Real-Time 3D Gaze Analysis in Mobile Applications

Jan Hendrik Hammer
Karlsruhe Institute of Technology
Department of Informatics
Institute for Anthropomatics
Vision and Fusion Laboratory
jan.hammer@kit.edu

Michael Maurus
Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB
michael.maurus@iosb.fraunhofer.de

Jürgen Beyerer
Fraunhofer Institute of Optronics, System Technologies and Image Exploitation IOSB
juergen.beyerer@iosb.fraunhofer.de

Figure 1: *Left image:* Visualization of 7381 raw gaze samples as violet spheres. The red spheres show the resulting 489 fixations. *Right image:* Normalized heat map visualization of the same gaze data. The data was collected during a study in the Valencian Kitchen.

ABSTRACT
This paper presents a system for real-time analysis of 3D gaze data arising in mobile applications. Our system allows users to freely move in a known 3D environment while their gaze is computed on arbitrarily shaped objects. The scanpath is analyzed fully automatically using fixations and areas-of-interest – all in 3D and real time. Furthermore, the scanpath can be visualized in parallel in a 3D model of the environment. This enables to observe the scanning behavior of a subject. We describe how this has been realized for a commercial off-the-shelf mobile eye tracker utilizing an inside-out tracking mechanism for head pose estimation. Moreover, we show examples of real gaze data collected in a museum.

Categories and Subject Descriptors
H.1.2 [**MODELS AND PRINCIPLES**]: User/Machine Systems – *Human factors*. H.5.1 [**INFORMATION INTERFACES AND PRESENTATION**]: Multimedia Information Systems - *Artificial, augmented, and virtual realities*. H.5.2 [**INFORMATION INTERFACES AND PRESENTATION**]: User Interfaces - *Input devices and strategies, Interaction styles*.

General Terms
Human Factors

Keywords
Mobile gaze analysis, 3D gaze points, 3D fixations, gaze-based interaction

ETSA '13, August 29 - 31 2013, Cape Town, South Africa

http://dx.doi.org/10.1145/2509315.2509333

1. INTRODUCTION
Gaze analysis provides detailed information on the visual attention of a person. Further insights into cognition can be revealed and the information can contribute to interest or intention deduction. That is why gaze analysis is interesting for a variety of applications and different fields of research. Especially due to the usage of mobile unobtrusive eye trackers, experimental setups can stay closer to reality and eye tracking becomes possible in settings where it was not imagined to be realizable years ago. But still, commercial off-the-shelf eye trackers without any expensive external tracking equipment do neither enable for 3D gaze point computation, nor have solutions for fully automated gaze analysis in environments with real 3D objects. That is why we developed a system enabling gaze analysis in mobile applications, to which commercial off-the-shelf eye tracker can be connected, if they fulfill several requirements. These are detailed in section 2.

Having real-time 3D gaze data at hand, a wide range of mobile applications can get started utilizing implicit and/or explicit gaze based interaction to make human computer interaction more intuitive. In the European project ARtSENSE[1] an active museum assistant is developed. The user wears an optical see-through head-mounted device with eye-tracking functionality[2]. Gaze analysis is used for implicit interest detection. While a user is freely viewing the environment, the system detects the visually most attended artworks. This information yields as basis for reasoning about what information can be provided visually and acoustically to the user. Explicit interaction is needed when mouse and keyboard are no option for interaction. This is usually the case in mobile applications. For example in a large-scale multi-display environment users interact with displays that are too far away to be touched. Selection by gaze will be a promising alternative to exhausting pointing gestures.

[1] Augmented RealiTy Supported adaptive and personalized Experience in a museum based oN processing real-time Sensor Events

[2] http://www.interactive-see-through-hmd.de/

In section 2 we talk about the issues that need to be regarded on the way to 3D gaze analysis in real environments with arbitrarily shaped objects. In section 3 we demonstrate how we interface a commercial off-the-shelf eye tracker to our gaze analysis system. Then, we explain the implemented algorithm for gaze movement computation in section 4. Finally, we show real 3D gaze data and analysis results collected in the Valencian Kitchen[3] in section 5 and conclude in section 6.

2. 3D GAZE POINT COMPUTATION

The approach we use to register 3D gaze points to the environment is known as geometry-based point of regard estimation [1]. The line-of-sight of one eye is reconstructed and intersected with a 3D model of the environment. Starting from the eye, the first intersection in viewing direction with the 3D model of the environment yields the 3D gaze point. Of course, this assumption can be wrong, especially when transparent objects are in the scene, but for applications like e.g. in museums or multi-display environments the approach is working quite well.

For the reconstruction of the line-of-sight three requirements exist: First, the *viewing direction* must be computed. Second, the *3D eye position* must be estimated in world coordinates to find a point on the line-of-sight (e.g. the nodal point of the optical system of the eye). Third, a correct *3D model of the environment* must be available. These three requirements comprise different challenges which are summarized below.

2.1 Viewing Direction

Computation of the viewing direction is part of the eye tracking process to which also pupil detection and nodal point estimation belong [2]. In case of static eye tracking devices the line-of-sight is already known since the eye tracker usually stays at a fixed location in the environment. The 6 degrees of freedom (DOF) of its pose can be measured manually.

2.2 Eye Position Estimation

Accurate 6 DOF estimation of the eye tracking device is one of the most difficult issues regarding mobile eye tracking. Pfeiffer [1] distinguishes between *inside-out* and *outside-in* optical tracking approaches. For inside-out tracking, the scene camera of the eye tracking device is utilized to enable e.g. a marker tracking based on image processing algorithms. The markers must be placed in the environment and their positions and orientations must be known to get 2D-3D correspondences for the estimation of the extrinsic parameters of the scene camera. Instead of fiducial markers, the environment itself can be used as markers [3]. The other approach for 6 DOF estimation of the eye tracker called outside-in tracking needs additional external hardware including cameras distributed over the environment. The extrinsic parameters of the external cameras must be calibrated in advance. Most systems further use a marker attached to the device to realize more accurate tracking. After the determination of the extrinsic parameters of the scene camera and the nodal point of the eye relative to the camera, the eye position can be computed.

While outside-in approaches with a marker attached to the device allow for a very accurate 6 DOF estimation, the size of the area in which a user can freely move is restricted much more and additional weight is put on the eye tracker. If the inside-out pose estimation of an eye tracker allows for good enough accuracy concerning the application, there is no need for installing expensive external tracking mechanisms. This makes inside-out tracking much more attractive. Nevertheless, if the inside-out tracking works with explicit markers, these act as additional stimuli and may disturb the gaze behavior.

2.3 Creation of a Model of the Environment

The third requirement for a geometry-based 3D gaze point computation is the availability of an up-to-date model of the environment. This can either be constructed manually in a 3D modeling software or more accurate and less time consuming be created using 3D reconstruction methods e.g. like those provided by KinectFusion [4]. Assuming that areas-of-interest (AOIs) are used for point of regard to object correspondence, the created models must be extended with information about the belongings of polygons to AOIs. This can for example be done in any 3D modeling software by creating each AOI as a virtual object surrounding its real counterpart.

3. INTERFACING THE DIKABLIS WIRELESS EYE TRACKER

The Dikablis Wireless[4] is a monocular eye tracker using an active sensor with an infrared diode to observe the left eye of a person. Additionally, a scene camera is attached to capture the scene in front of the user. Both scene and eye camera can be rotated to adjust the eye tracker to different head shapes. Along with the eye tracker the person wears a battery and a sender. The images of the scene and eye camera are transmitted wirelessly to a notebook where they are processed. Unfortunately, the Dikablis Wireless eye tracking software and equivalents from other companies do not provide the 3D eye position in world coordinates and the viewing direction out-of-the-box. However, the Dikablis software contains a live tracking module which detects markers attached to the scene. The reconstruction of the line of sight using the results of this marker tracking is described in the following section.

3.1 Calibration of the Scene Camera

First, the eye position, which can roughly be seen as start of the line-of-sight, is estimated. The live tracking module of the eye tracker can be categorized as an inside-out approach. For each marker the eye tracking data provided from the Dikablis software contains the position of the four corners of each detected marker as 2D coordinates in the images of the scene camera. By knowing the corresponding 3D points, one detected marker yielding four of such correspondences is sufficient to determine the extrinsic parameters of the camera. For this purpose we utilize the image processing library OpenCV[5]. The eye position is estimated manually at the time of writing and defined relatively to the camera coordinate system. The accuracy of the pose estimation depends mainly on the accuracy of the marker tracking. The images of Figure 1 show a configuration with markers being distributed equally over the scene. In our visualization tool detected markers can be highlighted with blue borders as can be seen in Figure 2.

3.2 Line-of-Sight Computation

Since no viewing direction is provided directly by the data delivered by the Dikablis software, we again make use of the marker tracking. Each marker spans a virtual 2D plane in the 3D space. These are visualized by green dots as can be seen in Figure 2. The live data of the eye tracking software contains the intersections of the line-of-sight with the virtual 2D plane of each detected marker. If a marker is attached to a wall, the marker's virtual 2D plane coincides with the wall's surface and the intersection of the line-of-sight with the marker's virtual plane can be interpreted as the gaze point. If the scene contains

[3] National Museum of Decorative Arts (Madrid, Ministry of Culture, Spain)

[4] http://www.ergoneers.com/de/products/dlab-dikablis/overview.html

[5] http://opencv.org

complex 3D objects whose surface is different from a 2D plane, the intersections with the 2D planes do not coincide with the real object. Instead, they lie somewhere in mid-air but not on the surface of an object. We therefore compute all 3D coordinates of the intersections of the line-of-sight with the marker planes. These intersections are visualized in Figure 2 as yellow spheres and should ideally lie on one line. Since this is not the case due to accuracy issues, we compute the mass value of the intersections. The mass value (grey sphere) and the previously computed eye position give the viewing direction. This is visualized in Figure 2 as a violet line and the resulting gaze point as violet sphere.

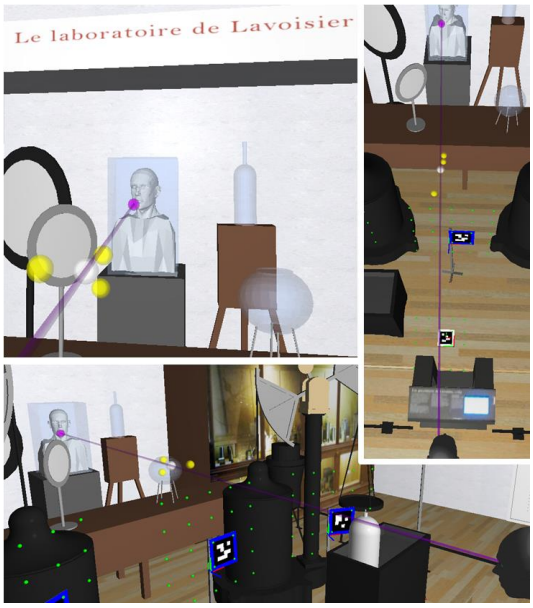

Figure 2: Gaze point (violet sphere) estimation by using the eye position and viewing direction computed by the mass value (grey) of intersections (yellow) with virtual marker planes (array of green spheres). The data of this example was recorded in the Musée des Arts et Métiers located in Paris.

4. FIXATION COMPUTATION

To reveal information from eye tracking data, the common, most important eye movement types for gaze analysis have to be extracted: Fixations and saccades. Visual perception of the environment almost only occurs during fixations [5], when the field of view is imaged on the retina by the optical system of the eye. Scanpaths of raw gaze points therefore contain much irrelevant data like gaze points computed during saccades. Furthermore, a cluster of gaze points can be processed faster when summarized as fixation. Detailed explanations of methods for computing fixations and saccades and their comparison concerning accuracy, execution speed and ease of implementation can be found in Salvucci and Goldberg [5] or in a more recent study focusing mainly on accuracy in Komogortsev et al. [6]. In the next section, we describe a velocity based fixation identification algorithm implemented in our system.

4.1 Velocity-Threshold Identification with 3D Gaze Points

Velocity-Threshold Identification (I-VT) is a fast and easy-to-realize algorithm. It is based on point-to-point velocities given in degrees per second. When two consecutive 3D gaze points and the position of the eye corresponding to the last gaze point are given, two visual axes from the eye position to both gaze points can be computed as well as the angle included by these lines. This angle divided by the time between both samples results in the actual angular velocity which is afterwards compared to a spatial threshold – we use a velocity threshold of 50°/s. If it is greater than this threshold, the last gaze point is assigned to a saccade and otherwise to a fixation. As long as points are assigned to a fixation, the new fixation center is computed as the mass value of the gaze points belonging to the fixation. Optionally, the I-VT can be extended by a temporal threshold that describes the minimal duration for fixations, which is about 100 milliseconds according to Goldberg and Schryver [7].

5. GAZE ANALYSIS IN THE VALENCIAN KITCHEN

Our tool allows for offline and online gaze visualization and analysis. In the following sections we will demonstrate how it can be used for offline analysis and real-time gaze-based interaction.

5.1 Offline Visual Attention Analysis

In the Valencian Kitchen two types of experiments were conducted. The first task for the subject was to have a look at the kitchen wall shown in Figure 1. The wall depicts a scene with a house lady (second person from the left) and several servants. The resulting gaze data of one subject has been visualized. The scanpath comprises around 5 minutes of viewing behavior. On the left image it can be seen that the gaze points are distributed over all objects and persons of the depicted scene. In the second experiment, the subject looked at the kitchen wall again, but this time with a headset, and was provided with audio information about the depicted scene. The raw gaze data and fixations as well as the resulting heat map are visualized in Figure 3 and Figure 4. Compared to the first experiment it can directly be seen that the audio content guided the subject's attention to the lower half of the kitchen wall depicting the important persons. This can also be emphasized with the sum of fixation durations on the AOIs of the food and kitchen tools (upper half of the wall) and those of the persons and objects (lower half). During the first task the sum of all fixation durations on the food and kitchen tools is 45 seconds and on the persons 64 seconds, whereas during the task with the audio guide it is 30 seconds on the food and kitchen tools and 181 seconds on the persons. In Figure 4, areas that are known to attract visual attention of humans like faces belong to the hot spots. The first fixation belongs to the AOI of the house lady, which has a sum of fixation durations of more than 20 seconds. Only the female servant along with the cat (first person from the right in Figure 4) has a higher sum of fixation durations. Further AOIs with similar high values for the sum of fixation durations are the black servant (first person from the left) and the tablet carried by another female servant (second person from the right) of which a cup is falling off.

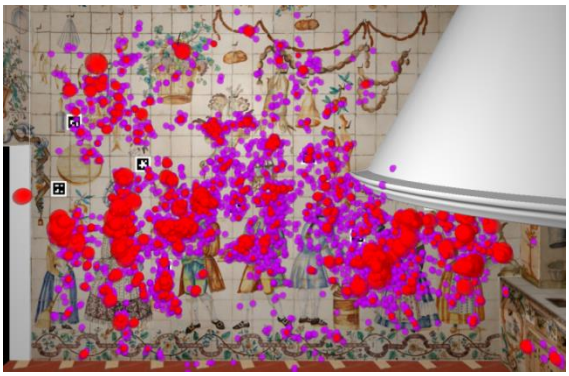

Figure 3: Visualization of 9741 raw gaze samples as violet spheres and resulting 693 fixations as red spheres.

Figure 4: Normalized heat map visualization of the same gaze data as in Figure 3.

5.2 Online Implicit Interest Detection

Implicit interest detection using gaze analysis is the main concern of eye tracking in the already mentioned European project ARtSENSE. It is assumed that the user's overall and visual attention is correlated and overt attention is prevalent. Fortunately, the latter can be assumed, because covert attention has been found out to mainly assist active vision and being unusual to occur as a substitute process [8]. In Figure 5 the visual attention of a subject at time t of the past 4 seconds is shown. The defined AOIs have an orange overlay. The system detects that AOI *tablet2* is hit which is highlighted in red color to give visual feedback. AOI *tablet1* has been attended before. In the 4 seconds time window the sum of fixation durations on all three tablets constitutes 62 % of the total sum of fixation durations of that time window. Figure 6 shows the visual attention at time t plus 1 second. The attention is now drawn to AOI *tablet3*. Again, the corresponding AOI is highlighted. The sum of fixation durations on all tablets increases to 71 %. Detecting which AOIs have been the most relevant during different time windows will allow the active museum assistant in the ARtSENSE project to reason about what information should be provided to the user next. A similar approach can for example be found in [9].

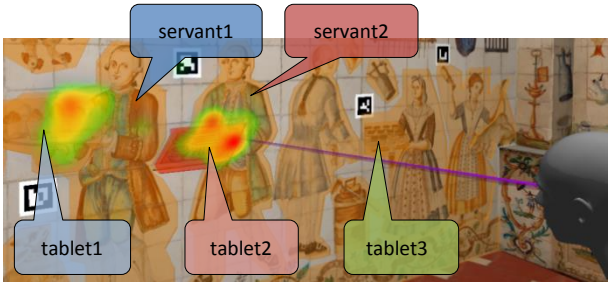

Figure 5: Visualization of hit and not hit AOIs of a time window of the past 4 seconds at time t.

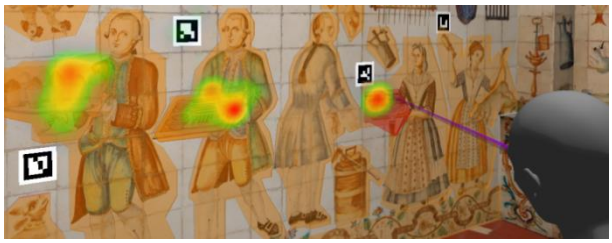

Figure 6: Same as Figure 5 but at time t + 1 second.

6. CONCLUSION

In this paper we presented a tool for real-time 3D gaze analysis and visualization. We began by structuring what is required for 3D gaze point computation. Then we demonstrated how a commercial off-the-shelf mobile eye tracker was connected to our software to harness its functionality. We detailed the embedded 3D gaze point computation and delineated our adaption of the I-VT algorithm for 3D fixation estimation. Afterwards we demonstrated the visualization and analysis capabilities by showing gaze data and first analysis results collected in a museum. In the future we will be realizing the computation of more metrics for scanpath analysis and their visualization in the 3D scene. We are still improving the 3D visualization capabilities and started connecting further eye trackers.

7. ACKNOWLEDGMENTS

This work was partially funded by the European Commission under the 7th Framework Program, Grant Agreement Number 270318, ARtSENSE.

8. REFERENCES

[1] Thies Pfeiffer. 2012. Measuring and visualizing attention in space with 3D attention volumes. In Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '12), Stephen N. Spencer (Ed.). ACM, New York, NY, USA, 29-36.

[2] E. D. Guestrin and M. Eizenman. 2006. General Theory of Remote Gaze Estimation Using the Pupil Center and Corneal Reflections. IEEE Transactions on Biomedical Engineering 53, 1124--1133.

[3] Tobias Schuchert, Sascha Voth, and Judith Baumgarten. 2012. Sensing visual attention using an interactive bidirectional HMD. In Proceedings of the 4th Workshop on Eye Gaze in Intelligent Human Machine Interaction (Gaze-In '12). ACM, New York, NY, USA, Article 16 , 3 pages.

[4] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. 2011. KinectFusion: Real-time dense surface mapping and tracking. In Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR '11). IEEE Computer Society, Washington, DC, USA, 127-136.

[5] Dario D. Salvucci and Joseph H. Goldberg. 2000. Identifying fixations and saccades in eye-tracking protocols. In Proceedings of the 2000 symposium on Eye tracking research & applications (ETRA '00). ACM, New York, NY, USA, 71-78.

[6] Oleg V. Komogortsev, Sampath Jayarathna, Do Hyong Koh, and Sandeep Munikrishne Gowda. 2010. Qualitative and quantitative scoring and evaluation of the eye movement classification algorithms. In Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications (ETRA '10). ACM, New York, NY, USA, 65-68.

[7] Joseph H. Goldberg and Jack C. Schryver. 1995. Eye-gaze-contingent control of the computer interface: Methodology and example for zoom detection. Behavior research methods, instruments, & computers, 27(3):338–350, 1995.

[8] J. M. Findlay. 2005. Covert attention and saccadic eye movements. In Neurobiology of attention. London ; New York: Elsevier Academic Press, 114-117.

[9] Ajanki, A.; Billinghurst, M.; Gamper, H.; Järvenpää, T.; Kandemir, M.; Kaski, S.; Koskela, M.; Kurimo, M.; Laaksonen, J.; Puolamäki, K.; Ruokolainen, T. & Tossavainen, T. 2011. An augmented reality interface to contextual information. Virtual Reality, Springer-Verlag, 15, 161-173.