Run-time verification of robot-assisted surgery using visual input

Kristina Gogoladze[1,*], Romy van Jaarsveld[3], Natasha Alechina[2,1], Ronald de Jong[3], Yasmina Al Khalil[4], Gino Kuiper[3], Brian Logan[5,1] and Jelle P. Ruurda[3]

[1]*Universiteit Utrecht, Utrecht, The Netherlands*
[2]*Open Universiteit, Heerlen, The Netherlands*
[3]*University Medical Centre Utrecht, The Netherlands*
[4]*Technical University Eindhoven, The Netherlands*
[5]*University of Aberdeen, UK*

Abstract

We present a preliminary study of the feasibility of using run-time verification to provide intraoperative warnings during robot-assisted surgery for cancer. Robot-assisted surgery offers several advantages over traditional surgery. However, it can be challenging for surgeons, particularly if they lack experience in the procedure being performed. We report the results of a survey to elicit from surgeons which clinically relevant properties should be monitored, and present a proof-of-concept evaluation of the feasibility of monitoring using the video image available to the surgeon as input, which suggests that real-time monitoring for violations of clinically relevant properties is possible.

Keywords

Robot-assisted surgery, Clinically relevant properties, Image segmentation, Run-time verification

1. Introduction

In this paper, we present a preliminary study of the use of run-time verification in robot-assisted surgery for cancer. Oncological operations are among the most complex surgical procedures, as the tumor and chemoradiotherapy can affect surrounding tissues. Changes in anatomy and tissue consistency can cause surgeons to have more difficulty operating effectively and efficiently. Robot-Assisted Minimally Invasive Surgery is an increasingly utilized method to treat operable cancer. Robot-assisted surgery offers several advantages over traditional surgery, including an enlarged three-dimensional view, articulated instruments, and improved ergonomics [1, 2]. However, for a surgeon to become proficient in robot-assisted surgery and operate without supervision requires considerable experience; for example, to become proficient in esophageal cancer surgery typically requires experience of 20 to 80 cases. Moreover, during surgery, the surgeon must take into account many tasks at the same time to achieve the best outcome for the patient. For surgeons in training, the mental and physical load of these tasks requires a great deal of energy and time [3]. We hypothesize that intraoperative warning systems based on run-time verification could prevent unwanted actions and thus possibly prevent complications, improving patient outcomes.

The objective of this study is to investigate the feasibility of warning systems based on run-time verification for intraoperative use. We focus on surgery performed using the *da Vinci Surgical System* robot (dVSS), which is widely used in robot-assisted surgery. The manufacturers of the dVSS, Intuitive Surgical, do not make the interface to the sensors of the robot available to end users or researchers. Some previous studies have been conducted using the da Vinci Research Kit (dVRK) which does allow

7th International Workshop on Artificial Intelligence and fOrmal VERification, Logic, Automata, and sYnthesis (OVERLAY 2025 ECAI 2025), October 26th, Bologna, Italy.
*Corresponding author.
k.gogoladze@uu.nl (K. Gogoladze)
https://www.uu.nl/staff/KGogoladze (K. Gogoladze)
0009-0007-6801-837X (K. Gogoladze)

access to the robot's sensors [4]. However, the dVRK only runs on out-of-date hardware that is no longer used in surgery. In this study, we therefore use the video image available to the surgeon during surgery as input. The key contributions of this paper are twofold: we report the results of a survey to elicit from surgeons which clinically relevant properties should be monitored; and we present a proof-of-concept evaluation of the feasibility of monitoring for one of the properties using the video image available to the surgeon as input.

2. Clinically Relevant Properties

A key problem in providing intraoperative warnings is to determine which properties should be monitored. In this section, we present the results of a survey to gauge whether surgeons would consider such warnings useful, and, if so, which clinically relevant properties should be monitored. The survey was conducted during an international surgical course on Minimally Invasive Gastrectomy and Esophagectomy. The properties contained in the survey are therefore grounded in surgery for gastrectomy and esophagectomy, but are also relevant to other forms of robot-assisted minimally invasive surgery. In conjunction with surgical experts from University Medical Center Utrecht we formulated eleven different properties relevant to preventing postoperative complications that could give rise to intraoperative warnings.

A total of 32 surgeons with differing levels of experience from novice to expert took part in the survey, including surgical fellows and consulting surgeons. The respondents were stratified into two groups for subgroup analysis: 7 experts (defined as someone who had performed more than 100 RAMIEs), and 25 trainees (who had performed fewer than 100 RAMIEs).

The surgeons were asked the following questions. The results were descriptively analyzed and the average response is shown after each question.

1. A surgical warning system could monitor and warn about the intraoperative events. Do you think you would experience benefit from such a warning system? [Average score 4.4 (where 1 was no benefit, 5 definitely benefit).]
2. Please indicate which events you believe are relevant (you may select multiple options). [The percentage of participants who selected the property is shown after each property.]
    - Excessive force on tissue [81%]
    - Activated energy source too near to vital structure (risk of thermal injury) [75%]
    - Irregular tool movement (e.g., sudden or unsure motion of instrument) [75%]
    - Thermal instrument out of camera view is activated [63%]
    - Surgical material left in body at end of procedure (e.g., gauze, needle) [66%]
    - Inadequate suture (e.g., knot tied is not square, incorrect needle position, direction of suture) [44%]
    - Non-addressed bleeding (camera moves away from active bleeding) [41%]
    - Instrument collision with vital structure [75%]
    - Moving instrument out of camera view [34%]
    - Poor tissue stabilization (misgrasping tissue, excessive grasping) [19%]
    - Incorrect use of camera (not focused on working area) [9%]

Interestingly, subgroup analysis shows that, on average, expert surgeons selected 6.4 events (range 4-9), whereas trainees selected 5.6 events (range 2-11), suggesting that experts could see benefit in more warnings than trainees.

3. Proof-of-Concept: Surgical Instrument Detection

Without access to sensor information from the dVSS, it is not possible to monitor for events such as "Excessive force on tissue" or "Activated energy source too near to vital structure". However, with

input from a video segmentation system that can reliably identify surgical instruments and anatomical structures, we can monitor for events such as “Irregular tool movement”, “Non-addressed bleeding”, “Instrument collision with vital structure”, and “Moving instrument out of camera view”. While possible in principle, detecting such events is challenging given the current state of the art in image segmentation. For a proof-of-concept study, we therefore focus on the simplest case of monitoring for instruments moving out of the camera view.

The proof-of-concept system consists of an image processing component and a run-time verification component connected by a simple Python-based client.

3.1. Instrument Segmentation

Recent advances in surgical instrument segmentation allow accurate real-time identification of instruments in minimally invasive surgery. The particular model used for in the proof-of-concept is based on the large-scale surgical foundation model introduced by Jaspers et al. [5]. We used the CAFormer-S18 [6] variant of their work, pre-trained with self-supervised learning on the SurgeNetXL dataset. This is a diverse collection of more than 4.7 million surgical video frames drawn from public, private, and curated YouTube sources [5]. This extensive pre-training supports strong feature learning in the domain, enhancing performance in downstream tasks such as instrument detection. The model was fine-tuned using the RAMIE dataset in de Jong et al. [7] to ensure strong performance in our intraoperative thoracoscopic data. The RAMIE data set consists of 879 annotated frames extracted from thoracoscopic robot-assisted minimally invasive esophagectomy (RAMIE) procedures, collected from 32 patients. Each frame is densely labeled across 12 classes, including several anatomical structures such as the aorta and the airways, and four surgical instruments (forceps, suction & irrigation, hook, and vessel sealer). The annotations were performed by trained researchers and validated by an experienced surgeon.

For our experiments, we extracted only the output segmentation maps for the four surgical instruments, as the anatomy was not relevant for the monitored property. The presence of each instrument in each frame is computed as follows:

$$c_i = \sum_x \sum_y S_i(x, y), \tag{1}$$

where $S_i(x, y)$ is the segmentation map for class i, defined over all pixel coordinates (x, y). The value of $S_i(x, y)$ is 1 if the pixel at (x, y) is predicted as class i, and 0 otherwise. Thus, c_i represents the total number of pixels predicted as class i. The binary presence indicator b_i for class i is defined as:

$$b_i = \begin{cases} 1, & \text{if } c_i > T \\ 0, & \text{otherwise} \end{cases} \quad \text{for } i \in \{1, 2, 3, 4\} \tag{2}$$

where T is the pixel count threshold used to determine the presence of the instrument associated with class i. In our experiments, T was set to 50, as this empirically led to stable predictions. If the number of pixels predicted as class i exceeds the threshold T, the instrument is considered present ($b_i = 1$); otherwise, it is considered absent ($b_i = 0$).

3.2. Run-time Verification

For the run-time verification component, we employed the Numerical Runtime Verification (NuRV) tool [8], specifically the "NuRV orbit" executable, designed for online run-time verification of properties expressed in Linear Temporal Logic (LTL). NuRV is a robust real-time verification tool developed for monitoring systems by evaluating user-defined temporal logic properties against live data streams. NuRV’s specification language allows the definition of properties using temporal logic operators such as Globally (G), Eventually (F), Next (X), and Until (U). Properties can express complex temporal constraints succinctly, providing flexibility to specify safety-critical properties. The tool supports efficient evaluation by compiling these properties into finite-state automata, enabling real-time verdict generation on system states.

For example, a property that gauze should not be left in the wound can be expressed as $G(\neg(gauze\ U(suturing\ \wedge\ gauze))$ (it is never the case that *gauze* proposition holds until and including the moment when *suturing* proposition becomes true.

For our specific monitoring scenario, ensuring that instruments remain within camera view, we employed a straightforward safety property, $G(inCameraView)$, i.e., that all four surgical instruments should always be in the camera view which is a clinically relevant safety requirement.

4. Evaluation

In this section, we present a preliminary evaluation of the proof-of-concept intraoperative warning system. The aim of the evaluation was twofold:

- to determine the reliability of surgical instrument detection, i.e., whether there were false positives or false negatives regarding the detection of tools in the camera view; and
- to determine the time required by both the visual processing component and the run-time verification software.

Together, these two factors determine the practical feasibility of real-time surgical monitoring. If the visual processing component is not sufficiently reliable and/or either the visual processing component or the run-time verification software requires time greater than the frame rate, the warnings will not be useful.

4.1. Experimental Setup

The experimental setup involved real-time monitoring during robot-assisted surgery at the University Medical Center Utrecht. Due to the need to accommodate the surgical schedule, the experiment was performed during stomach cancer surgery rather than RAMIE on which the model was fine-tuned. From the point of view of the monitored property there is no difference in difficulty between the procedures; however, the reliability of the segmentation model may have been impacted.

The surgical procedure was performed using the da Vinci Surgical System. The dVSS was directly connected to a computing system running both the neural model for instrument segmentation and the NuRV orbit monitoring tool. Intraoperative video was captured from the surgical console using an HDMI-to-USB video capture card with a resolution of 640x480 pixels and a frame rate of 25 frames per second. The model segmented the surgical instruments using one CUDA core of a GeForce GTX 1050 Ti Mobile graphics card. The setup also included a Python-based client developed specifically for interacting with the NuRV monitoring tool and logging the monitoring results. The client managed data capture, pre-processing, segmentation model execution, and communication with NuRV.[1]

The surgical procedure lasted more than two hours. However, the run-time verification experiment was restricted to a 15 minute period during the procedure. During this time the Python client continuously logged and evaluated the property $G(inCameraView)$. As the experiment was designed solely to establish the feasibility of monitoring, no warnings were given to the surgeon.

4.2. Results

During the surgical procedure, the property $G(inCameraView)$ was frequently violated, indicating that instruments at least momentarily exited the camera view. In some cases, violations were due to the segmentation model failing to identify tools which were actually present in the view. In addition, in a few cases, instruments were identified as being in the camera view when there were no instruments visible. We conjecture that this was due to the model being trained with videos of a different surgical procedure (RAMIE). Both false positives (instruments not detected when they are present) and false

[1]The Python code and experimental data are available at https://github.com/UtrechtUniversity/RAS-verification/tree/main/experiments/runtime-monitoring-er.

negatives (instruments detected when they are not present) are problematic for monitoring. Too many false positives are distracting and may cause surgeons to ignore real warnings. False negatives mean that property violations are not being detected.

	Hook	Forceps	Suction & Irrigation	Vessel Sealer
TP	20	10	10	18
TN	8	18	12	10
FP	2	1	6	2
FN	0	1	2	0
Precision	0.91	0.91	0.63	0.90
Recall	1.00	0.91	0.83	1.00
F1-score	0.95	0.91	0.71	0.95

Table 1
Performance metrics for segmentation of different instruments.

Table 1 reports the overall detection accuracy of the segmentation model. These values were computed by comparing the instruments identified by the model for a set of 30 sampled video frames with human expert "ground truth" annotations. The values are interpreted as follows. True Positive (TP) means the model correctly predicts an instrument is present when it actually is present; False Positive (FP) means the model predicts an instrument is present when it is not present; True Negative (TN) means the model correctly predicts an instrument is not present when it's not present; False Negative (FN) means the model fails to detect an instrument that is actually present. Precision indicates how many of the tools the model predicted as present were actually present. Recall indicates how many of the instruments actually present did the model successfully detect. Finally, the F1-Score combines precision and recall using their harmonic mean. We believe this level of performance is sufficient for monitoring the property used in the experiment; however, further studies are necessary to determine the level of false positives surgeons find acceptable.[2]

The maximum time required to process a frame of video in the experiment, including both segmentation and run-time verification, was 36 milliseconds, and the average processing time was 31.25 ms. This suggests that, for the segmentation model and property used, real time monitoring for property violations is possible at the frame rate of the vDSS video feed. The time required to process a frame of video was dominated by the segmentation model: the maximum time required to check the property using NuRV was 6.1 ms. While the time required for segmentation is a significant fraction of the 40 ms available to process each frame, the current implementation uses only one core of the graphics card, and using more cores and/or a more powerful GPU is likely to reduce the time required for segmentation. In addition, the current implementation assumes that both segmentation and run-time verification occur in the time available to process a single frame. It may be acceptable to relax this assumption; for example, run-time verification could take as input the segmentation produced at the previous frame; however, this requires further study to determine if the delay in detecting a violation is acceptable.

5. Conclusions and Future Work

We presented the results of a study to investigate the feasibility of intraoperative warning systems based on run-time monitoring. A survey of expert and novice surgeons confirmed the clinical relevance of warnings of property violations which could result in adverse events during surgery. Although preliminary, our proof-of-concept experiment demonstrated the feasibility of coupling advanced neural segmentation with existing runtime verification tools to improve intraoperative safety. Many challenges remain. While we expect the time required to detect anatomical structures to be similar to that for instrument detection (as the neural model inherently predicts all relevant classes simultaneously), improving reliability of the image segmentation component may be necessary to reduce the level of

[2]The relatively low precision when detecting the suction & irrigation tool may result in too many "false" warnings, but this tool is less likely to cause damage to tissues when out of camera view.

false positives to levels that surgeons find acceptable. Another direction for future work is developing mitigation techniques to handle potential transient "hallucinations" or misclassifications produced by the neural network when the type of surgery and/or surgical instruments differ from the training data. We also plan to investigate the suitability of other run-time monitoring tools for intraoperative monitoring, including, for example, the LoLa family of tools [9] and RTAMT [10].

Acknowledgement We gratefully acknowledge funding by NWO project Run-time verification for robot-assisted surgery, OCENW.M.21.377.

References

[1] M. Watanabe, K. Kuriyama, M. Terayama, A. Okamura, J. Kanamori, Y. Imamura, Robotic-assisted esophagectomy: Current situation and future perspectives, Annals of thoracic and cardiovascular surgery : official journal of the Association of Thoracic and Cardiovascular Surgeons of Asia 29 (2023) 168–176. doi:10.5761/atcs.ra.23-00064.

[2] M. Watanabe, K. Kuriyama, M. Terayama, A. Okamura, J. Kanamori, Y. Imamura, Robotic-assisted esophagectomy: Current situation and future perspectives, Annals of thoracic and cardiovascular surgery : official journal of the Association of Thoracic and Cardiovascular Surgeons of Asia 29 (2023) 168–176. doi:10.5761/atcs.ra.23-00064.

[3] R. Nagyné Elek, T. Haidegger, Non-technical skill assessment and mental load evaluation in robot-assisted minimally invasive surgery, Sensors (Basel, Switzerland) 21 (2021). doi:10.3390/s21082666.

[4] K. Gogoladze, N. Alechina, Z. J. Hu, H. Xu, R. C. van Jaarsveld, J. P. Ruurda, Run-time monitoring for robot-assisted surgery, in: D. Porello, C. Vinci, M. Zavatteri (Eds.), Short Paper Proceedings of the 6th International Workshop on Artificial Intelligence and Formal Verification, Logic, Automata, and Synthesis, OVERLAY 2024, Bolzano, Italy, November 28-29, 2024, volume 3904 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024, pp. 127–133. URL: https://ceur-ws.org/Vol-3904/paper16.pdf.

[5] T. J. M. Jaspers, R. L. P. D. de Jong, Y. Li, C. H. J. Kusters, F. H. A. Bakker, R. C. van Jaarsveld, G. M. Kuiper, R. van Hillegersberg, J. P. Ruurda, W. M. Brinkman, J. P. W. Pluim, P. H. N. de With, M. Breeuwer, Y. A. Khalil, F. van der Sommen, Scaling up self-supervised learning for improved surgical foundation models, 2025. URL: https://arxiv.org/abs/2501.09436. arXiv:2501.09436.

[6] W. Yu, C. Si, P. Zhou, M. Luo, Y. Zhou, J. Feng, S. Yan, X. Wang, Metaformer baselines for vision, IEEE Transactions on Pattern Analysis and Machine Intelligence 46 (2023) 896–912.

[7] R. L. P. D. de Jong, Y. al Khalil, T. J. M. Jaspers, R. C. van Jaarsveld, G. M. Kuiper, Y. Li, R. van Hillegersberg, J. P. Ruurda, M. Breeuwer, F. van der Sommen, Benchmarking pretrained attention-based models for real-time recognition in robot-assisted esophagectomy, 2024. URL: https://arxiv.org/abs/2412.03401. arXiv:2412.03401.

[8] A. Cimatti, C. Tian, S. Tonetta, Assumption-based runtime verification with partial observability and resets, in: B. Finkbeiner, L. Mariani (Eds.), Runtime Verification, Springer International Publishing, Cham, 2019, pp. 165–184.

[9] B. D'Angelo, S. Sankaranarayanan, C. Sánchez, W. Robinson, B. Finkbeiner, H. B. Sipma, S. Mehrotra, Z. Manna, LOLA: runtime monitoring of synchronous systems, in: 12th International Symposium on Temporal Representation and Reasoning (TIME 2005), IEEE Computer Society, 2005, pp. 166–174.

[10] T. Yamaguchi, B. Hoxha, D. Ničković, RTAMT – Runtime Robustness Monitors with Application to CPS and Robotics, International Journal on Software Tools for Technology Transfer 26 (2024) 79–99. doi:10.1007/s10009-023-00720-3.