SICE Annual Conference 2010
August 18-21, 2010, The Grand Hotel, Taipei, Taiwan

Fully-Automatic Annotation of Scene Vidoes

Establish Eye Tracking Effectively in Various Industrial Applications

Kai Essig[1,2], Norbert Sand[3], Thomas Schack[1,2]
[1]Center of Excellence "Cognitive Interaction Technology"
[2]Faculty of Psychology and Sport Sciences
Bielefeld University
PO Box 100131, 33615 Bielefeld, Germany
kai.essig@uni-bielefeld.de

Jörn Künsemöller[3], Matthias Weigelt[4], Helge Ritter[1,3]
[3]Faculty of Technology
Bielefeld University
PO Box 100131, 33615 Bielefeld, Germany
[4]Institute of Sport Science, Saarland University, Germany

Modern mobile eye-tracking systems record participants' gaze behavior while they move freely within the environment and have haptic contact with the objects of interest. Whereas these mobile systems can be easily set up and operated, the analysis of the stored gaze data is still quite difficult and cumbersome: the recorded scene video with overlaid gaze cursor has to be manually annotated - a very time consuming and error-prone process - preventing the use of eye-tracking techniques in various application fields. In order to overcome these problems, we developed a new software application (JVideoGazer) that uses translation, scale and rotation invariant object detection and tracking algorithms for a fully-automatic video analysis and annotation process. We evaluated our software by comparing its results to those of a manual annotation using scene videos of a typical day-by-day task. Preliminary results show that our software guarantees reliable automatic video analysis even under challenging recording conditions, while it significantly speeds up the annotation process. To the best of our knowledge, the JVideoGazer is the first software for a fully-automatic analysis of gaze videos. With it, modern eye-tracking techniques can be effectively applied to real world situations in various application fields, like research, control, quality management, human-machine interactions, as well as information processing and other industrial applications.

Keywords: eye tracking; object recognition; information processing; quality management; machine control;

I. INTRODUCTION

With the technical progress and the miniaturization of electronic devices, mobile eye tracking became more and more popular in recent years. The term eye tracking signifies the temporal recording of eye movements when participants look at 2D- or 3D stimuli. The analysis of eye movements yields valuable insights into the cognitive processes underlying scene perception ("eyes are a window to the mind") [1,2] – providing answers to questions like: "Which information are perceived as valuable?", "What is the temporal order of perceived objects?", "Where had the participants problems to understand important scene information (signified by a high number of fixations or long fixation durations)?". Traditional eye-tracking studies are applied with stimuli on a computer screen allowing a controllable setup [1]. These setups have quite a lot of disadvantages: they are restricted to indoor applications, artificial stimuli are used, and the participants have to sit in front of a screen, preventing them to interact naturally with their environment. We live in a dynamic environment and have to extract relevant information quite fast from the surroundings in order to come up with the right decisions. For example, in sports, players have to deduce opponents' intensions from the perceived body movements for an adequate counterattack. Thus, recently academic research shifted towards the analysis of human perceptual and cognitive processes in dynamic environments, i.e., under more realistic conditions [11,12,13]. This leads to a bunch of new problems amongst others: every participant perceives the world from different perspectives, the lighting conditions change, relevant objects move over time and they may be occluded. Furthermore, in order to track participants' gaze positions in dynamic environments, eye-tracking systems have to meet complete different demands.

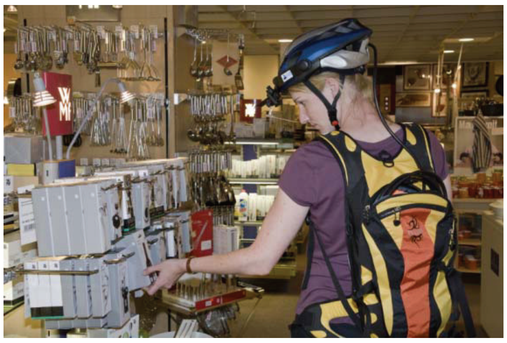


Figure 1. Mobile eye-tracking at the point-of-sale (POS) [7].

Fully mobile, head mounted eye-tracking systems were recently developed to overcome these demands [7]. Mobile systems consist of a scene and an eye camera attached to a (bicycle-) helmet or headset (see Fig. 1). The computer for the control of the eye tracker as well as for data recording is stored in a backpack and is connected via a cable to the cameras. This allows the participant to move freely in an unlimited working range and to grasp, touch and manipulate the objects of interest. Thus, with the fully mobile systems, eye-tracking studies can be shifted from the laboratory into the real world. Typical application fields are amongst others: shop studies, analysis of athletes' gaze behavior in various sport games, human-human and human-machine interactions.

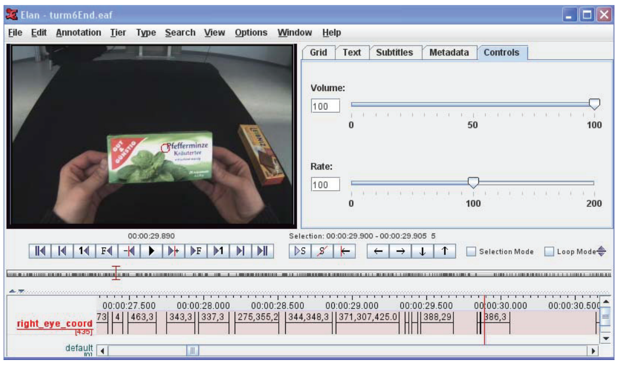

Elan - turm6End.eaf
File Edit Annotation Tier Type Search View Options Window Help
Grid Text Subtitles Metadata Controls
Volume:
Rate:
Selection Mode
Loop Mode
right_eye_coord
default

Figure 2.The ELAN annotation software [6].

Whereas the hardware of the mobile systems is quite developed in order to allow the recording of eye movements under various environmental conditions (indoors and outdoors), methods for the analysis of the recorded data are still in their infancy. The eye tracker records a scene video (see Fig. 2) with overlaid gaze positions (marked by a cross) and a file containing the x- and y-position of the gaze point (in the coordinate system of the scene video), as well as the pupil size together with the corresponding timestamps. The recorded gaze positions are not related to the objects in the scene video, i.e., it is not evident how many fixations (including their time frames) are directed toward an object of interest over the whole video sequence.

II. Related Work

In order to extract this information, the scene video has to be manually annotated - a very time-consuming and error-prone process. A professional tool that supports the creation of complex annotations on video and audio resources is ELAN (EUDICO Linguistic Annotator) [6], developed by the Max Planck Institute for Psycholinguistics in Nijmegen, Netherlands (see Fig. 2). ELAN's GUI (graphical user interface) shows multiple, hierarchically interconnected layers to create various annotation units, the so called tiers. For the annotation of gaze videos, the scene video synchronized with the corresponding fixation data (converted to the ELAN format) has to be loaded into the program (see Fig. 2). The object of interest and the corresponding gaze data (marked by a red line in the fixation tier) can then be observed by the user. Compared to the frame-by-frame annotation, ELAN depicts the relevant fixation data in contiguous block, on which the user can concentrate during the annotation process.

A more automated approach uses the shortest distance rule [10, 11]: at each point in time the distance between the gaze position and the central position of each object in the scene video is calculated. The observed object is the one with the shortest distance to the actual gaze position. Thus, the central positions of all objects in a scene must be extracted, while object extensions are not considered. Furthermore, every fixation is matched to the closest object – even when the participant does not look at an object at all.

Other approaches make extensive use of dynamic AOIs (areas of interest) to represent the position and content of important objects over time. The professionally distributed BeGaze2^{TM} software [7] uses semi-automatic annotation of gaze videos. It reduces the tedious annotation effort by interpolation of object positions over frame intervals. Important objects are marked by manually superimposed polygons at particular scene positions (e.g., when the object is clearly visible) over the considered time interval. The positions of the objects in the single frames over the considered time interval are then interpolated by the size and positions of these polygons. This approach works best when the object moves linearly within the considered scene sequence. If this is not the case, much more polygon markers on different frames have to be manually provided by the user, or the markers have to be shifted with the mouse in order to better fit the object boundaries - a quite time-consuming process. Additionally, the user has to select suitable starting, intermediate and end polygon markers that reliably describe the objects' movements within the scene video.

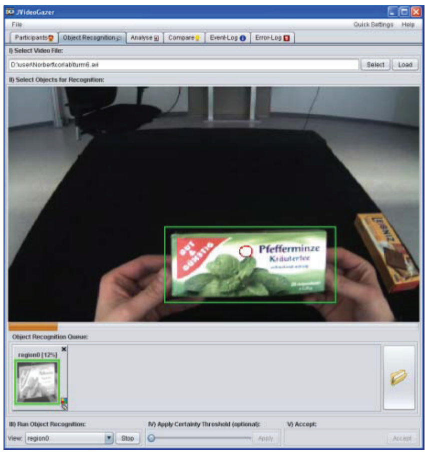
Another approach using areas of interest is DynAOI [8], a collection of scripts under the GNU General Public License. The automatic annotation of gaze videos is realized via the animation of 3D virtual models. First, however, adequate models and animations must be manually created with the complex 3D graphics application Blender [9], which is not bundled with DynAOI. The recorded 2D gaze coordinates of the eye tracker are first transformed into the 3D animation space. Then DynAOI matches the recorded gaze positions with the 3D models underlying the presented videos and points out the observed objects. In order to annotate real scene videos recorded by a mobile eye-tracking system, first adequate 3D models of the interesting objects have to be designed and animated with the Blender software. This is quite a time-consuming process. It is only worthwhile, when the material has to be prepared once and the video is shown to several participants, or when the objects are tracked automatically, i.e., in this case the animation does not have to be manually generated. Thus, DynAOI is suitable for artificial stimuli and in virtual environments, but not for real world applications. Furthermore, the complex program Blender is not intuitively to use, especially for laymen.

Therefore, we developed a fully-automatic and modular annotation tool for the analysis of gaze videos where the objects can be intuitively marked in the scene video. This approach is described in the following section. In order to prove its applicability, we compared the results of our program with those of a manual annotation in a typical day-by-day scenario in Section IV.

III. Fully-Automatic Annotation Software

A. The Program JVideoGazer

The JVideoGazer is a modular, multi-threaded software for fully-automatic video analysis written in Java [15]. The program applies an automatic object recognition and tracking algorithm (see below). The user can select multiple target objects of interest, which will then be automatically tracked over all frames of the scene video, by "roping" a lasso with the mouse around them. Selected objects are shown as thumbnails in the lower part of the program, where they can be labeled and marked by a bounding box in a particular color (see Fig. 3). Via two sliders, the quality of the object recognizer (and with it



I) Select Video File: D:\user\Norbert\coriabturm6.avi

View: region0

Figure 3. JVideoGazer: Full-automatic object tracking of the tea package.

the speed) and the threshold for the detection quality can be adjusted, to optimally configure the object recognizer to various recording conditions. The program then automatically tracks the objects over the whole scene video – optionally a bounding box is drawn around the object to provide visual feedback (see Fig. 3). As the result, the software provides for each selected object the coordinates of the bounding box (within the scene video) and the detection certainty, as well as the corresponding frame number and timestamp. In a final step, the software compares the recorded gaze data with the output of the tracking process and displays the results, i.e., number of fixations, average and cumulative fixation duration, separately for each user-selected object. The data of several participants can be sequentially analyzed and their results are comparatively displayed for the single parameters, allowing further statistical analysis. The modular program design allows an easy integration of other object recognition algorithms or the analysis of further eye-tracking parameters.

B. *The Object Recognizer*

The object recognition algorithm is based on SURF (Speeded-Up Robust Features), a high-performance scale and rotation invariant interest point detector and descriptor [3, 4]. Interest points are computed for the user-selected region (i.e., tracked object) and each frame in the scene video by a Fast-Hessian detector. This is done for every frame in scale space for scale invariance of the interest points. In order to guarantee rotation invariance, a Haar-wavelet-transformation is calculated in x- and y-direction within a circular neighborhood of radius $6s$ around the interest point, where s is the scale where the interest point was detected. Then all the horizontal and vertical wavelet responses (weighted with a Gaussian) in a sliding orientation window of size $\frac{\pi}{3}$ are summed up, yielding a local orientation vector. The direction of the sliding window with the longest vector defines the orientation of the interest point [3]. In a next step, a square Q is drawn around the interest point, with a side length of $20s$ and aligned to the prior calculated orientation. Q is then divided into 16 (4 x 4) sub-regions. For each sub-region, Haar-wavelet responses at 5 x 5 regularly spaced sample points are calculated. The wavelet responses (in horizontal and vertical direction) are summed up over each sub-region resulting in a four-dimensional descriptor vector v for its underlying intensity structure. Concatenating the descriptor vectors v of all 16 sub-regions result in the 64 dimensional feature vector (descriptor) for each interest point. The energy of the whole descriptor is normalized to obtain some invariance to illumination. Matching interest points are determined by using the Euclidean distance on the feature descriptors as a similarity measure [1]. In order to find suitable source and target points for the homography matrix [5], which is used to determine the bounding rectangle of the detected object in the video, a principle component analysis (PCA) [14] is calculated separately for every frame only on the matching points of the user-selected region and the detected interest points in the scene video. The suitable points for the transformation, as well as information about rotation and scaling of the tracked objects, are determined from the first two principle component vectors.

IV. EVALUATION STUDY

In order to evaluate our software, we recorded the eye movements of four participants when they sit in front of a table, grasp three objects of different size (i.e., book, cookies and tea) from the left side, moving them slightly and finally put them down on the right side of the table (see Fig. 3). We used a SMI iViewX HED mobile eye-tracker at a sampling rate of 200 Hertz, with a gaze position accuracy $< 0.5°$-$1°$ [7]. Each scene video has a resolution of 376 x 240 pixels at 25 fps and a duration of 2-3 minutes. The scene videos were manually labeled with the ELAN annotation software [6]. Additionally, all videos were analyzed fully-automatically using the JVideoGazer on a quad core computer with 3.2 GHz and 4 GByte DDR3-RAM.

A. *Results*

Fig. 4 compares the results for the number of fixations on the three objects in the four scene videos between the manual and fully-automatic analysis. The results of the JVideoGazer closely resemble those of the manual annotation. The deviation is on average 7.8%, ranging from 0 to 7 fixations in absolute values. Note that we compare at this point the results of the JVideoGazer to a subjective and imprecise manual annotation – we do not address which result resembles more closely the exact number of fixations on the three objects in the videos. In order to answer this question, we would need further manual annotations from different people. As expected, the results are best for larger objects (book), because the SURF algorithm finds more interest points improving the detection process, but it works also quite good for smaller ones (tea). The absolute number of the counted fixations on each object are slightly lower for the automatic annotation, because the objects' contours and identification labels blur, when they are put down quite fast on the right side of the table. Thereby matches of interest points are difficult to detect and the quality of the object tracking decreases.

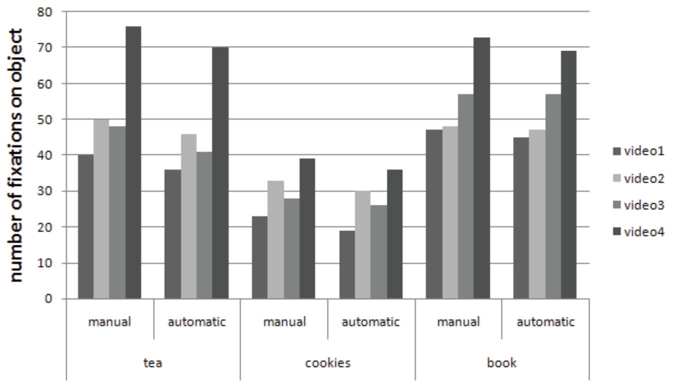


Figure 4. Results of the manual and fully-automatic analysis.

V. DISCUSSION

We implemented a new software application for the fully-automatic analysis of gaze videos in order to overcome the time consuming and error-prone disadvantages of manual annotation. Preliminary results showed that our software closely resembles the data of a manual annotation even when the resolution of the scene video is quite small, which makes the identification and matching of interest points quite difficult for the object recognizer. The manual annotation of each video took around 60-75 minutes, depending on its length. The automatic annotation lasts around 5-10 minutes for three objects on the whole scene video, resulting in significant time-savings (factor 9). Additionally, the software can be run around-the–clock, whereas humans get tired after a while and make more annotation errors.

Now, eye-tracking techniques can be used effectively in various industrial application fields, which was not possible before because of the high cost and time expenses. The additional analysis of eye movements provide new insights into the visual perception processes of customers, users and employees in order to improve user experience, workplace safety, customer and job satisfaction. Typical application fields are amongst others: car driving, machine operation, safety and ergonomics in the workplace, orientation in multi-storey car parks, and shop usability.

To the best of our knowledge, our software is the first one providing a fully-automatic video analysis of gaze recordings based on object recognition and tracking algorithms. We are only at the beginning to address this quite complex topic - much more research, development and evaluation has to be done. In the analysis phase we encountered different problems: I.) Manual annotation is very subjective, especially when the eye tracker is not correctly calibrated, II.) The performance of the object recognizer strongly depends on the recording conditions, such as illumination, occlusion, scale and viewpoint, as well as on the quality and resolution of the recorded scene video. In order to tackle these problems, the program allows to choose between several object recognizers being more suitable for particular scene recordings. For example, a histogram based object recognizer may give better results when the objects are blurred because of fast movements. Additionally, the coordinates of the user-selected regions can be saved in order to repeat the tracking process with more or less close-fitting user-selected regions. We already started further investigations with a higher number of participants, annotators, additional object recognition algorithms and under different recording conditions. This allows us to apply eye tracking as a central method for investigating the perceptual and cognitive processes underlying visual scene perception under dynamic conditions, i.e., under natural conditions – a topic of increasing importance for the field of industrial use and academic research.

ACKNOWLEDGMENT

This work was supported by a grant of the Center of Excellence “Cognitive Interaction Technology” (CITEC). Thanks to Thies Pfeiffer for his technical support.

REFERENCES

[1] K. Essig, Vision-Based Image Retrieval (VBIR) – A new eye-tracking based approach to efficient and intuitive image retrieval. Saarbrücken, Germany: VDM Verlag, 2008.

[2] A.T. Duchowski, Eye Tracking Methodology: Theory and Practice. London, England: Springer, 2003.

[3] H. Bay, T. Tuytelaars, and L. Van Gool,“Surf: Speeded up robust features,” European Conference on Computer Vision, vol. 1, pp. 404-417, 2006.

[4] C. Evans, “Notes on the OpenSURF Library,” CSTR-09-001, University of Bristol. January 2009.

[5] D. Kriegman, “Homography Estimation,” Lecture Computer Vision I, CSE 252A, Winter 2007.

[6] ELAN – Annotation Tool: http://www.lat-mpi.eu/tool/elan/.

[7] Sensomotoric Instruments: http://www.smivision.com.

[8] F. Papenmeier, and M. Huff, “DynAOI: A tool for matching eye-movement data with dynamic areas of interest in animations and movies,” Behavior Research Methods, vol. 42, pp. 179-187, 2010.

[9] Blender a 3D Grafics Applicarion :http://www.blender.org

[10] H.M. Fehd, and A.E. Seifert, “Eye movements during multiple object tracking: Where do particioants look?”, Cognition, vol. 108, pp. 201-209, 2008.

[11] G.J. Zelinsky, and M.B. Neider, “An eye movement analysis multiple object tracking in realistic environment,” Visual Cognition, vol. 16, pp. 553-566, 2008.

[12] J.M. Henderson, Real world scene perception. New York:, USA: Psychology Press, 2005.

[13] K. Rayner, T.J. Smith, G.L. Malcolm, and J.M. Henderson, “Eye movements and visual encoding during scene perception,” Psychological Science, vol. 20, pp. 6-10, 2009.

[14] I.T. Jolliffe, Principal Component Analysis, 2nd ed. New York, USA: Springer, 2002.

[15] Java Programming Language: http://java.sun.com.