Expert Driven Semi-Supervised Elucidation Tool for Medical Endoscopic Videos

Zeno Albisser[1], Michael Riegler[1], Pål Halvorsen [1], Jiang Zhou[2],
Carsten Griwodz[1], Ilangko Balasingham[3], Cathal Gurrin[2]
[1]Media Performance Group, Simula Research Laboratory, Norway
[2]Insight, Dublin City University, Ireland
[3]Intervention Center Oslo University Hospital, University of Oslo, Norway
zenoa@ifi.uio.no, {michael, paalh, griff}@simula.no, {jiang.zhou, cgurrin}@dcu.ie, ilangkob@medisin.uio.no

ABSTRACT

In this paper, we present a novel application for elucidating all kind of videos that require expert knowledge, e.g., sport videos, medical videos etc., focusing on endoscopic surgery and video capsule endoscopy. In the medical domain, the knowledge of experts for tagging and interpretation of videos is of high value. As a result of the stressful working environment of medical doctors, they often simply do not have time for extensive annotations. We therefore present a semi-supervised method to gather the annotations in a very easy and time saving way for the experts and we show how this information can be used later on.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Experimentation, Human Factors

Keywords

video annotation, medical multimedia information systems, semi-supervised, object tracking

1. INTRODUCTION

Detecting irregularities in intestines is a difficult and very time-consuming task, and there are several different kinds of irregularities a doctor can detect visually using colonoscopy or camera pills. For the untrained eye, such irregularities are, however, not always easy recognizable. Depending on the length of the video acquired by, e.g., a camera pill, this can be a very time-consuming and therefore expensive task. It seems natural to try to automate this task using computers. To be able to train an algorithm to detect such irregularities, a comprehensive data set, containing video sequences with and without irregularities, is necessary. Collecting this data requires recording of video sequences and tagging every occurrence of an irregularity in every video frame, e.g., marking a polyp in the colon as shown in Fig. 1. This work requires a specialist to make sure that no false positives or false negatives examples occur. Tagging all the occurrences is an especially tedious piece of work, as it requires stepping through single video frames and adding, moving and resizing tags. The experts usually do not have a lot of time for this kind of work. Thus, makes it necessary to create tools that reduce the amount of time needed to process a video. Such tools must meet the following requirements: (i) Save as much of the specialist's time as possible, (ii) allow efficient collection of big amounts of data, (iii) easy to use with very little introduction time and (iv) deployment of the system in restricted hospital environment.

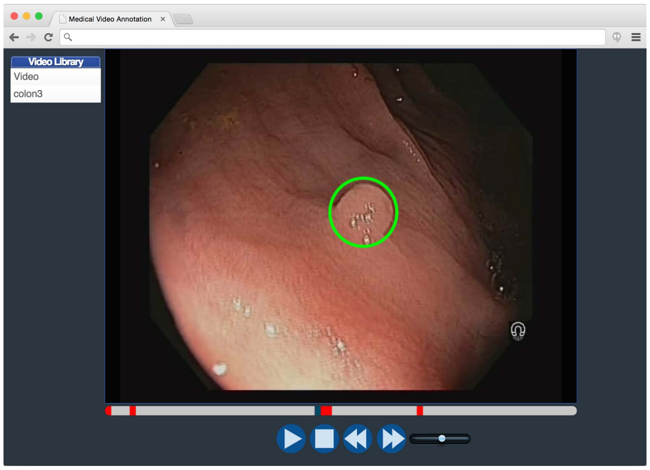


Figure 1: Web based video annotation software for medical purpose.

To tackle this problem we have been prototyping and experimenting with different technologies to cater these specific requirements. We present our semi-supervised annotation system, see Fig. 1, which is divided in two parts. The first part (i) is a web based tagging tool that should be used by a specialist to create a coarse selection of regions of interest. The second part (ii) is a tool that can be used subsequently by a regular user to generate a complete data set using object tracking algorithms and manual correction if necessary. The system is already in use for generating informative and large data sets for medical multimedia con-

ACM MMSys '15, Mar 18-20, 2015, Portland, OR, USA
ACM 978-1-4503-3351-1/15/03
http://dx.doi.org/10.1145/2713168.2713184.

tent analysis. The remainder of the paper gives an overview on related work and presents the architecture and the implementation of the application. Furthermore, we show how the obtained data can be used afterwards, and we give an outlook of ongoing and future work. A demo video of the tool can be found at `http://goo.gl/Fhd0J6`.

2. RELATED WORK

Previous research related to annotating videos can be split into manual video annotation tools and semi-supervised approaches. In this section, we will discuss their relation to our tool and point out the differences. A way of annotating videos is the use of different elements on top of the video frames like speech bubbles, hand drawn annotation and a lot of other different overlays. Furthermore, annotation by speech is also a widely used method. That these annotations have in common is that they are manually added to the video to describe the content. Examples for state of the art applications are, for instance, YouTube, VideoWiki and Popcorn Maker. A tool that combines complex annotations together is *Videojot*. For the medical use case, the *MedAnnotation Tool* is the latest related work in this area [14, 11, 4, 2, 13, 12]. The usage of these tools ranges from very complicated to very easy to use for, trained or untrained users. All these tools require a significant amount of time for creating annotations. In some areas, this is not a big problem, but in others like, the medical sector where the doctors are constantly under a lot of pressure and lack of time, the currently existing tools are not really usable, i.e., especially when the goal is to collect a huge amount of data for computer vision or retrieval algorithms [7].

Our tool tackles this problem by providing a very easy and quick way to annotate important parts. It then uses these tiny annotations to automatically generate the data that we need for further computation. There already exists some work about these kind of semi-supervised annotation tools, but they do not annotate specific parts of the video for the usage in a later training set. They are more general semantic annotation [5, 16, 15] tools, which cannot be used for example to detect cancer in regions of the video, etc. The biggest difference to existing tools is that the tool presented here is easy and time expeditious to handle, and it is able to automatically create a huge data set of medical conditions from a subset of expert annotations. Therefore, it supports the doctors to provide as much information as possible with very humble effort. To the best of our knowledge, there exist no such tool that provides the same functionality.

3. ARCHITECTURE

The architecture of our solution is divided into two steps *Manual Annotation* and *Object Tracking*. Fig. 2 gives an overview of the whole system. This is mainly to reduce the amount of time specialists are needed in the whole process due to the fact that they only have to provide elucidation in a single frame. We do require the specialist's knowledge during the first step to do a very basic identification of irregularities and to tag them accordingly. The *Manual Annotation* step is to precisely select any regions of interest in a video sequence. We also refer to this step as *Object Tagging*. The *Object Tracking* step is to track the regions of interest on previous and subsequent frames, based on the previously manually created tags. This step is more about tracking an object and adjusting the size and position of the tracked region than about identifying or recognizing irregularities. Specialist's knowledge is therefore not required for the second step. Another reason to divide the process into these two steps is the technologies available for implementing the required software. A specialist is usually located in a hospital with special restrictions to security due to sensitive patient information. Deployment of software is therefore a problem because of privacy issues. Nevertheless, internet access and a browser are usually available. This makes standard web technologies a convenient way of circumventing deployment related issues for the manual annotation software. It also implies storing all information on the server side and moves the responsibility of maintaining the system and data integrity from the user to the server administrator.

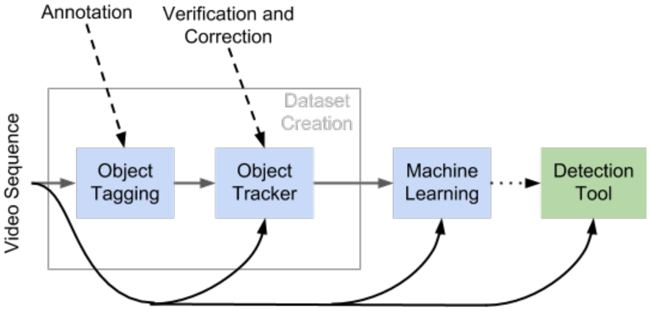


Figure 2: The processes for dataset creation are a prerequisite for building and a detection tool.

Manual Annotation. The manual annotation is the first step in the whole data gathering process. In this step, a specialist uses rubber band selection (mark a bounded area) to create a coarse selection of regions of interest and annotates every selection with a name for classification. Every region needs to be marked once only. To keep the specialist's time spent on this task minimal, we do not require the region to be marked in the very first video frame it appears. Information on first appearance and change of shape or position within the picture will be added later using object tracking and manual correction. This approach allows a rather rapid way of working for the specialists. They might even watch the video at a higher playback speed and only stop or slow down the playback when really necessary. The information collected in this step includes the position and dimensions of irregularities in pixel coordinates, a classification and a timestamp relative to the beginning of the video for each selected region. We have decided to implement the manual annotation component using JavaScript and HTML5 video which is available in most recent web browsers. We use a standard username and password authentication mechanism and transfer all the date using HTTPS to ensure secure data access and transmission during the whole process.

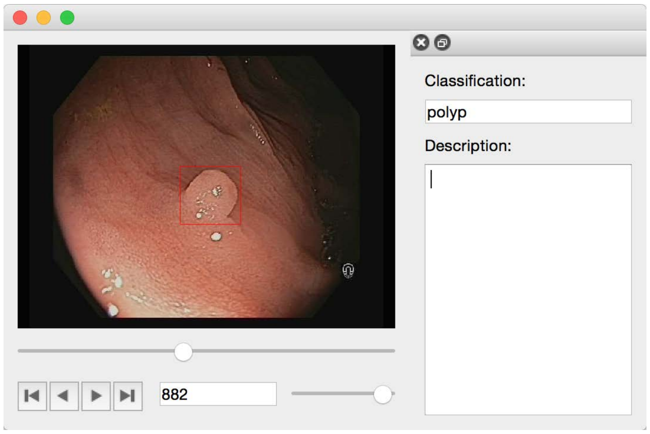
Object Tracking. The output from *Manual Annotation* only contains a single tag for every region of interest in the video sequence. Using this information, we can now apply object tracking algorithms and manual correction to generate a complete data set. Most of the work in this step is done by the software. The user just needs to step to the previously marked irregularities and playback the video from that point for the software to track the marked region on subsequent frames. Depending on the quality of the video and the speed of camera movement, user intervention is needed to assure a high quality of tracking. As the irregularity most likely has not been marked on the very first frame it appears in, the video must also be played in reverse direction from the first position a region was marked. This is needed to track the region towards the beginning of the video. There is of course still a fair amount of manual work involved in this task. However, using a suitable tracking algorithm, the time needed to create a complete dataset can be reduced significantly. Moreover, specialist skills are usually no longer required here as the whole task is simply about tracking regions and adjusting rectangular dimensions rather than actually detecting or recognizing irregularities. The output generated in this step is a list of rectangles for every previously marked region. Every rectangle in such a list is described by the index of the video frame it belongs to, its position in pixel coordinates and its dimensions.

4. IMPLEMENTATION

Experimenting with several different technologies we came to the conclusion that a solution divided in two steps has several advantages. It allows us to minimize the time a specialist is needed, and it also significantly simplifies the deployment and maintenance of the software. The only requirements for the first step are an HTML5 compliant web browser and an internet connection.

Manual Annotation. The web application we implemented is mostly written in HTML5 and JavaScript. Specifically, it makes use of the HTML5 video element. Listing and uploading videos and storing tagging information is implemented in Java and running in an Apache Tomcat servlet container[1]. All video sequences will be uploaded to the server through the web interface. On the server, we are using a Java servlet, which spins off a job to transcode the video to H.264. For transcoding we are using *libav* and *avconv*[2]. Transcoding is necessary in case the original video file is not encoded in a codec that is supported by the browser. H.264 seems to be a good choice as it is currently supported by all major web browsers. The transcoding job is running asynchronously, so a connection to the server is not needed to keep the job alive.

The web interface of our tagging application provides the usual start, stop and pause controls of a regular video player. Additionally, we added a seek bar that highlights the playback position and any regions of interest in colors. We also added a "seek-forward" and a "seek-backward" button that allows stepping to the next/previous region of interest. As the video playback in HTML5 is running outside of the JavaScript execution thread, we do not have a strict control over the video frames being displayed. The playback position is only provided as a floating point value property *currentTime* in seconds. The property can be read and it can also be written in order to seek to a specific position. When executing JavaScript code this property can be read at an arbitrary point in time. And since a single video frame is usually being displayed for about 40ms [3] this means that when playing a previously tagged video sequence, we will most likely not read the same value from the *currentTime* property again as we were reading while tagging. Therefore visibility of a previously created tag cannot be guaranteed during playback and we must use the seek buttons to seek to the next or previous region of interest. Whenever a region of interest has been selected, an editor shows up and allows the specialist to enter a classification and a comment. This information will be stored together with the tagged rectangle in JSON[4] format on the server.



Classification:
polyp

Description:

882

Figure 3: Native software for modifying tags and tracking of regions of interest.

Object Tracking. For the second step in the process, we implemented the object tracking tool in C++ using Qt[5] for the user interface and OpenCV[6] for reading and processing the video data. We further integrated with Struck [8] for tracking the tagged regions. The user interface for this tracking software is similar to the web interface described previously and can be seen in Fig. 3. It features a video widget, play, seek-forward and backward buttons as well as a seek bar with identical behavior. Moreover, a slider to increase or decrease the playback speed and an editor for classification and description is present. Further, a button for playing the video in reverse direction and context menus for modifying regions of interest is provided. After starting the application, a JSON file created using the tagging web application can be opened and the respective video file must be selected. We use the original video file instead of the H.264 encoded one. This is because we need to be able to play the video forwards and backwards frame by frame. Recreating frames in reversed direction is very expensive with H.264, because frames can be encoded referencing previously encoded ones. The original files uploaded to our server are usually simple MJPEG video files and are very well suited for playing both directions. The users use the seek buttons to seek to the next or previous regions of interest. Then they use the context menu to select one or multiple regions for tracking. Playing the video in either direction will then track the region in the video frames being displayed. Alternatively, the arrow keys can be used to step forward or backward frame by frame. The playback can be paused at any time to adjust size or position of the tracked region.

Using double buffering allows reading and processing the next frame while the previous frame is still being displayed. The processing (reading of frames and tracking of regions) is therefore running in a separate thread. The communication between the user interface and the worker thread is imple-

[1] http://tomcat.apache.org

[2] https://libav.org/avconv.html

[3] assuming a usual frame rate of 25 frames per second

[4] http://goo.gl/Oi5kIF

[5] http://www.qt.io

[6] http://www.opencv.org

mented using Qt's events delivery mechanism. Whenever the tracking algorithm fails to track a region, the playback stops automatically. It is then up to the user to decide if the tracked region should be removed or if the tracking should be re-initialized with an updated region. The user can seek forwards and backwards freely to review the tagging and tracking results and adjust, move or restart tracking of a region at any point during the process. Once the dataset is complete, it can be saved to a JSON file.

5. APPLICATIONS OF THE DATASET

The primary application of the annotated images is training algorithms for automatic medical screening. As stated at the begin, reviewing images or videos and making diagnostic decisions in screening are very time-consuming and the accuracy is subject to the experience and concentration of the physicians [6]. For example, in a camera pill endoscopy exam, there are about 60,000 images per examination for one patient, and it costs an experienced medical clinician about 2 hours on average to view and analyse all the video data [10]. Therefore, it becomes necessary to reduce the heavy burden on physicians and speed up the screening process with computer aided diagnosis. In terms of colonoscopy videos, the objective would be training a classifier and automatically detecting the colon cancer, or its precursor lesions, colorectal polyps in videos. To build the classifier, the annotated irregularity regions are pooled together as positive samples and random selected regions without any irregularity are used as negative samples. Colour, texture and shape features [1, 3] are extracted from the training samples. A Support Vector Machine (SVM) is used to train the classifier with the combinatorial features, and the Radial Basis Function is applied as the kernel [9]. To tune the parameters in SVM and prevent model over-fitting, k-fold cross validation is performed. A separated set of positive and negative samples, which have never been seen during the training, is prepared as a testing set. The classification performance is then measured by the Receiving Operating Characteristic (ROC) curve. With a shifting-window method, the built classifier can not only tell the presence of irregularities but also give their locations within an image. Beside the automatic screening, with our semi-supervised annotation tool, segments within a medical video are marked and labeled with specialists' knowledge input. Such annotated videos can be directly used in medical video archive for surgical documentation.

6. CONCLUSION AND FUTURE WORK

In this paper, we presented an application for annotation of any kind of videos that need expert knowledge for the elucidation. We focused on the medical use case of endoscopic videos. The time that doctors have to spend with this tool to annotate the videos is extremely low. Furthermore, we showed that the tool is able to automatically create more annotations based on the initial annotation by the experts and how these annotations can be used. It provides a possibility for easy annotation for further analysis, documentation or lecturing. In the future, we will focus on gathering a large dataset and the usage of it in machine learning or computer vision algorithms. We further would like to expand the use case to other domains like sport.

7. ACKNOWLEDGMENT

This work is founded by the FRINATEK project "EONS" (#231687) and the iAD Center for Research-based Innovation (#174867) by the Norwegian Research Council.

8. REFERENCES

[1] S. Ameling, S. Wirth, D. Paulus, G. Lacey, and F. Vilarino. Texture-based polyp detection in colonoscopy. pages 346–350, 2009.

[2] W. Bailer and P. Schallauer. Detailed audiovisual profile: enabling interoperability between mpeg-7 based systems. In *Proc. of MMM '06*, 2006.

[3] M. K. Bashar, K. Mori, Y. Suenaga, T. Kitasaka, and Y. Mekada. Detecting informative frames from wireless capsule endoscopic video using color and texture features. In *Proc. of MICCAI'08*, pages 603–610, Berlin, Heidelberg, 2008. Springer-Verlag.

[4] M. Bastan, H. Cam, U. Gudukbay, and O. Ulusoy. Bilvideo-7: an mpeg-7-compatible video indexing and retrieval system. *MM, IEEE*, 17(3):62–73, 2010.

[5] M. Bertini, A. Del Bimbo, and C. Torniai. Automatic video annotation using ontologies extended with visual information. In *Proc. of ACM MM'05*, pages 395–398. ACM, 2005.

[6] B. Giritharan, X. Yuan, J. Liu, B. Buckles, J. Oh, and S. J. Tang. Bleeding detection from capsule endoscopy videos. In *Proc. of EMBS'08*, 2008.

[7] M. Guugenberger, M. Riegler, M. Lux, and H. Paal. Event understanding in endoscopic surgery videos. In *Proc. of ACM HuEvent'14*. ACM, 2014.

[8] S. Hare, A. Saffari, and P. H. S. Torr. Struck: Structured output tracking with kernels. In *Proc. of ICCV'11*, 2011.

[9] B. Li and M.-H. Meng. Tumor recognition in wireless capsule endoscopy images using textural features and svm-based feature selection. *ITBM, IEEE*, 2012.

[10] B. Li and M. Q. H. Meng. Computer-based detection of bleeding and ulcer in wireless capsule endoscopy images by chromaticity moments. *CBM*, 39(2):141–147, 2009.

[11] C.-Y. Lin, B. L. Tseng, and J. R. Smith. Videoannex: Ibm mpeg-7 annotation tool for multimedia indexing and concept learning. In *ICME '03*, pages 1–2, 2003.

[12] M. Lux and M. Riegler. Annotation of endoscopic videos on mobile devices: a bottom-up approach. In *Proc. of ACM MMSys'13*, pages 141–145. ACM, 2013.

[13] M. Riegler, M. Lux, V. Charvillat, A. Carlier, R. Vliegendhart, and M. Larson. Videojot: A multifunctional video annotation tool. In *Proc. of ACM ICMR'14*, page 534. ACM, 2014.

[14] R. Schroeter, J. Hunter, J. Guerin, I. Khan, and M. Henderson. A synchronous multimedia annotation system for secure collaboratories. In *Proc. of e-Science'06*, pages 41–41. IEEE, 2006.

[15] R. Socher and L. Fei-Fei. Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora. In *Proc. of CVPR '10*, pages 966–973. IEEE, 2010.

[16] T. Zhang, C. Xu, G. Zhu, S. Liu, and H. Lu. A generic framework for video annotation via semi-supervised learning. *MM, IEEE*, 14(4):1206–1219, 2012.