Scenes extraction from videos of telementored surgeries

Paolo Buono, Giuseppe Desolda, Rosa Lanzilotti
Dipartimento di Informatica
Università degli Studi di Bari “Aldo Moro”
via Orabona, 4, 70125 - Bari, Italy
{paolo.buono, giuseppe.desolda, rosa.lanzilotti}@uniba.it

Abstract - **The huge amount of videos, available for various purposes, makes video editing software very important and popular among people. One of the uses of video in medicine is to store surgical operations for educational or legal purposes. In particular, in telemedicine, the exchange of audio and video plays a very important role. In most cases, surgeons are inexpert in video editing; moreover, the user interface of such software tools is often very complex. This paper presents a tool to extract important scenes from surgery videos. The goal is to enable surgeons to easily and quickly extract scenes of interest.**

Keywords: Telemedicine, video editing, scene extraction

I. INTRODUCTION

Video editing software is extremely popular today. Both, commercial [11] and open source [8] software present many useful features that allow their users to cut scenes, merge scenes, apply transitions as well as add soundtracks, video and audio effects, etc. The user interface of video editing software is often not easy to use for occasional users.

In medicine, and especially in laparoscopy surgery, videos are becoming common in surgical practices. However, many surgeons do not have adequate skills for video editing. Laparoscopic surgery, also called minimally invasive surgery (MIS), is a modern surgical technique in which the surgeon performs several small incisions (typically 3 or 4 incisions 0.5–1.5 cm in length) in the abdomen of a patient through which tools and the endoscopic camera are inserted. During the surgery, images are displayed on large monitors that magnify the area of interest. The whole surgery video is stored in order to be used for educational and/or legal purposes.

Surgery videos are usually long (at least 2 hours) and often only a few scenes are relevant. Moreover, typical functions of video editing software, like video filters, transition effects between scenes, advanced export functions (e.g. publishing on YouTube, Facebook, etc.), multi-audio/video tracks, etc. are not needed in the case of the surgery videos and may only confuse surgeons. We are involved in a research project aimed at developing a telemedicine system for supporting surgeons learning new surgical techniques. The system will provide different tools to support the surgeons work. Such tools are designed in order to accommodate the needs of their users. In particular, the aim is to reduce the complexity of the software tools by providing surgeons only those functions that they need. In this paper, we describe a tool that allows surgeons to easily and quickly perform the main task they are interested in, i.e. extract scenes of interest from a surgery video.

In order to create a tool able to satisfy needs and requirements of the specific category of surgeons, we adopted a participatory design approach [5]. A contextual enquiry was performed at the “Perrino” Hospital in Brindisi (Italy), which is actually a partner of our project. We carried out interviews and focus groups with several surgeons. Naturalistic observation of surgeons while performing laparoscopic surgeries or using video to teach to younger surgeons was also performed.

These studies showed the difficulties surgeons had in using software for video editing and provided hints for the design of a tool that can support surgeons in analyzing video and retrieving scenes of interest.

Next section presents a brief state of the art in video editing. Afterwards, we illustrate two alternative interfaces of the tool for extracting scenes. Then, the results of a formative evaluation in which these two alternative interfaces have been compared are described. The running version, implementing the prototype that resulted more effective and usable is later presented. Last section concludes the paper.

II. RELATED WORK

Video manipulation and, in particular, video summarization are gaining increasing interest due to the proliferation of digital camcorders, online video databases (e.g. YouTube), videos collected on large storage device, etc. Video summarization aims at extracting, from a long video, scenes that are more relevant for a certain purpose.

Some commercial tools for video summarization are already available. Examples are Windows Movie Maker, Pinnacle Studio, and Adobe Premiere. Such tools provide many functions, whose complexity confuses users that, occasionally, use them. Research on video summarization presents many different approaches. We briefly report here some of them.

AVST (Automatic Video Summarizing Tool) utilizes MPEG-7 visual descriptors to generate video thumbnails to search for similar scenes and cluster scenes [15]. AVST splits scenes also according to abrupt and/or gradual transitions. It works well for videos characterized by scene changes, like in sports or movies.

Jang et al. propose an algorithm that uses visual and audio content to automatically generate improved video summaries [13]. The system performs audio segmentation and classification according to audio and visual diversity, face quality, and overall image quality. These characteristics are gathered from users, who provide feedbacks on summarized videos. The system has been applied in the contexts of birthdays, weddings, shows, and parades.

Bailer et al. propose TRECVID, an interactive video browsing tool based on a multimedia content abstraction model [1]. The tool clusters scenes according to: camera motion, visual activity, audio volume, face occurrences, global color similarity, repeated takes and relations in multi-view content, in order to reduce the content to a manageable number of scenes.

Novel visual techniques in the field of video surveillance have been explored, an example is in [2] and [3] in which, in order to speed-up the selection process of interesting scenes, an interactive image of movements is created.

The previously mentioned approaches do not specifically address laparoscopic surgery videos, in which the camera is not stationary, videos are characterized by very similar scenes, the audio is produced by surgeons talking among them, and no face, or people characteristics are present in the videos.

Among several works carried out for summarizing videos in laparoscopic surgery, Leszczuk and Duplaga present a prototype that creates summaries of bronchoscopy video recordings [16]. The summarization algorithm removes poor quality frames due to blurry images. Such frames are unavoidable due to the relatively tight endobronchial space, rapid movements of the respiratory tract, and secretions that occur commonly in the bronchial tube, especially in suffering patients. During a classification phase, the algorithm identifies non-informative frames, which are discarded in the summarized video.

According to the opinions of surgeons we have worked with, shortening videos, by removing blurry or uninteresting scenes, can be useful but is not their primary goal. The problem addressed in our work is to choose a few relevant scenes that are representative for a video. It is worth remarking that the approach we propose does not rely on automatic algorithms only, but requires human intervention in order to reduce the possibility of errors and make sure that the final scenes identification are those required by the surgeons.

III. THE SCENE EXTRACTION TOOL

The tool, described in this paper, supports the extraction of scenes from video produced by a telementoring system that we have been developing in the last few months. Telementoring is gaining momentum. It is useful for surgeon training and can be used for consultancy requests to mentors working in different hospitals, cities or even continents [14].

Our telementoring system allows surgeons (learner) to be assisted by experienced remote surgeons (tutor) during a laparoscopic surgery. The system main components are the *Learner* and the *Tutor* (we distinguish Learner and Tutor devices from learner and tutor surgeons by capitalizing the first letter when referring to the devices). The Learner is a small device installed in the surgery room that sends video signals produced by the endoscopic camera and the audio of the surgery room to the Tutor. The Tutor is a different device available at the remote tutor surgeon location having the functionalities of audio and video I/O and a pointing feature (mouse, pen, touch, etc.). The Tutor sends audio, telestration and images to the surgery room. A high-level architecture is shown in Figure 1.

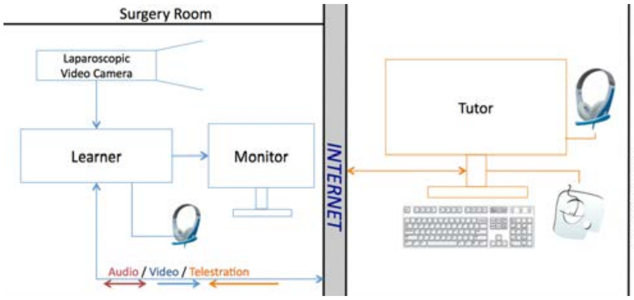

Figure 1: System architecture

In this paper, we refer to telestration [9] as the activity of drawing, sending images or applying a still image on a monitor from a remote location. Specifically, we use telestration for improving the remote communication between the learner and tutor.

During the discussions with surgeons in our participatory design team, important requirements emerged that were implemented in our telementoring system. Surgeons asked for the possibility to mark interesting moments of the surgery, like adding a bookmark while reading a book, in order to easily retrieve such moments that they would like to show later. To comply with this need, we provided the Tutor interface with a marking function that allows the remote surgeon to indicate relevant moments by pressing a button in the Tutor user interface.

Since surgeons, in specific moments, needed higher precision, provided the Tutor with a pause feature, which consists of visualizing, at the same moment, a high-definition still image of the learner camera to both tutor and learner monitors, on which the telestration is displayed.

Another feature, we added after a specific request of the surgeons, is the possibility to send images. Often, experienced tutors also teach, so they need to send images to improve their explanations. This picture is displayed like a paused image.

Each telementored laparoscopy surgery is video recorded by the telementoring system, which records also the audio interaction between learner and tutor. Surgeons often need to extrapolate a few meaningful scenes in order to teach, provide consultancy on a specific topic, show a novel technique at a conference, etc. Beside the recorded videos, our telementoring system stores XML files that contain all details of telestration data, marks, pauses and images sent. In order to support scenes extraction, our approach exploits data available in the XML file.

During the laparoscopic surgery, the tutor performs different actions, such as marking a certain time, pausing the video, performing a telestration, sending an image to the learner. A telestration can be a free-hand drawing or an arrow. For the tutor, such actions are performed in specific moments. Thus, in our tool, the identification of such actions is the starting point for retrieving scenes of interest and extracting them. Our approach is based on the visualization of actions in a compact form, which aims at helping the user to identify scenes of interest and save them in a personal area for future uses.

During the participatory design, several alternative designs and several low fidelity prototypes were proposed. In particular, two alternatives were better investigated: 1) visualize actions on a fixed area and show the details into another timeline; 2) visualize the different types of actions occurred in a given moment using a zoomable timeline. Furthermore, these two alternative prototypes present two different scene selection modalities, due to the different timeline visualization.

The two running prototypes were developed using Axure, a software for creating high fidelity interactive prototypes [10]. Figure 2 and Figure 3 show the two prototypes. In both cases, the main area is devoted to the video play and actions are visualized in a timeline. The panel on the right side is used to collect interesting scenes and see their previews.

A. Prototype 1

As shown in Figure 2, the interface of the Prototype 1 is composed of three main areas: the *play* area at the center of the screen; the *timeline*, at the bottom of the screen; the *preview* area, on the right side of the screen containing the selected videos.

The video is played in the *play* area. The play/pause button is located at the left-bottom corner of the *play* area.

The *timeline* was inspired by [6], which uses two timelines: one showing the overview of the video actions and the other one focusing on a small time span of the video. Indeed, as shown in detail in Figure 5, this area is divided into two parts: at the bottom there is an *overview timeline* visualizing data of the whole video, and on top of it, a *details timeline* containing details of the selection performed in the *overview timeline*. The blue pins in the *overview timeline* indicate all actions performed by the tutor. When the user clicks somewhere on the *overview timeline*, the details of about 1 minute video, centered at the clicked time point, is visualized in the *details timeline*. In the example of Figure 5, the user has clicked at minute 30:00. The system visualizes the details of the video from minute 29:30 to minute 30:42 in *details timeline*. Three icons represent three actions made by a tutor, i.e. a free hand draw, a pause action and an arrow, respectively. Moving the mouse pointer over these icons, a balloon shows a preview; by clicking on these icons, the corresponding part of video is played in the *play* area. To help the user to understand which part of the *overview timeline* is shown into the *details timeline*, the selection has a green border, like the *detailed area* border.

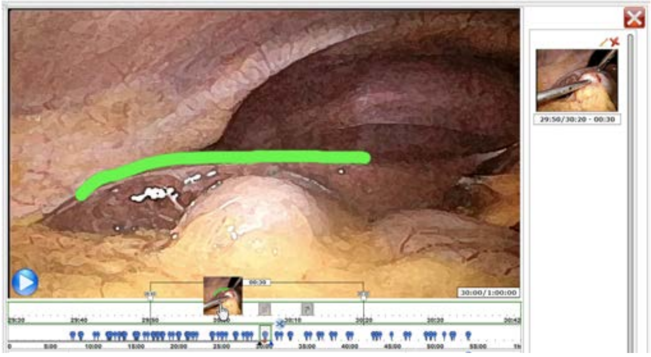

Figure 2: Screenshot of Prototype 1

One of the main goals of this paper is to bring out an interaction modality to easily extract scenes. The steps that the user has to perform with this prototype are:

1. Clicking on *overview timeline* to visualize the details;
2. Clicking the button represented by the scissor icon to visualize the selection function;
3. Resizing, if needed, the selection by using the handles;
4. Accepting or discarding the selection by clicking on SAVE button or X button respectively.

The *preview* area, on the right side of the user interface shown in Figure 2, contains thumbnails of saved scenes. At the top-right corner of each preview, the X button allows users to delete the selection, while the *pencil* button permits to change the video interval. By clicking on the *pencil* button, the previously saved selection appears again on the timeline in order to allow the user to change the begin/end of the video.

B. Prototype 2

As shown in Figure 3, the interface of the Prototype 2 is very similar to the first one, except for the timeline. In this prototype actions are visualized like in [4], in which the timeline shows different rows, each representing a type of action. Actions are organized into invisible tracks over the timeline, with each action allocated to a given segment of its track.

Moreover, these actions are vertically grouped into columns according to the concept of *indication*. We consider an *indication* as a group of actions performed by the tutor, useful to give suggestions to the learner about surgical procedures. An *indication* starts when the tutor draws something (free hands or arrow), pause the video or sends an image on which drawing something; the *indication* ends when the user deletes all actions or plays the video (if it was in pause or if picture was sent). Color, size and actions priority has been considered. We used the ColorBrewer online tool [7] to choose a set of 5 quantitative colors in order to associate the color to the type of action.

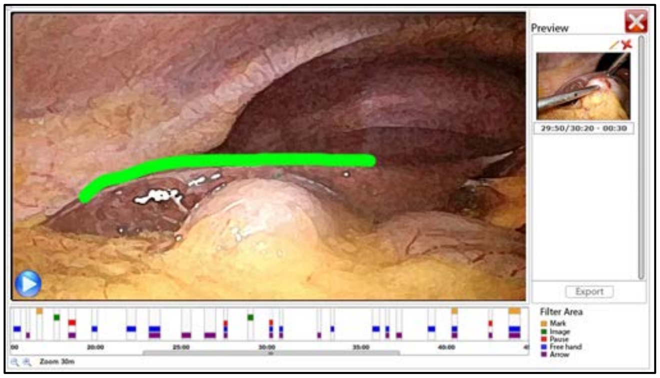

Figure 3: Screenshot of Prototypes 2

The size of actions is problematic. The columns width can be too tight. Typically, actions last in average for less than 20 seconds. On a common monitor with a screen resolution of 1024x768 pixels, the timeline is composed of about 800 pixels. In a video that lasts 2 hours (the average length in the surgical domain), 20 seconds can be represented in about 2 pixels wide. In order to make the information visible, we adopted two solutions. The first one is to start the video editing with a default timeline zoom that visualizes an interval of 30 minutes. In this way, the columns have an average width of about 10 pixels.

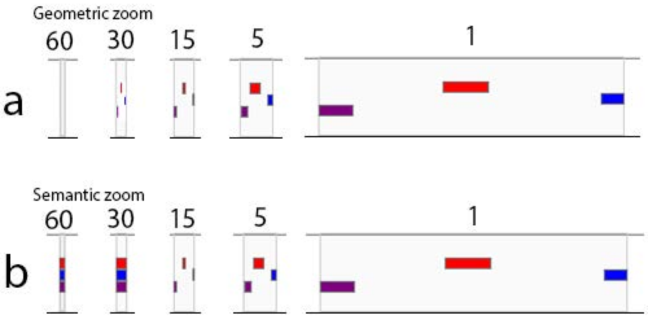

Figure 4: Comparison between action visualizations with (a) geometric and semantic (b) zoom

The second solution implemented a semantic zooming, which visualizes different levels of detail in a view when zooming in and out. If the system implements a geometric zoom (case A of Figure 4), when the user zooms, out over a certain level, the visual information about the actions disappears.

In the adopted semantic zoom, when the user zooms out and the visual information about the actions is too small to be visible, the actions have the same width of the containing column (e.g. Figure 4, case B, see details at 60 and 30 minutes). In any case, the group of actions is at least 2 pixels wide.

Finally, we ordered the action track according to an importance criteria expressed by interviewed surgeon. They considered the *mark* as the most important action, since it is the only one that the tutor performs explicitly. After, they considered the *image* as the next important, then *pause* function, lastly, *free hand* and *arrow*.

Similarly to Prototype 1, we designed a feature to select a scene. The green rectangle in Figure 6 starts at minute 30:00 and contains a scene with two bars composed of 3 and 2 actions, respectively. The green rectangle appears when the user clicks on the timeline and the 20 pixel default time span of the selection can be modified acting on the handles on both sides of the rectangle. A click on the floppy stores the selected scene and a corresponding thumbnail appears in the area at the right of the user interface. The user can zoom into the timeline and a preview of the scene is visible as a popup when the mouse pointer is moved over a specific action.

IV. FORMATIVE EVALUATION

As a part of formative evaluation during the early development phase, a user study was performed in February 2013 to get feedback from the intended users about which one of the two different prototypes is more usable and more appropriate for their main tasks. The study involved 6 surgeons and was performed in the field, i.e. in the surgeons' office at the Perrino hospital in Brindisi.

The 6 surgeons (5 males) were tested separately. The thinking aloud technique was used to evaluate the prototypes. Each test consisted of two phases, each one for analyzing a prototype. In order to avoid the learning effect, 3 surgeons first interacted with the Prototype 1 and, then, with the Prototype 2; the other 3 surgeons used the two prototypes in reverse order. At the beginning of each phase, the surgeon was given a brief introduction on the prototype to be used and its main functions. After this, the surgeon performed five predefined tasks and, finally, s/he was interviewed to collect data about her/his opinions on the used prototype. At this point, the other phase, in which the surgeon had to interact with the other prototype, started. This latter phase followed the same procedure of the former one: surgeon had to perform the same tasks, but with the support of the second prototype.

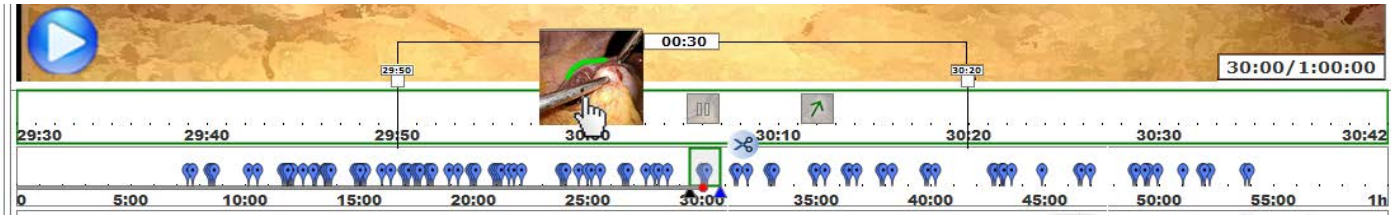

Figure 5: Detail of Prototype 1 with two timelines

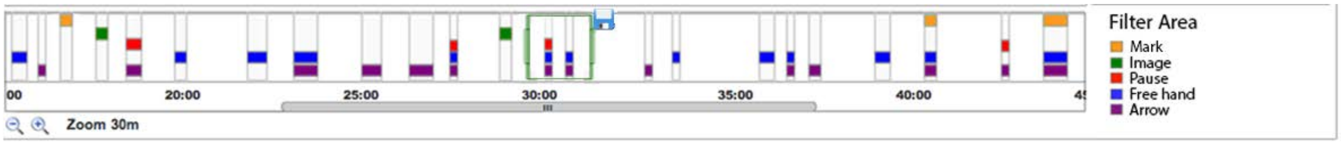

Figure 6: The timeline in Prototype 2 with semantic zoom

The surgeon was observed by two HCI evaluators and was videotaped. A video analysis was performed to collect data on the number of tasks successfully completed. Each test lasted about 30 minutes.

The five tasks were defined in order to allow surgeon to use the functions implemented in each prototype (i.e. play/pause, select, modify, save, delete). They were of different complexity. In order to analyze the ease of learning of the two prototypes, two tasks were very similar. In order to accomplish a task, surgeons have to do more than 2 steps.

The success rate was calculated for all tasks performed with the two prototypes [18]. Specifically, it resulted 54% for the Prototype 1 and 58% for the Prototype 2. Generally, no significant difference emerged. However, it is worthwhile to highlight that Prototype 2 better supports its users in performing the selection of a scene from the telementored video. In fact, only one user successfully completed this task with the Prototype 1, 3 users partially accomplished it and 2 users did not finish it. Regarding the Prototype 2, 4 users successfully completed the task and 2 users did not able to finish it.

Another important result concerns the ease of learning that was analyzed by the difference between the success rates of the two similar tasks for each prototype. Specifically, the success rate increased of the 8% for the Prototype 1 and of the 42% for the Prototype 2. This showed that Prototype 2 seems to be more easily learnable than the Prototype 1.

The thinking aloud was instrumental to identify usability problems of both prototypes. Specifically, all surgeons did not understand in what the two timelines visualized in the Prototype 1 differs. On the other side, it was not so clear to the surgeons that the labels, on the right side of the Prototype 2, were not only a legend, but they also were filters, which permit to refine the scene search. Other interaction difficulties surgeons concerned limits of the rapid prototyping software. In other words, if surgeons did not tightly follow the time indications given in the task definition (for example, “Select a scene starting at 29.50 min and ending at 30:20) they did not able to accomplish the specific task.

The interviews were useful to collect opinions and, especially, suggestions to improve the prototypes. Surgeons were agreed that the Prototype 2 was easier to use than the Prototype 1. They said that the Prototype 2, differently from the Prototype 1, not only allowed its users to accomplish the tasks without serious difficulties, but also to have at a glance an idea of actions available in the video.

Two surgeons required a scrubbing mechanism to facilitate the scene detection. In fact, being inspired by the Prototype 1, they would like that the Prototype 2 provided a scene preview, which appeared in a small popup window when the user goes through the timeline. In this way, the video analysis could become more rapid.

Another surgeons’ request concerns the way in which the video can be annotated. They explicitly said that, during a classical laparoscopic surgery without the telementoring support, they would prefer to have a vocal command to annotate a scene of the video. For example pronouncing: “System, mark now!” and the system stores a vocal mark.

V. THE FINAL PROTOTYPE

Starting from the results of the user test, we developed a final prototype in Java. We used JavaCV framework to visualize the video and export the final summary. The input of the application is a video of telementoring and its XML file produced by the telementoring software. Figure 7 shows the interface, composed by different areas: (1) video player, (2) previews; (3) filter by action type; (4) Timeline. The software is similar to Prototype 2, since it has been shown to be more adequate for surgeons.

Another evaluation of the application user interface was performed with three surgeons chosen among those that participated to the prototype user testing. The surgeons carried out the same five tasks performed to evaluate the first two prototypes. All tasks were correctly accomplished without any problem.

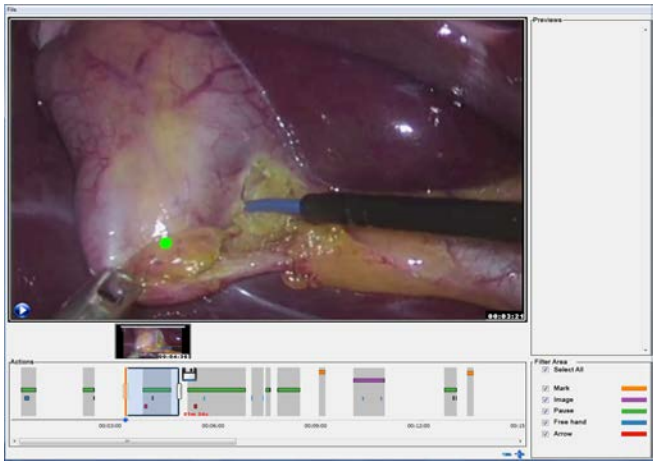

Figure 7: The current version of the video-editing tool

VI. Conclusions

In medicine and, especially, in laparoscopy surgery, surgeons have to edit surgery videos both for educational and legal purposes. It often happens that surgeons have difficulties in using video editing software given to its functional complexity. In fact, video editing software provides several functions, such as transition effects, advanced export function, etc., which surgeons do not use.

This work has presented the development of a tool that allows surgeons to extract scenes from a surgery video. Two different prototypes have been implemented and evaluated with end users in order to identify which one of them better supports the work of surgeons in extracting the important scenes.

The Prototype 1 provides two timelines: one visualizes data of the whole video and contains blue pins indicating actions performed by tutor; the other one shows details of a selected scene. The Prototype 2 shows a timeline containing different rows that represent the tutor's actions. Such actions are vertically grouped into columns, which provide actions performed by the tutor useful to give suggestion to the learner about the surgical procedures.

The performed usability testing revealed that end users preferred the Prototype 2; also some improvements to be implemented in the new version of the tool were suggested. For example, the use of advanced video scrubbing techniques [17] will be investigated to enhance the detection of interesting scenes. We are planning to implement in the telementoring system a speech recognition module [12] to give surgeons the possibility to vocally mark the videos.

In the immediate future, we will preform a comparison study to investigate which one of the two visualization techniques implemented in the two prototypes is more efficient to detect interesting moment in a video.

The analysis of a corpus of telementored surgery videos could be conducted to reveal actions patterns that allow the system to provide suggestions for the selection of important scenes.

Another improvement could be to export the selected videos as SCORM learning objects, in order to allow surgeons to easily integrate interesting videos into e-learning systems.

Acknowledgments

We thank Professor Maria Francesca Costabile for her valuable guidance in our research.

References

[1] Bailer, W., Weiss, W., Schober, C. and Thallinger, G. 2012. A video browsing tool for content management in media post-production. In Proceedings of the Proceedings of the 18th international conference on Advances in Multimedia Modeling (Klagenfurt, Austria, 2012). Springer-Verlag, 2189122, 658-659.

[2] Buono, P. 2011. Analyzing video produced by a stationary surveillance camera. In Proceedings of the DMS (2011). Knowledge Systems Institute, conf/dms/Buono11, 140-145.

[3] Buono, P. and Simeone, A. L. 2010. Video abstraction and detection of anomalies by tracking movements. In Proceedings of the Proceedings of the International Conference on Advanced Visual Interfaces (Roma, Italy, 2010). ACM, 1843036, 249-252.

[4] Costa, M., Correia, N. and Guimarães, N. 2002. Annotations as multiple perspectives of video content. In Proceedings of the Proceedings of the tenth ACM international conference on Multimedia (Juan-les-Pins, France, 2002). ACM, 641065, 283-286.

[5] Douglas, S. and Aki, N. Participatory Design: Principles and Practices, 1993.

[6] Hampapur, A., Brown, L., Connell, J., Ekin, A., Haas, N., Lu, M., Merkl, H. and Pankanti, S. 2005. Smart video surveillance: exploring the concept of multiscale spatiotemporal tracking. Signal Processing Magazine, IEEE, 22, 2 (2005), 38-51.

[7] Harrower, M. and Brewer, C. A. ColorBrewer.org: An Online Tool for Selecting Colour Schemes for Maps. In The Map Reader (2011). John Wiley & Sons, Ltd.

[8] Hsiang-An, W., Yen-Chun, L., Hua-Ting, L. and Shang-Te, L. Open Source for Web-Based Video Editing. Institute of Information Science, Academia sinica, Taipei, Taiwan, Province de Chine, 2012.

[9] http://en.wikipedia.org/wiki/Telestrator. Telestrator(

[10] http://www.axure.com. City.

[11] http://www.toptenreviews.com. Video Editing Software Review(

[12] Jacob, M. G., Li, Y.-T., Akingba, G. A. and Wachs, J. P. 2013. Collaboration with a robotic scrub nurse. Commun. ACM, 56, 5 (2013), 68-75.

[13] Jiang, W., Cotton, C. and Loui, A. C. 2011. Automatic consumer video summarization by audio and visual analysis. In Proceedings of the Proceedings of the 2011 IEEE International Conference on Multimedia and Expo (2011). IEEE Computer Society, 2193850, 1-6.

[14] Latifi, R., Peck, K., Satava, R. and Anvari, M. 2004. Telepresence and telementoring in surgery. Stud Health Technol Inform(2004), 200--206.

[15] Lee, J.-H., Lee, G.-G. and Kim, W.-Y. 2003. Automatic video summarizing tool using MPEG-7 descriptors for personal video recorder. IEEE Transaction on Consumer Electronics, 49, 3 (2003), 742-749.

[16] Leszczuk, M. I. and Duplaga, M. 2011. Algorithm for video summarization of bronchoscopy procedures. Biomed Eng Online, 10(2011), 110.

[17] Matejka, J., Grossman, T. and Fitzmaurice, G. 2013. Swifter: improved online video scrubbing. In Proceedings of the Proceedings of the 2013 ACM annual conference on Human factors in computing systems (Paris, France, 2013). ACM, 2466149, 1159-1168.

[18] Nielsen, J. Usability Engineering. Morgan Kaufmann Publishers Inc., 1993.