Automatic annotation of surgical activities using virtual reality environments

Arnaud Huaulmé, Fabien Despinoy, Saul Alexis Heredia Perez, Kanako Harada, Mamoru Mitsuishi, Pierre Jannin

Noname manuscript No.
(will be inserted by the editor)

Automatic annotation of surgical activities using virtual reality environments

Arnaud Huaulmé · Fabien Despinoy · Saul Alexis Heredia Perez · Kanako Harada · Mamoru Mitsuishi · Pierre Jannin

Received: date / Accepted: date

Abstract
Purpose Annotation of surgical activities becomes increasingly important for many recent applications such as surgical workflow analysis, surgical situation awareness and the design of the operating room of the future, especially to train machine learning methods in order to develop intelligent assistance. Currently, annotation is mostly performed by observers with medical background and is incredibly costly and time-consuming, creating a major bottleneck for the above-mentioned technologies. In this paper, we propose a way to eliminate, or at least limit, the human intervention in the annotation process.
Methods Meaningful information about interaction between objects is inherently available in virtual-reality environments. We propose a strategy to convert automatically this information into annotations in order to provide as output individual surgical process models.
Validation We implemented our approach through a peg-transfer task simulator and compared it to manual annotations. To assess the impact of our contribution, we studied both intra- and inter-observer variability.
Results and conclusion In average, manual annotations took more than 12 minutes for one minute of video to achieve low-level physical activity annotation whereas automatic annotation is achieved in less than a second for the same video period. We also demonstrated that manual annotation introduced mistakes as well as intra- and inter-observer variability that our method is able to suppress due to the high precision and reproducibility.

Keywords Automatic annotation · Surgical process model · Surgical simulation

This work was funded by ImPACT Program of Council for Science, Technology and Innovation, Cabinet Office, Government of Japan.

A. Huaulmé · F. Despinoy · P. Jannin
Univ Rennes, INSERM, LTSI - UMR 1099, F35000, Rennes, France
E-mail: arnaud.huaulme@univ-rennes1.fr

S. A. Heredia Perez · K. Harada · M. Mitsuishi
Department of Mechanical Engineering, the University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

1 Introduction

Recent developments in computer-assisted surgical (CAS) systems rely on explicit understanding of the surgical procedure through the use of surgical process models (SPMs). SPM methodology may help surgical learning and expertise assessment [1, 2], operating room optimization and management [3, 4], robotic assistance [5] and also decision support [6].

A SPM is a description of a surgical procedure at several levels of granularity: phases, steps and activities [7] along with surgemes and dexemes[8]. A SPM breaks down the surgical procedure into a succession of phases corresponding to the main periods of the intervention (e.g. abdominal closure). Each phase is composed of one or more steps corresponding to a surgical objective (e.g. resect the pouch of Douglas). A step is composed of a sequence of activities which describe the physical actions performed by an actor. An activity is broken down into different components: the action verb (e.g. cut), the target involved in the action (e.g. the pouch of Douglas) and the surgical instrument used to perform the action (e.g. a scalpel). Lower granularity levels are closer to kinematic data, such as surgemes and dexemes. A surgeme is defined as a surgical motion with explicit semantic meaning (e.g. grab), where, a dexeme is a numerical representation of the performed physical motion (e.g. go left). A SPM is usually acquired manually thanks to human observers [9]. Recent advances in machine and specifically deep learning suggest limiting human intervention in SPM acquisition. For example, automatic recognition methods have been studied for phases [10, 11], steps [12, 13] or activities [5, 14]. However, most of the proposed methods require prior manual annotations for training purposes.

Manual annotations of surgical procedures are mostly performed by observers with medical background where their efforts are incredibly costly, time-consuming and could bring variability into the data because of the subjective nature of the task. Use of public data-set such as JIGSAW [15], DIONE [16] or Cholec80 [11] may reduce the number of needed annotations by relying on transfer learning strategies for instance. Other strategies are currently studied to reduce the amount of manual annotations by learning from weakly annotated data (e.g. Auto-Encoder) or generating data from existing ones (e.g. Generative Adversarial Networks (GAN)). Recently, Zisimopoulos et al. [17] demonstrated the feasibility to train a neural networks from manually annotated surgical simulated data and validated it on a real data-set. Following such idea, we hypothesize it is possible to develop an automatic recognition method trained on simulated data.

In this paper, we propose a new approach for automatic generation of SPM from a simulated environment. We also compare performances of manual annotations from various observers with automatic annotation that we developed for this purpose.

2 Material and Method

The proposed method, called ASURA (Automatic SimUlatoR Annotator) is summarized in figure 1. It consists in converting information provided by a virtual-reality (VR) based simulator into a SPM at different granularity levels: phases,

steps, and activities. The final transcription is processed, after filtering, based on rules described in a configuration file.

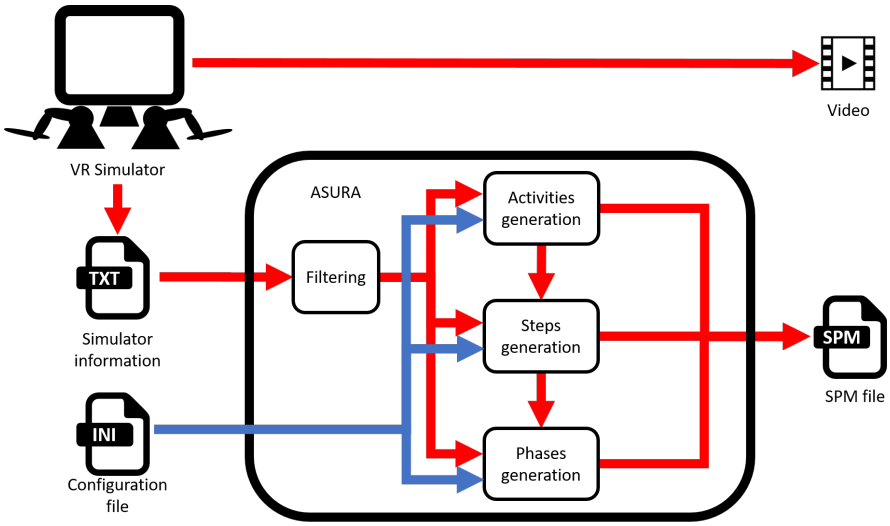

Fig. 1: Automatic SimUlatoR Annotator (ASURA) workflow.

We propose to use VR scene information provided by VR based simulators, including pose of each VR object, collision between them, as well as kinematic information. With this information, we extract boolean "flags", called "simulator information", for each time-step. These flags are set to true if the conditions they represent occur (e.g. collision between objects A and B).

2.1 Pre-filtering

To limit artifacts due to the noise in simulator information, signals are filtered. All signals being boolean, the new binary value at instant t is defined thanks to a majority voting in the window $[t-N;t+N]$, where N is an user-defined parameter.

2.2 Activities, steps and phases generation

The configuration file is used to the different rules for transcription. It is composed of:

- the meaning and type of each information from the simulator information file;
- the vocabulary describing the performed task;
- the rules allowing the transcription into a SPM.

Using this configuration file, the software is able to interpret, at each time step, the binary filtered data into pseudo-activities. A pseudo-activity is an activity which only takes into account the interaction between objects without any

contextual information. To add this contextual information, the software interprets additional information such as present and past others filtered flags, and the past of the activity sequence.

Figure 2 illustrates this process with an example. The configuration file allows interpretation of the flags relative to the collisions with the left grasper ("0 1 0 0") as the pseudo-activity "tool-tip-1 of left grasper touch object B". Without any contextual information, it is not possible to completely understand the meaning of this pseudo-activity. The contextual information is composed by the filtered flag relative to the closure of left grasper and the activity at $t-1$. We see a modification of the closure flag from true at $t-1$ to false at t and the previous activity consists of holding the object B, i.e. both tool-tips were in contact with this object. From this information: opening of the grasper, previously two contacts with object B and now only one, we could deduce that the current activity is "left grasper drops the object B".

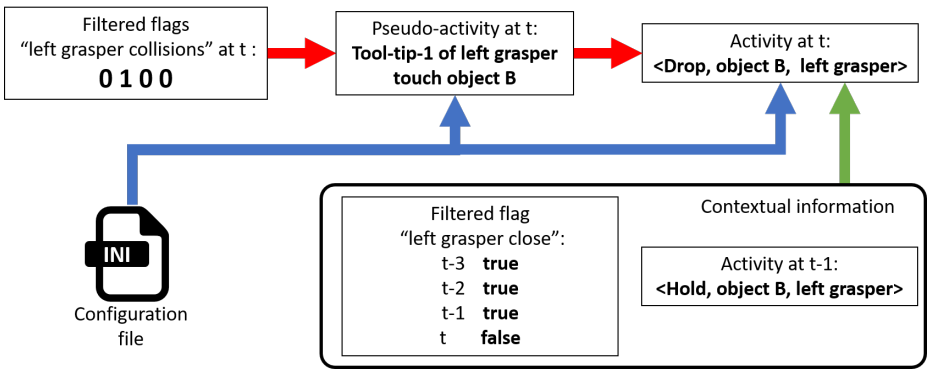

Fig. 2: Example of activity generation at instant t. The configuration file allows the interpretation of the filtered information into a pseudo-activity. Then, the pseudo-activity is converted into activity thanks to contextual information.

As a reminder, a step is a sequence of activities corresponding to a surgical objective. We deduce the current step using both the current activities and the filtered information. The rules define the conditions of the beginning and the ending of each step according to activities and scene information. For instance, one step of the peg-transfer task consists in the transfer of one block from the left to right (see subsection 3.2) and could be characterized as follows:

- Start: first activity concerning the block to transfer;
- End: the block is correctly place at the right part of the board.

The same principle is used for phase annotation, where a phase is composed by steps and deduced from current step and filtered information.

3 Validation

We compare performances of automatic and manual annotations in the context of a peg-transfer task performed on a VR simulator.

3.1 Peg-transfer simulator

The VR simulator (figure 3) used for the data acquisition was developed at the department of mechanical engineering at the University of Tokyo [18] and is composed of:

- A core laptop (i7-700HQ, 16Go RAM, GTX 1070);
- A 3D rendering setup: 3D screen (24 inches, 144Hz) and 3D glasses;
- Two haptic user interfaces.

A non-medical participant familiarized with the simulator performed five sessions of the peg-transfer task on the VR simulator. The mean duration of the sessions was 178±24 seconds (table 1).

Table 1: Duration of peg-transfer sessions.

session	1	2	3	4	5	Mean
Duration (s)	216	181	163	175	155	178±24

For each session, the simulator provide video and simulator information as output, synchronized at 30Hz.

Fig. 3: VR simulator setup.

3.2 Validation data

The peg-transfer task [19] consists in transferring six colored objects, called blocks, from left to right first and then reverse through bimanual manipulations. We

identified two phases, twelve steps, six action verbs, two targets, and one surgical instrument (Table 2). Each phase corresponds to one transferring direction. Each step (six by phase) corresponds to the transfer of one block in one direction (e.g. "Block1 L2R" corresponds to the transfer of the first block from the left to the right). For the activities, we differentiated two targets: "block" and "other block". The "block" corresponds to the one which is currently transferred. "Other block" is an additional target used to differentiate when the user accidentally interacts with another block.

Table 2: Peg-transfer vocabulary.

Phases	Steps
Transfer Left To Right (L2R)	Block 1 L2R Block 2 L2R Block 3 L2R Block 4 L2R Block 5 L2R Block 6 L2R
Transfer Right To Left (R2L)	Block 1 R2L Block 2 R2L Block 3 R2L Block 4 R2L Block 5 R2L Block 6 R2L

Activities		
Verb	Target	Tool
Catch Drop Extract Hold Insert Touch	Block Other block	Grasper

3.3 Surgical process model annotations

Seven observers performed manual annotations with the "Surgery Workflow Toolbox [annotate]" software [20] based on the videos provided by the simulator. One observer performed six annotations of each session to highlight intra-observer variability. The six others performed one annotation of each session to highlight inter-observer variability. Each observer was previously trained to the annotation software and learnt a complete description of each phase, step, and activity of the peg-transfer task. To limit variability due to annotation mistakes, each annotation was corrected retrospectively by an expert. As described in table 3, 18 mistakes were done by the observer for the intra-observer variability study and 47 for the inter-observer variability study. These corrections focused on vocabulary mistakes (such as wrong target or wrong step) and left-right inversion only. Other characteristics such as duration, numbers of occurrences, and observer interpretations were not modified.

Session id	1	2	3	4	5	**All**
Intra-observer variability study mistakes	9	4	1	1	3	**18**
Inter-observer variability study mistakes	16	4	11	12	4	**47**

Table 3: Number of annotation mistakes corrected by an expert

Automatic annotations were generated following the described method using a desktop computer (E5-1607 v3 @ 3.10 GHz with 8Go RAM). The parameter N used for the filtering window was set to 5. For each sessions, 100 automatic annotations were generated to measure the variability of the computing time. Table 4 summarizes the configuration for each type of annotations.

Table 4: Configuration details for each annotations type.

Annotation type	Manual		Automatic
	Intra-variability	Inter-variability	
Observer id	1	2-7	none
Number of annotation(s) for one session and one observer	6	1	100
Total number of annotations	30	30	500

3.4 Validation metrics

To monitor consistency of each annotation type, we focused our analysis on the following metrics:

- Time required to annotate one minute of video with the annotation software or by the automatic annotation;
- Duration to achieves phases, steps, and actions according to observers' interpretation, i.e. the duration took to perform, on the simulator, the task according to observers' surgical process model annotations.

For each metric and each session, we computed both mean and relative standard deviation (RSD). To highlight statistical differences between results, we performed ANOVA tests were the hypothesis is validated for a p-value inferior to 0.01.

4 Results

This section describes only the most relevant results. The entire analysis is available as supplementary material.

4.1 Manual annotations: intra-observer variability study

Table 5 presents the most relevant results for intra-observer variability computed thanks to six annotations of each session.

The average annotation duration for one minute of video is 11.8 minutes with a (RSD) of 16.03%.

Phases, respectively "transfer left to right" and "transfer right to left", have a mean duration of 85.4 and 86.4 seconds with a RSD of 0.37% and 0.10%. Mean

Table 5: Results from manual annotation for intra-observer variability study. *L2R: Left to Right; R2L: Right to Left.*

Session id		1	2	3	4	5	all
Time required to annotate 1 min. of video (min.)		11.5 ±21.13%	10.9 ±18.80%	12.5 ±10.00%	11.9 ±13.06	12.5 ±16.22%	**11.8 ±16.03%**
Mean duration to achieve the task (s)							
Phases	Transfer L2R	92.1 ±0.24%	100.6 ±0.20%	73.1 ±0.90%	88.0 ±0.15%	73.3 ±0.36%	**85.4 ±0.37%**
	Transfer R2L	115.2 ±0.10%	76.3 ±0.10%	84.2 ±0.06%	82.9 ±0.13%	73.6 ±0.36%	**86.4 ±0.10%**
Steps	Block 1 L2R	16.1 ±0.95%	25.6 ±0.79%	12.1 ±5.69%	21.4 ±1.73%	19.1 ±1.26%	**18.9 ±2.08%**
	Block 1 R2L	13.6 ±1.20%	13.8 ±0.63%	13.4 ±0.66%	13.9 ±1.00%	10.7 ±0.91%	**13.1 ±0.88%**
Action verbs	Catch	1.2 ±8.02%	0.9 ±8.79%	1.1 ±9.22%	1.1 ±9.78%	0.9 ±10.56%	**1.0 ±9.27%**
	Hold	3.4 ±4.65%	3.3 ±2.35%	2.9 ±4.65%	3.3 ±1.42%	3.1 ±1.67%	3.2 ±2.95%
	Touch	0.5 ±29.94%	0.0 ±0.00%	0.4 ±9.54%	0.2 ±40.37%	0.5 ±21.95%	**0.3 ±20.36%**
Average of occurrences for action verbs							
Action verbs	Catch	30.5 ±4.97%	25.7 ±5.87%	27.2 ±2.77%	30.3 ±3.99%	25.2 ±5.85%	**27.8 ±4.69%**
	Hold	26.7 ±3.06%	26.0 ±2.43%	24.3 ±2.12%	25.0 ±0.00%	24.0 ±0.00%	25.2 ±1.52%
	Touch	3.7 ±22.27%	**0.0 ±0.00%**	2.0 ±0.00%	1.2 ±34.99%	2.5 ±33.47%	**1.9 ±18.15%**

duration of step "block 1 L2R" is 18.9 seconds with a RSD of 2.08%. For step "block 1 R2L", it is 13.1 seconds with a RSD of 0.88%.

Action verb "catch" has on average $27.8 \pm 4.69\%$ occurrences by session with an average duration of $1 \pm 9.27\%$ second. The action verb "touch" is not always present in sessions (e.g. in session 2 there is no occurrence). On average, there is only $1.9 \pm 18.15\%$ occurrences of this action verb for a average duration of 0.3 second with a RSD of 20.36%.

4.2 Manual annotations: inter-observer variability study

Table 6 presents the most relevant results for inter-observer variability computed using one annotation for each session performed by six observers.

The average annotation duration for one minute of video is 12.7 minutes with a RSD of 17.25%.

Phases, "transfer left to right" and "transfer right to left", have respectively a mean duration of 85.2 and 86.3 seconds with a RSD of 0.79% and 0.45%. Mean duration of step "block 1 L2R" is 18.6 seconds with a RSD of 3.83%. For step "block 1 R2L", it is 13.1 seconds with a RSD of 2.68%.

Table 6: Results from manual annotation for inter-observer variability study. *L2R: Left to Right; R2L: Right to Left.*

Session id		1	2	3	4	5	all
Time required to annotate 1 min. of video (min.)		14.5 ±21.74%	11.0 ±13.40%	12.3 ±12.10%	13.6 ±17.00	11.6 ±19.78%	**12.7 ±17.25%**
Mean duration to achieve the task (s)							
Phases	Transfer L2R	91.8 ±0.91%	100.3 ±0.49%	72.9 ±1.60%	87.8 ±0.44%	72.9 ±0.53%	**85.2 ±0.79%**
	Transfer R2L	115.0 ±0.64%	76.2 ±0.16%	84.2 ±0.61%	82.8 ±0.60%	73.5 ±0.22%	**86.3 ±0.45%**
Steps	Block 1 L2R	15.9 ±2.60%	25.6 ±3.17%	11.3 ±4.67%	20.9 ±6.68%	19.1 ±2.02%	**18.6 ±3.83%**
	Block 1 R2L	13.5 ±2.60%	13.9 ±3.17%	13.5 ±4.67%	13.7 ±6.68%	10.8 ±2.02%	**13.1 ±2.68%**
Action verbs	Catch	1.0 ±23.57%	0.7 ±15.13%	0.9 ±18.65%	1.0 ±14.21%	0.8 ±21.68%	**0.9 ±18.65%**
	Hold	3.5 ±6.40%	3.6 ±11.49%	2.9 ±5.25%	3.3 ±4.36%	3.0 ±3.23%	3.3 ±6.15%
	Touch	0.5 ±46.33%	0.0 ±0.00%	0.8 ±56.51%	1.0 ±36.02%	0.4 ±21.96%	**0.5 ±32.16%**
Average of occurrences for action verbs							
Action verbs	Catch	30.5 ±5.77%	26.5 ±6.20%	28.0 ±3.19%	29.33 ±3.52%	25.5 ±5.41%	**28.0 ±4.82%**
	Hold	26.2 ±1.56%	25.0 ±3.58%	24.0 ±0.00%	24.2 ±1.69%	24.0 ±0.00%	24.7 ±1.37%
	Touch	3.7 ±14.08%	**0.0 ±0.00%**	2.2 ±67.94%	2.5 ±51.64%	2.3 ±22.13%	**2.1 ±31.16%**

Action verb "catch" has on average $28.0 \pm 4.82\%$ occurrences by session with an average duration of $0.9 \pm 18.65\%$ second. As for the intra-observer variability study, the action verb "touch" has no occurrence in session 2. On average, there was only $2.1 \pm 31.16\%$ occurrences of this action verb for a duration of 0.5 second with a RSD of 32.16%.

4.3 Automatic annotation

Table 7 presents the most relevant results for automatic annotation. The mean computation time, measured over 100 repetitions of each session, to generate annotation corresponding to one minute of video was 700 milliseconds with a RSD of 0.44%.

Phases, respectively "transfer left to right" and "transfer right to left", has a mean duration of 85 and 86 seconds. Mean duration of step "block 1 L2R" is 19 seconds and 13 seconds for step "block 1 R2L". Around 30.8 actions with verb "catch" are present by session with a mean duration of 0.6 second. With automatic annotation, the action verb "touch" is never present in the sessions.

Except computation time, there is no variation within the 100 automatic annotations.

Table 7: Results from automatic annotation. *L2R: Left to Right; R2L: Right to Left.*

Session id		1	2	3	4	5	all
Time required to annotate 1 min. of video (ms)		698 ±0.68%	700 ±0.33%	700 ±0.38%	701 ±0.35%	702 ±0.40%	**700 ±0.44%**
Mean duration to achieve the task (s)							
Phases	Transfer L2R	91.9	101.0	73.2	88.2	72.6	**85.4**
	Transfer R2L	115.2	75.7	83.8	82.6	73.5	**86.2**
Steps	Block 1 L2R	16.1	25.6	11.9	22.0	19.0	**18.9**
	Block 1 R2L	13.9	13.8	13.5	13.9	11.4	**13.3**
Action verbs	Catch	0.6	0.5	0.7	0.7	0.5	**0.6**
	Hold	3.1	3.2	3.1	3.4	3.0	3.2
	Touch	0	0	0	0	0	0
Average of occurrences for action verbs							
Action verbs	Catch	38	25	31	32	28	**30.8**
	Hold	30	28	24	25	24	26.2
	Touch	**0**	**0**	**0**	**0**	**0**	**0**

4.4 Automatic versus manual annotations

The most relevant results of the statistical analysis are shown on table 8.

One of the main differences between automatic and manual annotations is the time required to perform the annotations. As shown in figure 4, manual annotation is 700 times more time-consuming than automatic annotation ($p<0.0001$). The differences in annotation duration between the intra-observer and inter-observer variability studies are not statistically significant ($p=0.2058$).

For both phases and steps the mean duration is independent of the type of annotation, e.g. the mean duration of phase "transfer left to right" is between 85.2 and 85.4 seconds. Even if inter-observer variability are more important than intra-observer one, differences are not significant (table 8), e.g. 2.68% vs. 0.88% for the duration of "block 1 L2R" with $p=0.8225$.

For action verb duration, differences are statistically significant between all studies for activities shorter than one second ("catch" and "touch"), and between automatic and inter-observer variability studies for the action verb "hold". For action verb occurrences, the differences were only significant for "catch" between automatic and manual annotations, and for "hold" between automatic annotation and inter-observer variability study.

Table 8: p-value between automatic and manual annotations (intra-observer and inter-observer variability) with ANOVA test. *L2R: Left to Right; R2L: Right to Left; Na: not applicable ; ∗: statistically significant.*

p-value		automatic vs. intra	automatic vs. inter	intra vs. inter
Time required to annotate 1 min. of video		**<0.0001** ∗	**<0.0001** ∗	**0.2058**
Mean duration to achieve the task				
Phases	Transfer L2R	0.9732	0.9217	0.9235
	Transfer R2L	0.9233	0.9482	0.9820
Steps	Block 1 L2R	0.9386	0.6942	**0.8225**
	Block 1 R2L	0.2736	0.2703	0.9964
Action verbs	Catch	**<0.0001** ∗	**<0.0001** ∗	**0.0005** ∗
	Hold	0.1508	**0.0025**∗	0.4964
	Touch	Na	Na	**0.0062** ∗
Average of occurrences for action verbs				
Action verbs	Catch	**0.0002** ∗	**0.0005** ∗	0.7517
	Hold	0.0240	**0.0005** ∗	0.0531
	Touch	Na	Na	0.7907

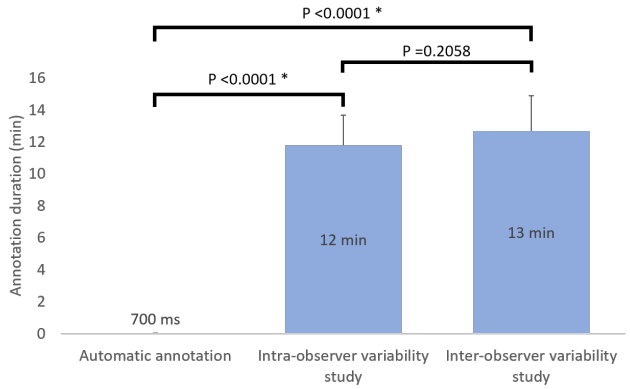

Fig. 4: Time required to annotate one minute of video for both automatic and manual annotations.

5 Discussion

In this paper, we propose a new automatic strategy to convert information provided by VR simulator into annotations and resulting individual surgical process

models. We compare the performances of automatic annotations to manual ones. Even if we validate this strategy with a peg-transfer simulator, it could be used with any VR environment, even a more complex one with multiple surgical tools and targets, as long as it is able to provide object collisions and kinematic data.

Manual annotations have several limitations. First, it is time-consuming, around 12 minutes are required to annotate one minute of video. Secondly, it could introduce annotation mistakes (table 3). Into our study, the most important part of the mistakes are due to multiple targets annotations for activities ("block" and "other block") or missing instrument that are mainly caused by the software design. Indeed, it is possible to have multiple targets for one activity and it is required to specify each element, even if only one is available. Others errors are due to inattention or to inversion between left and right. We also noticed a decreasing of errors along the learning of the software. For inter-observer variability study, all observer performed annotations on the following order: sessions 1, 4, 3, 2 and 5. The number of mistakes decreased progressively with this order (respectively 16, 12, 11, 4, and 4 mistakes). Thirdly, manual annotation introduces variability. This variability is more important if several observers are involved in the process; most of the RSD are superior for the inter-observer variability study (table 6) than for the intra-observer one (table 5). This variability is not significant for the granularity level of phase and step, but is for activities and especially for their duration. Indeed, for activities shorter than one second ("catch" and "touch"), the RSD was superior to 9% on the intra-observer variability study and superior to 18% on the inter-observer one. Differences in the number of occurrences are not significant (respectively p=0.7517 and p=0.7907) whereas the duration differences are (p=0.0005 for "catch" and p=0.0062 for "touch") and could be the consequence of different causes such as:

- Each observer has its own interpretation of the beginning and the end of each activity, even if a precise description is provided;
- The software used to annotate is not able to navigate frame by frame within the video. It could be difficult for observers to start or stop activity at the exact transition frame.

Automatic annotation is 700 less time-consuming than manual one and is very reproducible. However, automatic annotation is restricted by the available information in the VR simulator. As shown on table 7, the automatic annotation is not able to detect any action verb "touch". This is due to the available flags which do not allow distinction between catching and touching actions. Indeed, both actions start when a tool-tip is in contact with a block but are differentiated by the user intention. This intention is reflected by the side of the tool-tip in contact with the block. If it is the interior side, the intention is to catch the block, in other cases is to touch it. Currently, available flags do not differentiate tool-tip sides, resulting in more occurrences of "catch". We are currently working on this issue. Moreover, since "touch" actions are very short, the mean duration is impacted by this miss-detection. This could explain the significant differences (table 8) between automatic and manual annotations for "catch" verb (number of occurrences and duration).

In addition to the annotation duration, automatic annotation requires effort on simulator development and configuration file creation. Nevertheless, the aim of this work is to propose a way to generate data thanks to existing simulators, thus

the efforts to develop simulator are not inherent to this work. On the other hand, the creation of the configuration file for automatic annotations requires similar effort than the task definition for manual ones. In both cases, vocabulary and rules are necessary. Automatic annotation requires rules definition in accordance with available flags. These additional efforts are only realized once, whereas efforts for manual annotation is achieved for each new session. Obviously, the amount of work is also proportional to the number of actions involved in the simulated task. When simulators address complex and more realistic tasks with a lot of different actions, the workload proportionally increases for creating the configuration file, as well as for manual annotation.

Surprisingly, for action verb "hold", the differences in term of duration and number of occurrences are significant for automatic annotation versus inter-observer variability study (respectively, 0.0025 and 0.0005) but not for other configurations. This could be explained by:

- The fact that, as for activities shorter than one second, each observer had its own interpretation of the beginning and the end of each activity;
- The observer who participated to the intra-observer variability study is the person who defined the rules for both types of annotation (automatic and manual).

Thus, it makes sense that the observer of this intra-observer variability study has an annotation behavior closer to the automatic one than the observers of the inter-observer variability study.

6 Conclusion

Automatic annotation has multiple advantages compared to manual. It is faster, accurate and not subject to any variability. Requiring collision and kinematic data only, it could capture actions even if the field of view is blocked or if they take place outside of it. Automatic annotation could also be computed online with a delay of N samples (parameter used for flag filtering) which eases the data capture during surgical training. Moreover, if all activities could be interpreted by the provided flags, it does not introduce any mistake into the annotations as regard to human-based observations. From the authors perspectives, easily providing semantic information with raw data (e.g. images, kinematics, etc.) from virtual environment helps to understand surgical behavior with a dual objective: improve pedagogical guidance for trainees and also generate meaningful dataset for future artificial intelligence development in the context of computer-assisted surgery.

Conflict of Interest

The authors declare that they have no conflict of interest.

Ethical approval

All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research

committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards.

Informed consent

This articles does not contain patient data.

Acknowledgements
This work was funded by ImPACT Program of Council for Science, Technology and Innovation, Cabinet Office, Government of Japan.
Authors thanks the IRT b<>com for the provision of the software "Surgery Workflow Toolbox [annotated]", used for this work.
Authors especially thank Ms. M. Le Duff, Mr. A. Derathé, Mr. T. Dognon, Mr. E. Maguet and Mr. B. Ndack for their help to the data annotation.

References

1. A. Huaulmé, K. Harada, G. Forestier, M. Mitsuishi, and P. Jannin. Sequential surgical signatures in micro-suturing task. *International Journal of Computer Assisted Radiology and Surgery*, 13(9):1–10, May 2018. ISSN 1861-6410, 1861-6429. doi: 10.1007/s11548-018-1775-x.
2. G. Forestier, L. Riffaud, F. Petitjean, P.-L. Henaux, and P. Jannin. Surgical skills: Can learning curves be computed from recordings of surgical activities? *International Journal of Computer Assisted Radiology and Surgery*, 13(5):629–636, May 2018. ISSN 1861-6410, 1861-6429. doi: 10.1007/s11548-018-1713-y.
3. W. S. Sandberg, B. Daily, M. Egan, J. E. Stahl, J. M. Goldman, R. A. Wiklund, and D. Rattner. Deliberate Perioperative Systems Design Improves Operating Room Throughput:. *Anesthesiology*, 103(2):406–418, August 2005. ISSN 0003-3022. doi: 10.1097/00000542-200508000-00025.
4. B. Bhatia, T. Oates, Y. Xiao, and P. Hu. Real-time identification of operating room state from video. volume 2, pages 1761–1766, 2007.
5. S.-Y. Ko, J. Kim, W.-J. Lee, and D.-S. Kwon. Surgery task model for intelligent interaction between surgeon and laparoscopic assistant robot. *International Journal of Assitive Robotics and Mechatronics*, 8(1):38–46, 2007.
6. G. Quellec, M. Lamard, B. Cochener, and G. Cazuguel. Real-Time Task Recognition in Cataract Surgery Videos Using Adaptive Spatiotemporal Polynomials. *IEEE Transactions on Medical Imaging*, 34(4):877–887, April 2015. ISSN 0278-0062. doi: 10.1109/TMI.2014.2366726.
7. F. Lalys and P. Jannin. Surgical process modelling: a review. *International Journal of Computer Assisted Radiology and Surgery*, 9(3):495–511, September 2013.
8. F. Despinoy, D. Bouget, G. Forestier, C. Penet, N. Zemiti, P. Poignet, and P. Jannin. Unsupervised trajectory segmentation for surgical gesture recognition in robotic training. *IEEE Transactions on Biomedical Engineering*, 63(6): 1280–1291, 2015.
9. T. Neumuth, R. Wiedemann, C. Foja, P. Meier, J. Schlomberg, D. Neumuth, and P. Wiedemann. Identification of surgeonindividual treatment profiles to

support the provision of an optimum treatment service for cataract patients. *Journal of Ocular Biology, Diseases, and Informatics*, 3(2):73–83, June 2010.
10. N. Padoy, T. Blum, S.-A. Ahmadi, H. Feussner, M.-O. Berger, and N. Navab. Statistical modeling and recognition of surgical workflow. *Medical Image Analysis*, 16(3):632–641, 2010.
11. 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*, 36(1):86–97, January 2017. ISSN 0278-0062. doi: 10.1109/TMI.2016.2593957.
12. L. Bouarfa, P. P. Jonker, and J. Dankelman. Discovery of high-level tasks in the operating room. *Journal of Biomedical Informatics*, 44(3):455–462, June 2011.
13. A. James, D. Vieira, B. Lo, A. Darzi, and G.-Z. Yang. Eye-Gaze Driven Surgical Workflow Segmentation. *Medical Image Computing and Computer-Assisted Intervention MICCAI 2007*, pages 110–117, October 2007.
14. F. Lalys, D. Bouget, L. Riffaud, and P. Jannin. Automatic knowledge-based recognition of low-level tasks in ophthalmological procedures. *International Journal of Computer Assisted Radiology and Surgery*, 8(1):39–49, April 2012.
15. Y. Gao, S. S. Vedula, C. E. Reiley, N. Ahmidi, B. Varadarajan, H. C. Lin, L. Tao, L. Zappella, B. Bejar, D. D. Yuh, C. C. G. Chen, R. Vidal, S. Khudanpur, and G. D. Hager. JHU-ISI Gesture and Skill Assessment Working Set (JIGSAWS): A Surgical Activity Dataset for Human Motion Modeling. *Modeling and Monitoring of Computer Assisted Interventions (M2CAI) MICCAI Workshop*, page 10, 2014.
16. D. Sarikaya, J. J. Corso, and K. A. Guru. Detection and Localization of Robotic Tools in Robot-Assisted Surgery Videos Using Deep Neural Networks for Region Proposal and Detection. *IEEE Transactions on Medical Imaging*, 36 (7):1542–1549, July 2017. ISSN 0278-0062. doi: 10.1109/TMI.2017.2665671.
17. O. Zisimopoulos, E. Flouty, M. Stacey, S. Muscroft, P. Giataganas, J. Nehme, A. Chow, and D. Stoyanov. Can surgical simulation be used to train detection and classification of neural networks? *Healthcare Technology Letters*, 4(5):216–222, September 2017. ISSN 2053-3713. doi: 10.1049/htl.2017.0064.
18. S.A. Heredia Perez, K. Harada, and M. Mitsuishi. Haptic Assistance for Robotic Surgical Simulation. *27th Annual Congress of Japan Society of Computer Aided Surgery*, 20(4):232–233, November 2018.
19. A. M. Derossis, G. M. Fried, M. Abrahamowicz, H. H. Sigman, J. S. Barkun, and J. L. Meakins. Development of a model for training and evaluation of laparoscopic skills. *American Journal of Surgery*, 175(6):482–487, June 1998. ISSN 0002-9610.
20. C. Garraud, B. Gibaud, C. Penet, G. Gazuguel, G. Dardenne, and P. Jannin. An Ontology-based Software Suite for the Analysis of Surgical Process Model. In *Proceedings of Surgetica'2014*, pages 243–245. Chambery, France, December 2014.