GazeProjector: Accurate Gaze Estimation and Seamless Gaze Interaction Across Multiple Displays

Christian Lander[1], Sven Gehring[1], Antonio Krüger[1], Sebastian Boring[2], Andreas Bulling[3]

[1]DFKI GmbH, Saarbrücken, Germany
[2]Departement of Computer Science, University of Copenhagen, Denmark
[3]Max Planck Institute for Informatics, Saarbrücken, Germany
[1]{firstname.lastname}@dfki.de, [2]sebastian.boring@di.ku.dk, [3]andreas.bulling@acm.org

ABSTRACT

Mobile gaze-based interaction with multiple displays may occur from arbitrary positions and orientations. However, maintaining high gaze estimation accuracy in such situations remains a significant challenge. In this paper, we present *GazeProjector*, a system that combines (1) natural feature tracking on displays to determine the mobile eye tracker's position relative to a display with (2) accurate point-of-gaze estimation. *GazeProjector* allows for seamless gaze estimation and interaction on multiple displays of arbitrary sizes independently of the user's position and orientation to the display. In a user study with 12 participants we compare *GazeProjector* to established methods (here: visual on-screen markers and a state-of-the-art video-based motion capture system). We show that our approach is robust to varying head poses, orientations, and distances to the display, while still providing high gaze estimation accuracy across multiple displays without re-calibration for each variation. Our system represents an important step towards the vision of pervasive gaze-based interfaces.

Author Keywords

Eye tracking; gaze estimation; calibration; large displays; multi-display environments; natural feature tracking

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

Gaze is a powerful modality for interacting with displays, as it naturally indicates what we visually attend to and what we are interested in [30]. Gaze is furthermore faster than other existing pointing devices (e.g., a mouse [22]). For that reason, gaze-based interaction received considerable attention with applications ranging from controlling desktops [13,33], to text entry [18], to target selection [27], and to entering passwords [7]. First prototypes used desktop-like settings and stationary eye trackers in which a user's head was fixed (regarding position and orientation).

Latest advances in head-mounted eye tracking point the way towards pervasive gaze-based interactions with situated displays in everyday settings [6]. These trackers are commonly equipped with two cameras: (1) a scene camera partly capturing a user's current field of view, and (2) an eye camera recording a close-up video of the user's pupil position. Such eye trackers have to be calibrated to a *specific* user for a *specific* display before first use to establish a mapping between the pupil's 2D positions in each of the cameras' coordinate systems. This calibration is time-consuming and cumbersome as it involves looking at several calibration points on the target display.

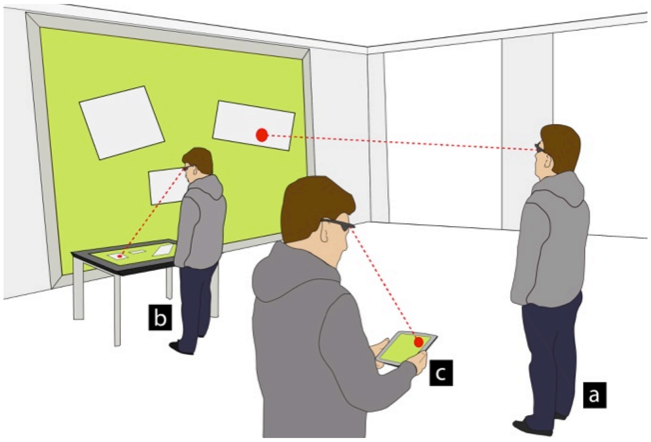

Figure 1. *GazeProjector* enables seamless gaze-based interaction with multiple displays from arbitrary locations and orientations, such as wall-sized displays (a), horizontal screens (b), and handheld devices (c) – without active recalibration.

Another problem is that calibration is typically performed for a *fixed* position and orientation of the user to a *single* display. While this is less of an issue for stationary settings and TV-sized displays, mobile settings and multiple – potentially large – displays evoke two types of motion: (1) user movements in front of a single display to inspect other parts of the display's content; and (2), head movements to reach targets outside the ocular motor range [10]. In addition, there might be multiple displays present, causing further movements. Both types of motion considerably reduce gaze estimation accuracy [8]. In order to achieve high gaze estimation accuracy, it is crucial to track a user's (and eye tracker's respectively) position and orientation relative to a display. Solutions that realize this include the augmentation

UIST '15, November 08-11, 2015, Charlotte, NC, USA © 2015 ACM. ISBN 978-1-4503-3779-3/15/11…$15.00

of the environment with visual markers [5,32] as well as using vision-based motion capturing systems (e.g., OptiTrack[1]). While such approaches can achieve high tracking and gaze estimation accuracy, the need to deploy them for every display the user might want to interact with currently severely limits uptake and truly pervasive and spontaneous gaze-based interaction.

In this paper we present *GazeProjector*, a system that allows for accurate gaze estimation on arbitrary displays independent of the user's position and orientation (see Figure 1). Similar to Touch Projector [4], the eye tracker continuously tracks its position and orientation relative to different displays using natural feature tracking on the scene camera's video stream. To do so, displays continuously stream their (potentially dynamic) content to a server, which performs the feature matching. Thus, *GazeProjector* does neither require a motion capturing system nor visual markers preinstalled in the environment. After a one-time calibration with an arbitrary display *GazeProjector* is able to transform pupil positions onto any connected display in the environment, as long as a part of that display is visible to the mobile eye tracker's scene camera. As the calibration is independent of a potential target display, our system allows for accurate gaze estimation and seamless interaction across multiple displays, and thus empowers users to freely move around within the environment.

In a controlled performance study with 12 participants we compared our approach to a state-of-the-art video-based OptiTrack motion capturing system as well as a marker-based approach. In the first task, participants looked at on-screen targets from various positions and orientations in front of a large display. In a second task, we compared *GazeProjector* to the marker-based approach on multiple displays (here: a wall-sized display, a tabletop, and a tablet PC). In both tasks, we found that our approach compensated well for head movements (i.e., change of orientation) and user relocation (i.e., change of location). Thus, our work offers the following three contributions:

- **Location- and orientation-independent** gaze estimation on a display (of varying size/resolution) without requiring any instrumentation in the environment.
- **Accurate gaze estimation on arbitrary displays** in the environment without requiring recalibrating the system for each display independently.
- **Maintaining high gaze estimation accuracy** without requiring recalibrating the system for varying positions, orientations and displays.

Overall, these features enable both application developers (who wish to employ gaze as additional input modality) as well as researchers (who wish to study a user's gaze) to rapidly do so in fully pervasive settings, without the need for augmenting the environment.

RELATED WORK

Our work builds on methods for (1) gaze approximation and estimation on displays, (2) gaze interaction using head-mounted eye trackers, as well as (3) tracking the spatial relationship between users and displays.

Gaze Approximation and Estimation on Displays

Several previous works used head orientation as an approximation of where people look. For example, Sippl et al. used a remote camera to detect facial features, such as eyes and nose tip, and estimate head pose on four areas on the display [23]. Nakanishi et al. relied on a stereo face tracking system and the 3D head pose as an approximation of gaze direction [20]. Finally, ViewPointer aimed to detect eye contact between users and devices using a wearable camera and IR tags placed in the environment [25]. While useful for coarse attention measurements, none of these approaches allowed for accurate gaze estimation on the display.

Accurate gaze estimation on displays remains a significant challenge – particularly when remote eye trackers (i.e., eye trackers placed at a display) are used. Such trackers only allow a single user to interact with a display at any point in time and any interaction is restricted to the tracking range of typically 50-80 cm in a central area in front of the display, thereby severely limiting users' mobility [24,27]. Previous work either focused on extending the tracking range of remote trackers [11,19], or on calibration-free (spontaneous) interaction but was either limited to interaction along a horizontal axis, i.e., without full 2D gaze estimation [34] or required dynamic interfaces [31]. Stellmach et al. addressed the mobility (interacting from different positions/orientations) of users [28] by using an additional external tracking system. *GazeProjector* differs in that it does not require such additional systems.

Gaze Interaction Using Head-mounted Eye Trackers

Head-mounted eye trackers are more flexible as they allow the user to move freely in front of the display. Early work on using head-mounted eye trackers for interaction still required calibration to a single, stationary display prior to first use [9]. More recent approaches aimed to estimate gaze dynamically but either required visual markers attached to the display [32] or in the environment to detect gaze on predefined interaction areas, e.g., to control a TV set [5].

With advances in computer vision, visual markers can be substituted with detecting the display directly in the scene camera's field of view. Mardanbegi et al. detect screens based on quadrilaterals found in the scene [16]. Turner et al. extended this to multiple displays (based on the displays' aspect ratios) by adding a second camera and a method for transparently switching between two calibrations [29]. Unlike *GazeProjector*, both approaches require the display to be fully visible to the scene camera, which cannot be guaranteed at all times in mobile settings. Also, relying on automatically selected feature points instead of screen borders is more robust to changing light conditions and generalizes better to displays of arbitrary shape and size.

[1] https://www.naturalpoint.com/optitrack/

Tracking Spatial Relationships of Users and Displays

Tracking the spatial relationship of users (and the users' devices respectively) can be done in two ways. First, external tracking equipment can be used to determine a device's exact position in 3D space (and thus its spatial relationship to a display in the environment). The Proximity Toolkit makes use of such high-precision tracking equipment and provides an interface to acquire spatial relationships [17]. While such a setup results in extremely high accuracy, it is often impractical for outdoor use.

Alternatively the device's camera can be used to identify its spatial relationship to a display. Many approaches exist, such as temporarily showing on-screen visual markers [2] or using dynamic markers following a camera's position [21]. More recently, natural feature tracking was used to determine spatial relationships. Herbert et al. used Scale-Invariant Feature Transform (SIFT) to determine the camera's spatial relationship to a display [12]. Their system tried to identify a screenshot of the display in the device's camera stream. Virtual Projection extended this approach to dynamically updated displays [3]. Touch Projector further allowed for tracking multiple displays provided that display contents differ sufficiently [4]. *GazeProjector* uses these underlying concepts, but advances them with respect to tracking efficiency: we use FAST/FREAK (with their significantly improved matching accuracy [1]). The combination of these two algorithms further increases the frame rate from 10 fps in *Virtual Projection* [3] to more than 20 fps – thus allowing for more interactive frame rates at higher precision than previous systems of that kind.

ENABLING GAZE INTERACTION ON LARGE DISPLAYS

As mentioned before, estimating a user's gaze on a large display and in multi-display environments using a head-mounted eye tracker faces two key challenges: the eye tracker has to be calibrated and used from fixed positions and orientations for all displays. During calibration, the entire display has to be visible in the eye tracker's scene camera. Ideally, the eye tracker only has to be calibrated once. This can be achieved by (1) calibrating pupil positions to the scene camera coordinate system; and (2) tracking the spatial relationship between the eye tracker and a specific display and (3) mapping 2D gaze positions in scene camera coordinate space to that display.

Eye Tracker Calibration

GazeProjector uses a one-time calibration to map pupil positions to the scene camera's coordinate system. Because of this, there is no need to perform the calibration on the display one intends to interact with. Instead, the system can be calibrated once on any display in the environment (e.g., a laptop). This independence of the target display has two advantages: The usage of the eye tracker is not restricted to the same distance and/or orientation to a display while calibrating as this is handled by the self-localization directly; and the calibration does not depend on a single display, thus allowing for seamless gaze estimation across several displays in multi-display environments.

Tracking the Spatial Relationship to Displays

To determine the spatial relationship between the eye tracker and a specific display, we use the approach described in [3], yet with different feature detection algorithms. Specifically, our system streams the scene camera's video to a server that is aware of all screens (and their displayed content) in the environment. All displays in the environment repeatedly stream screenshots to the server to reflect their current content (i.e., in case of quickly updated content, such as videos). The server is thus only aware of the physical dimensions of each display (i.e., size and resolution) as well as their current content, but not their physical location. This is especially important for mobile devices, which frequently change their position and orientation over time.

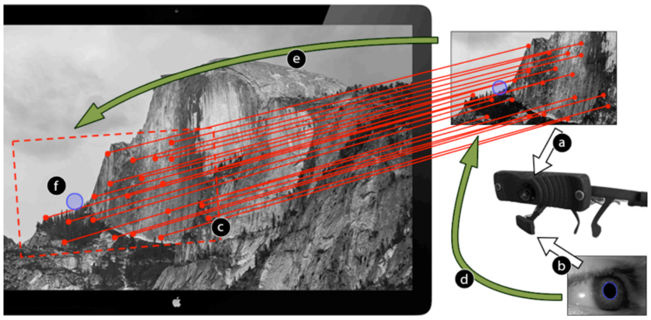

Figure 2. Estimation of gaze on a display using the eye tracker's scene camera (a) and pupil camera (b): the software determines the transformation between the scene camera's image plane and the display (c), combines it with the calibration (d) between the eye camera's and the scene camera's coordinate systems (e) to obtain the location on the display (f).

The server then processes the incoming screenshots as well as incoming frames from the field camera using FAST feature detectors [15] and FREAK feature descriptors [1]. The idea is to use current screenshots as *template* images, which the server tries to find in the *observed* images (here: the field camera's video). If a template matches an observed image, the algorithm calculates the transformation matrix (i.e., a homography), which describes the transformation of points from one image plane (say: a video frame) into another image plane (say: the display's screenshot).

Gaze Estimation

The transformation matrix allows for bidirectional mapping between locations in the scene camera's and target display's coordinate system. Note that the display does not have to be visible in full in the scene view. Instead, a unique sub-region (a region that not occurring anywhere else on that display) is sufficient given enough features within it to allow for robust tracking. Likewise, unique sub-regions and their features on multiple displays present in the scene camera allow for detecting each of the displays within one frame. As in Touch Projector (where touch positions are transformed), *GazeProjector* uses the transformation matrix to estimate gaze positions on the display. Figure 2 illustrates this procedure in more detail. If multiple displays are present, a transformation matrix is calculated for each display and the gaze position is transformed accordingly.

Implementation

Our system consists of three components: (1) a monocular head-mounted PUPIL eye tracker[2] connected to a laptop [14]; (2) a one or more planar displays of arbitrary size; and (3) a desktop computer driving the displays. Laptop and desktop computer are connected via WiFi. The eye tracking software on the laptop is written in Python and is based on PUPIL's open source mobile eye tracking platform[2]. The software running on the desktop computer is written in C# (.NET Framework 4.5). For feature detection, description and matching, we use EmguCV[3] as wrapper for OpenCV[4]. For faster processing, we downscale display screenshots to 384 × 240 pixels and camera frames to 320 × 240 pixels. We achieve up to 30 fps on one display (20 fps with three displays) and are only limited by WiFi bandwidth.

The system allows for distances ranging from 0.5 times the display's diagonal up to six times the display's diagonal. When being further away, the accuracy decreases as the display observed in the camera's field of view decreases in size (thus, removing several features). We believe that a multi-scale approach of screenshots will increase the operational range, yet we decided not to include it in this proof-of-concept implementation. In addition, the tracking compensates for an angular offset of ±60°. While this is sufficient for most interactions, fast eye/head movements will have a slight impact on accuracy. However, we believe that the increasing processing capabilities of future devices will allow for both faster image processing on larger images (i.e., less or no scaling required) for higher accuracy.

Example Application

We built an example application to demonstrate the use of *GazeProjector* in a multi-display setting. It showcases how people can seamlessly interact with multiple displays while freely moving around in the environment.

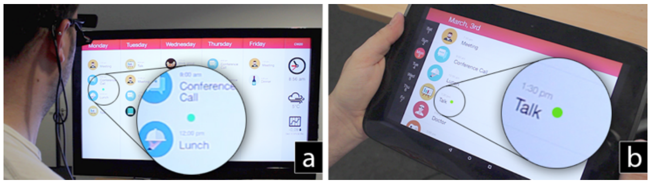

Figure 3. Our example application: after a user select an event from a public calendar (a), that information is shown on the personal mobile device, where gaze estimation also works (b).

We envision an office building where public information screens are distributed showing a calendar application (see Figure 3a). We use *GazeProjector* to interact with these displays in a 'walk-up-and-use' fashion. The only requirement is to calibrate the eye tracker (e.g., at the beginning of a work day using a tablet). The calibration procedure uses a round marker displayed at nine different positions around the screen at which the user has to gaze on. In our interaction scenario users are able to transfer information between public and private displays (e.g., a handheld). Looking at a specific event (e.g., a talk) for a certain dwell time (say: 2 seconds) selects that event, which is then shown on the user's handheld device. *GazeProjector* also works on that device with the same calibration (see Figure 3b). The video figure illustrates the example in more detail. Note, that the flickering of the gaze point stems from the pupil not being detected, which then prevents correct gaze mapping.

EXPERIMENT I: GAZE ESTIMATION ACCURACY

We first conducted a controlled laboratory study to assess *GazeProjector*'s gaze estimation accuracy in comparison to existing but more heavyweight tracking approaches.

Independent Variables

We had two independent variables in this experiment: *Mode* (i.e., the gaze estimation method used), and *Location* (i.e., where participants stood in front of the display).

Mode: We chose three different modes for gaze estimation: *GazeProjector* (*GP*) implemented as described before; *Marker Tracking*[5] (*MT*), which uses a set of on-screen markers for tracking the orientation between the eye tracker and the display provided by the PUPIL framework; and a simple *Head Orientation* (*HO*) approach, which tracks the participant's head using an external OptiTrack system. For each of these modes, we calibrated the eye tracker from two different locations to investigate the effect of distance during calibration. Both were placed centrally in front of the display with one location being close to the display and one being further away. We further calibrated the eye tracker for each participant separately instead of using one calibration (see limitations section for further details).

Location: We chose six different locations in front of the display to simulate a more realistic setting. Three of these locations were close to the display and three were further away. The eye tracker was only calibrated for the central *near* and *far* central locations. This is more realistic, as users would not calibrate for every position in a walk-in-and-use scenario. Note that we calibrated the eye tracker for each participant separately instead of using one calibration (see limitations section for further details). Since no visual feedback was given to them and to keep the experiment at a reasonable length, participants had to perform the set of tasks only once. We then computed the gaze estimation accuracy post-hoc for each of the calibrations.

Task & Procedure

We implemented a gaze pointing task in which participants had to fixate nine different target locations represented as red circles (50 pixels or 98 mm) on the display with equal distances between them (see Fig. 4). A pilot study showed that participants were affected by visualizing their gaze point on the display. Especially if the gaze position was incorrect, people tended to "move" the gaze point to com-

[2] http://pupil-labs.com/pupil/
[3] http://www.emgu.com/
[4] http://opencv.org/
[5] http://www.pupil-labs.com/blog/2013/12/036-release.html

pensate for the error. We therefore opted to not provide any visual feedback to the participants. Participants were instructed to look at each target as quickly and accurately as possible. Each target location was shown for five seconds.

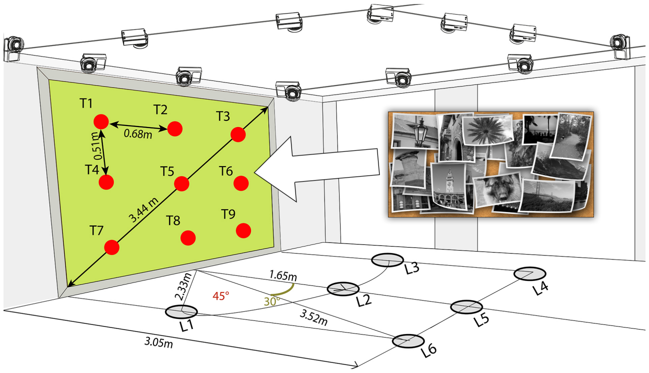

Figure 4. Experimental setup showing all locations (L1-L6) and orientations relative to the display, the nine different positions (T1-T9) of on-screen visual targets, and the background image used for feature tracking.

For each *Mode*, participants first calibrated from the *near-center* location and performed the tasks for all other locations. Afterwards, the calibration for the *far-center* location was recorded and gaze positions as well as errors were evaluated post-hoc. Following best practices in gaze estimation experiments, we validated all calibrations by asking participants to fixate once on each point on a 9-point pattern. Finally, we asked for demographic information.

We collected gaze data from the eye tracker and transformation matrices calculated by *GP* as well as *MT*. Furthermore, we recorded data about the head position and orientation with *OptiTrack*. Data was sampled at 30 Hz (i.e., 150 samples per on-screen target) leading to a total of 1,350 samples for each *Mode* and *Location* combination. We discarded samples for which participants' pupil was not detected (7.5%). We dropped the first two seconds of the five seconds per target (60 samples, 40%) for each target, which was the maximum time required to find the target All together we dropped 276,985 out of 583,200 samples (3 modes × 6 locations × 2 calibrations × 12 participants × 1,350 samples), leaving 306,215 samples recorded: 140,532 for *GP*, 165,683 for *MT* (the sample set used for *HO*).

Experimental Design

We used a within-subject design with the independent variables *Mode* (*GP-near*, *GP-far*, *MT-near, MT-far, HO-near, HO-far*) and *Location* (*front-left, front-center*, *front-right*, *back-left*, *back-center*, *back-right*).

We counterbalanced the order of *Location* across participants using a Latin Square. Although it is possible to record all position information in parallel, we opted to have *GP* and *MT* separate, as markers would favor *GazeProjector*'s tracking. The *HO* mode was recorded while participants were using the *MT* mode. Half of our participants started with the *MT*, and the other half with *GP*. Thus, each participant performed the task twice per location. For each mode and location, the nine targets (equally distributed in a 3 × 3 grid on-screen) were presented in random order.

Apparatus

Figure 4 shows our experimental setup: we used a large front-projected wall with a size of 2.75 × 2.07 meters (diagonal: 3.44 meters). The six locations were distributed within a nine square meter area in front of the display as follows: three locations at a distance of 1.65 m (*near*), and three locations at a distance of 3.05 m (*far*). The left and right locations for *near* were exactly 2.33 meters away from the display's centerline (i.e., an angular offset of ±45°); those for *far* were located 3.52 m away from the display's centerline (i.e., an angular offset of ±30°). Naturally, the two center locations for *near* and *far* had an angular offset of 0°. Locations located *far* allow participants to observe the entire screen at once (the display covers 48.52°), while for locations located *near* the display covers 79.60° – thus exceeding the full-scale ocular motor range of ±55° [10]). The maximum visual angles were 3.4° (*near*) and 1.84° (*far*), and the minimal ones were 1.5° (*near*) and 1.3° (*far*).

Participants

Twelve participants (three female) between 22 and 32 years (mean = 27.45 years, SD = 3.1 years) were recruited from a local university campus. All participants had normal or corrected to normal vision; none reported any form of visual impairments (e.g., color blindness).

GAZE ESTIMATION RESULTS

We corrected all reported gaze estimation accuracies by subtracting the mean calibration error (2.04° with SD = 0.69°). To verify this, we performed a one-way ANOVA with a Bonferroni-corrected post-hoc analysis on calibration accuracies across all *Modes*, and found no significant differences. In subsequent post hoc analyses, we used Bonferroni-corrected confidence intervals to retain comparisons against $\alpha = 0.05$. Furthermore, we used Greenhouse-Geisser correction in cases where sphericity had been violated.

Gaze Estimation Error

To assess the gaze estimation error, we calculated the average gaze estimation error in degrees of visual angle. That is, the difference of the visual angle between the predicted on-screen gaze point and the actual fixation targets for all *Modes* and *Locations*. We then performed a 6 × 6 (*Mode* × *Location*) within subjects ANOVA on gaze estimation errors and found a main effect for *Mode* ($F_{1.989,21.879}$ = 8.526, $p < .002$), a main effect for *Location* ($F_{5,55} = 7.363$, $p < .001$), but we did not find an interaction between the two.

We performed post-hoc tests to further understand the main effect of *Mode*. Most importantly, we found significant differences within *MT* and *HO* for the two calibrations near and *far* (all $p < .033$). In both cases, the *near* calibration led to lower estimation errors. *GP*, on the other hand, did not show such an effect, suggesting that the point of calibration does not effect its gaze estimation error significantly, and the difference in means was also lower than for the other two (*GP*: 0.281°; *MT*: 0.931°; *HO*: 0.948°) – yet, also for *GP*, the mean estimation errors were slightly lower for the *near* calibration than for the *far* one.

This is further reflected when comparing across *Modes*: *GP-near* differed significantly from both *MT-far* and *HO-far* (all $p < .01$). However, there was no significant difference between the *Modes* for the *near* calibration. Furthermore, *GP-far* did not differ significantly from any other *Mode* despite having relatively large differences in error.

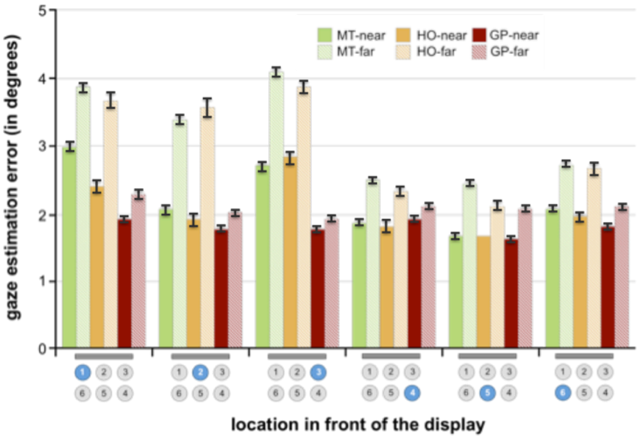

Figure 5. Mean gaze estimation error for every location for MT-near, MT-far, HO-near, HO-far, GP-near and GP-far. Error bars indicate ± standard error of the mean.

GP-near showed the lowest error (*M* = 1.80°, *SD* = 0.20°), followed by *GP-far* (*M* = 2.08°, *SD* = 0.27°), and *HO-near* (*M* = 2.09°, *SD* = 0.23°). *MT-near* (*M* = 2.23°, *SD* = 0.31°) also has an estimated gaze error of less than 3 degrees. The other *Modes* performed slightly worse: *MT-far* (*M* = 3.16°, *SD* = 0.32°) and *HO-far* (*M* = 3.04°, *SD* = 0.31°). Figure 5 summarizes theses results.

Post-hoc tests on *Location* revealed that the significant main effect stems from participants' distance to the display: *front-left* differed significantly from *back-center* and *back-right* (all $p < .019$). *Front-right* also differed significantly from *back-center* ($p < .011$). Overall, *back-center* led to the least estimation errors (*M* = 1.93°, *SD* = 0.23°), followed by *back-right* (*M* = 2.01°, *SD* = 0.17°), and *back-left* (*M* = 2.22°, *SD* = 0.30°). The *front* locations performed worse with *front-center* having the least errors (*M* = 2.45°, *SD* = 0.28°), followed by *front-left* (*M* = 2.86°, *SD* = 0.29°) and *front-right* (*M* = 2.86°, *SD* = 0.30°). On average, the *back* locations had a lower estimation error of 2.08° (*SD* = 0.23°) compared to the *front* locations with 2.72° (*SD* = 0.29°).

Differences for On-screen Target Positions

We did not expect high gaze estimation errors for each of the *Modes*. However, we wanted to analyze whether the on-screen targets resulted in different estimation errors and thus analyzed the results separately for each on-screen target. For *MT*, we found no significant main effects on gaze estimation error for *Target.* We found the same for *HO.* Only for *GP* we found significant differences for gaze estimation for *Target.* Our analysis revealed that predominantly the *bottom-left* target T7 differed significantly from few others (T2, T3, T6 and T8) and led to higher estimation errors. We assume that this is due to the scene camera seeing too few features, which in turn increased the error of the transformation matrix. Figure 6 shows gaze estimation errors for the different modes averaged over all targets.

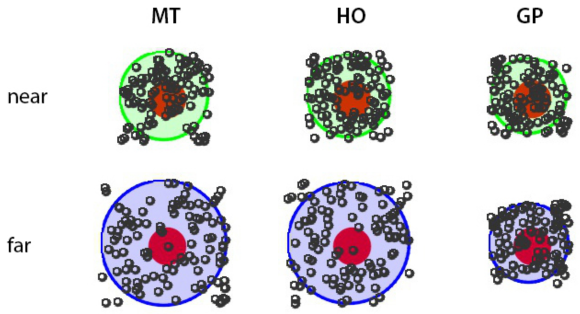

Figure 6. Visualization the mean gaze error (ellipses) for the three modes MT, HO and GP and all calibrations averaged over all targets. Black circles visualize the mean gaze points.

Eye and Head Movements

We were further interested in whether participants mainly moved their head or their eyes to point at an on-screen target location. As expected [9], we found that the average normalized gaze position in the field camera's video was $x = 0.44$ and $y = 0.47$ ($SD_x = 0.21$; $SD_y = 0.25$). Thus, gaze positions remained near the center of the participants' field of view. We subsequently analyzed the gaze position for every *Location* in front of the display and found no significant differences between them. The largest average difference was 0.03. Table 1 lists these results for each *Location*.

Location	Mean (x,y)	SD (x,y)	Var (x,y)
front-left	0.43,0.45	0.19,0.24	0.038,0.060
front-center	0.45,0.47	0.20,0.25	0.040,0.062
front-right	0.46,0.46	0.22,0.27	0.052,0.076
back-left	0.46,0.48	0.23,0.25	0.054,0.065
back-center	0.45,0.48	0.20,0.24	0.044,0.058
back-right	0.43,0.48	0.20,0.23	0.043,0.057

Table 1. Mean, standard deviation and variance for *x,y*-coordinates of normalized gaze positions in the participants' field of view.

The *OptiTrack* data provided detailed information of participants' head orientation (*HO*). We found that the largest head turns covered the entire width of the display (*far*: 51.2°, *near*: 83.66°). On average head motions covered an angle of 31.61° (*SD* = 2.04°). This further confirms our results in that *HO* might be a suitable approximation for gaze estimation with an average error of 2.09° (*SD* = 0.23°) for *HO-near* and 3.04° (*SD* = 0.31°) for *HO-far*.

EXPERIMENT II: MULTIPLE DISPLAYS

We conducted a second controlled laboratory study to assess *GazeProjector's* gaze estimation accuracy across multiple displays of varying form factors – with only a single calibration performed on one of the displays.

Independent Variables

We had two independent variables: *Mode* (i.e., the gaze estimation method used), and *Screen* (i.e., on which display the target was shown). There were no fixed positions to mimic a more realistic scenario where participants were free to move in the environment.

Mode: In this experiment we chose to use only *GazeProjector* (*GP*) and *Marker Tracking* (*MT*), but not head orientation, as we believe it will perform similarly across displays. We calibrated for two locations (as in the first experiment), but additionally recorded calibrations on a 40" tabletop display (*Surface*), and on a 9.7" iPad Air (*iPad*). We chose to do so to investigate the effects on gaze estimation accuracy of calibrating (1) on surfaces not orthogonal to the participant, and (2) on personal devices with a considerably smaller display. The latter resembles a more realistic scenario where users calibrate the eye tracker once on a personal device. Again, calibrations were analyzed post hoc.

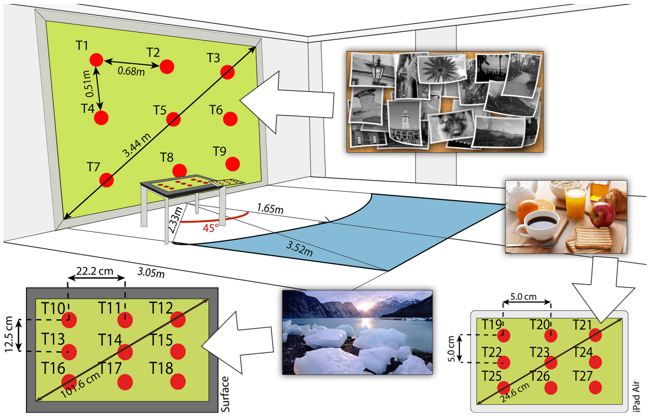

Figure 7. Our setup showing the three screens (including their background images for feature tracking) used during the experiment, as well as the placement of the nine targets per screen. Note that all participants were free to choose a location within the blue area throughout the experiment.

Screen: In addition to the large display used in the first experiment (*Wall*), we chose to add the other two displays used for calibration as well (here: *Surface*, and *iPad*).

Task & Procedure

The task used in this experiment was the same as in the first one: participants had to fixate on-screen targets. However, since we had three displays, participants now had to acquire nine targets per display (27 in total) as shown in Figure 6 As mentioned before, participants could freely choose and change their position between the displays. We again opted to not provide any feedback to participants for the same reasons as before. Participants were instructed to look at each target as quickly and accurately as possible. Each target location was shown for ten seconds to give the participants enough time to find the target on the correct display. There was only *one* target on *one* display shown at a time.

The procedure was nearly the same as for the first experiment but with an additional calibration for *Surface* and *iPad* after all tasks were completed. On the additional displays we used the same 9-point calibration pattern. At the end of the study we asked for demographic information.

We used the same data collection method as in the first experiment. Data was sampled at 30 Hz (i.e., 300 samples for each target, 8100 samples for each *Mode*), and samples were discarded if the participants' pupil was not detected. As we expected an increase in search time for the target, we dropped the first five seconds (150 samples) for each target, leaving 259,745 samples (*GP*: 124,421; *MT*: 135,324).

Experimental Design

We used a within-subject 8 *Mode* (*GP-near*, *GP-far*, *GP-Surface*, *GP-iPad*, *MT-near, MT-far, MT-Surface, MT-iPad*) × 3 *Screens* (*Wall, Surface, iPad*) design. Half of our participants started with *GP*, the other half with *MT* (as in experiment I). The targets were randomized, thus the next target appeared on any of the three *Screens*. The 27 targets were again placed in 3 × 3 grids (i.e., nine per display, 50 pixels in radius, or 10 mm on *iPad*, 23 mm on *Surface*) on each display. In total, participants acquired 54 targets.

Apparatus

We used the same front-projected *Wall* as in the first experiment. In addition, we had a 40" Microsoft Surface 2 (*Surface*), and a 9.7" iPad Air tablet (*iPad*). Figure 7 shows our setup. The tabletop display was placed in front of the projection wall in an area where the participant would occlude the beamer projection. Participants held the tablet in hand during the experiment. They could freely choose their location within a nine square meter area.

MULTI-DISPLAY RESULTS

We again corrected gaze estimation accuracy by subtracting the mean calibration error. The mean calibration error was 2.18° (*SD* = 0.69°). We again verified that we could do so by performing an ANOVA with a Bonferroni-corrected post-hoc analysis on calibration accuracies across all *Modes*, and found no significant differences. As in experiment I, we used Bonferroni-corrected confidence intervals in all post hoc analyses and Greenhouse-Geisser correction in cases where sphericity had been violated.

Gaze Estimation Error

We calculated the average gaze estimation error as in experiment I and subsequently performed a 8 × 3 (*Mode* × *Screen*) within subjects ANOVA on them. We found main effects for *Mode* ($F_{7,77} = 21.733$, $p < .001$), and for *Screen* ($F_{2,22} = 82.705$, $p < .001$) as well as an interaction effect between the two ($F_{14,154} = 9.100$, $p < .001$).

Post-hoc pairwise multiple means comparisons revealed that *GP-near* and *GP-far* differed significantly from *MT-far, MT-Surface* and *MT-iPad* (all $p < .001$). Furthermore, *GP-Surface* differed significantly from *MT-Surface* and *MT-iPad* (all $p < .007$). And finally, *GP-iPad* also differed significantly from *MT-far*, *MT-Surface*, and *MT-iPad* (all *p*

< .039). It is noteworthy, however, that both *GP* and *MT* did not show any significant differences between their different calibrations, suggesting that the device on which they were calibrated on did not impact accuracy.

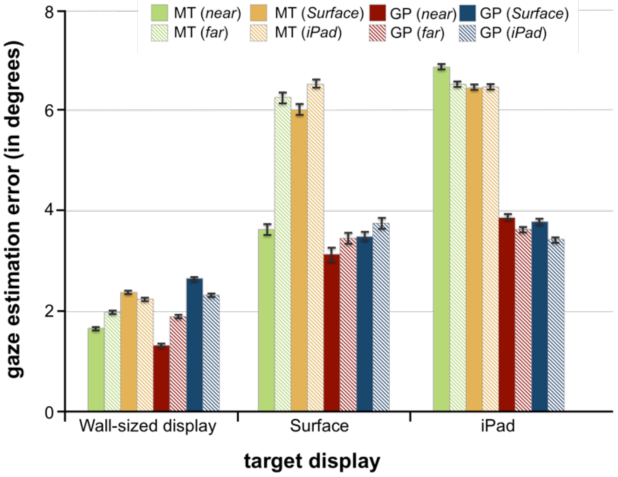

Figure 8. Mean gaze estimation error of each mode for each display. Error bars indicate ± standard error of the mean.

Overall, *GP-near* had the lowest estimation error (*M* = 2.77°, *SD* = 0.20°), followed by *GP-far* (*M* = 3.01°, *SD* = 0.16°), *GP-iPad* (*M* = 3.24°, *SD* = 0.17°) and *GP-Surface* (*M* = 3.31°, *SD* = 0.16°) across all *Screens*. For all *MT* variations, the estimated gaze errors were larger than 4 degrees. Figure 8 summarizes theses results.

As for the main effect for *Screen*, post-hoc multiple means comparisons revealed that *Wall* was significantly different from the other two *Screens* (all p < .001). However, there was no significant difference between *Surface* and *iPad*. Overall, targets on the *Wall* had the least estimation error (*M* = 2.07°, *SD* = 0.07°), followed by *Surface* (*M* = 4.52°, *SD* = 0.22°) and *iPad* (*M* = 5.12°, *SD* = 0.23°).

As shown in Figure 8, the source of the *Mode* × *Screen* interaction is the increased difference between *MT* and *GP* (all calibration modes) between the *Wall* and *Surface*/*iPad*, with the *Wall* resulting in much lower estimation errors than the other two. It is noteworthy, that *MT-near* performs similarly to all *GP* modes on the *Surface*, but its estimation error increases drastically on the *iPad*, although all *GP* modes remain at their level. We subsequently ran separate ANOVAs on *Modes* for each *Screen*, and found several significant effects. On the *Wall*, only *MT-iPad* and *GP-near* differed significantly (p < .008) indicating that nearly all modes performed similarly.

On the *Surface*, the differences become more prominent, with *GP-near* and *GP-far* outperforming all *MT* modes except *MT-near* (all p < .016). Furthermore, *GP-Surface* and *GP-iPad* differed significantly from *MT-iPad* (all p < .012). We found the most differences on the *iPad*, where all *GP* modes are significantly less error-prone than all *MT* modes (all p < 0.03). Here we again did not find any differences within *GP* and *MT* for different calibration modes. Figure 9 visualizes the mean gaze estimation errors for each of the screens and targets for both *MT* and *GP*.

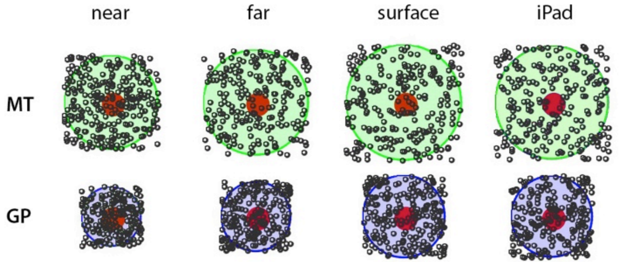

Figure 9. Visualization of the mean gaze error (ellipses) for all different modes, MT and GP, and all calibrations averaged over all targets over all screens. Additionally, black circles visualize the mean gaze estimations.

DISCUSSION

Our results show that – on a single display – *GazeProjector* achieves an average gaze estimation accuracy of 1.78° compared to 2.64° for *MT*, and 2.65° for *HO*. We used the same calibration grid for both the near and far calibration resulting in different visual angles in view space: the size of the calibrated visual field decreases when distance increases. Thus, the near calibration achieves better results than the far one. When used on multiple displays (and only being calibrated on a single screen), *GazeProjector* achieves an average gaze estimation accuracy of 2.47° compared to 3.60° for *MT* over all modes and target screens. Although this accuracy is slightly lower than the 0.5°–1° reported for the PUPIL eye tracking glasses under ideal conditions (i.e., in a stationary desktop setting with a 27” screen and optimal lighting conditions [14]), we achieve this accuracy in a fully unconstrained, pervasive interaction setting.

Pervasive Settings

The first advantage of *GazeProjector* is its suitability for pervasive gaze interaction settings [6]. Current approaches that allow for gaze interaction on multiple displays using monocular mobile eye trackers require heavyweight external motion capturing systems or visual markers. While motion capture systems allow for high-precision tracking, they are (1) costly and (2) cannot easily be installed in public environments. Markers reduce this, but have another drawback: all displays have to be augmented with them – either with printed ones attached to a display’s frame [32, 5], or digital ones shown on the display. However, printed markers quickly clutter the environment, in particular in settings with a large number of displays. While digital markers could only be shown on demand, they still take away display space and “compete” with the main content.

While binocular systems can automatically compensate for vergence error, estimating gaze in display coordinates still requires to track changes in the user’s position and orientation relative to these displays. This severely limits the use of these devices to instrumented environments. *GazeProjector*, however, allows users to interact from arbitrary locations and orientations relative to multiple displays without this need – and, as our experimental results show, *GazeProjector* does so without lowering accuracy. Thus, our approach allows for unconstrained and seamless gaze interaction with multiple displays while on the move.

Display Visibility & Multi-Display Interactions

Display tracking using visual markers requires the whole target display to be visible in the eye tracker scene camera's field of view during calibration and interaction. In contrast *GazeProjector* relies on natural feature tracking and provides competitive gaze estimation accuracy even if only a fraction of the target display is visible. Naturally, the larger the visible portion of the display, the lower the tracking error. However, we found that a quarter of the display is usually sufficient, provided that enough features (e.g., high frequencies) are found in that portion. This allows *GazeProjector* to work on much larger displays as well as with more extreme head movements than current eye trackers.

Our results show that *GazeProjector* provides robust gaze estimation accuracy for different displays of different sizes without a need for recalibrating the eye tracker to each of the displays. Instead, the eye tracker only needs to be calibrated once (on *any* display) and gaze estimates are then automatically mapped to the other displays during runtime. Applying our calibration method in the presented experiments, we were still able to achieve an accuracy of 3.24° when the eye tracker had been calibrated on a 9.7" iPad Air screen. This is a significant advancement over state-of-the-art gaze estimation approaches.

Head Movement and Orientation

We further found that head movements are more prevalent in gaze interaction with large displays compared to smaller displays (e.g., mobile devices). This finding is in line with controlled laboratory studies on human vision: humans employ head movements for gaze shifts with ocular orbital eccentricity exceeding 20° [26]. While head movements pose a significant challenge to current head-mounted eye trackers, *GazeProjector* proved to be robust to head movements, which is an essential feature for using head-mounted eye tracking systems for large screens.

Limitations

Despite its numerous advantages over state-of-the-art eye-tracking systems, *GazeProjector* also comes with some limitations: first, our current implementation requires continuous snapshots of the target displays to be transferred to a central server. Consequently, all displays need to be registered with such a server a priori. Furthermore, increasing the number of displays also increases the network load for transferring real-time updates of a display's content. However, we believe that future network technologies may overcome this limitation.

Second, *GazeProjector*'s gaze estimation accuracy depends on the quality of image data from the scene camera: these cameras usually come with wide angle lenses to cover a larger field of view, resulting in smaller representations of a target display. This may increase errors in the transformation matrix due to insufficient image features. This technical limitation can be overcome by using different lenses for scene cameras. Furthermore, as with all optical tracking systems, environmental conditions such as changes in lighting will affect our system as this influences the video quality of the scene camera. And finally, *GazeProjector's* accuracy is dependent on the number of features of a display's content [12], thus requiring feature-rich content on displays. For multiple displays, we found that wallpapers in Windows 8 are sufficiently different. Here, the server can detect potential similarities across displays through feature matching of their respective content.

Nevertheless, we believe that *GazeProjector* is a promising system that realizes continuous gaze-based interaction in pervasive settings and multi display environments.

CONCLUSIONS & FUTURE WORK

In this paper, we presented *GazeProjector*, an approach for accurate gaze estimation and seamless interaction with multiple large displays using head-mounted eye trackers. In contrast to existing systems, *GazeProjector* only requires a single calibration performed with an arbitrary display and is robust to the user's location and orientation to the displays as well as head movements. Furthermore, *GazeProjector* works without external tracking equipment, such as motion capturing systems or markers attached to display.

We conducted two experiments in which we compared *GazeProjector* to existing, well-established techniques (which require additional equipment), and found that our approach compares well to these techniques. When being used on multiple displays, the results are even more promising. Overall, our results underline the significant potential to finally bring gaze-based interaction into pervasive settings that involve gaze interaction with multiple displays.

We tested *GazeProjector* in a laboratory environment to gain first insights into its performance compared to existing techniques. However, we want to take our approach one step further. One obvious step is to take it to the real world and evaluate its performance on (1) ultra-large displays, such as media façades, and (2) do so with multiple users simultaneously. This will further add to the eye tracking community, as it has virtually been impossible to test eye-tracking systems in such large scales.

ACKNOWLEDGMENTS

We would like to thank our participants in the experiments for their time and effort. Furthermore, we deeply thank our reviewers for their valuable feedback.

REFERENCES

1. Alahi, A., Ortiz, R. and Vandergheynst, P. FREAK: Fast retina keypoint. *Proc. CVPR* 2012, 510-517.
2. Ballagas, R., Borchers, J., Rohs, M. and Sheridan, J. The smart phone: A ubiquitous input device. *IEEE Pervasive Computing* 5(1), 70-77, 2006.

3. Baur, D., Boring, S. and Feiner, S. Virtual projection: exploring optical projection as a metaphor for multi-device interaction. *Proc. CHI 2012*, 1693-1702.
4. Boring, S., Baur, D., Butz, A., Gustafson, S. and Baudisch, P. Touch projector: mobile interaction through video. *Proc. CHI 2010*, 2287-2296.
5. Breuninger, J., Lange, C., and Bengler, K. Implementing Gaze Control for Peripheral Devices. *Proc. PETMEI 2011*, 3-8.
6. Bulling, A., and Gellersen, H. Toward Mobile Eye-based human-computer interaction. *IEEE Pervasive Computing* 9(4):8-12, 2010.
7. Bulling, A., Alt, F., and Schmidt, A. Increasing the security of gaze-based cued-recall graphical passwords using saliency masks. *Proc. CHI 2012*, 3011–3020.
8. Cerrolaza, J. J., Villanueva, A., Villanueva, M. and Cabeza, R. Error characterization and compensation in eye tracking systems. *Proc. ETRA 2012*, 205-208.
9. Eaddy, M., Blasko, G., Babcock, J. and Feiner, S. My own private kiosk: Privacy-preserving public displays. *Proc. ISWC 2004*, 132-135.
10. Guitton D. and Volle M. Gaze control in humans: eye-head coordination during orienting movements to targets within and beyond the oculomotor range. *Journal of Neurophysiology* 58:427–459, 1987.
11. Hennessey, C. and Fiset, J. Long Range Eye Tracking: Bringing Eye Tracking into the Living Room. *Proc. ETRA 2012*, 249-252.
12. Herbert, L., Pears, N., Jackson, D. and Olivier, P. Mobile device and intelligent display interaction via scale-invariant image feature matching. *Proc. PECCS 2011.*
13. Jacob, R. J. K. What you look at is what you get: eye movement-based interaction techniques. *Proc. CHI 1990*, 11–18.
14. Kassner, M., Patera, W. and Bulling, A. Pupil: An Open Source Platform for Pervasive Eye Tracking and Mobile Gaze-based Interaction. *Adj. Proc. UbiComp 2014*, 1151-1160.
15. Lowe, D.G. Object recognition from local scale-invariant features. *Proc. ICCV 1999*, 1150-1157.
16. Mardanbegi, D. and Hansen, D.W. Mobile Gaze-based Screen Interaction in 3D Environments. *Proc. NGCA 2011*, 2:1--2:4.
17. Marquardt, N., Diaz-Marino, R., Boring, S., and Greenberg, S. The proximity toolkit: prototyping proxemic interactions in ubiquitous computing ecologies, *Proc. UIST 2011*, 315-326.
18. Majaranta, P., and Kari-Jouko R. Twenty years of eye typing: systems and design issues. *Proc. ETRA 2002*, 15-22.
19. Model, D. and Eizenman, M. A General Framework for the Extension of the Tracking Range of User-Calibration-Free Remote Eye-Gaze Tracking Systems, *Proc. ETRA 2012*, 253-256.
20. Nakanishi, Y., Fujii, T., Kiatjima, K., Sato, Y. and Koike, H. Vision-based face tracking system for large displays. *Proc. UbiComp 2002*, 152–159.
21. Pears, N., Jackson, D.G. and Olivier, P. Smart phone interaction with registered displays. *IEEE Pervasive Computing* 8(2), 14-21, 2009.
22. Sibert, L. E. and Jacob, R. J. K. Evaluation of eye gaze interaction. *Proc. CHI 2000*, 281–288.
23. Sippl, A., Holzmann, C., Zachhuber, D. and Ferscha, A. Real-time gaze tracking for public displays. *Proc. AmI 2010*, 167–176.
24. San Agustin, J., Hansen, J. P. and Tall, M. Gaze-based interaction with public displays using off-the-shelf components. *Adj. Proc. Ubicomp 2010*, 377–378.
25. Smith, J. D., Vertegaal, R. and Sohn, C. ViewPointer: lightweight calibration-free eye tracking for ubiquitous handsfree deixis. *Proc. UIST 2005*, 53-61.
26. Stahl, J. S. Amplitude of human head movements associated with horizontal saccades. *Experimental Brain Research* 126.1: 41-54, 1999.
27. Stellmach, S. and Dachselt, R. Look & touch: gaze-supported target acquisition. *Proc. CHI 2012*, 2981-2990.
28. Stellmach, S. and Dachselt, R. Still looking: investigating seamless gaze-supported selection, positioning, and manipulation of distant targets. *Proc. CHI 2013*, 285-294.
29. Turner, J., Bulling, A. and Gellersen, H. Extending the visual field of a head-mounted eye tracker for pervasive eye-based interaction. *Proc. ETRA 2012*, 269-272.
30. Vertegaal, R. Attentive user interfaces. *Communications of the ACM* 46(3):30–33, 2003.
31. Vidal, M., Bulling, A. and Gellersen, H. Pursuits: Spontaneous Interaction with Displays based on Smooth Pursuit Eye Movement and Moving Targets. *Proc. UbiComp 2013*, 439-448.
32. Yu, L., and Eizenman, E. A new methodology for determining point-of-gaze in head-mounted eye tracking systems. *IEEE Transactions on Biomedical Engineering* 51(10):1765-1773, 2004.
33. Zhai, S., Morimoto, C. and Ihde, S. Manual and gaze input cascaded (MAGIC) pointing. *Proc. CHI 1999*, 246–253.
34. Zhang, Y., Bulling, A. and Gellersen, H. Sideways: A Gaze Interface for Spontaneous Interaction with Situated Displays. *Proc. CHI 2013*, 851-860