure and provides as output the detection boxes frame by frame, along with the confidence scores. The supervisory controller, implemented as a Finite State Machine (FSM), decides the action the robot has to perform depending on the recognized action performed by the main surgeon. In the following we will describe each of these modules.

3.1 Action Recognition Module

Network for action recognition The proposed system needs to detect the actions performed in real-time by the main surgeon, in order to command the corresponding actions to the robot as soon as the main action is recognized. Indeed, the robot must cooperate with the surgeon in a smooth and coordinated way, while avoiding every kind of delay that could make the system inefficient. To this aim, we exploited and implemented a Single Shot Multibox Detector (SSD) Convolutional Neural Network since it allows to perform regression and classification in a single-stage efficiently. Then, we trained the SSD with the dataset already presented in Section 2. SSD is based on a feed-forward CNN scores for the presence of object category instances in those boxes, followed by a non-maximum suppression step to generate the final detections. It is based on a so-called base network used for high-quality image classification, which is composed of the early layers of VGG 16 (Simonyan and Zisserman (2015))and truncated before any classification layer. At the end of the base network, some convolutional features layers that progressively decrease in size are added: this peculiarity allows to predict detections at multiple scales. Each of these convolutional features layers is able to provide a fixed set of detection predictions by using a set of convolutional filters. For a feature layer of size $m \times n$ with p channels, there is a $3 \times 3 \times p$ small kernel that produces either a score for a category or a shape offset relative to the coordinates of the default box. The kernel is applied to each of the $m \times n$ locations and it generates an output value.
The bounding box offset output values are measured in comparison to a default box position relative to each feature map location.

Since in the emulated setup the environment is highly controlled and the actions to be recognized are quite different from each other, we did not use optical flow images, since the information provided by them would be redundant. However, they could be integrated and exploited in case of more complex environments, like for example in a real surgical procedure, to detect even the motion information.

Evaluation metrics Before introducing the results obtained by the action recognition module, we need to introduce the metrics we used to evaluate the network performance: the precision, the recall, the average precision, the mean average precision, and the intersection over union.

The **Precision** of the network represents the percentage of instances correctly classified as positive over the total number of instances classified as positive, and it is defined as:

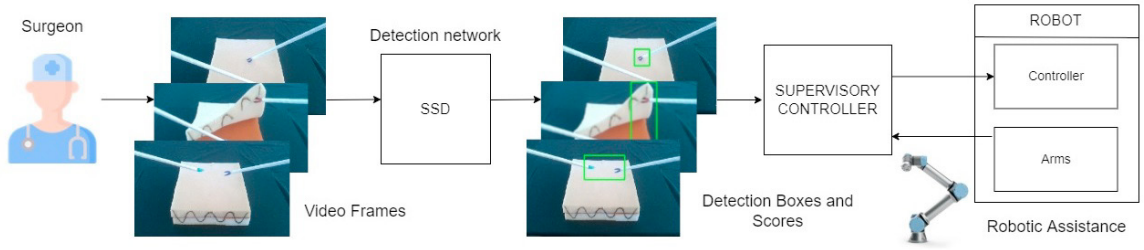

Fig. 4. Proposed control architecture. An action recognition module detects the action performed by the main surgeon. Then, the supervisory controller enables the robot to provide assistance to the main surgeon.

$$\mathcal{P} = \frac{TP}{TP + FP} \tag{1}$$

where $\mathcal{P} \in \mathbb{R}$ represents the precision, $TP \in \mathbb{R}$ represents the number of true positives, and $FP \in \mathbb{R}$ represents the number of false positives.

The **Recall** of the network represents the rate of positive instances correctly recognized as such, and it is defined as:

$$\mathcal{R} = \frac{TP}{TP + FN} \tag{2}$$

where $\mathcal{R} \in \mathbb{R}$ represents the recall, and $FN \in \mathbb{R}$ represents the number of false negatives.

The **Average Precision** $AP \in \mathbb{R}$ is obtained by plotting precision against recall yielding to a precision-recall curve and then integrating the area under the curve.

The **mean Average Precision** $mAP \in \mathbb{R}$ over a set of classes is the mean of the average precision scores for each class.

Finally, in order to understand if a predicted detection is right, we can measure the percentage overlap between the predicted and the ground truth bounding boxes. The **Intersection over Union** $IoU \in \mathbb{R}$ is the ratio between the area of the intersection of the two bounding boxes over the area of their union. If its value is 0, it means that there is no overlap between the predicted and ground truth boxes; otherwise, if its value is 1, then the predicted and ground truth boxes are completely overlapping.

Training and Testing We adopted the SSD architecture with an input image size of 300×300 and an ImageNet pre-trained VGG 16 network. The network has been trained for 5000 iterations with a learning rate of 0.001,a batch size of 16, a total number of 4086 frames and 4 action classes. The model has been implemented in PyTorch and trained using CUDA.

The goal of training is to find the set of weights and biases that reports a low average value of localisation loss and confidence loss across all the examples. The **localisation loss** represents the mismatch between the ground truth box and the predicted boundary box while the **confidence loss** is the loss of making a class prediction. From Fig. 5 we can observe that, during the iterations of the training phase, both the localisation and confidence loss decrease until reaching a value that is very close to 0, which is a very promising result.

We tested the network with a total number of 1012 frames and by choosing a confidence threshold equals to 0.9. Results are reported in Table 2. We can observe that

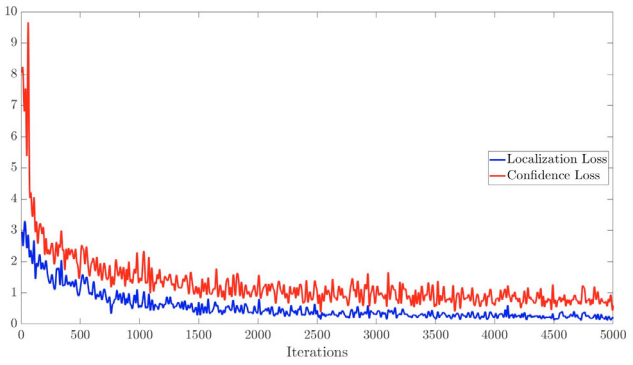

Fig. 5. Plot of training losses (confidence loss and localisation loss).

Table 2. mAP results of the SSD trained on the created dataset.

IoU threshold δ	**0.5**	**0.8**	**0.85**	**0.9**
AP Idle	1.0	0.9576	0.5903	0.3989
AP FlapHoldDown	1.0	0.9941	0.9941	0.4044
AP FlapHoldUp	1.0	1.0	1.0	1.0
AP Suture	0.9597	0.9147	0.7944	0.4631
mAP	0.9899	0.9666	0.8447	0.5666

the AP related to each class of action and the mAP over all the classes vary according to the chosen IoU threshold $\delta \in \mathbb{R}$. It can be noted also that AP decreases as δ increases. However, even by keeping a more stringent δ, such as 0.8, the results are very good and reach an mAP value of approximately 0.97. On the other hand, by choosing a δ very stringent, such as 0.9, the results are not trustworthy. This is due to the fact that, in general, it is extremely unlikely that the coordinates of the predicted bounding box are going to exactly match the coordinates of the ground truth bounding box. Usually, a $\delta \geq 0.5$ is considered a good prediction.

3.2 Supervisory Controller

The results obtained by the action recognition module determine the tasks the robot must carry out. The supervisory controller is required to coordinate the recognized actions performed by the main surgeon and the motion of the robotic arm. The supervisor has been implemented as a Finite State Machine (FSM), and its schematics is reported in Fig. 6. The initial state is INIT, in which the robot is located by the surgeon in the operational area. If the action recognized by the neural network corresponds

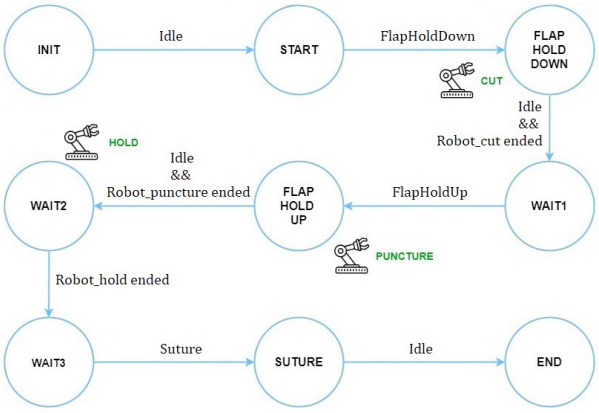

Fig. 6. Finite State Machine implemented as supervisory controller.

to **Idle**, a transition from INIT to START occurs. The system remains in this state until the recognized action is **FlapHoldDown**, the one corresponding to holding the flap down. This action triggers the transition to the state FLAP HOLD DOWN, which will command the robot to perform the incision of the tissue. Once the cut is over and the action performed by the surgeon is recognized as **Idle**, the transition to state WAIT1 happens. The system remains in this state until the action **FlapHoldUp**, corresponding to holding the flap up, is detected and causes the transition to the state FLAP HOLD UP. Once entered this state, the robot has to perform the injection. Once the puncture is over, the transition to state WAIT2 happens when **Idle** is recognized. Then, the robot is commanded to hold down the flap and the system moves to the WAIT3 state. At this stage, when an action corresponding to **Suture** is recognized, the transition to the state SUTURE occurs and just when the neural network recognizes an action corresponding to **Idle**, the transition to the last state END happens. Here the robot has accomplished all its tasks and comes back to its initial position.

3.3 Robot control

To allow the emulation of the robotic surgery, a 3D printed laparoscopic instrument was mounted to the end-effector of the robot. This tool allows at the same time to emulate the cut performed by a scalpel, a puncture, and to be used to immobilize tissues. Since the automation of movements according to the action to be performed is not the focus of this work, the actions performed by the robot are pre-programmed, using the robot's position control.

4. EXPERIMENTAL VALIDATION

In order to validate the effectiveness and the proper functioning of the developed system, an experimental validation on the emulated setup has been performed. In particular, a real-time video of the emulated subcutaneous injection is provided as input to the control system. The main surgeon starts performing actions of his responsibility and expects the robot to replace the assistant's tasks and cooperate properly with him. A videoclip showing the setup and the entire experiment can be found at `https://youtu.be/U3n4KdO-e1Q`.

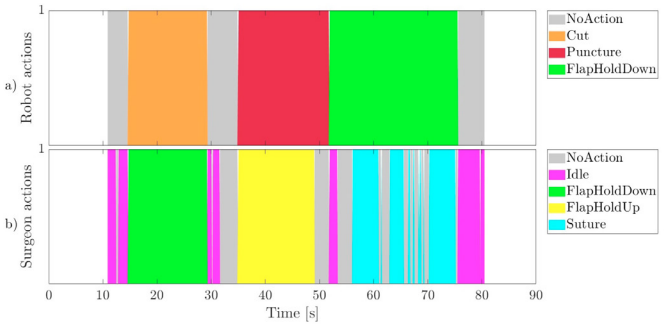

Fig. 7. Evolution over time of the surgical actions performed by the surgeon and recognized by the system and the corresponding robot actions.

It is important to specify that, as regards the SSD, a confidence threshold equal to 0.9 has been chosen. This choice is dictated by the fact that we labeled only the actions which are of interest for triggering the robot and making it cooperate with the surgeon in a synchronous way. For this reason, the intermediate movements among actions, such as approaching the operational area or moving away from it, could be mistaken for the actual annotated actions if a low confidence threshold is chosen. The value of 0.9 has been experimentally chosen.

To evaluate the performance of the system, we compared the actions performed by both the surgeon and the robot over time, as reported in Fig. 7. The results confirm the right execution of the operations. Indeed, as shown in Fig. 7, when the surgeon starts holding the tissue of the body part involved in the surgical operation (green area in Fig. 7(b)), then the robot starts performing the incision of the tissue (orange area in Fig. 7(a)). Once the incision has been successfully completed, the surgeon releases the tissue, comes back to the idle position and starts holding the flap of cut tissue (yellow area in Fig. 7(a)). At this point, the robot starts making a puncture into the subcutaneous tissue (red area in Fig. 7(a)) and once this operation has been concluded, the surgeon releases the lifted tissue coming back to the idle position and this triggers the robot to start holding the tissue (green area in Fig. 7(a)). Then, the surgeon can now proceed with the suture (blue area in Fig. 7(b)). Once the suture is over, the robot has accomplished all its tasks and the surgical operation terminates.

We can thus conclude that surgeon and robot cooperate as expected and their tasks are synchronized, resulting in a successful surgical operation. Furthermore, it can be noticed that the SSD is actually able to recognize accurately and in real-time the surgeon's actions over time. However, it is possible to see that it suffers from a bit of uncertainty in the recognition of the suture action, which sometimes is mistaken for no action. This aspect does not influence the correct functioning of the system since it is properly managed by the FSM. Nevertheless, it has to be taken into consideration for future developments and to improve the performance of the SSD.

Fig. 8 reports the robot end-effector position and orientation (pose) among with the robot action. We can notice that the pose change according to the actions performed by the robot, which in turn depend on the surgeon's

actions recognized by the SSD. In the same way, when the robot is not acting (grey areas in Fig. 8(c)), position and orientation remain the same, as expected. The evolution over time of the robot end-effector pose could be exploited also for understanding other information about the surgical procedure, such as the length and the depth of an incision or a puncture, or to check collisions with other organs not involved in the operation. Moreover, some post-surgical complications may be due to medical errors and the monitoring of the trajectories followed by the autonomous robotic surgeon's assistant might be useful to understand what went wrong in the operation.

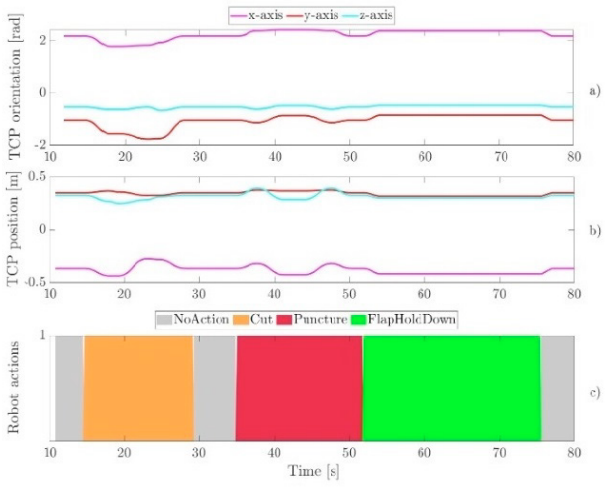

Fig. 8. Evolution over time of robot actions along with Tool Center Point position and orientation.

5. CONCLUSION

In this work we developed an autonomous system that allows the execution of a RAMIS procedure by a single surgeon. The results achieved in this work have revealed the correct functioning of the system, and its capacity to accurately recognize the surgeon's actions over time, providing an efficient and well-coordinated cooperation between surgeon and robot.

It must, however, be emphasized that the use of a dataset composed of a few elementary surgical tasks, captured in the same artificial environment, precludes the consideration of different issues that could arise in a real setup. Among these, we find the deformable nature of organs, which could vary from person to person, the diversity of the same organ depending on the orientation of the camera that captures the surgical scene, and the surgeon's skill level and operative style. All these aspects could prevent the SSD, or more in general neural networks, to learn how to discriminate different categories of actions. Despite this, the results of this work allow us to affirm the effectiveness of the proposed strategy in a controlled environment, as a first step towards the application of autonomous robotic assistance in surgical environments.

Possible future developments in this direction could be the creation of an exhaustive dataset inclusive of several surgical operations performed in complex environments with blood, camera motions, illumination changes, occlusions and variability in motion and action order. This could help towards generalizing and avoiding learning just a few specific training samples, leading to overfitting and lack of accuracy against unseen data. Moreover, the use of optical flow images, together with RGB ones, could help the network to improve the learning of how to differentiate actions, exploiting the information of motion among frames.

REFERENCES

Bawa, V.S., Singh, G., KapingA, F., Skarga-Bandurova, I., Oleari, E., Leporini, A., Landolfo, C., Zhao, P., Xiang, X., Luo, G., et al. (2021). The saras endoscopic surgeon action detection (esad) dataset: Challenges and methods. *arXiv preprint arXiv:2104.03178*.

Chiu, A., Bowne, W.B., Sookraj, K.A., Zenilman, M.E., Fingerhut, A., and Ferzli, G.S. (2008). The role of the assistant in laparoscopic surgery: important considerations for the apprentice-in-training. *Surgical Innovation*, 15(3), 229–236.

De Rossi, G., Minelli, M., Sozzi, A., Piccinelli, N., Ferraguti, F., Setti, F., Bonfé, M., Secchi, C., and Muradore, R. (2019). Cognitive robotic architecture for semi-autonomous execution of manipulation tasks in a surgical environment. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 7827–7833. doi:10.1109/IROS40897.2019.8967667.

Funke, I., Bodenstedt, S., Oehme, F., Bechtolsheim, F.v., Weitz, J., and Speidel, S. (2019). Using 3d convolutional neural networks to learn spatiotemporal features for automatic surgical gesture recognition in video. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 467–475. Springer.

Gao, Y., Vedula, S.S., Reiley, C.E., Ahmidi, N., Varadarajan, B., Lin, H.C., Tao, L., Zappella, L., Béjar, B., Yuh, D.D., et al. (2014). Jhu-isi gesture and skill assessment working set (jigsaws): A surgical activity dataset for human motion modeling. In *MICCAI workshop: M2cai*, volume 3, 3.

Khatibi, T. and Dezyani, P. (2020). Proposing novel methods for gynecologic surgical action recognition on laparoscopic videos. *Multimedia Tools and Applications*, 79(41), 30111–30133.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., and Berg, A.C. (2016). Ssd: Single shot multibox detector. In *European conference on computer vision*, 21–37. Springer.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*.

Singh, G., Saha, S., Sapienza, M., Torr, P., and Cuzzolin, F. (2017). Online real-time multiple spatiotemporal action localisation and prediction. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 3657–3666. doi:10.1109/ICCV.2017.393.

van Amsterdam, B., Clarkson, M.J., and Stoyanov, D. (2021). Gesture recognition in robotic surgery: A review. *IEEE Transactions on Biomedical Engineering*, 68(6), 2021–2035. doi:10.1109/TBME.2021.3054828.

Winston, P.H. (1992). *Artificial intelligence*. Addison-Wesley Longman Publishing Co., Inc.