

EyeMirror: Mobile Calibration-Free Gaze Approximation using Corneal Imaging

Christian Lander¹ Sven Gehring¹ Markus Löchtefeld² Andreas Bulling³ Antonio Krüger¹

¹German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany

²Department of Architecture, Design and Media Technology, Aalborg University, Aalborg, Denmark

³Max Planck Institute for Informatics, Saarbrücken, Germany

¹firstname.lastname@dfki.de, ²mloc@create.aau.dk, ³bulling@mpi-inf.mpg.de

ABSTRACT

Gaze is a powerful measure of people's attracted attention and reveals where we are looking within our current FOV. Hence gaze-based interfaces are gaining in importance. However, gaze estimation usually requires extensive hardware and depends on a calibration that has to be renewed regularly. We present EyeMirror, a mobile device for calibration-free gaze approximation on surfaces (e.g., displays). It consists of a head-mounted camera, connected to a wearable mini-computer, capturing the environment reflected on the human cornea. The corneal images are analyzed using natural feature tracking for gaze estimation on surfaces. In two lab studies we compared variations of EyeMirror against established methods for gaze estimation in a display scenario, and investigated the effect of display content (i.e. number of features). EyeMirror achieved 4.03° gaze estimation error, while we found no significant effect of display content.

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

Author Keywords

Corneal Image; mobile device; gaze approximation; feature tracking; pervasive

INTRODUCTION

The visual system allows us to perceive a highly detailed reflection of our environment and plays an important role when interacting with the real world. The point of gaze reflects our overt visual attention and naturally indicates what we are interested in [44]. Gaze has therefore long been used as a modality for human-computer interaction since the early 90s [9], among other things, for eye typing [15], target selection [36] and cross-device object transfer [42].

Advances in mobile eye tracking point toward pervasive eye gaze interfaces for daily usage [3]. These mobile eye-trackers

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MUM 2017, November 26–29, 2017, Stuttgart, Germany

© 2017 ACM. ISBN 978-1-4503-5378-6/17/11...\$15.00

DOI: <https://doi.org/10.1145/3152832.3152839>

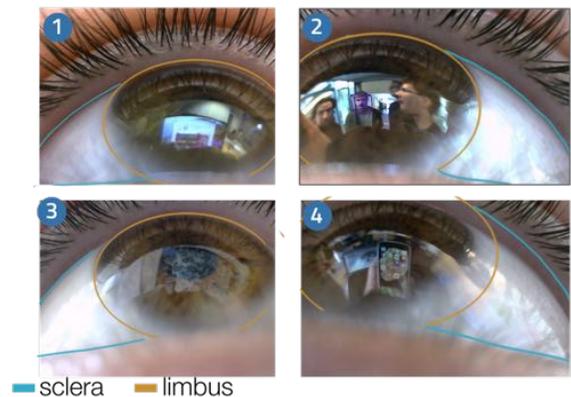


Figure 1. Corneal Images, those show (1) a computer monitor, (2) faces, (3) a poster and (4) an iPhone6 display.

are usually equipped with two cameras. An eye camera captures a close-up view of a person's eye to track its pupil and movements using active infrared illumination, while a scene camera records a person's field of view. Gaze estimation is the process of mapping pupil positions from eye into world camera coordinates. An important aspect of head-mounted eye tracking systems is the calibration, used to create a function that maps eye to gaze positions. This step creates serious problems and makes the usage of these devices cumbersome if not impossible in pervasive scenarios [16]. The usual calibration procedure requires the user to fixate a sequence of visual stimuli. Although current commercial devices (e.g., Tobii Pro Glasses [39]) are built on model-based (geometric) gaze estimation methods, they require at least a one-point calibration, and more to increase its accuracy [45]. The problem of calibration drift [12], caused by different factors (e.g., eye physiology, environmental factors, or the device's position), worsens gaze estimation over time. Hence a periodic re-calibration is mandatory. Although various research exists to enhance the calibration procedure, it remains the major problem of interactive eye tracking systems [4, 30]. The latest commercial head-mounted eye trackers remain regrettably expensive and rely on a direct connection to a powerful computer for real-time data processing. Thus they are usually not usable in real-world scenarios or in-the-wild experiments.

In this paper we present EyeMirror, a mobile wearable system for corneal imaging. It allows for calibration-free,

moderate gaze approximation on surfaces (e.g., displays) in the environment, while tolerating changes in distance and orientation to them. Our eyes literally serve as a mirror of our everyday doings and whereabouts. The parts that we can see externally are the white sclera, the iris, and the black pupil – the latter two of which are covered by the cornea. The corneal surface is covered by the tear fluid, which turns it into a highly reflective surface.

We placed a lightweight camera in front of the eye to capture close-up images of the human eye. They contain a distorted partial reflection of the user’s current field of view (see Figure 1). These corneal images are compared with the content of surfaces in the environment (e.g., displays, posters or books) using natural feature tracking. We developed two approaches to approximate a person’s gaze. The first uses the k-means cluster of all extracted key feature pairs as the gaze point on the surface. The second roughly extracts the pupil center and maps it to the display using a homography matrix, based on the extracted key feature pairs [11, 41]. Both of our methods require knowledge about the surface’s content. In the case of displays, the screen content can be streamed; books’ and posters’ content has to be available. The underlying concept is solely based on natural feature tracking and a single head-mounted camera, which makes a calibration procedure unnecessary. As the device is connected to a wearable computer, users are able to freely walk around, which is empowering for pervasive scenarios.

We conducted two consecutive laboratory experiments to evaluate our approach. In the first experiment with 10 participants we compared four versions of EyeMirror – the two already described approaches, and each approach using distortion-corrected corneal images – against a state-of-the-art Pupil Labs eye tracker [10], and using head orientation as gaze direction based on a Microsoft Kinect v2 sensor. In the first experiment we were primarily interested in gaze estimation accuracy in a single display scenario. The task was to look at different on-screen targets from multiple distances and orientations in front of a projected display. EyeMirror achieves moderate gaze estimation accuracy of about 5° in each version. The second experiment explored the effect of the number of features on gaze estimation accuracy. This is of great importance, as the system is based on natural feature tracking. In a single-desktop setting, we repeated the same task as in the first experiment, but changed between six different content types. We found no significant change in gaze estimation accuracy among five of them. One content type did not work, since it contained too few features. Hence, our work provides the following contributions:

- **Fully implemented wearable corneal reflection system** enabling **calibration-free** gaze mapping on ambient surfaces in real time.
- **Investigation of two algorithms** based on established concepts (natural feature tracking) executable on-board for use in the wild.
- **Guidelines for the quality of surfaces** (i.e., content); the device can be used according to the results of the evaluations.

We envision corneal imaging using monocular cameras readily integrated into smart eyewear (e.g., Google Glass). This qualifies EyeMirror to be suitable for real-world applications, experiments in the wild, and gaze estimation on surfaces (e.g., displays, books, advertisements), containing sufficient features, and other objects (e.g. human faces). The paper is structured as follows: After an overview of existing research related to our work, we will present EyeMirror’s approach. In the second half of the paper, we present the two experiments to assess EyeMirror’s gaze estimation accuracy and the effect of the display content in a single-display scenario. We conclude the paper by discussing our results, pointing out the current limitations and giving an outlook for future work.

RELATED WORK

Our work builds upon methods for (1) eye tracking and gaze estimation, as well as (2) corneal reflection analysis.

Eye Tracking and Gaze Estimation

For a detailed review of eye gaze tracking methods, we refer the reader to Young and Sheena [47]. We restrict the following to video-based eye tracking, as it is the most-used approach nowadays. This can be divided into remote and head-mounted eye tracking systems. Remote eye trackers use one or more cameras to track a user’s eyes and are fixed to the surface (e.g., a display) one intends to estimate gaze on. Recently, gaze-based interfaces in stationary settings became a valid option, as these devices became easily available at a low price point. The latest remote eye trackers for usage with desktop computers are available for under \$200 (Tobii 4C [69]). However, making gaze-based interfaces ubiquitous remains an open challenge that cannot be resolved using remote technologies.

Head-mounted eye trackers are equipped with at least two cameras, one capturing the eye, one the field of view. Many approaches exist to track a person’s eye; most systems use the Pupil Center Corneal Reflection (PCCR) technique [8]. Gaze estimation is always about finding a suitable mapping from pupil to gaze positions and requires a user-dependent calibration [5]. This mapping has to be renewed on a regular basis to keep a constant highly accurate gaze estimation over time [12]. Approaches utilizing visual markers [2] in the environment or detecting surfaces (e.g., displays) directly in the scene camera’s frame [43] allow for gaze estimation on specific interaction areas. However, these systems either need an instrumentation of the environment, or rely on a multi-camera system to track objects on the environment. EyeMirror can directly estimate gaze on surfaces (e.g., displays) visible in the corneal images. People are thereby free to move around, as it is a mobile wearable device. At the same time, our method does not require any prior calibration procedure and runs in real time.

Corneal Reflection Analysis

The cornea of the human eye has mirror-like characteristics. Different research areas took advantage of the specular reflections on the eye. Backes et al. [1] revealed that display reflections could be used to access sensitive data (e.g., passwords) using a telescope. Nishino and Nayar [23, 24] pioneered in corneal imaging. They developed the corneal catadioptric

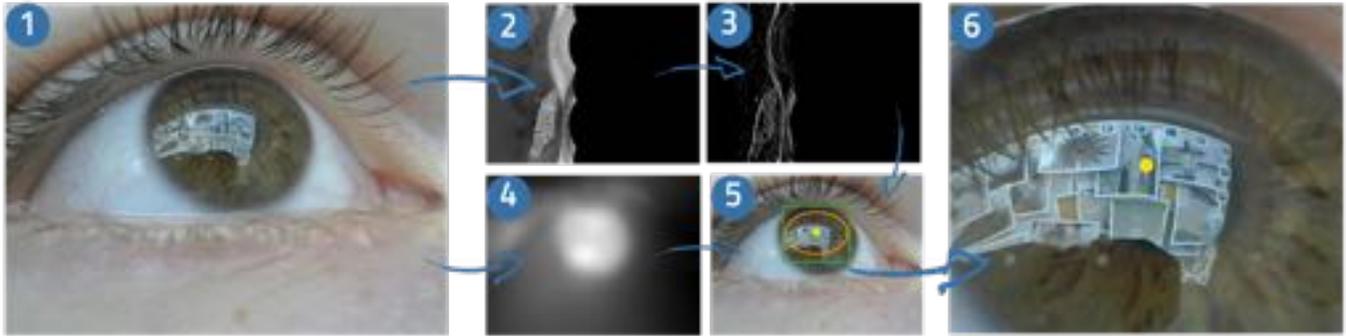


Figure 2. Limbus detection is done via ellipse fitting (5, marked by the green rectangle and the orange ellipse) after a polar transformation (2) and radial derivation (3). The eye center is extracted using image gradients (4). The final region of interest (6) contains the limbus and the eye center location (yellow circle).

imaging system using a 3-dimensional geometric model of the cornea. The derived system can be used for several applications like facial reconstruction and relighting [22], face recognition [21] and the calibration of display-camera setups [26]. EyeMirror follows an approach aiming for a simpler yet fast method that allows for real-time application in pervasive scenarios.

Besides the applications in the field of computer vision [27], there are also several approaches that utilize the reflection of the eye to do gaze estimation [28]. Schnieders et al. [34] use a remote camera to detect a display in the reflected eye image using its special properties (e.g., curved edges in the reflection). Nakazawa et al. [19] used infrared light to track the eye’s iris and pupil as well as to create patterns in the environment visible in the reflection image. Nitschke et al. [25] further improved the method by omitting the active illumination in the environment. However, all these approaches limited the user’s mobility by using a remote camera. Moreover, additional components, like active illumination or a 3D geometric eye model, were needed. Nakazawa et al. [20] used a head-mounted device to capture corneal reflections and achieved reasonable results for gaze estimation on a 23-inch display. In their study they did not investigate changes in orientation and distance to the display.

Takemura et al. [37] developed a mobile prototype to estimate the object a user is focusing on. They utilized natural feature tracking to detect objects visible in the corneal images. In [38] they extended their prototype with a scene camera. Thus, a calibration of the system is again needed. Both approaches are based on 3D eye pose estimation and geometric modeling of the eye, making them computationally expensive. Their implementations achieve only 7.3 and 1 fps respectively. In addition, they apply a color correction and unwarped the corneal images. They evaluated different eye models in a very constrained setup with fixed head position at one distance to the screen and achieved worse results for gaze estimation than EyeMirror (9.5° in [38]). Hence, both systems cannot be used for gaze based interaction in pervasive settings.

With EyeMirror we built a wearable mobile corneal imaging system using a single off-the-shelf webcam without any additional components, such as active infrared light [35] or optical parts [37, 38]. Iris contour detection, as well as tracking the eye center point, is based only on processing the close-up eye images using computer vision methods, and works without any highly complex computations based on a 3D model of the eye [6]. In our approach we solely investigate the method of natural feature tracking for calibration-free gaze approximation in real time. We use lightweight algorithms of a low computational complexity, executable on a single-board computer. Thus EyeMirror is a system made for the exploration of gaze estimation in the wild. We evaluated our approach against established methods for gaze estimation, as well as the effect of the number of features to align EyeMirror.

THE EYEMIRROR SYSTEM

The EyeMirror system is designed to detect known aspects visible in a person’s field of view to estimate gaze on ambient surfaces (e.g., it is usable to measure attention on displays). The only hardware required for a working system is a single off-the-shelf RGB webcam. It is positioned underneath the eye to capture a close-up video, revealing objects in the near environment, reflected on the eye’s pupil and iris. The camera is slightly rotatable and movable to center the eye in the image, as needed for an optimal reflection image. The camera frames are analyzed with image processing and computer vision methods. Figure 2 illustrates the processing pipeline for extracting the limbus (iris contour) as well as the eye center, used to approximate the pupil center. The output is a cropped version of the raw input image, containing the region within the iris contour, used for further operations. In the following we explain the limbus extraction, the eye center localization, and gaze approximation on displays.

Limbus Detection

Figure 2(1-3) visualizes the pipeline used for limbus extraction. The algorithm continuously receives close-up images, shown in Figure 2(1). These raw video frames contain a lot of information not needed for later processing. The sclera can take up to one-third of the image, depending on the eye pose and camera position, and does not reveal any relevant

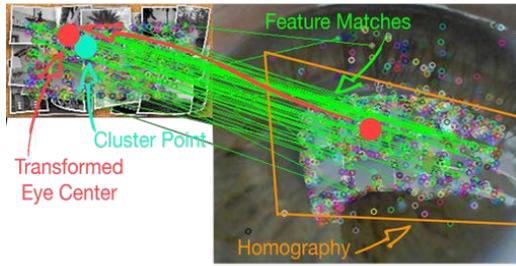


Figure 3. EyeMirror’s two approaches for gaze estimation on surfaces, here displays. One transforms the eye center point (red circle) onto the screen using a homography (orange rectangle). The second uses the cluster of the key feature pairs (green lines) as a gaze point (cyan circle).

information. The eyelashes may corrupt the result of later processing steps (e.g., feature tracking). Hence, these areas are removed from the raw image through a pre-defined region of interest. The limbus of the eye has two main image characteristics: (1) it has an elliptical shape, and (2) it can be distinguished well from the surrounding structures (sclera and eyelids). A well-known approach is to look for the radial edges and classify them as the limbus boundary using ellipse fitting. EyeMirror’s algorithm is based on the approach by Wood et al. [46]. As suggested by them, the derivative of the polar transformation of the raw image is used, shown in Figure 2(2-3). The maximum of each row is marked as a potential limbus point and fitted using a least-squares method for ellipse fitting. This approach is highly robust across different users and under varying lighting conditions, as it is not based on pre-defined thresholds for edge detection.

Eye Center Localization

To realize gaze estimation, it is necessary to have a reference point in the environment of what the user is currently looking at. One of our approaches in EyeMirror declares the reflection at the eye center as the gaze reference point. This point often correlates with the actual pupil center. For eye center localization, the method of image gradients, proposed by Timm et al. [40], is used. Figure 2(3) shows the input image processed with the function they developed, which extracts the location where the most gradient vectors intersect. Like limbus detection, this method is robust under changing lighting conditions and wide eye movements.

Gaze Estimation on Displays

With EyeMirror, we propose a novel approach for calibration-free gaze estimation on surfaces without using a geometric 3D eye model [34], any additional hardware like active IR illumination [19, 37], or a user calibration [38]. The corneal images contain the real information about what a person is currently looking at. In the case of interacting with a surface, e.g. a display, its content is partially reflected on the eye’s iris and pupil, as shown in Figure 2(1). In EyeMirror, we developed two techniques for realizing gaze estimation on surfaces, such as public displays using the corneal images:

Our first approach is similar to GazeProjector [11] and Gaze+RST [41]. Instead of utilizing the world camera of

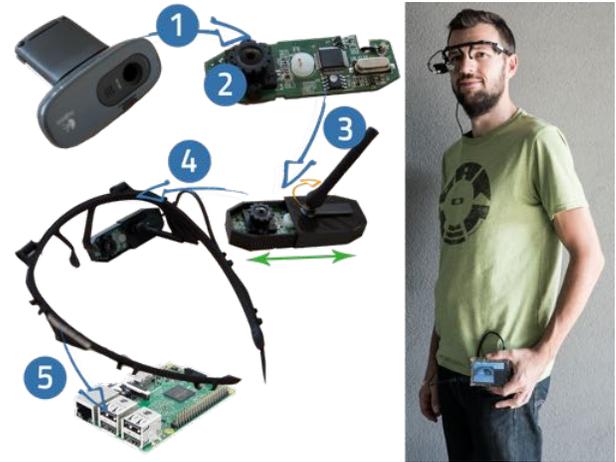


Figure 4. Building steps: (1) Extraction of camera board; (2) removing glue around the lens to adjust focus; (3) camera is built into a custom enclosure – rotatable and movable – and (4) mounted on a glasses frame, connected to a (5) RaspberryPi. On the right, the final wearable and fully functional prototype is shown.

a mobile eye tracker, we use the area within the limbus, containing the reflection of the real world. These are matched to the display’s content, recorded via screenshots. In both image streams, key features are extracted using FREAK [29] and matched using FLANN [18]. The basic idea is to use the screenshots of the displays as template images, which are searched for in the pre-processed corneal images. To estimate the spatial relationship, the system uses the found key feature pairs to compute a homography matrix, describing a transformation of points from the display’s image plane to the corneal image plane. The inverse of the homography matrix is then used to transform the eye center point to the corresponding location on the screen. The orange rectangle together with the red points in Figure 3 illustrate the procedure.

The second approach works in a similar manner in terms of calculating the relation between the eye reflection and the displays. In contrast to the previous version, we are computing a k-means cluster for the found key feature pairs (with $k = 1$) to extract the gaze point instead of detecting the pupil center. The result is immediately chosen as the current gaze point, as it is usually located near to the pupil’s center. The result is shown by the cyan point in Figure 3. Note that most of the features are found and matched within the area of the pupil. The reasons are that (a) the distortion of the reflection in that part is relatively small for most eye poses, and (b) the corneal reflection is less noisy at the pupil than on the iris. This is caused by the fact that the reflection on the iris is mixed with its color, making it blurrier. Also, other distraction factors (e.g., contact lenses) can corrupt the reflection there.

Both described methods are applicable to any surface providing feature-rich content (e.g., posters or books).

Implementation

The EyeMirror system consists of three components: (1) the head-mounted prototype built from a 3D-printed glasses frame

[10] and a 3D-printed camera mount to place the webcam underneath a person’s eye; (2) a single-board computer, running the EyeMirror software component and (3) the surfaces in the environment. The software is designed to be easily extendable through plugins (e.g., further image processing algorithms). In the following, we will outline the implementation to use EyeMirror with displays. Each mobile device and the displays are connected over WiFi. To keep the prototype device as lightweight as possible, we are using a Logitech c270 camera (Figure 4(1)), capturing frames with a maximal resolution of 1280 x 960 px at 30 fps. The camera covers a 60° field of view and has a fixed focus of 4 mm. The camera has been stripped of its original housing. To enable the camera to capture with macro resolution, the glue around the lens had to be removed to adjust the focus manually (Figure 4(2)). The camera board is mounted on a 3D-printed glasses frame using a custom enclosure (Figure 4(3-4)). In this way, the camera is movable and rotatable; thus we do not need any further optical parts like prisms [37]. As the joint between the frame and a custom camera hold is lockable with a screw, the camera will not move without further effort. Consequently, the relationship between eye, camera and scene is rather fixed and re-adjustments are not done more often than for other head-mounted eye tracking devices. Figure 4 illustrates the building steps and the final prototype.

All software components are developed in Python. For image processing methods, the OpenCV 3.0¹ library is used. It provides methods for FAST and FLANN to extract and match key features. For simple image operations (e.g., rotation and maxima search) we are using numpy², as it provides a fast way to process arrays. Eye images are captured with a resolution of 1280 x 960 px, resulting in an image with varying resolution after limbus extraction. Depending on the eye pose (i.e., the location of the limbus) the resolution ranges from 410 x 400 pixels to 400 x 360 pixels. Hence, these images are not further down-scaled, to preserve a high number of features. All image processing steps, i.e., limbus detection, eye center localization and gaze estimation, are implemented as separate sub-processes, integrated as plugins. The software component is running on a Raspberry 3 single-board computer, based on Raspbian OS, with up to 25fps. It is mountable to a belt, enabling the whole system to be portable, while it is powered by an power bank. Camera frames are captured by using pyuvc³, a python wrapper for libuvc, a cross-platform library for USB video devices, built atop libusb. For a super-fast jpeg decompression, the wrapper uses libjpegturbo. This makes this library more robust and faster than the built-in OpenCV camera plugin. Using EyeMirror with displays, their content has to be streamed to the software component. We developed a python script that broadcasts screenshots with a resolution of 240 x 180 px to the RaspberryPi 3. In the case of other ambient surfaces (e.g., books), the images have to be known beforehand.

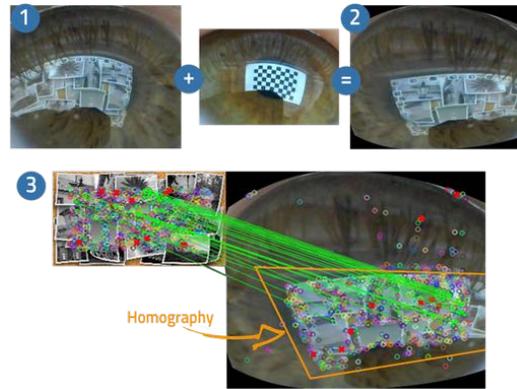


Figure 5. Calibration with chessboard pattern (1) to undistort the corneal image (2). Computation of a homography may be more exact (3).

EXPERIMENT – GAZE ESTIMATION ON A DISPLAY

We conducted a controlled laboratory study to evaluate EyeMirror’s accuracy in estimating gaze on a projected display. We compared our two approaches with corresponding versions using distortion-corrected corneal images, a head-mounted Pupil Labs eye tracker that uses marker tracking to estimate its relative position to the display [11], and an approach with a Kinect v2 sensor that uses only the head position and orientation.

Independent Variables

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

Mode: We used six different modes for gaze estimation on the projected display: *EyeMirror-Pupil (EM-P)*, transforming the pupil center using a homography, and *EyeMirror-Cluster (EM-C)* taking the cluster of the key feature pairs as gaze point, both described above; *EyeMirror-Pupil-Undistorted (EM-P-U)* and *EyeMirror-Cluster-Undistorted (EM-C-U)*, both using a corrected corneal image to investigate the effect of the spherical distortion of the corneal reflection (as in [31]); *Marker Tracking (MT)*, using a set of visual markers shown on the screen to track the orientation between the display and the eye tracker provided by the Pupil framework⁴; and a simple *Head Orientation (HO)* approach, tracking the participant’s head with a Kinect v2 sensor, placed underneath the projected display. For *MT* we calibrated the eye tracker for each participant separately from the centered location in front of the display. For all *EM* modes we adjusted the camera (position and focus). For *EM-P-U* and *EM-C-U* we sampled data while the people were looking at a chessboard pattern for 5 seconds to dewarp the images. The pattern filled out the whole projected display. To compute the distortion map, we used OpenCV’s camera calibration tool, using 20 samples for each calibration. Figure 5 highlights the calibration procedure and feature matching as well as homography computation of the undistorted corneal image.

¹<http://opencv.org>

²<http://numpy.org>

³<https://github.com/pupil-labs/pyuvc>

⁴<http://www.pupil-labs.com/blog/2013/12/036-release.html>

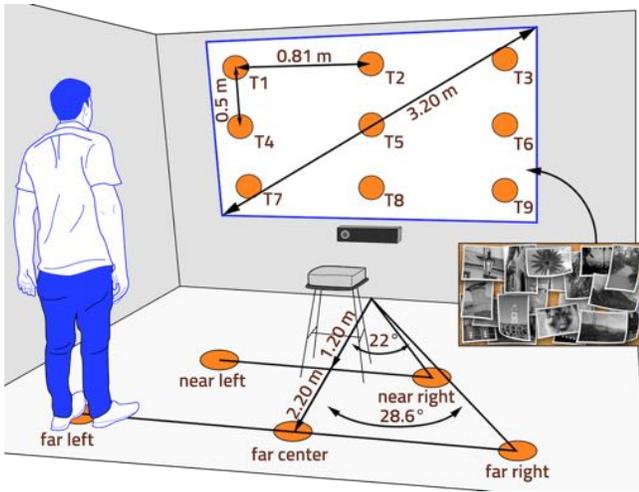


Figure 6. Study setup: A large projected screen and the Kinect sensor v2. In addition all locations (*near-left*, *near-right*, *far-left*, *far-center* and *far-right*) and targets (T1-T9) are shown. .

Locations: We chose five different locations (2 near, 3 far) to investigate the effect of varying positions and orientations in front of the display. In doing so, we obtained images containing many different reflections of the display (i.e., different size and distortion), while simulating a more realistic setting. The eye tracker was calibrated only for the central location, as it is not likely that users re-calibrate for every position in a dynamic setting. We did not give any visual feedback to the participants, to prevent false positives. In addition, this allowed us to keep the length of the experiment reasonable, as all the data was sampled in two runs. We had to record the HO data separately, as the head tracking was error-prone while the mobile devices (eye tracker and prototype) were being worn. We computed the gaze estimation accuracy as well as the correction post-hoc for every mode.

Task & Procedure

We implemented a simple gaze-pointing task in which participants had to focus on targets, shown at nine different positions (T1-T9). These were represented as red circles (40 pixels, or approx., 58 mm) on the projection with equal distances between them (see Figure 6). A pilot study showed that artificial lighting conditions could affect the quality of the reflected eye image. We therefore created a realistic scenario by illuminating the room with natural light. Every participant was first asked to calibrate the head-mounted eye tracker while standing at the center location in front of the projected screen. Each *Mode*, except *HO*, was recorded in parallel, as it is possible to wear both mobile devices at once. The participants were instructed to look at each target as quickly and accurately as possible, while being free to move their heads. The targets were shown for six seconds each. At the end, participants were asked for demographic information including the color of their eyes.

The task was done while standing in front of the projected display at five locations (see Figure 6). Looking straight

ahead, they looked approximately at the vertical center of the projection. We collected gaze data from the eye tracker for *MT* as well as the location of the on-screen targets and their time-stamps needed for post-hoc analysis. Furthermore, we recorded raw video material for all *EM* modes including calibration frames, as described above. Data was sampled at 30 Hz for all six modes (i.e., 180 samples per on-screen target = 6sec x 30 Hz), leading to 1620 samples for each Mode and Location combination. In total we recorded 30 (Hz) x 6 (sec) x 9 (targets) x 6 (Modes) x 5 (Locations) x 10 (participants) = 486000 samples. We dropped the first 2 out of 6 seconds per target, leading to 30 x 2 x 9 x 6 x 5 x 10 = 162000 (30% of all samples), resulting in 324000 samples in total (i.e. 54000 for each Mode). We discarded all data points of the first two seconds for each target, as this was the maximal required timespan to find the current target.

Experimental Design

We used a within-subject design for our experiment with two independent variables, *Mode* (*EM-P*, *EM-C*, *EM-P-U*, *EM-C-U*, *MT* and *HO*) and *Location* (*near-left*, *near-right*, *far-center*, *far-left*, *far-right*). We counterbalanced the order of *Location* between all participants using a Latin square. All modes except *HO* were recorded in parallel. Thus each participant performed the task twice. For each location, the nine targets were displayed in a random sequential order, different between participants, but identical within them (i.e., for both runs).

Apparatus

Figure 6 illustrates the experimental setup: we used a large front-projected display with a size of 2.80x1.56 meters (diagonal: 3.20 meters) using a short-throw projector. The five locations (L1-L5) were distributed as follows: the *near* locations at a distance of 1.2 m, and the *far* locations at a distance of 2.20 m. The *near-left* and *near-right* locations were 1.35 meters away, and the *far-left* and *far-right* locations were 2.50 meters away from the display's center with an angular offset of $\pm 26.5^\circ$ and $\pm 28.6^\circ$, respectively. The *far-center* location had an angular offset of 0° . Standing at the *far* locations, the display covers 64.8° , while at the *near* locations the display covers 98.8° . Choosing the locations this way forces participants to move their heads, as the region covered by the display exceeds the ocular motor range of $\pm 55^\circ$ [7]. To record the *EM* and *MT* in parallel, we mounted EyeMirror's camera underneath the eye camera of the monocular Pupil Labs eye tracker, capturing the right eye. The Pupil Labs system was running on a Thinkpad X201, transmitting the data via WiFi to a MacBook Pro driving the display and capturing the EyeMirror camera frames. For feature tracking we used the background image (containing 9262 features, shown in Figure 6) without markers as a template to prevent any effect on *EM*'s performance.

Participants

Ten participants (3 female) between 23 and 37 years old ($M = 27.5$ years, $SD = 4.55$ years) and having three different iris colors (5 brown, 4 blue, 1 green) were recruited from a local university campus. All participants had corrected or normal vision; none reported any visual impairments (e.g., color blindness).

		<i>MT</i>			<i>EM-C</i>			<i>EM-P</i>			<i>EM-C-U</i>			<i>EM-P-U</i>			<i>HO</i>		
		<i>x</i>	<i>y</i>	<i>2D</i>	<i>x</i>	<i>y</i>	<i>2D</i>	<i>x</i>	<i>y</i>	<i>2D</i>	<i>x</i>	<i>y</i>	<i>2D</i>	<i>x</i>	<i>y</i>	<i>2D</i>	<i>x</i>	<i>y</i>	<i>2D</i>
<i>near</i>	<i>M</i>	2.27°	1.39°	2.92°	3.89°	2.35°	4.90°	4.45°	3.50°	6.17°	4.23°	2.50°	5.28°	4.36°	3.12°	5.91°	3.93°	2.71°	5.32°
<i>left</i>	<i>SD</i>	1.46°	0.94°	1.27°	3.16°	1.75°	3.12°	3.35°	2.15°	3.14°	3.43°	1.81°	3.38°	3.79°	2.13°	3.57°	2.69°	2.95°	3.23°
<i>near</i>	<i>M</i>	2.44°	1.41°	3.12°	3.97°	2.38°	4.96°	4.08°	3.33°	5.79°	4.23°	2.57°	5.31°	4.53°	3.21°	6.08°	3.92°	3.01°	5.55°
<i>right</i>	<i>SD</i>	1.74°	1.02°	1.51°	2.92°	1.78°	2.93°	3.34°	2.22°	3.23°	3.19°	1.89°	3.17°	3.68°	2.21°	3.51°	2.58°	3.14°	3.20°
<i>far</i>	<i>M</i>	1.67°	0.96°	2.10°	2.80°	1.62°	3.48°	3.61°	2.27°	4.60°	2.92°	1.66°	3.61°	3.01°	2.29°	4.16°	2.43°	2.47°	3.97°
<i>left</i>	<i>SD</i>	1.07°	0.68°	0.96°	1.93°	1.12°	1.83°	2.63°	1.52°	2.50°	2.11°	1.18°	2.03°	2.49°	1.54°	2.35°	1.94°	4.59°	4.60°
<i>far</i>	<i>M</i>	1.50°	0.97°	1.99°	2.70°	1.71°	3.44°	3.27°	2.37°	4.39°	2.84°	1.80°	3.62°	2.79°	2.28°	3.99°	2.71°	2.37°	4.03°
<i>center</i>	<i>SD</i>	0.98°	0.68°	0.80°	1.96°	1.11°	1.86°	2.40°	1.59°	2.31°	2.12°	1.18°	2.03°	2.37°	1.53°	2.25°	1.70°	3.86°	3.82°
<i>far</i>	<i>M</i>	1.70°	.89°	2.10°	2.79°	1.55°	3.41°	3.24°	2.32°	4.33°	2.84°	1.61°	3.50°	3.08°	2.29°	4.23°	2.65°	2.74°	4.35°
<i>right</i>	<i>SD</i>	1.20°	1.01°	1.32°	1.76°	1.05°	1.66°	2.37°	1.53°	2.26°	1.90°	1.09°	1.80°	2.52°	1.57°	2.40°	1.74°	4.92°	4.78°

Table 1. Means and standard deviations of the overall, horizontal and vertical gaze estimation for all modes and locations.

Results

To evaluate the EyeMirror concept, we calculated the average gaze estimation error in degrees of visual angle. This value states the difference between the position of the estimated gaze point and the actual on-screen target of the six Modes (*EM-P*, *EM-C*, *EM-P-U*, *EM-C-U*, *MT* and *HO*) and the five Locations (*near-left*, *near-right*, *far-center*, *far-left*, *far-right*). We performed a 6×5 (*Mode* \times *Location*) within-subjects ANOVA on gaze estimation errors and found a main effect for *Mode* ($F_{5,25} = 34.52, p < .001$), and for *Location* ($F_{4,20} = 30.73, p < .001$), but not for an interaction between them.

In a subsequent post-hoc analysis in gaze estimation accuracy across all Modes, we found that *MT* differed significantly from all *EM* modes (all $p < .001$) as well as *HO* ($p < 0.05$). We found no significant difference in gaze estimation accuracy between *HO* and all *EM* modes. Comparing the gaze estimation accuracy concerning different targets between *EM* modes and *HO* also revealed no significant difference. All *EM* modes performed better than *HO*, as shown in Table 1. *EM-C* differed significantly from *EM-P* ($p < .01$), *EM-C-U*

and *EM-P-U* (both $p < .05$). Finally, we found a significant difference between *EM-P* and *EM-C-U* ($p < .01$).

Overall *EM-C* achieved the highest accuracy for all *EM* modes ($M = 4.03^\circ, SD = .04^\circ$), followed by *EM-C-U* ($M = 4.22^\circ, SD = .03^\circ$). *EM-P* ($M = 4.99^\circ, SD = .07^\circ$) and *EM-P-U* ($M = 4.87^\circ, SD = .13$ degree), showed the worst results overall. *HO* ($M = 4.66^\circ, SD = .36^\circ$) yields better results than *EM-P* and *EM-P-U*, but worse than *EM-C* and *EM-C-U*. Finally, *MT* achieved the best results overall, i.e., the lowest gaze estimation error ($M = 2.41^\circ, SD = .06^\circ$). All values were averaged over all locations and summarized in Table 1. Transformed to absolute rounded pixels, these values correspond to 64px ($SD = 31$ px) for *MT*, 106px ($SD = 59$ px) for *EM-C*, 133 ($SD = 72$ px) for *EM-P*, 111px ($SD = 64$ px) for *EM-C-U*, 128px ($SD = 73$ px) for *EM-P-U* and 122px ($SD = 116$ px) for *HO*.

Figure 7 depicts the average gaze estimation error for every Mode and Location. *MT-far-center* performed best ($M = 1.99^\circ, SD = 0.22^\circ$), followed by *MT-far-right* ($M = 2.08^\circ, SD = 0.35^\circ$), *MT-far-left* ($M = 2.09^\circ, SD = 0.25^\circ$), *MT-near-left* ($M = 2.95^\circ, SD = 0.47^\circ$) and *MT-near-right* ($M = 3.10^\circ, SD = 0.44^\circ$). All *EM* modes performed worse than *MT*, but within *EM-C* showed the lowest error for *EM-far-right* ($M = 3.42^\circ, SD = 0.18^\circ$), followed by *EM-far-center* ($M = 3.44^\circ, SD = 0.40^\circ$), *EM-far-left* ($M = 3.48^\circ, SD = 0.19^\circ$), *EM-near-left* ($M = 4.86^\circ, SD = 0.43^\circ$) and *EM-near-right* ($M = 4.95^\circ, SD = 0.36^\circ$). The other Modes performed slightly worse.

Post-hoc tests on *Location* revealed that the significant main effect stems from the participants' distance to the display: *near-left* differed significantly from *far-left*, *far-center* and *far-right* (all $p < .01$). *Near-right* also differed significantly from *far-left* ($p < .01$), *far-center* and *far-right* (all $p < .05$). Overall, *far-center* showed the highest gaze estimation accuracy, i.e., the lowest error ($M = 3.62^\circ, SD = 0.1^\circ$), followed by *far-left* $M = 3.64^\circ, SD = 0.11^\circ$, and *far-right* ($M = 3.75^\circ, SD = 0.17^\circ$). The *near* locations led to worse results: *near-right* had the highest error in gaze estimation accuracy ($M = 4.97^\circ, SD = 0.15^\circ$), followed by *near-left* ($M = 4.96^\circ, SD = 0.09^\circ$).

To investigate further the gaze estimation of all *EM* modes, we split the error measured in degrees of visual angle into two val-

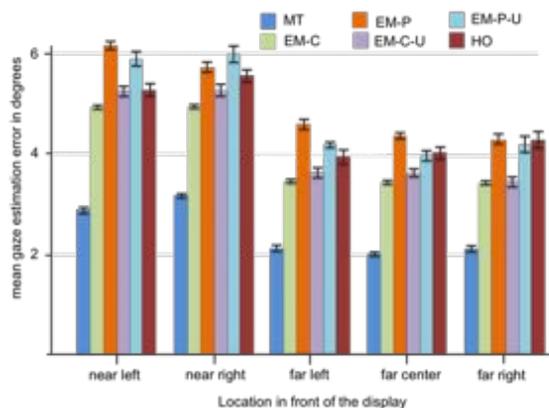


Figure 7. Mean gaze estimation error for every location and mode. Error bars indicate \pm standard error of the mean.



Figure 8. The different display contents used in the experiment: C1 with 9262, C2 with 6988, C3 with 3314, C4 with 9476, C5 with 4239 and C6 with 934 features. The right picture shows the setup for the second experiment.

ues showing horizontal (x-direction) and vertical (y-direction) errors separately. We found that for all *Locations*, the vertical gaze estimation error is lower than the horizontal across all *Modes*. Table 1 summarizes the results. On average *MT* showed the lowest difference between horizontal and vertical gaze estimation error, followed by *EM-C*.

To complete the analysis we further wanted to find out whether the screen targets and thus screen regions resulted in different gaze estimation accuracy. In doing so we analyzed the results separately for each on-screen target. We found significant differences between most of the targets across all *Modes*. For *MT* we found significant differences between all targets except (T1, T6), (T3,T4), (T3,T7), (T3,T9), (T4,T7), (T4,T9) and (T7,T9). *EM-C* showed no significant difference for gaze estimation accuracy between targets (T1,T3) and (T7,T9). For *HO* the target pairs that showed no significant difference are (T2,T4), (T2,T9), (T2,T6), (T4,T6) and (T5,T7). Every *Mode* performed best for T5, whereas *EM-C* achieved the lowest gaze estimation error overall ($M = 1.43^\circ$, $SD = 1.71^\circ$), followed by *EM-C-U* ($M = 1.54^\circ$, $SD = 1.95^\circ$), *MT* ($M = 1.67^\circ$, $SD = 0.92^\circ$), *HO* ($M = 3.85^\circ$, $SD = 5.36^\circ$), *EM-P-U* ($M = 4.08^\circ$, $SD = 2.07^\circ$) and *EM-P* ($M = 4.70^\circ$, $SD = 2.08^\circ$).

Finally, we found no significant difference of gaze estimation error in iris color for all *EM* modes. *EM-C* performed best, with 3.84° ($SD = 2.31^\circ$); brown eyes achieved the lowest gaze estimation error on average, followed by blue eyes with 4.12° ($SD = 2.56^\circ$). Green eyes achieved the highest gaze estimation error with 4.20° ($SD = 2.41^\circ$) on average for *EM-C*. We found the same order for all other *EM* modes.

EXPERIMENT II – INFLUENCE OF DISPLAY CONTENT

In a second laboratory experiment we wanted to investigate the connection between EyeMirror’s gaze estimation approach and the content of the display. Therefore, we compared the gaze estimation accuracy of *EM-C* across various display content, where the content used differed in the number of natural features.

Independent Variables

We had one independent variable in this experiment, *Content*: we used six different display content images (C1–C6), while computing the gaze estimation accuracy for the display. The different images are depicted in Figure 8 together with their

number of features. We used three different wallpapers (C1–C3) as well as realistic desktop scenes where different kinds of applications were opened (C4–C6).

Task & Procedure

We reused the same gaze-pointing task as in the first experiment to be able to compare the findings to our initial results. Looking at red circles, shown consecutively at nine different positions on a desktop monitor, was done while sitting in front of the display (shown in Figure 8). Raw video material for EyeMirror’s clustering approach (*EM-C*) was recorded at 30hz. Again, we discarded the first 2 seconds for each target.

Experimental Design

We used a within-subject design with one independent variable, *Content*. We counterbalanced the order of *Content* between all participants using a Latin square. As we had six different images, each participant did the task six times. For each content image, the targets (T1–T9) were shown in a random sequential order.

Apparatus

Figure 8 illustrates the experimental setup: we used a multi-touch enabled monitor with a size of 0.59×0.33 meters (diagonal: 0.68 meters). The participant was sitting centered in front of the display at a distance of 0.6 meters, as she would be sitting when interacting with the touch-enabled display. At this position the display covers 31.6° of the field of view. The EyeMirror camera frames were recorded by plugging it into a MacBook Pro that also was driving the display.

Participants

Six participants (2 female) between 23 and 37 years old ($M = 27.5$ years, $SD = 4.55$ years) were recruited from a local university campus. All participants had normal vision; none reported any visual impairments (e.g., color blindness).

Results

To evaluate EyeMirror’s accuracy, we calculated the average gaze estimation error in degrees of visual angle. This value states the difference between the position of the estimated gaze point and the actual on-screen target for all different *Content* images using the cluster approach of *EM-C*. While computing the gaze estimation for all different *Content* images,

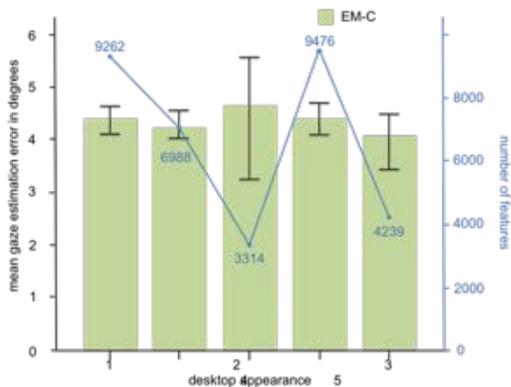


Figure 9. Mean gaze estimation error for the desktop content images C1-C5, combined with the number of features.

we discovered that our approach did not work for C6. Hence we performed a 5×1 (Content (C1–C5) \times EM-C) within-subjects ANOVA on gaze estimation errors and found no effect for *Content*. In Figure 9 the gaze estimation error across all targets and participants is plotted against the number of features.

We achieved the highest accuracy, i.e., the lowest gaze estimation error, for C2 (wallpaper with 6988 features) ($M = 4.30^\circ$, $SD = 0.04^\circ$), followed by C5 (Mac desktop: opened browser and IDE, 4239 features) ($M = 4.27^\circ$, $SD = 0.02^\circ$), C1 (wallpaper used in first experiment with 9262 features), C3 (wallpaper, 3314 features) ($M = 4.34^\circ$, $SD = 0.06^\circ$) and C4 (Mac desktop, opened pdf, 9476 features) ($M = 4.36^\circ$, $SD = 0.06^\circ$).

DISCUSSION

Our results show that – using corneal reflections (i.e., area covering iris and pupil) – EyeMirror is capable of achieving a gaze estimation accuracy at 4.03° using the clustering approach (*EM-C*) compared to 2.41° for *MT* and 4.66° for *HO*.

Gaze Estimation

We used different approaches within EyeMirror to compute the gaze point of the participants. Our results reveal that it is sufficient to extract the cluster of all key feature pairs to achieve still-reasonable results. Taking the eye center into account and transforming it onto the display by using homography matrices performs 0.77° worse (*EM-P*). While it seems counterintuitive, it can be explained when we have a closer look at the difference between both approaches. Each method uses as input image the extracted limbus area (i.e., the pupil and the iris). The reflection at the area of the pupil is less noisy and brighter at the iris. Overall, features are found on both the iris and the pupil. Most of the matches are detected at the pupil, leading to key feature pairs. The main drawback of *EM-P* and *EM-P-U* is the use of the eye center as a gaze reference point. The method used for eye center localization does not strictly correspond to the actual pupil center. Consequently, it does not always correspond to a person’s real gaze, and only transforms rough estimations.

In addition, we investigated the use of a camera calibration to correct the input images for our approaches. As the eyeball has a spherical structure, the corneal images are distorted. As expected, we found a small effect when using the homography matrix for transformation, such that *EM-P-U* is 0.12° more accurate than *EM-P*. Having a look at Figures 5(3) and 3, the approach is shown based on the raw and the corrected image. The shape of the reflected display is more rectangular if using the undistorted version. This supports the concept of computing a homography matrix, as the template (i.e., the screen’s content) is also of a rectangular shape. Otherwise, we found the opposite for the cluster approach. Using the undistorted version of the corneal image (*EM-C-U*) significantly decreases the gaze estimation accuracy by 0.19° . Correcting the images changes the distribution of the key feature pairs. As most of the matches are found at the pupil, they are arranged in an elliptical shape supporting the structure of the pupil. After a correction this is no longer the case, leading to worse results.

As we used different locations in front of the screen, we were able to investigate the effect of different orientation and distances to the screen. We calibrated once for *MT* from the far-center position for each participant to simulate a realistic scenario. Hence, we achieved the optimal results for a one-calibration scenario, as the calibration plane was orthogonal to the participants. Using different locations results in various distorted reflections on the human eye. EyeMirror is able to partly deal with these changes, since its accuracy remains constant among different orientations for the same distance. In the case of *EM-C* and *EM-C-U*, this is primarily caused by the sharp angle between participant and display for both *near* locations. There, the highest gaze estimation error was found for targets at the opposite side of the location (i.e., for *Location near-left* the targets T3, T6 and T9). With increasing distance, the angle between participant and display also increases. For *EM-P* and *EM-P-U* there is also another reason: the larger the distance to the display, the larger the reflected area. This means that the template of the surface – here the display – is detected more accurately, and thus the computation of the homography matrix is more robust. The results for *MT* are in line with existing findings [11]. Calibrating from the *far* location leads to a smaller calibration plane than for *near* locations. Hence the eye tracker extrapolates for gaze positions lying outside that region, causing worse gaze estimation.

Although we found no significant difference between *HO* and all *EM* modes, *EM* achieved a constantly better gaze estimation accuracy than *HO*. The good results of *HO* stem from the experiment design, i.e., the location layout that forced participants to move the head quite a lot, mostly for *near* locations. This fits with the very high standard deviation (see Table 1) in gaze estimation for *far* locations. This is mainly caused by the large variability in head movement propensity [17]. If we also take a look into at the gaze estimation between targets with minimal distance (e.g., far-left *EM-C* T4: 3.39° , T5: 1.14° compared to *HO* T4: 7.52° , T5: 2.08°), we can see that *HO* performs much worse than *EM*. This fact indicates that the gaze of users who do a lot of eye movements might be better approximated with *EM* than *HO*. In general, *EM* gives a direct connection to the objects in the environment,

as they can be extracted from the corneal images. *HO* only provides the position and orientation of the head in 3D space, which has to be combined with knowledge about the position of surrounding objects.

Thus, *HO* is not preferable for gaze approximation, especially in settings where gaze is computed across a variety of surfaces (e.g., different multiple displays). The first experiment investigated the gaze estimation accuracy of EyeMirror in a lab setting. In real-world scenarios, *HO* is still not usable. If the head tracking is done remotely, it requires the trackers to be attached on every surface in the environment. Doing head tracking via an IMU sensor will require a scene camera to create a connection to the environment. However, EyeMirror has a major advantage, as we get a realistic representation of the human vision, combined with a value for gaze approximation.

Surface Content

We evaluated *EM-C*'s accuracy across six display content images, all different in the kind of information shown and the numbers of provided features. Overall we found no significant difference in gaze estimation error, having content with a number of features between 3314 and 9476. Obviously, our approach is not working with surfaces providing too few features (e.g., C6 with 934 features). Interestingly, we achieved the best results for C5 (4239 features). This may be caused by the distribution of the features being rather uniform. Hence, our results show that EyeMirror enables gaze approximation in a normal desktop setting.

Applications

To summarize, EyeMirror is sufficient for approximate gaze estimation and thus usable for settings in which a spontaneous measurement of a person's gaze is sufficient and no highly accurate gaze estimation is required. The system can be used to detect if someone is looking at a screen in a multi-display setting and at which region, useful for gaze-contingent displays. EyeMirror is built as a mobile wearable system, so it could be used as a tool for exploring of the human gaze (e.g., attention measuring) in more unconstrained and mobile settings [14, 13].

For example, information like detected faces (shown in Figure 1(2)) can be used to estimate social interaction. Tracking other objects can be used to measure attention on many different things. EyeMirror might be easily integrated into AR and VR devices. A virtual reality headset provides the best conditions, since nothing other than the screen content (i.e., the VR environment) is reflected on the cornea of the eye. Approximating someone's gaze in VR can be used for foveated rendering [32]. With EyeMirror we developed a system that provides a base for further investigations in a pervasive setting. The system will be made open-source. As EyeMirror is easily extendable, new algorithms and methods for different purposes can be plugged in and explored.

Limitations

Apart from its advantages over state-of-the-art eye tracking systems and applicability to pervasive application scenarios, EyeMirror also comes with some limitations: First, our concept of gaze estimation on surfaces is based on natural feature

tracking that requires an information stream about the content (e.g., display content). However, with the rise of IoT, we believe that information like display content will be accessible. For example, information about the environment outside is available through Google Street View. Moreover, we think that objects can be detected via a trained image classifier and specific neural networks [33]. Second, EyeMirror's performance is influenced by the quality of the corneal images. The camera is placed close underneath the eye to capture most of the eye movements. To handle large eye movements, the use of more cameras should be explored. In our current prototype, we have to adjust the focus manually by rotating the lens in the thread in order to retrieve sharp corneal images. Whereas limbus and pupil center extraction are tolerant of various lighting conditions, they might have an effect on the overall approach. Obviously our proposed method does not work in dark environments, as no information will be reflected on the cornea.

Besides lighting, the eye-object distance is the another source of blurred corneal images. If the user focuses an object at a far distance, its representation on the corneal image is rather unsharp, caused by the mirror properties. To adapt the system to this new setting, the focus of the camera would have been adapted. Using cameras with adjustable parameters, like ISO sensitivity, aperture size and auto-focus, might help to counteract the above limitations.

So far we have not explored the long-term usage of EyeMirror. We can only argue about the duration of the experiment, which lasted for at least 27 minutes (6 (sec) x 9 (targets) x 6 (Modes) x 5 (Locations)). The presented results are representative for a usage time within half an hour.

Nevertheless, we believe that EyeMirror is a promising case toward a new generation of less-invasive head-mounted eye trackers and a good baseline for using corneal reflections in HCI.

CONCLUSION & FUTURE WORK

In this paper, we presented EyeMirror, a mobile system using a novel approach for calibration-free gaze approximation. We built a low-budget prototype, which in contrast to existing head-mounted eye-trackers, only requires a single camera. Capturing the environment through corneal images, it enables gaze approximation on various surfaces (e.g., displays), solely based on natural feature tracking. Therefore a representation of the object's content has to be known and provide enough visual features.

In a laboratory experiment we compared different modifications of our approach against a Pupil Labs eye tracker and using the head orientation as gaze. We found better results for EyeMirror compared to *HO* in gaze estimation accuracy, although they were not significant. The head-mounted eye tracker gives the best results, but needs to be calibrated.

As next steps, we aim to further improve the technical part of the method to realize better gaze estimation. We also plan to develop a large platform to record and store corneal reflection data. In that way a large data set can be created and used to explore different kinds of computer vision algorithms, as well as potential application cases.

REFERENCES

1. Michael Backes, Markus Dürmuth, and Dominique Unruh. 2008. Compromising Reflections-or-How to Read LCD Monitors Around the Corner. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy (SP '08)*. IEEE Computer Society, Washington, DC, USA, 158–169. DOI : <http://dx.doi.org/10.1109/SP.2008.25>
2. Jurek Breuninger, Christian Lange, and Klaus Bengler. 2011. Implementing Gaze Control for Peripheral Devices. In *Proceedings of the 1st International PETMEI Workshop (PETMEI '11)*. ACM, New York, NY, USA, 3–8. DOI : <http://dx.doi.org/10.1145/2029956.2029960>
3. Andreas Bulling and Hans Gellersen. 2010. Toward Mobile Eye-Based Human-Computer Interaction. *IEEE Pervasive Computing* 9, 4 (Oct. 2010), 8–12. DOI : <http://dx.doi.org/10.1109/MPRV.2010.86>
4. Jixu Chen and Qiang Ji. 2011. Probabilistic Gaze Estimation Without Active Personal Calibration. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*. IEEE Computer Society, Washington, DC, USA, 609–616. DOI : <http://dx.doi.org/10.1109/CVPR.2011.5995675>
5. Andrew T. Duchowski. 2007. *Eye Tracking Methodology: Theory and Practice*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
6. L El Hafi, M Ding, J Takamatsu, and T Ogasawara. 2016. Gaze Tracking Using Corneal Images Captured by a Single High-Sensitivity Camera. (2016). DOI : <http://dx.doi.org/10.1049/ibc.2016.0033>
7. Daniel Guitton and Michel Volle. 1987. Gaze control in Humans: Eye-Head Coordination During Orienting Movements to Targets within and beyond the Oculomotor Range. *Journal of Neurophysiology* 58, 3 (1987), 427–459. <http://jn.physiology.org/content/58/3/427>
8. Dan Witzner Hansen and Qiang Ji. 2010. In the Eye of the Beholder: A Survey of Models for Eyes and Gaze. *IEEE Trans. Pattern Anal. Mach. Intell.* 32, 3 (March 2010), 478–500. DOI : <http://dx.doi.org/10.1109/TPAMI.2009.30>
9. Robert J. K. Jacob. 1990. What You Look at is What You Get: Eye Movement-based Interaction Techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '90)*. ACM, New York, NY, USA, 11–18. DOI : <http://dx.doi.org/10.1145/97243.97246>
10. Moritz Kassner, William Patera, and Andreas Bulling. 2014. Pupil: An Open Source Platform for Pervasive Eye Tracking and Mobile Gaze-based Interaction. In *Adjunct Proceedings of UbiComp 2014 (UbiComp '14 Adjunct)*. ACM, New York, NY, USA, 1151–1160. DOI : <http://dx.doi.org/10.1145/2638728.2641695>
11. 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>
12. Christian Lander, Frederic Kerber, Thorsten Rauber, and Antonio Krüger. 2016. A Time-efficient Re-calibration Algorithm for Improved Long-term Accuracy of Head-worn Eye Trackers. In *Proceedings of the Ninth ACM Symposium on Eye Tracking Research & Applications (ETRA '16)*. ACM, New York, NY, USA, 213–216. DOI : <http://dx.doi.org/10.1145/2857491.2857513>
13. Christian Lander, Felix Kosmalla, Frederik Wiehr, and Sven Gehring. 2017. Using Corneal Imaging for Measuring a Human's Visual Attention. In *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers (UbiComp '17)*. ACM, New York, NY, USA, 947–952. DOI : <http://dx.doi.org/10.1145/3123024.3124563>
14. Christian Lander, Antonio Krüger, and Markus Löchtefeld. 2016. "The Story of Life is Quicker Than the Blink of an Eye": Using Corneal Imaging for Life Logging. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct (UbiComp '16)*. ACM, New York, NY, USA, 1686–1695. DOI : <http://dx.doi.org/10.1145/2968219.2968337>
15. Päivi Majaranta and Kari-Jouko Räihä. 2002. Twenty Years of Eye Typing: Systems and Design Issues. In *Proceedings of the 2002 Symposium on Eye Tracking Research & Applications (ETRA '02)*. ACM, New York, NY, USA, 15–22. DOI : <http://dx.doi.org/10.1145/507072.507076>
16. Carlos H. Morimoto and Marcio R. M. Mimica. 2005. Eye Gaze Tracking Techniques for Interactive Applications. *Comput. Vis. Image Underst.* 98, 1 (April 2005), 4–24. DOI : <http://dx.doi.org/10.1016/j.cviu.2004.07.010>
17. Omar Mubin, Tatiana Lashina, and Evert Loenen. 2009. How Not to Become a Buffoon in Front of a Shop Window: A Solution Allowing Natural Head Movement for Interaction with a Public Display. In *Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction: Part II (INTERACT '09)*. Springer-Verlag, Berlin, Heidelberg, 250–263. DOI : http://dx.doi.org/10.1007/978-3-642-03658-3_32
18. Marius Muja and David G. Lowe. 2009. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. In *In VISAPP International Conference on Computer Vision Theory and Applications*. 331–340. DOI : <http://dx.doi.org/10.1.1.160.1721>

19. Atsushi Nakazawa and Christian Nitschke. 2012. Point of Gaze Estimation Through Corneal Surface Reflection in an Active Illumination Environment. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part II (ECCV'12)*. Springer-Verlag, Berlin, Heidelberg, 159–172. DOI: http://dx.doi.org/10.1007/978-3-642-33709-3_12
20. Atsushi Nakazawa, Christian Nitschke, and Toyoaki Nishida. 2015. Non-Calibrated and real-time Human View Estimation Using a Mobile Corneal Imaging Camera. In *Multimedia & Expo Workshops (ICMEW), 2015 IEEE International Conference on*. IEEE, 1–6. DOI: <http://dx.doi.org/10.1109/ICMEW.2015.7169846>
21. Ko Nishino, Peter N. Belhumeur, and Shree K. Nayar. 2005. Using Eye Reflections for Face Recognition Under Varying Illumination. In *Proceedings of the Tenth IEEE International Conference on Computer Vision - Volume 01 (ICCV '05)*. IEEE Computer Society, Washington, DC, USA, 519–526. DOI: <http://dx.doi.org/10.1109/ICCV.2005.243>
22. Ko Nishino and Shree K. Nayar. 2004a. Eyes for Relighting. In *ACM SIGGRAPH 2004 Papers (SIGGRAPH '04)*. ACM, New York, NY, USA, 704–711. DOI: <http://dx.doi.org/10.1145/1186562.1015783>
23. Ko Nishino and Shree K Nayar. 2004b. The World in an Eye. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1. IEEE, 1–444. DOI: <http://dx.doi.org/10.1109/CVPR.2004.248>
24. Ko Nishino and Shree K. Nayar. 2006. Corneal Imaging System: Environment from Eyes. *Int. J. Comput. Vision* 70, 1 (Oct. 2006), 23–40. DOI: <http://dx.doi.org/10.1007/s11263-006-6274-9>
25. Christian Nitschke, Atsushi Nakazawa, and Toyoaki Nishida. 2013. I See What You See: Point of Gaze Estimation from Corneal Images. In *Proceedings of the 2013 2Nd IAPR Asian Conference on Pattern Recognition (ACPR '13)*. IEEE Computer Society, Washington, DC, USA, 298–304. DOI: <http://dx.doi.org/10.1109/ACPR.2013.84>
26. Christian Nitschke, Atsushi Nakazawa, and Haruo Takemura. 2011a. Display-camera Calibration Using Eye Reflections and Geometry Constraints. *Comput. Vis. Image Underst.* 115, 6 (June 2011), 835–853. DOI: <http://dx.doi.org/10.1016/j.cviu.2011.02.008>
27. Christian Nitschke, Atsushi Nakazawa, and Haruo Takemura. 2011b. Image-Based Eye Pose and Reflection Analysis for Advanced Interaction Techniques and Scene Understanding. *Computer Vision and Image Media (CVIM)* (2011), 1–16. <http://ci.nii.ac.jp/naid/110008584047/en/>
28. Christian Nitschke, Atsushi Nakazawa, and Haruo Takemura. 2013. Corneal Imaging Revisited: An Overview of Corneal Reflection Analysis and Applications. *Information and Media Technologies* 8, 2 (2013), 389–406. DOI: <http://dx.doi.org/10.11185/imt.8.389>
29. Raphael Ortiz. 2012. FREAK: Fast Retina Keypoint. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*. IEEE Computer Society, Washington, DC, USA, 510–517. <http://dl.acm.org/citation.cfm?id=2354409.2354903>
30. Ken Pfeuffer, Melodie Vidal, Jayson Turner, Andreas Bulling, and Hans Gellersen. 2013. Pursuit Calibration: Making Gaze Calibration Less Tedious and More Flexible. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. ACM, New York, NY, USA, 261–270. DOI: <http://dx.doi.org/10.1145/2501988.2501998>
31. Alexander Plopski, Yuta Itoh, Christian Nitschke, Kiyoshi Kiyokawa, Gudrun Klinker, and Haruo Takemura. 2015. Corneal-Imaging Calibration for Optical See-Through Head-Mounted Displays. *IEEE transactions on visualization and computer graphics* 21, 4 (2015), 481–490. DOI: <http://dx.doi.org/10.1109/TVCG.2015.2391857>
32. Daniel Pohl, Xucong Zhang, Andreas Bulling, and Oliver Grau. 2016. Concept for Using Eye Tracking in a Head-mounted Display to Adapt Rendering to the User's Current Visual Field. In *Proceedings of the 22Nd ACM Conference on Virtual Reality Software and Technology (VRST '16)*. ACM, New York, NY, USA, 323–324. DOI: <http://dx.doi.org/10.1145/2993369.2996300>
33. Joseph Redmon and Ali Farhadi. 2016. YOLO9000: Better, Faster, Stronger. (2016).
34. Dirk Schnieders, Xingdou Fu, and Kwan-Yee K Wong. 2010. Reconstruction of Display and Eyes from a Single Image. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1442–1449. DOI: <http://dx.doi.org/10.1109/CVPR.2010.5539799>
35. John D. Smith, Roel Vertegaal, and Changuk Sohn. 2005. ViewPointer: Lightweight Calibration-free Eye Tracking for Ubiquitous Handsfree Deixis. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology (UIST '05)*. ACM, New York, NY, USA, 53–61. DOI: <http://dx.doi.org/10.1145/1095034.1095043>
36. Sophie Stellmach and Raimund Dachsel. 2012. Look & Touch: Gaze-supported Target Acquisition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 2981–2990. DOI: <http://dx.doi.org/10.1145/2207676.2208709>
37. Kentaro Takemura, Shunki Kimura, and Sara Suda. 2014. Estimating Point-of-regard Using Corneal Surface Image. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '14)*. ACM, New York, NY, USA, 251–254. DOI: <http://dx.doi.org/10.1145/2578153.2578197>

38. Kentaro Takemura, Tomohisa Yamakawa, Jun Takamatsu, and Tsukasa Ogasawara. 2013. Estimating Focused Object using Corneal Surface Image for Eye-Based Interaction. In *3rd International Workshop on Pervasive Eye Tracking and Mobile Eye-Based Interaction*.
39. Tobii Technology. 2017. Tobii Pro Glasses 2. (2017). Retrieved Oct 2017 from <https://www.tobiiipro.com/product-listing/tobii-pro-glasses-2/>
40. Fabian Timm and Erhardt Barth. 2011. Accurate Eye Centre Localisation by Means of Gradients. *VISAPP 11* (2011), 125–130.
41. Jayson Turner, Jason Alexander, Andreas Bulling, and Hans Gellersen. 2015. Gaze+RST: Integrating Gaze and Multitouch for Remote Rotate-Scale-Translate Tasks. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 4179–4188. DOI : <http://dx.doi.org/10.1145/2702123.2702355>
42. Jayson Turner, Andreas Bulling, Jason Alexander, and Hans Gellersen. 2014. Cross-device Gaze-supported Point-to-point Content Transfer. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '14)*. ACM, New York, NY, USA, 19–26. DOI : <http://dx.doi.org/10.1145/2578153.2578155>
43. Jayson Turner, Andreas Bulling, and Hans Gellersen. 2012. Extending the Visual Field of a Head-mounted Eye Tracker for Pervasive Eye-based Interaction. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '12)*. ACM, New York, NY, USA, 269–272. DOI : <http://dx.doi.org/10.1145/2168556.2168613>
44. Roel Vertegaal and others. 2003. Attentive User Interfaces. *Commun. ACM* 46, 3 (2003), 30–33.
45. A. Villanueva and R. Cabeza. 2008. A Novel Gaze Estimation System With One Calibration Point. *Trans. Sys. Man Cyber. Part B* 38, 4 (Aug. 2008), 1123–1138. DOI : <http://dx.doi.org/10.1109/TSMCB.2008.926606>
46. Erroll Wood and Andreas Bulling. 2014. EyeTab: Model-based Gaze Estimation on Unmodified Tablet Computers. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '14)*. ACM, New York, NY, USA, 207–210. DOI : <http://dx.doi.org/10.1145/2578153.2578185>
47. Laurence R. Young and David Sheena. 1975. Survey of eye movement recording methods. *Behavior Research Methods & Instrumentation* 7, 5 (01 Sep 1975), 397–429. DOI : <http://dx.doi.org/10.3758/BF03201553>