Mobile gaze-based screen interaction in 3D environments

Diako Mardanbegi
IT University of Copenhagen
Rued Langgaards Vej 7, 2300 KBH. S.
Phone: +45 50245776
dima@itu.dk

Dan Witzner Hansen
IT University of Copenhagen
Rued Langgaards Vej 7, 2300 KBH. S.
Phone: +45 72 18 50 88
witzner@itu.dk

ABSTRACT

Head-mounted eye trackers can be used for mobile interaction as well as gaze estimation purposes. This paper presents a method that enables the user to interact with any planar digital display in a 3D environment using a head-mounted eye tracker. An effective method for identifying the screens in the field of view of the user is also presented which can be applied in a general scenario in which multiple users can interact with multiple screens. A particular application of using this technique is implemented in a home environment with two big screens and a mobile phone. In this application a user was able to interact with these screens using a wireless head-mounted eye tracker.

Categories and Subject Descriptors

H.5.2 [**User Interfaces**]: Input devices and strategies, Interaction styles, Evaluation/Methodology; H.5.3 [**Group and Organization Interfaces**]: Collaborative computing.

General Terms

Human Factors

Keywords

Head-mounted eye tracker, Screen interaction, Gaze-based interaction, Domotics

1. INTRODUCTION

This paper presents a robust method to use head-mounted eye trackers for interaction with different screens in a 3D environment. Through this paper it is shown that gaze interaction can be generalized for usage in 3D environments where multiple screens and users can interact simultaneously thus allowing users to move around freely in a 3D environment while using gaze for interaction.

Eyes are meant for 3D navigation tasks, yet most gaze-aware applications are focused on 2D screen-based interaction. With the increasing number of displays (TVs, computer monitors, mobile devices and projectors) used ubiquitously in our 3D daily lives, and with the current developments in small high quality cameras that transmit data wirelessly it seems obvious that gaze-based interaction holds potential for more than tools aimed for limited user groups (e.g. disabled) and there is a long range of novel gaze-based applications waiting to be investigated with improved principles for gaze-based interaction in 3D environments.

Gaze interaction is mostly done with a single user sitting in front of a screen using a remote eye tracker. An attractive property of remote eye tracking is that it is quite accurate and allows for non-invasive interaction. Remote eye trackers are restricted by only allowing interaction with a single screen. Besides it only has a limited field of view. Multiple screen interaction can be obtained with multiple remote eye trackers but may induce high costs and it will despite the multiple eye tracker and novel synchronization schemes still not facilitate the user with a complete freedom to move. A high degree of flexibility can be obtained with remote eye trackers, where the eye tracker is mounted on the user and thus allows gaze to be estimated when e.g. walking and driving. Even though head mounted eye trackers have reported higher accuracies than remote eye trackers [11], head mounted eye trackers only give gaze estimates on the scene image and not on the object used for interaction e.g. the screen. Using head mounted eye trackers for screen-based interaction is also complicated by the fact that the screen may be viewed from multiple viewpoints. Head mounted eye trackers can be used with multiple screens without synchronization of eye trackers but requires some method for knowing which screen is in the field of view. Head mounted eye tracking may potentially allow multiple users share the same screen without additional requirements on the eye tracker.

This paper addresses the particular problem of using head mounted eye trackers for interaction with planar objects (such as screens and visual projections on planar surfaces). While the general problem of recognizing objects in images is challenging this paper presents a novel and effective method to determine which particular screen the user is looking at without heavy computational demands yet without cluttering the interaction space with tags attached to the objects. The proposed method also supports multiple users interacting with the screens simultaneously.

Section 2 describes previous work and section 3 gives a brief introduction to head mounted eye trackers. Section 4 describes the method for detecting and recognizing screens in the scene image and transferring gaze estimates from the scene image to the object space. Section 5 presents a particular application of using the generalized technique for a home environment and section 6 concludes the paper.

NGCA '11, May 26-27 2011, Karlskrona, Sweden

2. PREVIOUS WORK

A wide variety of eye tracking applications exist. These can broadly be categorized into diagnostic and interactive applications [8]. Interactive applications were initiated in the early 1980's [2] and further developed by [21]. A large body of novel applications has been proposed to use gaze information for improved interaction with screen-based applications.

Gaze interaction with screens is mostly done through remote eye trackers and significant attention has been given to applications that assist disabled people [13].

Eye interaction has also been used to control objects in the 3D environment, like turning lamps on and off via the monitor [5, 3], which is a rather indirect way of interacting with 3D objects. Head mounted eye trackers have been intended for environmental control. Gale [10, 20] proposes to use head-mounted eye trackers as a device for monitoring the attended objects in a 3D environmental control application. However, this work did not actually use the head mounted eye trackers for direct interaction with user interface and objects, and they relied on alternate sources to do the interaction e.g. remote eye trackers.

Some other applications include attentive user interfaces [14] (e.g., gaze contingent displays [7] and EyePliances [19]). Although remote eye trackers can be use for interaction with attentive user interfaces on public screens [1] or large screens, there are still the lack of mobility and multiple user interaction, and head mounted eye trackers may be better suited for this purpose. Eddy (2004) suggests using the head-mounted eye trackers together with a head-tracking device for monitoring the user's gaze when viewing large public displays [9], however head-mounted eye tracker was not used for gaze interaction.

Object identification can be done thorough visual markers and can either be visible [17] or invisible. Visible markers include QR-Codes, Microsoft color tag, and ArToolKit [15] and invisible tags can be obtained by using polarization [16] or infrared markers [18]. While being simplifying detection, the visual markers are limited by the need to place the markers on the objects.

3. HEAD MOUNTED EYE TRACKER

There are generally two types of video-based eye trackers: remote gaze trackers and head-mounted eye trackers [6]. Head mounted eye trackers (HMET) have at least one camera for capturing eye movements and another for capturing scene images (Figure 1-a). The cameras are mounted on the head to allow the user to move freely. This is in contrast to remote eye trackers that have only one camera located away from the user for capturing the eye image. Remote eye trackers estimate the point of regard on the screen while head-mounted eye trackers estimate the user's point of regard in the scene image (displayed in figure 1-b with a cross-hair).

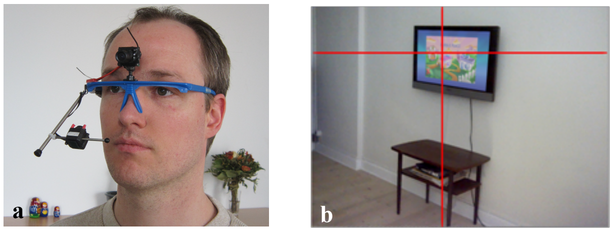

Figure 1.(a) HMET system and (b) scene image with a red cross-hair to indicate point of regard.

The head mounted eye tracker used in this paper is shown in figure 1 and was made by the authors. The eye tracker transmits image data to a server wirelessly for further processing.

4. FRAMEWORK

The general framework addressed in this paper contains several screens (clients), a server and one or more eye trackers. An example of a potential multi screen scenario is shown in figure 2. The user is wearing the HMET holding a mobile phone (screen) in the hand. There are two other screens in the background that could also be used for interaction.

Communication between system components (eye trackers, screens/clients and servers) builds upon TCP/IP. The purpose of the server is to facilitate communication between the eye tracker and the screens. Images from the eye tracker are sent wirelessly for further processing on a remote PC. The remote PC locates the screen, estimate gaze on the screen and subsequently sends the information to the server.

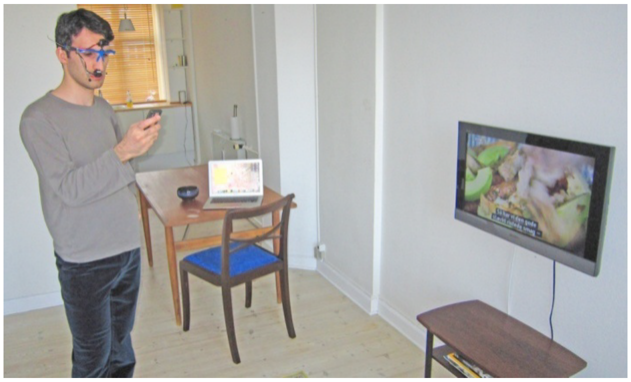

Figure 2. An example of a potential multi screen scenario with a user wearing a HMET, and able to interact with a TV screen (on the wall), a computer screen (on the table) and a mobile phone.

The following sections describe the screen detection method (section 4.1) and how the gaze coordinates from the scene image is transformed to screen coordinates (section 4.2).

4.1 Screen detection

The scene image is the prime resource for obtaining information about the surroundings in head mounted eye trackers unless other position devices are available. The eye tracker should potentially be able to detect and discern multiple screens. There is a multitude of image-based methods that could be used to detect a screen in the scene image. The ideal method is able to detect the screen in different light conditions and when the screen is turned on or off and should simultaneously be sufficiently fast to allow for real-time processing.

Another challenge is to be able to discern screens with identical appearance and when these are viewed from different angles. Fixed visual markers could be placed on the screen to allow for easy identification e.g. a QR-Code around the screen. The visual tag is only needed for identification of the screen and is not needed during interaction. Hence, fixed visual tags are needless most of the time and could be disturbing for the user while they also clutter the scene. Besides, fixed visual tags are not suitable for use when employing a large number of screens since someone needs to be placing the tags where most appropriate.

Potential screen candidates are detected using quadrilateral contour information (illustrated in figure 3). Whenever a quadrilateral, Q, appears in the scene image the eye tracker notifies the server to show identification tags. For initialization, the server issues a command to all the screens to show their identification tag (similar to a QRCode) for short period of time. The tag is shown until the eye tracker has identified the tag in the scene image. The tag possesses information about screen identity and may contain other screen and application dependent information. The screen is tracked over time after identification, but the identification procedure is reinitiated when other screens appear in the scene image. During re-initialization the server only issues commands to the currently inactive screens.

This approach allows a low degree of maintenance and offers an efficient way of identifying screens. Notice that this approach is sufficiently general and scalable to situations with multiple eye trackers and multiple screens located in individual networks (e.g. located over large distances). Several users may even share the same screen.

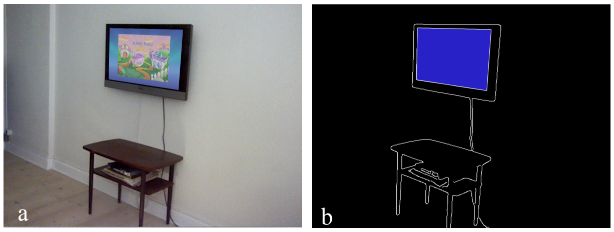

Figure 3. (a) Scene image with a screen (b) edge image and the detected screen

4.2 Mapping point of regard (PoR) to object space

The eye tracker provides only gaze estimates in the scene image, but what is needed is to be able to determine where on the screen the user is looking. This means that a mapping from the image coordinates, **s**, to the screen coordinates, **m**, are needed. In this paper we assume the objects used for interaction (the screens) are planar. Under these circumstance there is a homographic mapping, H_s^m from the screen in the scene image, to the screen coordinates [12]. H_s^m needs to be calculated in each frame since the position of the screen is not fixed in the scene image. The homography from the screen corners S_i to M_i (figure 4) is estimated in each time instance. Information about the screen dimensions are obtained from the visual tag during screen identification. The gaze point in the scene image is then mapped to the screen coordinates through $\mathbf{m} = H_s^m \cdot \mathbf{s}$. Figure 4 shows the mapping of the PoR (center of the red cross-hair) from the scene image to the screen plane and the real coordinates of the PoR in the screen by a black cross-hair (left image).

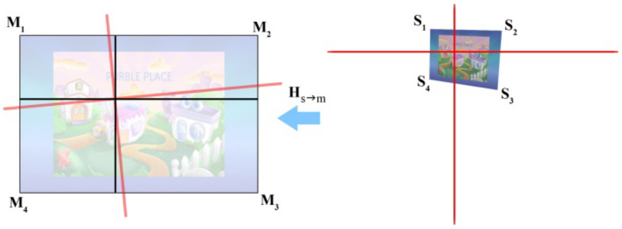

Figure 4. Mapping from the scene plane (right) to the real screen plane (left)

Eye trackers do not have pixel precision. Each gaze measurement in the scene image is therefore associated with an error. A convenient property of this approach is that the assumed precision and point of regard can be mapped to the screen image by mapping the uncertainty ellipse from the scene image to the screen image [12]. Figure 5 illustrates this process.

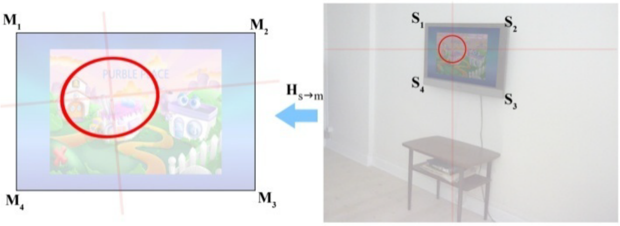

Figure 5. (left) The point of regard (cross hair) and the estimated uncertainty (ellipse) in the screen. (Right) The screen as viewed from the scene camera, the estimated point of regard (cross hair) and the assumed eye tracker precision.

5. EXPERIMENTAL APPLICATION

The experimental setup is intended for a home environment where the users are be able to communicate and interact with screens and control objects (e.g, fan, door, window and radio) via the screens.

Three screens are located in a house, each with a TCP/IP connection to the server. Two screens (S1 and S2) are 55” LG flat panel TVs. The third screen, S3, is a 4” Sony Ericsson Xperia X10 screen. Three different markers are used for identifying the screens. The applications are running on the screens, only allow single-user inputs and the experiments are therefore conducted with single user at the time. Each screen application is made to illustrate different applications of head mounted eye tracking for domotics [4] scenarios, namely controlling devices, the computer and small mobile devices.

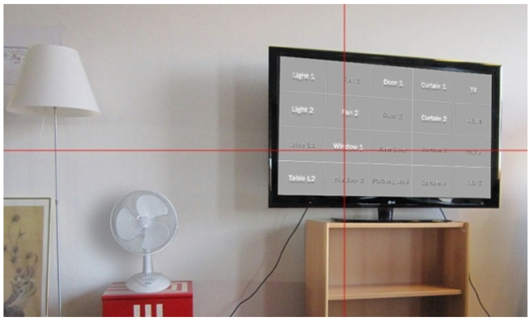

Figure 6. the user is interacting with S1

S1 is running an application which allows 20 devices to be turned on or off in the home environment. The user can control the devices by double blinking while gazing on the on-screen buttons. Each button spans a 17x24 cm rectangle on the screen and the user can see the status of each object by the changing of the color (figure 6). S2 is connected to a computer via RS232 port to allow the computer to communication with the functionalities of the TV. The user can change the channels up and down by double blinking on the left hand side corners and similarly for the volume (right hand side corners). Each of the corners regions is (20x20cm) on the screen. S3 is mobile phone screen and a java application is running on it that has 4 on screen-buttons and wirelessly connects to the server. Each button can be used to control a subset of the objects of S1 through double blinking. The eye tracker is feature-based using homographic mappings from the eye image to the scene image, thus requiring a 4point calibration procedure. The eye tracker runs at 15 fps on 640x480 images with an accuracy of about 1 degree of visual angle for the calibration distance. Screen detection is done using quadrilaterals in the scene image based on the contour-based features.

Head mounted eye trackers are usually prone to errors when objects in the scene image are on different depths than calibration distance (due to the parallax of the scene camera and eye). When the calibration was performed at 1.5 meters, the eye tracker had an accuracy about 1.5° in the scene image when the user was at 4 meters from the screen, and about 3° when the user was at 40 cm. The inaccuracy of the eye tracker consequently propagates to the screen and is therefore dependent on the distance and angle between the user and screen (figure 5). However the accuracy on the screens was sufficient for interaction with mobile device and large screens (5 x 4 grid on the screen).

6. CONCLUSION

The background for this work was how interaction with multiple screens can be done with a head mounted eye tracker, We have presented a general framework that allows screens to be detected efficiently and identified without cluttering the scene or disturb the user significantly. The method is easily extendible to multiple locations, with many screens and is still easy to maintain. The low-cost head-mounted eye tracker that does not support parallax error, limits the difference between working plane and the calibration plane, however using the accurate systems calculate the point of regard accurately as the screen is viewed from close or far distances.

A significant limitation of our system, however, is that the current method for screen detection and mapping of the gaze point cannot be used when the screen is not completely inside the scene image (e.g. viewing the big screens from close distance). However with more advanced techniques this would be possible.

The method has been tested on 3 different applications intended for domotics using a low cost wireless head-mounted eye tracker. The users were able to interact with a TV and a computer screen located in different places in the home environment and with a mobile phone.

Through this work we have demonstrated that head mounted eye trackers can be used for interaction with the screens in 3D spaces.

7. REFERENCES

[1] Agustin, J.S., Hansen, J.P., Tall, M. 2010. Gaze-Based Interaction with Public Displays Using Off-the-Shelf Components. In Proceedings of the 12th ACM International Conference on Ubiquitous Computing (UBICOMP2010), Copenhagen, Denmark. ACM, New York, pp. 377-378.

[2] Bolt, R.A. 1982. Eyes at the Interface. In Proc. of Human Factors in Computer Systems Conference, 360-362.

[3] Bonino, D.; Castellina, E., Corno, F. &Garbo, A. 2006. Control Application for Smart Housethrough Gaze interaction," Proceedings of the 2nd COGAIN Annual Conference onCommunication by Gaze Interaction, Turin, Italy.

[4] Bonino, D., Castellina, E., & Corno, F. 2008. The DOG gateway: enabling ontologybasedintelligent domotic environments. Consumer Electronics, IEEE Transactions on, 54 (4), 1656-1664.

[5] Castellina, E., Razzak, F., Corno, F. 2009. "Environmental Control Application. Compliant with Cogain Guidelines," The 5h Conference on Communication by gaze interaction (COGAIN 2009).

[6] Duchowski, A.T. 2007. Eye Tracking Methodology: Theory and Practice. Springer, London. (2th edn)

[7] Duchowski, A. T., Cournia, N., and Murphy, H. 2004. Gaze-contingent displays: A Review. CyberPsychology and Behaviour, 7(6), 621‑634.

[8] Duchowski, A.T. 2002. A breadth-first survey of eye tracking applications. In Behavior Research Methods, Instruments, & Computers (BRMIC), 34(4), 455-470.

[9] Eaddy, M., Blasko, G., Babcock, J., and Feiner, S. 2004. My own private kiosk: Privacy-preserving public displays. In ISWC '04: Proceedings of the Eighth International Symposium on Wearable Computers, 132‑135, Washington, DC, USA, IEEE Computer Society.

[10] Gale A.G., 2005. Attention Responsive Technology and Ergonomics. In Bust P.D. & McCabe P.T. (Eds.) Contemporary Ergonomics 2005, London, Taylor and Francis, 273-276.

[11] Hansen, D. W. and Ji, Q. 2010. In the Eye of the Beholder: A Survey of Models for Eyes and Gaze. IEEE Trans. Pattern Anal. Mach. Intell, 478-500.

[12] Hartley, R., and Zisserman, A. 2000. Multiple view geometry in computer vision. Cambridge University Press, Cambridge, UK.

[13] Hutchinson, T. E., White, K. P., Martin, W. N., Reichert, K. C., and Frey, L. A. 1989. Human‑computer interaction using eye-gaze input. Systems, Man and Cybernetics, IEEE Transactions on, 19(6),1527‑1534.

[14] Hyrskykari, A., Majaranta, P., and Raiha, K. J. 2005. From gaze control to attentive interfaces. In Proceedings of the 11th International Conference on Human–Computer Interaction (HCII 2005). IOS Press.

[15] Kato, H., and Billinghurst, M. 1999. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In Proc.IEEE International Workshop on Augmented Reality, 125–133.

[16] Koike,H., Nishikawa, W., Fukuchi, K. 2009. Transparent 2-D Markers on an LCD Tabletop System, ACM Human Factors in Computing Systems (CHI 2009), 163-172.

[17] Palmer, R.C. 2001. The Barcode Book, 4th edition, Helmers Pub.

[18] Park, H., and Park, J. 2004. Invisible marker tracking for AR. Proc. 3rd IEEE/ACM Int. Symp. on Mixed and AugmentedReality, 272–273.

[19] Shell, J. S., Vertegaal, R., and Skaburskis, A.W. 2003. Eyepliances: attention-seeking devices that respond to visual attention. In CHI '03: Extended abstracts on Human factors in computing systems, pages 770‑771, New York, NY, USA. ACM Press.

[20] Shi, F., Gale, A.G. & Purdy, K.J. 2006. Eye-centric ICT control. In Bust P.D. & McCabe P.T. (Eds.) Contemporary Ergonomics 2006, 215-218.

[21] Ware, C. and Mikaelian, H.T. 1987. An evaluation of an eye tracker as a device for computer input. In Proc. of the ACM CHI + GI-87 Human Factors in Computing Systems Conference, 183-188.