EyeScout: Active Eye Tracking for Position and Movement Independent Gaze Interaction with Large Public Displays

Mohamed Khamis[1], **Axel Hoesl**[1], **Alexander Klimczak**[1], **Martin Reiss**[1],
Florian Alt[1], **Andreas Bulling**[2]
[1]Ubiquitous Interactive Systems Group, LMU Munich, Germany
[2]Max Planck Institute for Informatics, Saarland Informatics Campus, Germany
{mohamed.khamis, axel.hoesl, florian.alt}@ifi.lmu.de, bulling@mpi-inf.mpg.de

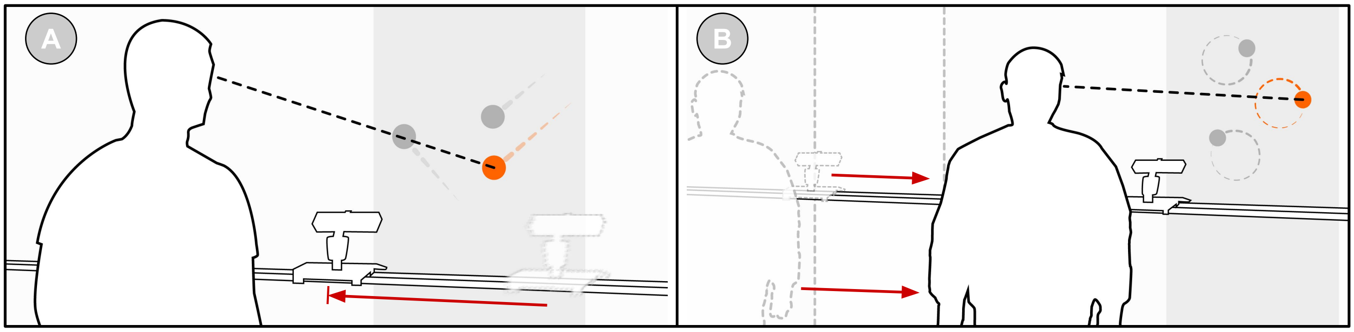

Figure 1. EyeScout is an active eye tracking system that enables gaze interaction with large public displays. It supports two interaction modes: In "Walk *then* Interact" the user can walk to a location in front of the display and the system positions itself accurately to enable gaze interaction (A). In "Walk *and* Interact" the user can walk along the display and the system follows the user, thereby enabling gaze interaction while on the move (B).

ABSTRACT
While gaze holds a lot of promise for hands-free interaction with public displays, remote eye trackers with their confined tracking box restrict users to a single stationary position in front of the display. We present EyeScout, an active eye tracking system that combines an eye tracker mounted on a rail system with a computational method to automatically detect and align the tracker with the user's lateral movement. EyeScout addresses key limitations of current gaze-enabled large public displays by offering two novel gaze-interaction modes for a single user: In "Walk *then* Interact" the user can walk up to an arbitrary position in front of the display and interact, while in "Walk *and* Interact" the user can interact even while on the move. We report on a user study that shows that EyeScout is well perceived by users, extends a public display's sweet spot into a sweet line, and reduces gaze interaction kick-off time to 3.5 seconds – a 62% improvement over state of the art solutions. We discuss sample applications that demonstrate how EyeScout can enable position and movement-independent gaze interaction with large public displays.

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI)

Author Keywords
Gaze Estimation; Body Tracking; Gaze-enabled Displays

UIST 2017, October 22–25, 2017, Quebec City, QC, Canada

ACM 978-1-4503-4981-9/17/10...$15.00
https://doi.org/10.1145/3126594.3126630

INTRODUCTION
Over the last years we have witnessed a significant increase in the number and size of displays deployed in public. Large displays are now commonly found in public communal spaces, such as shopping malls or transit areas in airports and train stations [13]. At the same time, as sensing technologies are becoming cheaper and more robust, various modalities for interacting with these displays have been explored. A particularly promising interaction modality is gaze, given that it is fast, natural, and intuitive to use [41].

However, in contrast to common desktop settings, public displays afford ad-hoc use over short periods of time by "passersby", i.e. users who move in front of the display [26, 27]. These characteristics pose three unique challenges that have so far forestalled wider adoption of gaze interaction on large public displays: 1) Gaze-enabled public displays cannot afford time-consuming eye tracker calibration for each user prior to interaction [21], 2) they have to allow passersby to interact from different positions [36] and 3) they have to support interactions while on the move [30]. Previous work mainly addressed the first challenge [22, 34, 40]. To date, addressing the latter two currently requires augmentation of each individual user with head-mounted eye trackers as well as interconnected displays [24].

To address the last two challenges we present EyeScout, a novel active eye tracking system that enables gaze interaction for a single user on large public displays from different lateral positions in front of the display and while on the move. EyeScout consists of a body detection and tracking module using a depth sensor, and an eye tracking module using an eye tracker mounted on a rail system. Our system detects the user's position in front of the display and then moves the eye tracker to face and follow the user. EyeScout thereby enables gaze interaction with large public displays that are (1) *position-independent*: the user can interact from different positions within 90 cm in front of the display and along the display's full extent, and (2) *movement-independent*: the user can interact via gaze while passing by the display. EyeScout builds on existing work that employed Pursuits [34], a popular calibration-free gaze interaction technique. It is important to note that EyeScout readily supports other gaze interaction techniques, such as gaze gestures [14] or pupil-canthi-ratio [40].

The specific contributions of this work are three-fold: First, we introduce the design and implementation of EyeScout, a novel active eye tracking system that enables gaze interaction for a single user on large displays. Second, we report on our findings from a controlled laboratory study (N=23) to evaluate the performance of EyeScout. We evaluate EyeScout for scenarios in which users "Walk *then* Interact" (to test for position independence), as well as "Walk *and* Interact" (to test for movement independence). Findings from our study show that EyeScout is well-suited for both interaction modes and well-perceived by users. In particular, EyeScout reduces the time required to kick-off gaze interaction (i.e., the time starting from the moment the user appears in front of the display until the time the user interacts) to 3.5 seconds – a 62% improvement over state-of-the-art methods [1, 41]. Finally, we discuss how active eye tracking using EyeScout can enable novel gaze-based applications that are not possible with current systems, such as gaze interaction with non-planar displays, and on escalators and moving walkways.

RELATED WORK

Our work builds on three strands of prior work: (1) Gaze interaction with public displays and (2) active eye tracking.

Gaze Interaction With Public Displays

Gaze holds particular promise for interaction with public displays given that it, for example, overcomes the embarrassment problems associated with mid-air gestures [8], reflects attention [32], and allows at-a-distance interactions [17]. However, gaze-enabled public displays also face several unique challenges. First, although eye tracker calibration may be acceptable in desktop settings where users interact for long periods of time, interaction time on public displays is short [26, 27], which makes time-consuming tasks, in particular calibration, undesirable [21]. Recent works have therefore either tried to improve calibration [22] or employed calibration-free interaction techniques [19, 34, 40].

Second, in contrast to desktop settings, users approach public displays from different directions and want to interact with them from different locations, distances, and relative orientations [36]. However, existing gaze-enabled public displays restrict users' position; users have to position themselves within the tracking box of the eye tracker for their eyes to be detected [20, 22, 34, 41]. A common approach to address this problem is to guide passersby to the right position in front of the eye tracker, for example, using on-screen mirrored video feeds [41], markers on the floor [20], or on-screen visual cues that are adapted based on the user's distance to the display [1]. In contrast, EyeScout moves the eye tracker to the user as soon as they approach the display or as they walk along it. An alternative approach is to use head-mounted eye tracking that allows for freedom of movement. However, this approach requires the eye tracker to (1) identify the position and borders of surrounding displays, (2) map gaze estimates to on-screen positions, and (3) communicate gaze data to the display. Prior work attached printed markers on the display [39] or used on-screen visual markers [15] to locate the display and map the gaze estimates onto it. These approaches usually rely on a tethered connection to the display. Lander et al. used visual feature tracking to determine the positions of surrounding displays, and exchanged gaze data over Wifi [24].

Although mobile trackers are starting to become ubiquitous and integrated into eyewear computers and despite the vision of pervasive display networks [13], pervasive integration on such a big scale would require taking concepts from lab settings to the field. In-field application is currently challenging, as participants need to be explicitly hired and asked to wear mobile eye trackers [12]. Until passersby wearing mobile eye trackers becomes the norm, there is a need to study user behavior on gaze-enabled public displays using other means, such as remote eye trackers.

Active Eye Tracking

One way to achieve position-independent gaze interaction with public displays is by using active eye tracking, i.e. systems that adapt to the user's eye position rather than restricting their head and/or body movements. Active eye tracking is particularly popular in medicine; it is used in eye surgery to account for eye and body movements during lasik operations [25]. A common approach is to use a single [9, 10, 11, 29] or multiple pan-and-tilt cameras [6], or pan-and-tilt mirrors [28] to adapt to the user's head position. Hennessey and Fiset used a Kinect to detect faces and adjust the angle of an eye tracker mounted on a pan and tilt mechanism accordingly [17]. While all of these methods demonstrated the potential of active eye tracking, EyeScout is first to *move* the eye tracker rather than only panning and tilting it in a single fixed position. This way, EyeScout actively accommodates for the user's body movements along large displays.

THE EYESCOUT SYSTEM

When interacting with large, cylindrical or spherical displays, users approach from different directions and do not necessarily interact from a static position in front of the display [1]. Instead, passersby expect to be able to walk-up to the display and interact from any position. We refer to this interaction mode as "Walk *then* interact". Similarly, passersby move at different speeds and often interact with displays while moving [30]. We refer to this interaction mode as "Walk *and* interact". The key challenges in both interaction modes are that the system needs to detect the user's eyes at arbitrary stationary positions or while the user moves in front of the display.

We designed EyeScout for single user gaze interaction specifically with these two interaction modes and associated challenges in mind. The design was inspired by camera motion

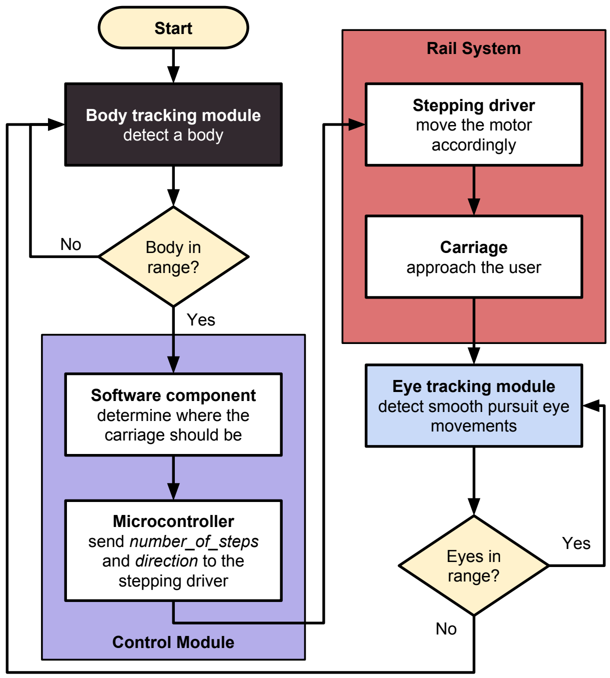

Figure 2. The body tracking module lies at the outset of EyeScout. Once a body is detected, the control module determines the target position of the carriage which carries the eye tracker, and instructs the rail system to move it to the body's position. The eye tracking module then tracks the user's eyes. If eyes are no longer detected, the system determines where the user is and moves the carriage to the new position.

technologies used in cinematography [18], and by actuated displays [2, 31]. EyeScout consists of four main modules (see Figure 2 and Figure 3): a body tracking module, a rail system, an eye tracking module, and a control module. Each module can be run and debugged independently allowing it to be modified or replaced. This maximizes ease of assembly and allows straightforward integration of new modules. In the following we describe each module in details.

Body Tracking Module

The body tracking module consists of a motion sensing camera (Microsoft Kinect One). The Kinect is mounted on a tripod (150 cm high) aligned with the center of the rail. This module determines the user's position in front of the display. We used the skeletal joints provided by the Kinect API to detect the user's position. The Kinect detects the user immediately as soon as they are in range. We placed the Kinect at the opposite side of the display to cover a large area (see Figure 3).

Rail System

The rail system consists of a 4-meter twin track aluminium rail[1] and a carriage[2] to move the eye tracker. Two 3D-printed end pieces were attached at both ends of the rail (see Figures 4A and 4C). The end pieces serve two main purposes: (1) To hold switches that are activated once the carriage reaches any of the ends. The switches are used for a one-time

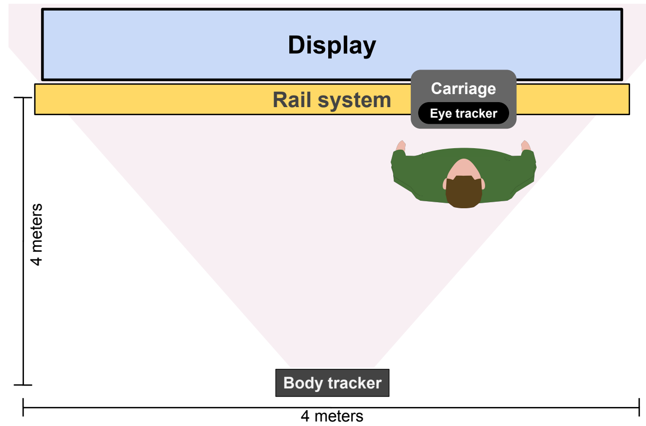

Figure 3. An illustration of the top view of the apparatus (not to scale). The body tracking module (Kinect One) is positioned 4 meters away from a 4 meters wide projected display. The carriage, which carries the eye tracker (Tobii REX), moves along the rail system according to the commands received from the body tracking module.

system calibration that determines the bounds of the rail's range to prevent the carriage from colliding with other components of the end pieces. (2) Each end piece harbors a steel axis that holds a pulley. A motor, whose axis is connected to the steel axis, is mounted on one of the end pieces. Thus, when the motor spins, the axis is spun and moves a tightened timing belt that moves the carriage. In addition to a DC motor[3], we used a 2-phase digital stepper driver[4] to convert digital signals to commands that can control the motor. The entire rail system was mounted above three evenly distributed tripods (height: 113 cm), as depicted in Figure 5A.

Eye Tracking Module

The eye tracking module consists of a remote eye tracker (Tobii Rex) and a custom 3D-printed mount. The mount is attached to a tripod head that allows adjusting the angle of the eye tracker. The tripod head is in turn attached to another 3D-printed base that is screwed into the carriage (Figure 4B). This module is responsible for tracking the eyes once they are in range. The minimum and maximum range of the eye tracker (40 cm to 90 cm in our case) are predefined in the body tracking module. This allows the body tracking module to detect when users are too close or too far away from the eye tracker. Similarly, the eye tracking module continuously detects whether or not the user's eyes are detected. This information can then be used to provide feedback to the user.

For gaze interaction with our system we use Pursuits [34], which has been widely adopted recently for calibration-free gaze interaction. Pursuits checks for motion correlation [33] between user's eye movements and trajectories of on-screen targets . The strength of the method lies in its ability to determine which object the user is gazing at without the need for calibrating the eye tracker to each user. As public displays require immediate usability and cannot afford the time spent for eye tracker calibration [21], Pursuits is well-suited for use

[1] igus® drylin® Double rail http://www.igus.eu/wpck/2003/drylin_w_doppelschiene

[2] igus® drylin® Double rail/carriage http://www.igus.eu/wpck/8980/drylin_w_Slider_Schienen

[3] igus® drylin® step motor NEMA23 http://www.igus.eu/wpck/7663/N11_6_14_2_Schrittmotor_NEMA23

[4] Leadshine DM556 http://www.leadshine.com/UploadFile/Down/DM556m.pdf

in the context of public displays. Moreover, as users will be moving, gaze estimates can be expected to have low accuracy as humans naturally bob up and down while walking. Its robustness to inaccurate gaze data makes Pursuits even more suitable for our deployment.

Our implementation of Pursuits is based on prior work; we used Pearson's product-moment correlation with a threshold of 0.85 and a window size of 500 ms [34]. This means that the system computes Pearson's correlation every 0.5 s. This is similar to previous work, some of which used a 0.5 s window size [20, 34], while Orbits [16] and TextPursuits [22] used 1 s and 2 s respectively. The stimulus whose movement correlates the most with the eye movement is deemed to be the one being looked at, if the correlation is higher than 85%.

Control Module

The control module handles the logic of EyeScout and the communication between the body tracking module and the rail system. It consists of a software component written in C# and a microcontroller (Arduino Due). The software component runs on a Microsoft Surface Pro that is connected to the Kinect via USB. It receives the coordinates of the user's body from the body tracking module. According to the predefined distance between the Kinect and the rail system (4 meters in our implementation), the software computes an optimal position for the carriage at which the user's eyes would be in the eye tracker's range. Based on the coordinates that are received from the eye and body tracking modules, text prompts are shown to instruct the user to stand back or come closer to the system if necessary. These coordinates can be sent to the microcontroller via Bluetooth or USB.

Given the current position of the carriage, the Arduino Due maps the new coordinates received from the software component to a number of motor steps in a specific direction. These values are then forwarded to the stepping motor, which moves the motor accordingly. The microcontroller also interacts with the switches that are attached to the end pieces. In the aforementioned calibration process, the microcontroller determines the bounds of the rail's range and updates them internally if necessary. After successful calibration, the carriage is never instructed to move far enough to touch the switches again (see Figure 4A). For example, if a user moves out of range the carriage will stop right before it touches the switch. For additional security, the microcontroller issues an emergency stop command in case the carriage touches the switches after calibration, and resets the boundary values. Although the tasks of the microcontroller could also be performed by the software module, we opted for separating the component that interacts with the rail system and the one that interacts with the body tracking module to further minimize the dependencies between different modules. Additionally, while the heavy traffic generated by the body tracking module often requires a tethered connection to the computer (in our case a Kinect One is connected via USB), microcontrollers can be communicated with wirelessly (e.g., via Bluetooth), which would not necessitate long cables between the Kinect and the rail system; the computer could stay next to the Kinect, and the microcontroller could stay next to the rail system (see Figure 3).

EVALUATION

We designed a controlled laboratory study to evaluate the performance of EyeScout for both interaction modes: (1) Walk *then* interact: the user approaches the display then interact while *stationary* and (2) Walk *and* interact: the user interacts with the display while *moving* at different speeds.

Participants

We recruited 26 participants (11 females) aged 19 to 37 years ($M = 26.77$, $SD = 4.46$). All participants had normal or corrected-to-normal vision. Four had previous experience with body tracking devices such as Kinect, out of which three participants had prior experience with eye tracking. One participant was not detected by the Kinect due to wearing a black outfit, and thus was excluded from the analysis.

Apparatus

We deployed EyeScout in one of our lab spaces (7.15 m $\times$ 5.65 m). The system was placed parallel to the wall at a distance of 117 cm (see Figure 5A). We used a short throw projector (1920 $\times$ 1080 pixels) positioned 92 cm from the wall. The eye tracker angle was adjusted at 50° to the display.

Study Design and Procedure

The study was split into two experiments, each evaluating one of the two interaction modes. All participants took part in both experiments. Half of the participants started with "Walk *then* Interact" while the other half started with "Walk *and* Interact". Each experiment also followed a within-subjects design in which all participants performed all conditions.

The experimenters started by introducing the study and asking the participant to sign a consent form. In both experiments, the system showed a white vertical rectangle (the "interaction frame") in which three dots moved in either linear or circular trajectory (see Figure 5). The participant's task was to select the red moving dot from among the grey ones via Pursuits, i.e., by simply following it with their eyes. We picked two trajectory types that are commonly used in implementations of Pursuits: circular [16, 20] and linear [22, 34] trajectories. All simultaneously shown moving dots were selectable and followed the same trajectory. We predefined 8 arrangements for moving dots (see sample arrangement in Figure 5B), 4 of which followed linear trajectories, while the other 4 followed circular trajectories. Figures 5C and 5D show one example of each. The participant was shown one arrangement (i.e., one set of three selectable dots moving in the same trajectory) at a time. In each selection, participants had to select 1 of 3 targets. Dots disappeared after being selected. After participants had performed all selections in both experiments, they filled in a questionnaire and participated in a semi-structured interview.

Experiment 1: Walk then Interact

To study the interaction mode where passersby approach a random spot in front of the display and then interact while stationary, the interaction frame appeared at a random position on the display in this experiment. The participant was asked to walk to the frame and then select the red dot via Pursuits. The carriage approached the participant as he/she approached the frame, to ultimately position the eye tracker in front of

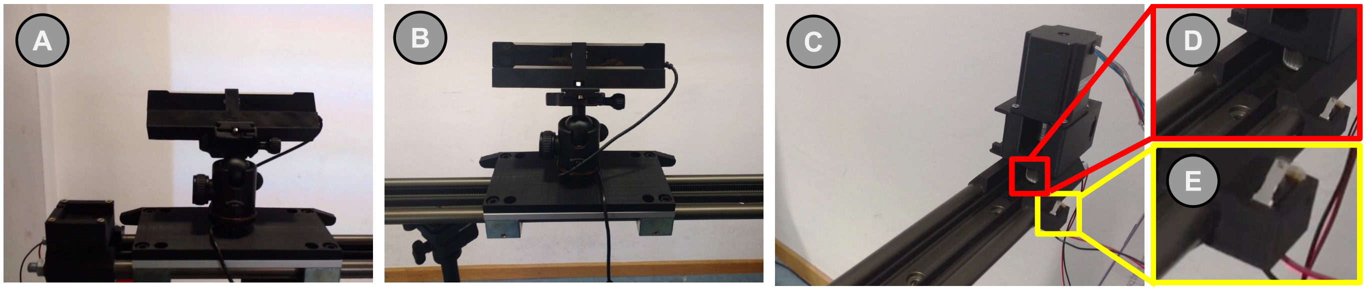

Figure 4. (A) and (C) show the 3D-printed end pieces attached at the ends of the rail. Each end piece harbors a pulley that moves the timing belt (D), and a switch (E) that prevents the carriage from accidentally colliding with the end piece. One end piece carries the motor (C). A 3D-printed base is screwed into the carriage (B), on which a tripod head is attached. The head allows the adjustment of the eye tracker, held by a 3D-printed holder.

the user. After a successful selection, the frame reappeared at another random position that was at least 50 cm away from the previous one. This was done to ensure that the participant had to approach a different spot in front of the display, rather than performing the selection from the current position. Each participant performed 3 blocks, each of which covered one selection per trajectory arrangement. Thus, every participant performed 8 trajectory arrangements $\times$ 3 blocks $=$ 24 selections. We consider these blocks an additional independent variable, referred to in the followingas "repetitions". Repetitions were studied to investigate learning or fatigue effects. The order of conditions was counter-balanced using a Latin-square.

Experiment 2: Walk and Interact

To study the interaction mode where passersby interact while moving, in this experiment the participant, the interaction frame, and the carriage were all moving. We focus on scenarios in which users interact with content that moves with them, as typically done in large interactive displays intended for moving users [30]. To evaluate if the carriage's speed had an impact on detection accuracy we introduced an independent variable "carriage speed" with three conditions: 0.36 m/s (maximum speed of EyeScout), 0.3 m/s, and 0.24 m/s. The interaction frame would follow the participant, but the participant would be able to make a selection only when in range of the eye tracker. Each participant performed 24 selections in this experiment (8 arrangements $\times$ 3 speeds). The order of conditions was counter-balanced using a Latin-square.

Limitations

In the current version of our prototype, taller participants are asked by means of the aforementioned textual prompts to step back in order for the eye tracker to detect them. In our study, the angle of the eye tracker to the head was between 35° and 50°. The exact value depends on the user's height and distance from the tracker. Future systems can adjust the eye tracker's angle dynamically to be within this range.

Another limitation of gaze interaction while on the move is that it might be affected by motion blur. Although we did not face this problem in our study; the performance of EyeScout was almost similar in "Walk *then* Interact" (baseline) compared to "Walk *and* Interact". However we acknowledge that higher carriage speeds might result in less accurate gaze data.

Like current stationary eye trackers, EyeScout only supports a single user. Multi-user support is one of the most important directions for future work. This can be realized by mounting multiple eye trackers on different belts or by using appearance-based gaze estimation methods [32] that use multiple or a single wide angle RGB camera. In the current implementation of EyeScout, there are three possible scenarios in which a person other than the user appears in range of EyeScout: 1) passersby show up near the user, 2) passersby step between the user and the Kinect, 3) passersby step between the user and the eye tracker. The eye tracker locks onto the user whose eyes are detected even if a passerby occludes the user from the Kinect's view. This prevents disrupting interaction in case of glitching position tracking, and means that 1) and 2) do not influence the system's usability. The Kinect can be placed at the top of the interaction area to account for 2) when the user is moving. However 3), while unlikely, would result in the passerby taking over the interaction.

Finally, we can validate EyeScout's performance with higher walking speeds only after upgrading its motor. However users are likely to slow down to interact when moving [30].

Quantitative Results

Although the Kinect performed fairly well in our setup, it failed to detect one participant wearing black. We further excluded the data of two more participants: One found the task to be overwhelming; he struggled to walk around and look at multiple objects at the same time. The second squinted his eyes too often, resulting in very few collected gaze points.

We measured the *cruise time* in the "Walk *then* Interact" experiment, i.e., the time it took the carriage from the moment the interaction window appeared till the moment eyes were detected. Because this is the first time a commercial IR-PCR eye tracker is used for active eye tracking, we wanted to investigate whether there is any degradation in the eye tracker performance when it is in motion. This was done by logging the *gaze points per second* during the Walk *and* Interact experiment. In both experiments, we additionally measured the *error count*, which we define as the number of grey dots that were selected before the red dot. There can be 0, 1, or 2 errors before selecting the red dot.

Error Count

As shown in Table 1, errors decreased in the "Walk *then* Interact" experiment as participants performed more repetitions. This suggests that there could be a learning effect, i.e., participants adapted to the system. In the "Walk *and* Interact" experiment we found a slight increase in the number of errors as the carriage moved at higher speeds.

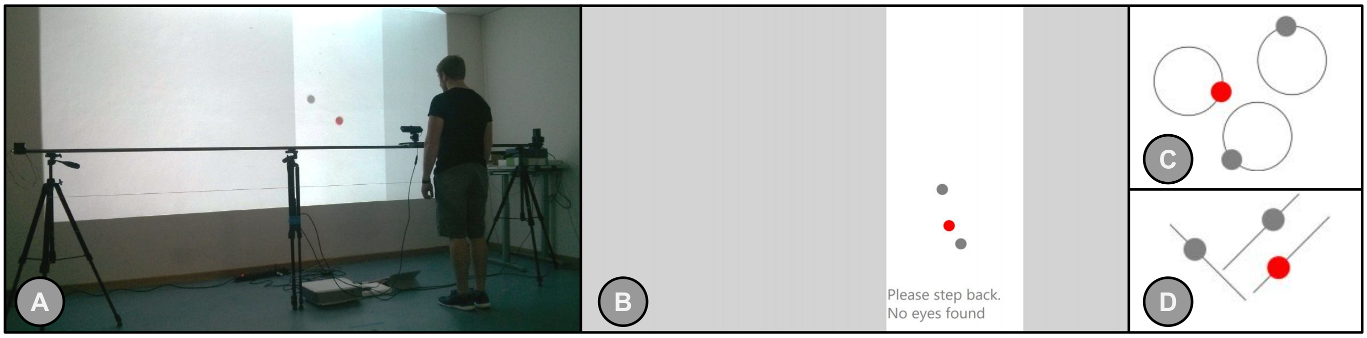

Figure 5. An "interaction window" appeared on the projected display with an arrangement showing three selectable moving dots. The dots moved either in circular trajectories (C) or in linear trajectories (D). The participant's task was to select the red dot. In the "Walk *then* Interact" experiment, the window appeared at a random place on the display; the participant had to walk to the window then perform the selection. While in the "Walk *and* Interact" experiment, the window moved along the display; the participant had to walk along the window and perform selections while moving.

	Walk *then* Interact$_{(baseline)}$			Walk *and* Interact		
ϵ	Repetition 1	Repetition 2	Repetition 3	Speed 1	Speed 2	Speed 3
0	73.9%	83.7%	79.4%	79.3%	76.1%	72.3%
1	26.1%	15.8%	20.1%	19.6%	21.7%	23.9%
2	0%	0.5%	0.5%	1.1%	2.2%	3.8%

Table 1. The table shows the percentage of times a successful selection of a red dot was preceded by 0, 1, or 2 errors (i.e., selection of a gray dot). As participants performed more selections, errors in "Walk *then* Interact" decreased. This suggests that there could be a learning effect. In "Walk *and* Interact", errors increased slightly with higher speeds.

	Walk *then* Interact			Walk *and* Interact		
	Repetition 1	Repetition 2	Repetition 3	Speed 1	Speed 2	Speed 3
μ	2.4 s	2.9 s	2.9 s	4.7 s	4.7 s	5.4 s
σ	2.1 s	2.2 s	2.2 s	6.1 s	5.8 s	7.0 s

Table 2. The table shows the mean selection time and standard deviation. Selection times in "Walk *and* Interact" are longer than in "Walk *then* Interact" due to tracking distortions caused by walking. While in "Walk *and* Interact" mean selection time is only slightly more than in previous work (e.g., 1.5 s to 2.0 s [20]).

Mean Correlation Coefficient

The mean correlation coefficient that was calculated during the experiments is 0.91 (Threshold = 0.6). This is comparable to previous work. Th mean correlation coefficient in previous work were 0.8 (Threshold = 0.6) in TextPursuits [22], 0.86 (Threshold = 0.7) in EyeVote [23], and 0.89 (Threshold = 0.8) in a study by Khamis et al. [20]. This shows that the accuracy of EyeScout is comparable to static systems.

Cruise Time

Although the maximum speed of the carriage was 0.36 meters per second, the carriage needs to accelerate and decelerate at the beginning and end of its cruise. The average cruise speed in the "Walk *then* Interact" experiment was 0.2 meters per second (SD=0.09), while the average distance between two consecutive selection areas was 0.7 meters (SD=0.34). This means that the overall mean cruise time was 3.5 seconds.

Gaze Points per Second

The eye tracker used in our prototype system returns a maximum of 30 gaze points every second. By measuring the number of collected gaze points at each carriage speed, we did not find large differences between the different conditions used in the "Walk *and* Interact" experiment. The eye tracker collected an average of 21.59 ($SD = 7.84$), 22.78 ($SD = 7.3$), and 22.57 ($SD = 7.33$) gaze points per second during the slow, medium and fast carriage speeds respectively. This means that we did not find any evidence that the performance of EyeScout is degraded by movements of up to 0.36 meters per second.

Selection time

Table 2 summarizes the selection times. The overall mean selection time in "Walk *then* Interact" is 2.7 seconds, which is only slightly more than in previous work (e.g., 1.5 s to 2.0 s [20]). In "Walk *and* Interact" mean selection time is 4.9 s, which is longer due to tracking distortions caused by walking.

Questionnaire

We asked participants about the perceived easiness and precision of selections on a 5-point Likert scale. In the "walk *then* interact" experiment, participants felt that walking up to the display *then* making selections was easy ($Mdn = 4$, $SD = 0.59$) and precise ($Mdn = 4$, $SD = 1.08$) (see Figures 6 and 7). They also agreed that the eye tracker was positioned properly in front of them ($Mdn = 4$, $SD = 0.66$). A Friedman test showed statistically significant differences in perceived easiness of selections depending on carriage speed $\chi^2(2) = 9.414, p = 0.009$ in the "walk *and* interact" experiment. Post hoc analysis with Wilcoxon signed-rank tests was conducted with Bonferroni correction, indicating significant differences ($p < 0.017$). Median perceived easiness of selection levels for the slow, medium, and fast carriage speeds were 5 (4 to 5), 4 (4 to 5) and 4 (4 to 4), respectively. There were no significant differences between medium and slow carriage speeds ($Z = -0.535$, $p = 0.593$). However, there was a statistically significant reduction in perceived easiness in fast vs slow carriage speed ($Z = -2.555$, $p = 0.011$), and in fast carriage speed vs medium carriage speed ($Z = -2.517$, $p = 0.012$). This means that selections done during slower and medium carriage speeds are perceived to be easier to perform (Figure 6). We performed a Friedman test to determine if there were differences in perceived precision across the different carriage speeds. Perceived precision was consistent ($Mdn = 4$) across the different carriage speeds and there were no statistically significant differences $\chi^2(2) = 3.073, p = 0.215$. This means there is no evidence that perceived precision changes depending on the carriage speed (Figure 7).

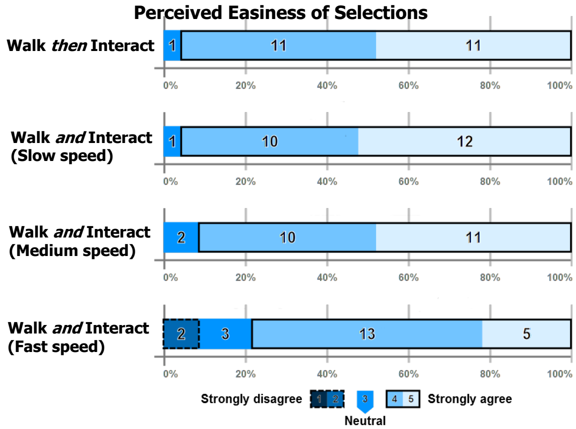

Figure 6. Overall, participants found it easy to make selections when using EyeScout. However, making selections on medium and slow speeds is perceived to be easier than on fast speeds.

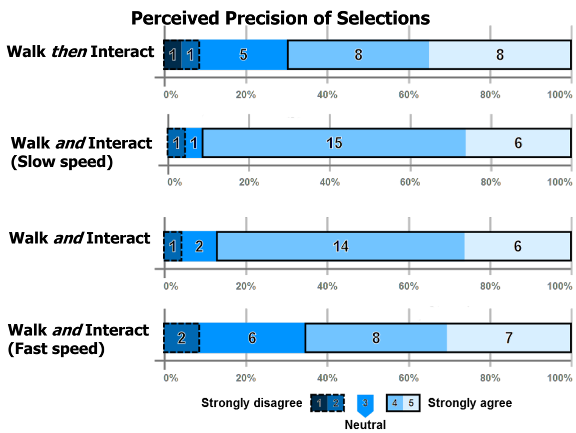

Figure 7. Participants perceived precision to be generally high. It seems that selections at higher speeds in the "Walk *and* Interact" experiment are perceived to be slightly less precise, however we found no significant differences to support this.

Observations and Qualitative Feedback

The overall feedback about the experience was very positive. Participants mentioned that they found the system "interesting" and thought it was "working surprisingly well". One participant found the experience of being followed by the eye tracker to be "futuristic". Another participant commented that he found the idea "novel" and "sci-fi". This is in-line with the novelty effect often experienced when interacting via gaze.

Participants also mentioned some aspects of EyeScout that could be improved.

Hardware Improvements

Although carefully placed to avoid tripping, three participants reported that the legs of the tripods sometimes distracted them. This suggests that future versions of EyeScout should be mounted differently. For example, the rail can be engraved into the wall underneath the display. Five participants reported that the cable that connects the eye tracker to the PC distracted them shortly when they saw it the first time. In field deployments of EyeScout, wireless technology (e.g., WiFi or Bluetooth) should be utilized instead. To further reduce distraction, the tracker could be embedded into a case with semi-transparent glass so that the current position of the tracker is not visible to users. Yet, knowledge about the position of the eye tracker might positively influence the position of the user – hence, this needs to be subject to future investigation.

A female participant with long dark hair was not always correctly detected by the Kinect due to the aforementioned problem with detecting dark objects. While she reported that the eye tracker was consequently not always in front of her, she did not report any problems regarding the responsiveness of the system. A possible direction of improvement is to place the motion sensing device at the top of the interaction area, or embed a wide range sensor into the display.

Interaction

Two participants reported they were uncomfortable with performing Pursuits against circular trajectories. Some participants also reported feeling tired after performing 48 selections using Pursuits. This feedback is in line with findings reported from lab studies of Pursuits [22]. Given that interactions in real deployments would not involve as many Pursuit selections as in our study, this fatigue effect can be expected to play a minor effect in real deployments. However, applications that expect multiple selections (e.g., games) should be designed with the fatigue effect in mind.

One participant reported not having noticed the text prompts used to guide the user closer or farther from the eye tracker. Previous work has shown that public display users sometimes miss on-screen content, and are more likely to notice it if it is attached to their user representation [35]. Another participant was not confident that the system recognized his eyes, which led him to look at the eye tracker during his first trials. However that was only before he realized that the projected screen shows feedback when eyes are not detected. This suggests that the system should always provide feedback to indicate that the eyes are detected, rather than only when they are not in range. Furthermore, future work should consider different visual feedback methods, such as continuously showing eye symbols on the screen and adapting them depending on the state of eye detection. Similar to previous work [1], the content can be adapted to subconsciously guide the user by making it completely visible only when the user is at a particular distance from the display.

Walking Strategies

When asked how to move, the experimenters told the participants that they are free to move however they liked. We noticed that participants walked in different ways in the "Walk *and* Interact" experiment. While the majority walked naturally with their head turned towards the display, some walked sideways with their entire body facing the display. Participants who walked sideways reported that it was uncomfortable, but they walked that way thinking that the system would not detect them otherwise. One participant tried both and eventually settled on the natural walk. User interfaces of commercial eye trackers explicitly tell their users to relax and act naturally; EyeScout can similarly provide such feedback when unnatural moving behavior is detected.

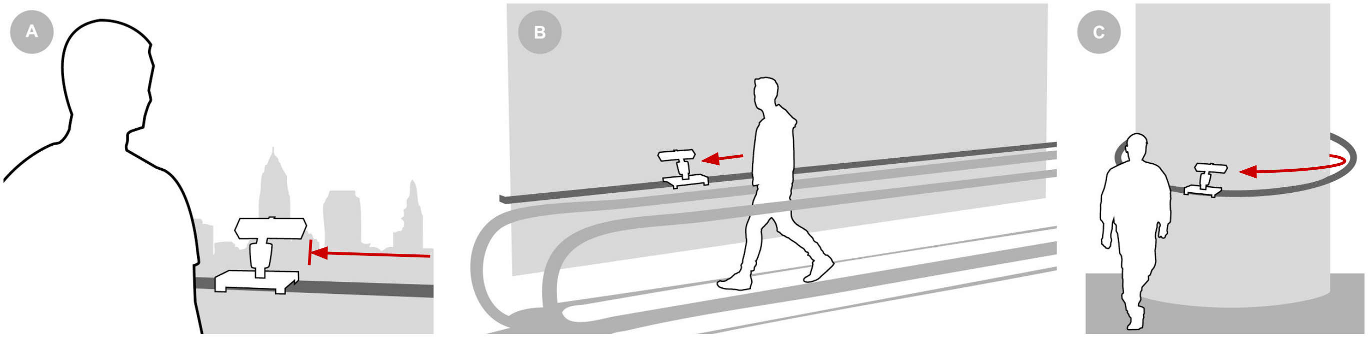

Figure 8. We envision multiple scenarios in which EyeScout can create new possibilities for gaze interaction with large displays. For example, EyeScout can enable interaction with objects or scenery (A); a user is standing at a touristic vantage point (e.g., observation deck), EyeScout can detect and track the user's gaze along the scene and provide information about the buildings and landmarks being gazed at. Users can interact while on a moving walkway (e.g., at an airport), where the tracker actively follows the moving user (B). EyeScout could also enable gaze-based interaction on non-planar displays such as cylindrical displays; the user approaches the display from any direction, then EyeScout moves the eye tracker to a position from which it can detect the user's eyes (C).

SAMPLE APPLICATIONS

We envision EyeScout to be used in a variety of scenarios. We describe three examples in which the use of EyeScout enlarges the interaction space and opens up novel possibilities for interaction via gaze that are otherwise infeasible.

Gaze Interaction with Scenery

EyeScout could be used for gaze-based interaction with a scenery (see Figure 8A). For example, observation decks and towers offer vantage points for scenic overviews (e.g., of a city or a certain landmark). On these platforms, tourists and visitors overlook a scene and are often provided with audio guides that, in a sequential manner, describe what can be seen from the platform (for example, "To your left, next to the red building, you can see the townhall."). However, this sequential feeding of information eliminates the exploratory nature of these platforms, thus hindering the tourist's experience. Even with the presence of a human tour guide, pointing at something (for example, a building or a landmark) and asking for information is not straight forward; communicating a point of interest to others by pointing is also inefficient.

We propose augmenting these platforms with EyeScout to enable eye tracking across the whole platform as shown in Figure 8A. In this scenario, the eye tracker would follow the user as he/she walks on the platform, the system would then be able to detect which buildings/landmarks the user is looking at, or allow the user to select landmarks. In-situ information about the area of interest can then be shown in the form of visual overlays or audio messages. The displayed information could be predefined in the system and loaded based on the positions of the areas of interest relative to the platform, and the position of the user. Since this application requires a precise gaze point, calibration might be required. A direction for future work is to investigate how well calibration-free techniques such as TextPursuits [22] perform in scenarios where users are moving. Another alternative is to use gaze gestures (e.g., right and left) to allow users to select the landmark they want to learn about.

Eye Tracking on Moving Walkways and Escalators

Moving walkways (aka travelators) and escalators can be found in large numbers in airports, supermarkets, ski resorts, museums, and public transport stations. People using them are usually presented with static content at one or both sides of the walkways, such as advertisements. Gaze interaction or attention measurements in walkways is infeasible using current systems and techniques, unless each passersby is augmented with a head-mounted eye tracker.

Walkways and escalators could be augmented with EyeScout, such that the eye tracker would follow the user (see Figure 8B). In static non-interactive contexts, eye tracking enabled through EyeScout could provide information about the passersby's attention (for example, which advertisements passersby look at). The content displayed on one or two sides can also be interactive, in this case interacting with a UI while standing on a moving walkway via touch would be challenging unless the UI moves with the user. On the other hand, in cases where the user "passes by" the content (for example, moving walkway surrounded by stationary exhibits[5]), interaction via gaze extends the user's reach, as the user's gaze vector could reach farther areas compared to interaction via touch or via mid-air gestures.

Gaze-based Interaction with Non-Planar Displays

There has been a recent interest in deploying and studying passersby behavior in front of non-planar displays such as cylindrical [4, 5, 7] and spherical [3, 37, 38] displays. Due to their form factor, the requirement of adapting to users approaching and interacting from different directions and positions becomes even more prominent. To date, the strict positioning constraints imposed by eye trackers make gaze interaction infeasible with such non-planar displays.

Although we evaluated EyeScout only in the context of a large planar display, the same concept is applicable to non-planar ones by using a circular rail system (Figure 8C). A camera mounted on the top of the display could detect surrounding motion and move the eye tracker to intercept passersby as they approach the display. Gaze could then be tracked to understand which content the passersby attend to, or to enable interaction as they move around the display. We believe such displays to be particularly useful in guiding users to less crowded areas of a public space.

[5]https://www.nga.gov/exhibitions/villarealinfo.shtm

DISCUSSION

Findings from our study show that EyeScout successfully overcomes the positioning requirements imposed by classical eye tracking systems, and is flexible to lateral movements in front of the display at different speeds. We also found that it is well perceived by users, who reported finding it generally easy and precise to perform selections using Pursuits.

Improvement over State-of-the-art

62% – 87% Faster in Kickstarting Gaze Interaction

State-of-the-art methods for guiding passersby to the sweet spot – which is, in our case, the area in which the user is detected by the eye tracker – were reported to require 4.5 to 23 seconds [1]. After reaching the sweet spot, users typically need to align their face to the correct position in front of the gaze-enabled display. Recent work reported that users required 4.8 seconds for the face alignment on a gaze-enabled display [41]. By adding these values, we can expect that even for state-of-the-art methods, passersby need 9.3 to 27.8 seconds before they can start interaction via gaze. EyeScout reduces this time to 3.5 seconds, which represents a 62% to 87% improvement. This improvement is due to EyeScout not requiring users to move to the sweet spot, nor to align their faces, but instead "doing the work for them". EyeScout still requires less time compared to previous approaches although it informs participants if they are too far from or too close to the eye tracker, and asks them to reposition accordingly.

Previous work showed that unless displays are immediately usable, users abandon them [26, 27]. Hence, we expect an increase in conversion rates due to EyeScouts reduction of kick-off time. In future work, this increase can be quantified through a field study.

From "Sweet Spot" to "Sweet Line"

EyeScout maximizes the horizontal flexibility of public displays. While previous work report an optimal interaction spot (the sweet spot [27]), our work extends the sweet spot to a *sweet line*: an area with the width of the display, and the length of the eye tracker's range. Future work can further extend the distance to the screen by incorporating 3D vector rigs[6].

Gaze-based Interaction on the Move

Although the Tobii REX eye tracker that we used is intended for stationary settings, it performed fairly well when in motion. The number of collected gaze points stayed almost the same across the different cruise speeds and was sufficient to perform Pursuits-based selections. Although there is a slight increase in error when using faster cruise speeds compared to slower ones (see Table 1), the accuracy of selections achieved in the "Walk *then* Interact" experiment do not differ much from those in the "Walk *and* Interact" experiment. We furthermore found that participants generally felt selections to be easy and precise but results were in favor of slower speeds compared to faster ones in the "Walk *and* Interact" experiment. The differences between the perception of easiness and precision of both experiments were not significant. Figures 6 and 7 suggest that participants perceived selections positively in all modes.

[6] http://www.vector-cam.com/services.html

Gaze Interaction Techniques Other than Pursuits

We opted to use Pursuits because it is the state-of-the-art method for calibration-free gaze interaction with public displays and it addresses the first of the three challenges mentioned at the beginning of this paper. However, EyeScout is not limited to this technique. It is straight forward to replace Pursuits by other calibration-free techniques such as eye gestures [14] or pupil-canthi-ratio [40]. Furthermore, future work can experiment with calibrating the eye tracker implicitly while users are interacting as in TextPursuits [22] to collect more accurate gaze points.

However, we cannot claim that all methods can be adapted into active eye tracking scenarios; Pursuits is robust to uncalibrated gaze points, which is likely one of the reasons it performs well while the eye tracker was in motion. It will be interesting to see whether techniques that require accurate gaze estimates, such as dwell time, can be used with EyeScout or whether the applications enabled by our system will remain infeasible if these techniques are used.

Upgrading EyeScout

The gaze-interaction technique is not the only upgradable component of EyeScout. The way EyeScout is designed allows straightforward upgrades and improvements to the different hardware and software modules. Basically all modules are upgradable, including the motor, the eye tracker, the body tracker, and even the control unit. For example, a stronger motor can be used to increase the cruise speed, a wide-angle RGB camera can be used instead of infrared-based eye trackers, and body positions can be detected via on-body sensors.

CONCLUSION

In this work we introduced the design and implementation of EyeScout, a novel active eye tracking system that addresses two challenges that were unsolved in research on gaze-enabled public displays to date. Findings from a user study show that EyeScout is not only well-perceived but also allows passersby to interact with large displays (1) from different positions and (2) while one the move. We furthermore introduced several sample applications that demonstrate how active eye tracking can enable new interactions with gaze that were not possible before. Our detailed description of EyeScout's implementation is valuable for researchers and practitioners alike who would like to employ active eye tracking into their public displays.

ACKNOWLEDGEMENTS

Thanks to Patrick Mörwald for his help in developing EyeScout. Work on this project was partially funded by the Bavarian State Ministry of Education, Science and the Arts in the framework of the Centre Digitisation.Bavaria (ZD.B).

REFERENCES

1. Florian Alt, Andreas Bulling, Gino Gravanis, and Daniel Buschek. 2015. GravitySpot: Guiding Users in Front of Public Displays Using On-Screen Visual Cues. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 47–56. DOI: http://dx.doi.org/10.1145/2807442.2807490

2. Gilles Bailly, Sidharth Sahdev, Sylvain Malacria, and Thomas Pietrzak. 2016. LivingDesktop: Augmenting Desktop Workstation with Actuated Devices. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 5298–5310. DOI: http://dx.doi.org/10.1145/2858036.2858208
3. Hrvoje Benko, Andrew D. Wilson, and Ravin Balakrishnan. 2008. Sphere: Multi-touch Interactions on a Spherical Display. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology (UIST '08)*. ACM, New York, NY, USA, 77–86. DOI: http://dx.doi.org/10.1145/1449715.1449729
4. Gilbert Beyer, Florian Alt, Jörg Müller, Albrecht Schmidt, Karsten Isakovic, Stefan Klose, Manuel Schiewe, and Ivo Haulsen. 2011. Audience Behavior Around Large Interactive Cylindrical Screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 1021–1030. DOI: http://dx.doi.org/10.1145/1978942.1979095
5. Gilbert Beyer, Florian Köttner, Manuel Schiewe, Ivo Haulsen, and Andreas Butz. 2013. Squaring the Circle: How Framing Influences User Behavior Around a Seamless Cylindrical Display. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 1729–1738. DOI: http://dx.doi.org/10.1145/2470654.2466228
6. David Beymer and Myron Flickner. 2003. Eye gaze tracking using an active stereo head. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, Vol. 2. II–451–8 vol.2. DOI: http://dx.doi.org/10.1109/CVPR.2003.1211502
7. John Bolton, Peng Wang, Kibum Kim, and Roel Vertegaal. 2012. BodiPod: Interacting with 3D Human Anatomy via a 360° Cylindrical Display. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems (CHI EA '12)*. ACM, New York, NY, USA, 1039–1042. DOI: http://dx.doi.org/10.1145/2212776.2212380
8. Harry Brignull and Yvonne Rogers. 2003. Enticing people to interact with large public displays in public spaces. In *In Proceedings of the IFIP International Conference on Human-Computer Interaction (INTERACT 2003*. 17–24.
9. Chao-Ning Chan, Shunichiro Oe, and Chern-Sheng Lin. 2007a. Active Eye-tracking System by Using Quad PTZ Cameras. In *Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE*. 2389–2394. DOI: http://dx.doi.org/10.1109/IECON.2007.4460366
10. Chao Ning Chan, Shunichiro Oe, and Chern-Sheng Lin. 2007b. Development of an active gaze tracking system in unrestricted posture. In *International Conference on Control, Automation and Systems, 2007. ICCAS '07*. 1348–1353. DOI: http://dx.doi.org/10.1109/ICCAS.2007.4406547
11. Dong-Chan Cho and Whoi-Yul Kim. 2013. Long-Range Gaze Tracking System for Large Movements. *IEEE Transactions on Biomedical Engineering* 60, 12 (Dec 2013), 3432–3440. DOI: http://dx.doi.org/10.1109/TBME.2013.2266413
12. Nicholas S. Dalton, Emily Collins, and Paul Marshall. 2015. Display Blindness?: Looking Again at the Visibility of Situated Displays Using Eye-tracking. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 3889–3898. DOI: http://dx.doi.org/10.1145/2702123.2702150
13. Nigel Davies, Sarah Clinch, and Florian Alt. 2014. *Pervasive Displays: Understanding the Future of Digital Signage* (1st ed.). Morgan & Claypool Publishers.
14. Heiko Drewes and Albrecht Schmidt. 2007. *Interacting with the Computer Using Gaze Gestures*. Springer Berlin Heidelberg, Berlin, Heidelberg, 475–488. DOI: http://dx.doi.org/10.1007/978-3-540-74800-7_43
15. Marc Eaddy, Gabor Blasko, Jason Babcock, and Steven Feiner. 2004. My own private kiosk: privacy-preserving public displays. In *Eighth International Symposium on Wearable Computers*, Vol. 1. 132–135. DOI: http://dx.doi.org/10.1109/ISWC.2004.32
16. Augusto Esteves, Eduardo Velloso, Andreas Bulling, and Hans Gellersen. 2015. Orbits: Gaze Interaction for Smart Watches Using Smooth Pursuit Eye Movements. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 457–466. DOI: http://dx.doi.org/10.1145/2807442.2807499
17. Craig Hennessey and Jacob Fiset. 2012. Long Range Eye Tracking: Bringing Eye Tracking into the Living Room. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '12)*. ACM, New York, NY, USA, 249–252. DOI: http://dx.doi.org/10.1145/2168556.2168608
18. Axel Hoesl, Julie Wagner, and Andreas Butz. 2015. Delegation Impossible?: Towards Novel Interfaces for Camera Motion. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '15)*. ACM, New York, NY, USA, 1729–1734. DOI: http://dx.doi.org/10.1145/2702613.2732904
19. Shahram Jalaliniya and Diako Mardanbegi. 2016. EyeGrip: Detecting Targets in a Series of Uni-directional Moving Objects Using Optokinetic Nystagmus Eye Movements. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 5801–5811. DOI: http://dx.doi.org/10.1145/2858036.2858584

20. Mohamed Khamis, Florian Alt, and Andreas Bulling. 2015. A Field Study on Spontaneous Gaze-based Interaction with a Public Display Using Pursuits. In *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers (UbiComp/ISWC'15 Adjunct)*. ACM, New York, NY, USA, 863–872. DOI: http://dx.doi.org/10.1145/2800835.2804335

21. Mohamed Khamis, Florian Alt, and Andreas Bulling. 2016a. Challenges and Design Space of Gaze-enabled Public Displays. In *Ajunct Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '16)*. ACM, New York, NY, USA, 10. DOI: http://dx.doi.org/10.1145/2968219.2968342

22. Mohamed Khamis, Ozan Saltuk, Alina Hang, Katharina Stolz, Andreas Bulling, and Florian Alt. 2016b. TextPursuits: Using Text for Pursuits-Based Interaction and Calibration on Public Displays. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '16)*. ACM, New York, NY, USA, 12. DOI: http://dx.doi.org/10.1145/2971648.2971679

23. Mohamed Khamis, Ludwig Trotter, Markus Tessmann, Christina Dannhart, Andreas Bulling, and Florian Alt. 2016c. EyeVote in the Wild: Do Users Bother Correcting System Errors on Public Displays?. In *Proceedings of the 15th International Conference on Mobile and Ubiquitous Multimedia (MUM '16)*. ACM, New York, NY, USA, 57–62. DOI: http://dx.doi.org/10.1145/3012709.3012743

24. Christian Lander, Sven Gehring, Antonio Krüger, Sebastian Boring, and Andreas Bulling. 2015. GazeProjector: Accurate Gaze Estimation and Seamless Gaze Interaction Across Multiple Displays. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 395–404. DOI: http://dx.doi.org/10.1145/2807442.2807479

25. Michael Mrochen, Mostafa Salah Eldine, Maik Kaemmerer, Theo Seiler, and Werner Htz. 2001. Improvement in photorefractive corneal laser surgery results using an active eye-tracking system. *Journal of Cataract & Refractive Surgery* 27, 7 (2001), 1000 – 1006. DOI: http://dx.doi.org/10.1016/S0886-3350(00)00884-1

26. Jörg Müller, Florian Alt, Daniel Michelis, and Albrecht Schmidt. 2010. Requirements and Design Space for Interactive Public Displays. In *Proceedings of the 18th ACM International Conference on Multimedia (MM '10)*. ACM, New York, NY, USA, 1285–1294. DOI: http://dx.doi.org/10.1145/1873951.1874203

27. Jörg Müller, Robert Walter, Gilles Bailly, Michael Nischt, and Florian Alt. 2012. Looking Glass: A Field Study on Noticing Interactivity of a Shop Window. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 297–306. DOI: http://dx.doi.org/10.1145/2207676.2207718

28. Borna Noureddin, Peter D. Lawrence, and C.F. Man. 2005. A non-contact device for tracking gaze in a human computer interface. *Computer Vision and Image Understanding* 98, 1 (2005), 52 – 82. DOI: http://dx.doi.org/10.1016/j.cviu.2004.07.005 Special Issue on Eye Detection and Tracking.

29. Takehiko Ohno and Naoki Mukawa. 2004. A Free-head, Simple Calibration, Gaze Tracking System That Enables Gaze-based Interaction. In *Proceedings of the 2004 Symposium on Eye Tracking Research & Applications (ETRA '04)*. ACM, New York, NY, USA, 115–122. DOI: http://dx.doi.org/10.1145/968363.968387

30. Constantin Schmidt, Jörg Müller, and Gilles Bailly. 2013. Screenfinity: Extending the Perception Area of Content on Very Large Public Displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 1719–1728. DOI: http://dx.doi.org/10.1145/2470654.2466227

31. Julian Seifert, Sebastian Boring, Christian Winkler, Florian Schaub, Fabian Schwab, Steffen Herrdum, Fabian Maier, Daniel Mayer, and Enrico Rukzio. 2014. Hover Pad: Interacting with Autonomous and Self-actuated Displays in Space. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. ACM, New York, NY, USA, 139–147. DOI: http://dx.doi.org/10.1145/2642918.2647385

32. Yusuke Sugano, Xucong Zhang, and Andreas Bulling. 2016. AggreGaze: Collective Estimation of Audience Attention on Public Displays. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '16)*. ACM, New York, NY, USA. DOI: http://dx.doi.org/10.1145/2984511.2984536

33. Eduardo Velloso, Marcus Carter, Joshua Newn, Augusto Esteves, Christopher Clarke, and Hans Gellersen. 2017. Motion Correlation: Selecting Objects by Matching Their Movement. *ACM Trans. Comput.-Hum. Interact.* 24, 3, Article 22 (April 2017), 35 pages. DOI: http://dx.doi.org/10.1145/3064937

34. Mélodie Vidal, Andreas Bulling, and Hans Gellersen. 2013. Pursuits: Spontaneous Interaction with Displays Based on Smooth Pursuit Eye Movement and Moving Targets. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '13)*. ACM, New York, NY, USA, 439–448. DOI: http://dx.doi.org/10.1145/2493432.2493477

35. Robert Walter, Andreas Bulling, David Lindlbauer, Martin Schuessler, and Jörg Müller. 2015. Analyzing Visual Attention During Whole Body Interaction with Public Displays. In *Proceedings of the 2015 ACM*

International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '15). ACM, New York, NY, USA, 1263–1267. DOI: http://dx.doi.org/10.1145/2750858.2804255

36. Miaosen Wang, Sebastian Boring, and Saul Greenberg. 2012. Proxemic Peddler: A Public Advertising Display That Captures and Preserves the Attention of a Passerby. In *Proceedings of the 2012 International Symposium on Pervasive Displays (PerDis '12)*. ACM, New York, NY, USA, Article 3, 6 pages. DOI: http://dx.doi.org/10.1145/2307798.2307801

37. Julie R. Williamson, Daniel Sundén, and Jay Bradley. 2015a. GlobalFestival: Evaluating Real World Interaction on a Spherical Display. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '15)*. ACM, New York, NY, USA, 1251–1261. DOI: http://dx.doi.org/10.1145/2750858.2807518

38. Julie R. Williamson, John Williamson, Daniel Sundén, and Jay Bradley. 2015b. Multi-Player Gaming on Spherical Displays. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '15)*. ACM, New York, NY, USA, 355–358. DOI: http://dx.doi.org/10.1145/2702613.2725447

39. Lawrence H Yu and Moshe Eizenman. 2004. A new methodology for determining point-of-gaze in head-mounted eye tracking systems. *IEEE Transactions on Biomedical Engineering* 51, 10 (2004), 1765–1773.

40. Yanxia Zhang, Andreas Bulling, and Hans Gellersen. 2014. Pupil-canthi-ratio: A Calibration-free Method for Tracking Horizontal Gaze Direction. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces (AVI '14)*. ACM, New York, NY, USA, 129–132. DOI: http://dx.doi.org/10.1145/2598153.2598186

41. Yanxia Zhang, MingKi Chong, Jörg Müller, Andreas Bulling, and Hans Gellersen. 2015. Eye tracking for public displays in the wild. *Personal and Ubiquitous Computing* 19, 5-6 (2015), 967–981.