Extraction of Read Text Using a Wearable Eye Tracker for Automatic Video Annotation

Mizuki Matsubara
Osaka Prefecture University
1-1, Gakuen-cho, Naka, Sakai, Osaka, Japan
matsubara@m.cs.osakafu-u.ac.jp

Marcus Liwicki
German Research Center for Artificial Intelligence
Kaiserslautern, Germany
liwicki@dfki.uni-kl.de

Joachim Folz
German Research Center for Artificial Intelligence
Kaiserslautern, Germany
Joachim.Folz@dfki.de

Andreas Dengel
German Research Center for Artificial Intelligence
Kaiserslautern, Germany
Andreas.Dengel@dfki.de

Takumi Toyama
German Research Center for Artificial Intelligence
Kaiserslautern, Germany
takumi.toyama@dfki.de

Koichi Kise
Osaka Prefecture University
1-1, Gakuen-cho, Naka, Sakai, Osaka, Japan
kise@cs.osakafu-u.ac.jp

Ubicomp/ISWC'15 Adjunct, September 07-11, 2015, Osaka, Japan
 ISBN 978-1-4503-3575-1/15/09...$15.00
DOI: http://dx.doi.org/10.1145/2800835.2804333

Abstract
This paper presents an automatic video annotation method which utilizes the user's reading behaviour. Using a wearable eye tracker, we identify the video frames where the user reads a text document and extract the sentences that have been read by him or her. The extracted sentences are used to annotate video segments which are taken from the user's egocentric perspective. An advantage of the proposed method is that we do not require training data, which is often used by a video annotation method. We examined the accuracy of the proposed annotation method with a pilot study where the experiment participants drew an illustration reading a tutorial. The method achieved 64.5% recall and 30.8% precision.

Author Keywords
eye tracking, video annotation, life-logging, document image retrieval

ACM Classification Keywords
H.5.2 [User Interfaces]: Input devices and strategies

Introduction
Recently, various types of smart-glasses have been introduced by several manufacturers[1]. Wearing such a device, the users can have access to digital information resources

[1] http://viewer.tips/best-smart-glasses/

in everyday ubiquitous environments. A benefit of wearing such devices is that they enable the deployment of so-called life-logging systems. Life-logging refers to computational logging of personal life activities[2]. For instance, a typical application of life-logging systems can be implemented with a pair of smart-glasses where it records a video of the user's whole day activities from his or her egocentric perspective [10]. In this paper, we focus on a challenge of this type of video life-logging. An issue of this type of video life-logging is that the lengths of recorded videos become be very long. If one records a seven hours long video per day, it will end up with 49 hours long in one week, which is already too long to review. Thus, we must annotate and index the recorded video so that the user can easily retrieve them later [8].

We propose an automatic video annotation method using an eye tracker. To demonstrate the potential of eye tracking for video annotation, we focus on particular situations where the text the user read is closely related to his or her following actions. This type of situations includes cooking where the user reads a recipe, system maintenance where the user reads a manual, etc. During these activities, the users often read a textual instruction and perform the actions described in the instruction. The proposed method identifies the video segments in which the user reads a text document (e.g., a manual or tutorial) using eye tracking signals and a document image retrieval engine. Then, read text is extracted from the reading segments. The text extracted from each reading segment is used to annotate the following non-reading segment.

We conducted a pilot study where the participants drew an instructed illustration reading a given tutorial text document. We recorded the videos with eye tracking signals and tested the feasibility and accuracy of the proposed video annotation system.

[2] http://en.wikipedia.org/wiki/Lifelog

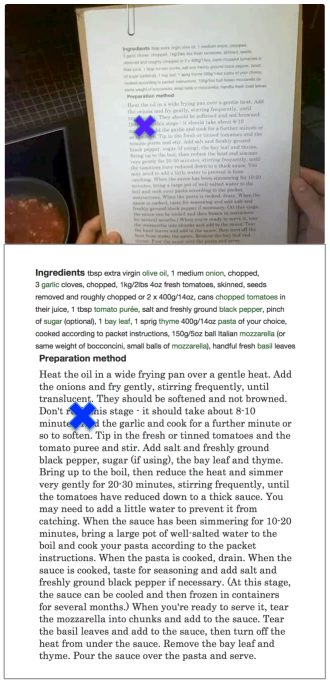
Related Work

Automatic video annotation and indexing has been a main topic for video data processing [8, 4]. Recent automatic annotation methods can reasonably perform well for known concepts [1]. Although traditional video annotations are done by using a single or few words, the recent work by Thomason et al. showed the promise of automatic annotation with natural language descriptions [11]. However, in order to annotate a video, this method requires training data with the labels of entities and activities occurring in the video. It is difficult to prepare such training data to provide all the entities and the activities in diverse real-world videos. Our proposed method can annotate a video with natural sentences and does not require training data since we extract text from a document.

Eye tracking has been utilized to detect and analyze the user's reading behaviour [3]. Recent improvements of wearable eye tracking devices, such as [5], pave the way of reading analysis in everyday ubiquitous scenarios [2]. Kunze et al. [7] showed that the gaze position on a printed document can be robustly calculated by combining document image retrieval and eye tracking. Prior work by Yoshitaka showed that analysis of eye movements benefits video indexing, retrieval and summarization [13]. We also show the potential of eye tracking for video annotation extending the document image retrieval-based reading analysis method presented in [7].

Proposed Method

To realize our automatic annotation system, we primarily have two questions to take into account, i.e., which part of a video and which text to be annotated. As previously



Ingredients tbsp extra virgin olive oil, 1 medium onion, chopped, 3 garlic cloves, chopped, 1kg/2lbs 4oz fresh tomatoes, skinned, seeds removed and roughly chopped or 2 x 400g/14oz, cans chopped tomatoes in their juice, 1 tbsp tomato purée, salt and freshly ground black pepper, pinch of sugar (optional), 1 bay leaf, 1 sprig thyme 400g/14oz pasta of your choice, cooked according to packet instructions, 150g/5oz ball Italian mozzarella (or same weight of bocconcini, small balls of mozzarella), handful fresh basil leaves

Preparation method

Heat the oil in a wide frying pan over a gentle heat. Add the onions and fry gently, stirring frequently, until translucent. They should be softened and not browned. Don't rush this stage - it should take about 8-10 minutes. Add the garlic and cook for a further minute or so to soften. Tip in the fresh or tinned tomatoes and the tomato puree and stir. Add salt and freshly ground black pepper, sugar (if using), the bay leaf and thyme. Bring up to the boil, then reduce the heat and simmer very gently for 20-30 minutes, stirring frequently, until the tomatoes have reduced down to a thick sauce. You may need to add a little water to prevent it from catching. When the sauce has been simmering for 10-20 minutes, bring a large pot of well-salted water to the boil and cook your pasta according to the packet instructions. When the pasta is cooked, drain. When the sauce is cooked, taste for seasoning and add salt and freshly ground black pepper if necessary. (At this stage, the sauce can be cooled and then frozen in containers for several months.) When you're ready to serve it, tear the mozzarella into chunks and add to the sauce. Tear the basil leaves and add to the sauce, then turn off the heat from under the sauce. Remove the bay leaf and thyme. Pour the sauce over the pasta and serve.

Figure 1: Gaze position mapping on a document image

mentioned, we focus on a video taken from the egocentric perspective. For this type of video, annotations that describe the activities or actions that the user performs are widely employed [9]. In this paper, we propose to annotate a particular video segment where the user performs an action. Segmentation is done by separating reading activities and others. First, we detect text reading states in the video. Then, the video is separated with reading and non-reading segments. Non-reading segments are annotated with the text from reading segments. In the following sections, we describe the individual process.

Reading Detection

To detect reading states, we use a wearable eye tracker which has an egocentric scene camera. In the experiments, we used a pair of SMI Eye Tracking Glasses[3] (ETG). For reading analysis, we extend the approach presented in [7]. This method uses a document image retrieval method called Locally Likely Arrangement Hashing (LLAH). First, we prepare a document image database which contains document images and document meta-data. The document meta-data contains the meta-information of the document, e.g., list of words occurring the document, region of individual word, etc. When the user reads a text document, the scene camera of ETG captures the document image. The captured image is used to retrieve the current reading document from the database using LLAH. If the document is retrieved from the database, we also calculate the pose of the document in the image. Accordingly, the gaze position given by the ETG is mapped to the retrieved document image as shown in Figure 1.

To separate reading segments from the video, we need to detect the user's reading state. When the user gaze is detected on the same document page for certain amount of duration, we assume that the user is reading the document. For the reading detection, we adapt the method presented in [12] where the user attention to a particular object is detected by counting the number of frames that have the same recognition output. Similarly, we detect that the user is reading the document X when the number of frames that have the retrieval results X exceeds T_{dur} without having T_{noise} frames of irrelevant results in between.

Extraction of Read Text

Once a reading state is detected, we extract read text from the document. As previously mentioned, the gaze position can be mapped to a document image when retrieval is successfully done. Since we have the meta-data of word region, we can calculate the geometrical (Euclidean) distance of the gaze position to individual words. We extract the sentences that contain the words nearest to the individual gaze positions as the read text.

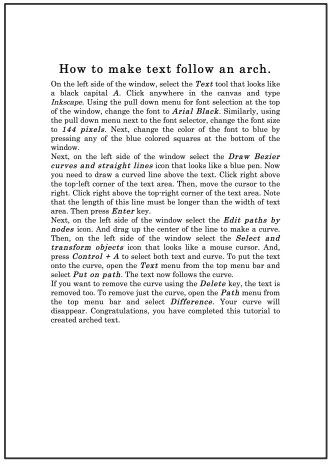
Video Annotation

In the situations that we focus on, the user reads a textual instruction, such as a system manual, a cooking recipe and a software tutorial, and performs the actions and tasks that are described in the instruction. Thus, we consider that the text that is read by the user before the individual action is closely related to the performed action. Our proposed method separates a video with reading and non-reading segments and annotate the non-reading segment with the text extracted from the preceding reading segment. Note that the read text may not always be appropriate to annotate the following segment. We test the feasibility of the proposed method in the following section.

Pilot Study

We conducted three pilot studies to evaluate the proposed method. In the studies, we chose an illustration drawing

[3] http://www.eyetracking-glasses.com/



How to make text follow an arch.

On the left side of the window, select the ***Text*** tool that looks like a black capital *A*. Click anywhere in the canvas and type *Inkscape*. Using the pull down menu for font selection at the top of the window, change the font to ***Arial Black***. Similarly, using the pull down menu next to the font selector, change the font size to ***144 pixels***. Next, change the color of the font to blue by pressing any of the blue colored squares at the bottom of the window.

Next, on the left side of the window select the ***Draw Bezier curves and straight lines*** icon that looks like a blue pen. Now you need to draw a curved line above the text. Click right above the top-left corner of the text area. Then, move the cursor to the right. Click right above the top-right corner of the text area. Note that the length of this line must be longer than the width of text area. Then press ***Enter*** key.

Next, on the left side of the window select the ***Edit paths by nodes*** icon. And drag up the center of the line to make a curve. Then, on the left side of the window select the ***Select and transform objects*** icon that looks like a mouse cursor. And, press ***Control + A*** to select both text and curve. To put the text onto the curve, open the ***Text*** menu from the top menu bar and select ***Put on path***. The text now follows the curve.

If you want to remove the curve using the ***Delete*** key, the text is removed too. To remove just the curve, open the ***Path*** menu from the top menu bar and select ***Difference***. Your curve will disappear. Congratulations, you have completed this tutorial to created arched text.

Figure 2: Text tutorial used in the pilot study

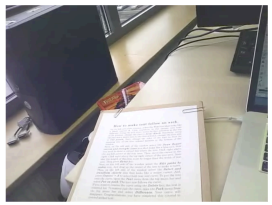


Figure 5: How participant 2 read document

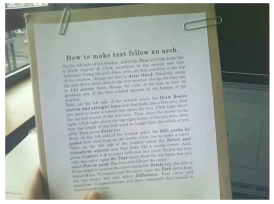


Figure 6: How participant 3 read document

scenario where the participants drew an arch-shape text with a graphic design software called *Inkscape* reading the textual tutorial shown in Figure 2. The tutorial was written in English. The number of pages was one. The font sizes of the title and the body were 22 and 14 pt, respectively. The number of sentences was 23. We had five participants in total. For each participant, we calibrated the eye tracker before the recording. After the calibration, we asked them to carry out the task written in the tutorial. In order to record the sentences actually read by the participants as ground truth data, we asked them to read aloud the text. Except this, we asked them to perform the task naturally. Note that the participants were allowed to carry out the task as they liked. Some participants read the tutorial and performed the task step by step, whereas some other participants first read the tutorial through and performed the task. The lengths of the recorded videos were 405, 458, 393, 370 and 354 seconds, respectively. The frame rate was 24.5 fps.

First, we evaluated the reading detection using the LLAH. The LLAH has the Gaussian parameter which controls the blurring scale. The larger the Gaussian parameter is, the more the scene image is blurred, which performs well when the document paper is close to the user. We tuned the parameter for each participant. As a result, we found that the appropriate parameters differ from one to another. We set the appropriate parameter for each participant and tested the reading detection. Figures 3 and 4 show the result of

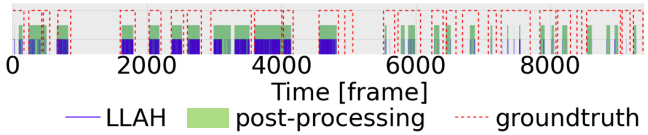


Figure 3: Reading detection and reading segment (participant 2)

the reading detection of participant 2 and 3, respectively. The x-axis shows the frame index of the video. The blue boxes show the results of document image retrieval. Furthermore, the green boxes show the results of reading detection (post-processing with the adaptation of [12]). On average, the recall, i.e., the ratio of the number of successfully retrieved document frames to the number of ground truth frames, was 80.0%[4]. However, the recalls varied between the participants. The recall of the participant 2 was 56.3%, which was the worst. As shown in Figure 3, the reading detection did not work well for the participant 2 after approx. 5000 frames. On the other hand, the proposed method can successfully detect reading segments for the participant 3 as shown in Figure 4 (the recall was 90.5%). Figures 5 and 6 show images from the recorded videos when the participant 2 and 3 read the document. The participant 2 put the document on to the desk during the experiment. Therefore, the distance and angle between the camera and document were too far and steep to retrieve the document by the LLAH. On the other hand, the participant 3 read the text by holding the printout with his hand, which is easy to retrieve for the LLAH. Therefore, one can infer that if the positions of the camera and the document are appropriate, the reading detection method can perform reasonably well.

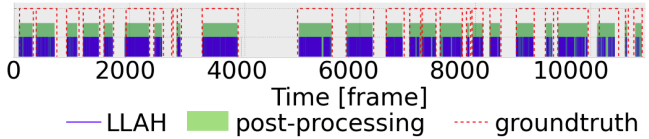


Figure 4: Reading detection and reading segment (participant 3)

[4]The LLAH never output incorrect documents in this study. Therefore, the precision was 100.0%.

Next, we examined the accuracy of the extraction of read text. In this experiment, we used manually segmented videos to rule out video segmentation errors caused by the failure of the reading detection to see the potential of the extraction method. Thus, we assume that these segments are ideal for extraction of read text. Since the participants read aloud the text, we regarded the verbal signals as the actual sentences read by them (i.e., ground truths)[5]. We compared the ground truths and the sentences extracted by the proposed method and calculated the recall and precision. The recall is the ratio of the number of the correctly extracted sentences to the number of sentences from the ground truths and the precision is the ratio of the number of the correctly extracted sentences to the total number of the extracted sentences. The recall was 69.5% and the precision was 46.2%. These results show that the extracted sentences may contain noisy results and sometimes miss read text.

Additionally, we also analyzed whether the sentences read by the participants actually described the following actions. We calculate the recall and precision as follows: the recall is the ratio of the number of the read sentences that actually describe the performed actions to the number of the sentences that describe the performed actions written in the tutorial. The precision is the ratio of the number of the read sentences that actually describe the performed actions to the number of the read sentences. On average, the recall was 86.8% and the precision was 61.0%. Thus, we consider that the read sentences describe the following actions to some extent.

Finally, we tested whether the extracted sentences can be used as the annotations. In this test, we replaced the read sentences used in the previous analysis with the extracted sentences by the proposed method and re-calculated the recall and precision. The recall was 64.5% and the precision was 30.8%. The recall shows that more than a half of the segments were annotated correctly by the proposed method. On the other hand, the precision was not really high. One can infer that if we implement a retrieval system using this method, the retrieval results may contain two irrelevant video segments out of three videos. Choosing one from three results is easier than manually searching it from very long video streams. For that reason, we consider that extraction of read text is effective for an automatic annotation system. The reason why the precision of this test was lower than 86.8% was that it extracted the text that the participants did not read. The recall dropped when the LLAH or the eye tracking (gaze position) had failures.

[5]We also checked the recorded gaze data to confirm that the participants actually read.

Summary

In this paper, we proposed a method for automatic video annotation using a wearable eye tracker with an egocentric perspective camera. The proposed method separates reading segments and non-reading segments and extracts read text from the reading segments. Annotations to non-reading segment can be selected from the sentences extracted from the preceding reading segment. We examined the feasibility and accuracy of the proposed method. In the annotation test, the recall was 64.5% and the precision was 30.8%. The future work is to improve the accuracy of read text extraction. For example, we can check whether the user is actually performing the written action by recognizing the user's activity.

Acknowledgements

This work has supported in part by the JST CREST, JSPS Kakenhi (25240028, 15K12172) and HySociaTea project, German Federal Ministry of Education and Research (BMBF),

grant no. 01IW14001.

REFERENCES

1. Nicolas Ballas, Benjamin Labbé, Borgne H. Le, Philippe Gosselin, David Picard, Miriam Redi, Bernard Merialdo, Boris Mansencal, Jenny Benois-Pineau, and Stéphane Ayache, IRIM at TRECVID 2014: Semantic Indexing and Instance Search, In Proceedings of TRECVID, 2014.
2. Andreas Bulling, Jamie A. Ward, Hans Gellersen, and Gerhard Tröster, Robust Recognition of Reading Activity in Transit Using Wearable Electrooculography, In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 19–37, 2008.
3. Andrew T. Duchowski, A breadth-first survey of eye-tracking applications. Behavior Research Methods, Instruments, & Computers 34,4, 455–470, 2002.
4. S.L. Feng, Raghavan Manmatha, and Victor Lavrenko, Multiple bernoulli relevance models for image and video annotation, In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2, II-1002–II-1009, 2004.
5. Moritz Kassner, William Patera, and Andreas Bulling, Pupil: an open source platform for pervasive eye tracking and mobile gaze-based interaction, In Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, ACM, 1151–1160, 2014.
6. Eriko Kawasaki, Cognitive psychology of document understanding, 47–98, 2014.
7. Kai Kunze, Hitoshi Kawaichi, Kazuyo Yoshimura, and Koichi Kise, Towards inferring language expertise using eye tracking, CHI'13 Extended Abstracts on Human Factors in Computing Systems, 217–222, 2013.
8. Hisashi Miyamori, and Shun-ichi Iisaku, Video annotation for content-based retrieval using human behavior analysis and domain knowledge, In Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition 2000, 320–325, 2000.
9. Mohammad Moghimi, Pablo Azagra, Luis Montesano, Ana C. Murillo, and Serge Belongie, Experiments on an RGB-D wearable vision system for egocentric activity recognition. In 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE, 611–617, 2014.
10. Kieron O'Hara, Mischa M. Tuffield, and Nigel Shadbolt, Lifelogging: Privacy and empowerment with memories for life. Identity in the Information Society 1,1, 155–172, 2008.
11. Jesse Thomason, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Raymond Mooney, Integrating language and vision to generate natural language descriptions of videos in the wild, In Proceedings of the 25th International Conference on Computational Linguistics (COLING), 2014.
12. Takumi Toyama, Thomas Kieninger, Faisal Shafait, and Andreas Dengel, Gaze guided object recognition using a head-mounted eye tracker, In Proceedings of the Symposium on Eye Tracking Research and Applications, ACM, 91–98, 2012.
13. Atsuo Yoshitaka, Image/Video Indexing, Retrieval and Summarization Based on Eye Movement, In Proceedings of the 4th International Conference on Computing and Informatics, ICOCI 2013, 15–21, 2013.