ScreenGlint: Practical, In-situ Gaze Estimation on Smartphones

Michael Xuelin Huang, Jiajia Li, Grace Ngai, Hong Va Leong
CHI Lab, Department of Computing, Hong Kong Polytechnic University
Hong Kong, China
{csxhuang,csjjli,csgngai,cshleong}@comp.polyu.edu.hk

ABSTRACT
Gaze estimation has widespread applications. However, little work has explored gaze estimation on smartphones, even though they are fast becoming ubiquitous. This paper presents *ScreenGlint*, a novel approach which exploits the glint (reflection) of the screen on the user's cornea for gaze estimation, using only the image captured by the front-facing camera. We first conduct a user study on common postures during smartphone use. We then design an experiment to evaluate the accuracy of ScreenGlint under varying face-to-screen distances. An in-depth evaluation involving multiple users is conducted and the impact of head pose variations is investigated. ScreenGlint achieves an overall angular error of 2.44º without head pose variations, and 2.94º with head pose variations. Our technique compares favorably to state-of-the-art research works, indicating that the glint of the screen is an effective and practical cue to gaze estimation on the smartphone platform. We believe that this work can open up new possibilities for practical and ubiquitous gaze-aware applications.

Author Keywords
Gaze estimation; mobile eye tracker; screen reflection; glint.

ACM Classification Keywords
H.1.2 [Models and Principles]: User/Machine Systems—Human factors; I.5.m [Pattern Recognition]: Miscellaneous

INTRODUCTION
Gaze estimation has attracted quite a lot of research attention. However, most of the current techniques are devised for desktop applications. Only a few attempts [2][11][30] have been made to explore mobile platforms, such as tablets. Prior studies on mobile gaze estimation [9][11][30] generally produce lower accuracy than their counterparts in the desktop settings, probably due to severe head pose variations.

As an alternative to precise gaze point estimation, some efforts have also explored gaze interaction techniques based on eye movements, notably smooth pursuit [2][26]. These methods focus on gaze movements or gaze gestures, rather than the exact gaze locations. However, their approaches still require accurate identification of gaze features or input from a reliable eye tracker. Therefore, there still remains a need for accurate gaze estimation on the mobile platform.

State-of-the-art eye trackers usually use the reflection(s) from a single or multiple infrared (IR) light source(s) on the cornea for gaze estimation. This reflection is often referred to as the *glint* [1] (see Figure 1). Despite the successful use of the glint in the desktop settings, few efforts have been made to extend it to smartphones. This may be because the small size of the smartphone constrains the setup of multiple well-spread IR emitters, and therefore decreases the informativeness of the glint(s) and the robustness of current methods. Furthermore, many accurate gaze estimation techniques require specialized equipment [6], such as multiple IR light sources or multiple cameras.

Gaze estimation methods that use signals from RGB cameras often treat corneal reflections as noise [30] and filter them out during preprocessing. However, inspired by the IR-based methods, we see an opportunity to leverage the glint of the screen for gaze estimation using the standard front-facing camera on the smartphone.

Generally, the use of a single camera and a single glint is not sufficient to overcome the head pose variations, mainly because of the displacement between the center of the eyeball and the sphere that fits the corneal surface [6]. However, normal smartphone use generally involves users facing a small screen straight-on from a close distance. Under such contexts, the glint of the screen locates in the

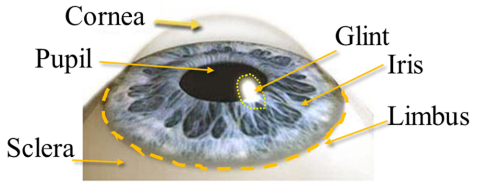

Figure 1. An illustration of an eye model. The cornea is the transparent front part of the eye that covers the iris and pupil. Limbus is the boundary between iris and sclera. Glint is the reflection on the surface of cornea.

CHI 2017, May 06-11, 2017, Denver, CO, USA
 ISBN 978-1-4503-4655-9/17/05…$15.00
DOI: http://dx.doi.org/10.1145/3025453.3025794

central area of the cornea. Therefore, the influence of head movement on the shift of the glint is minimal. This potentially makes it possible to perform gaze estimation with only a single glint from the screen.

This paper proposes *ScreenGlint*, which uses the glint of the screen to estimate the gaze location. ScreenGlint uses only the unmodified front-facing camera on the smartphone, and does not rely on any add-on or specialized equipment. We present an in-depth evaluation on the performance of ScreenGlint against different face-to-screen distances. To the best of our knowledge, this is the first work that uses the glint of the screen to estimate gaze on smartphones. This work suggests a practical cue to gaze estimation on smartphones. It also has the potential of making a significant contribution to the human-computer interaction community.

The contributions of this paper are as follows: we (1) study common postures in which smartphones are held when in use and collect a dataset of 18 subjects with which to evaluate the accuracy of gaze estimation on the smartphone platform; (2) propose *ScreenGlint*, a technique that exploits the glint of the screen for gaze learning; (3) design an effective algorithm to detect the glint from the captured image; (4) evaluate the performance and effectiveness of ScreenGlint and suggest a guideline for practical usage.

RELATED WORK

Although gaze estimation is a perennially popular research topic and eye trackers on desktop platforms have achieved remarkable successes, estimating gaze point on the smartphone platform, especially without the use of specialized equipment, is still a challenging issue. This section discusses related research on mobile gaze estimation, followed by general techniques that use RGB cameras or glints for gaze estimation.

Gaze Estimation on the Mobile Platform

Only a few studies have explored gaze estimation on the mobile platform using the built-in, unmodified camera. Holland et al. [9] used a neural network to learn the mapping from the pixel intensity of the eye region into the gaze point. Huang et al. [11] applied the widely-used feature descriptors to extract more effective appearance information. Rather than using the raw low-level image features, researchers have also explored the eye geometric attributes, including the eye corners, pupil center, limbus, and eyelids. Swirski et al. [25] used random sample consensus (RANSAC) [3] to identify the pupil contour and center. Similarly, Wood and Bulling [30] developed the EyeTab system that extracts the iris contour and calculates the 3D intersection between the optical axis and screen. Hohlfeld et al. [8] presented an evaluation on EyeTab. They pointed out the current gaze accuracy on tablets is not good enough for many HCI applications, such as the evaluation of interface design.

Since it is difficult to achieve high gaze accuracy on smartphones, there are emerging interaction applications that do not rigidly rely on the gaze location. For instance, instead of forcing the cursor to follow the eye movement, Zhai et al. [33] proposed the MAGIC system that warps the cursor to the vicinity of the potential gaze target. Rozado et al. [19] showed that saccadic gaze gestures are promising for mobile interactions. Vidal et al. [26] suggest pursuit-based interaction for large-size public displays. Correlation between the eye pursuit (estimated by an uncalibrated eye tracker) and the trajectory of the visual object is used to identify the user intention. Esteves et al. [2] further explored this technique on the small-size smartwatch platform. Although precise gaze estimation may not be necessary for these applications, acquiring the accurate gaze movement or eye features is still generally preferable.

Accurate Eye Tracker Based on RGB Camera

With a few exceptions, most of the studies on gaze estimation rely on data-driven methods to learn the mapping from gaze features into gaze point, and these methods generally rely on massive training data. Huang et al. [11] prepared a large dataset called TabletGaze for gaze learning, which contains over 100 thousand images. Zhang et al. [34] constructed a convolution neural network (CNN) to recognize the gaze angle from the MPIIGaze dataset, which consists of over 200 thousand images. Xu et al. [32] presented TurkerGaze, which collects large-scale gaze data through crowdsourcing. Krafka et al [13] also used a CNN to model gaze from 1.5 million frames of face and eye regions. Similarly, Zhang et al. [35] presented a CNN that learns a gaze model using only full face images. As CNNs and regression models are competing technologies for eye tracking, a comparison between these two key techniques is presented near the end of this paper, right before our concluding remarks on our work.

Instead of using real images, Wood et al. [29] leveraged graphical rendering techniques to generate a million realistic eye images for gaze angle estimation. However, previous work [13] shows that gaze models calibrated by user-specific data can markedly outperform their user-independent counterpart, implying that it is still hard to overcome individual differences by simply learning from large amounts of data. Lu et al. [14] synthesized user-specific training samples for unseen head poses from images with certain reference head positions. However, this kind of computer-generated data may not fully coincide with the real user-specific data.

An alternative is to collect user-specific data implicitly through interactions. Sugano et al. [24] applied the saliency map of video frames to estimate the gaze points according to the eye appearances. Wang et al. [28] proposed to minimize the overall difference between the estimated fixations and the saliency map. However, the consistency between image saliency and real gaze locations can be affected by the semantics and complexity of visual stimuli.

The use of mouse and keyboard as cues to gaze has also been investigated. Sugano et al. [23] used the mouse-click locations as ground truth to update the gaze model. Similarly, Papoutsaki et al. [16] presented a browser-based eye tracker

that learns from mouse-clicks and mouse movements. To ensure consistency between gazes and interactions, Huang et al. [10] identified the temporal and spatial alignments between keypresses, mouse-clicks, and the gaze signals. These methods achieve impressive performance, but they are all based on the desktop platform. They also require sufficient well-aligned data from user interactions, which can be slow to acquire, which therefore limits their applicability.

Using Glints in Eye Trackers

Given a light source that the user is looking at, the vector from the pupil center to the corneal reflection, or the *glint*, shows the distance and angle differences between pupil center and glint on the cornea [1]. It has been shown to be one of the most indicative features for gaze estimation [6]. In contrast to appearance-based methods, glint-based gaze estimation methods require only a short calibration with a few fixation points [27]. However, they generally require the use of infra-red (IR) light source(s).

One major challenge for almost all vision-based methods, including the glint-based methods, is head pose variance. Most single-glint methods approximate the corneal surface as a perfect mirror [6]. However, the location of the glint shifts as the head pose varies and the cornea rotates. Villanueva and Cabeza [27] showed that modeling the pupil contour as an ellipse can be useful in reducing the dependency between glint features and head poses. Guestrin and Eizenman [5] studied gaze estimation from pupil center corneal reflections under different setups. They pointed out that increasing number of cameras and light sources can reduce the undetermined parameters for calibration and relax the constraints on head movements. Similarly, Hennessey et al. [7] presented a system based on a single camera and two glints, which allows for free head movements. Hansen and Ji [6] concluded that adding light sources is a small change that contributes significantly to achieving head pose invariance. Nevertheless, little work has investigated the use of glint on the smartphone platform. This may be explained by the size constraint of smartphones; any light sources will be necessarily located close to each other and may not be able to produce clear and robust cues to eye gaze.

Besides the reflection from IR light sources, a few studies have explored using the reflection from the screen. Nitschke et al. [15] used the screen reflection on the cornea to determine the relative location of the screen and camera. Only two studies have presented tentative results in exploiting screen reflection for gaze estimation, both on desktops. Iqbal and Lee [12] located the glint and eye by the screen flicker pattern detected by a specialized high-speed camera. Their method allows for no head movement and is inapplicable to smartphones. Schnieders et al. [20] used the edges of screen reflection for calibration. However, in the context of smartphone use, the shape of the reflection is easily distorted under illumination variations.

To summarize, gaze estimation on mobile platforms generally leverages conventional techniques based on the RGB camera. A well-performing eye tracker may require a lengthy calibration or training data collection. Even though the use of glints (of IR light sources) can simplify the process, it seems to be infeasible on the smartphone platform.

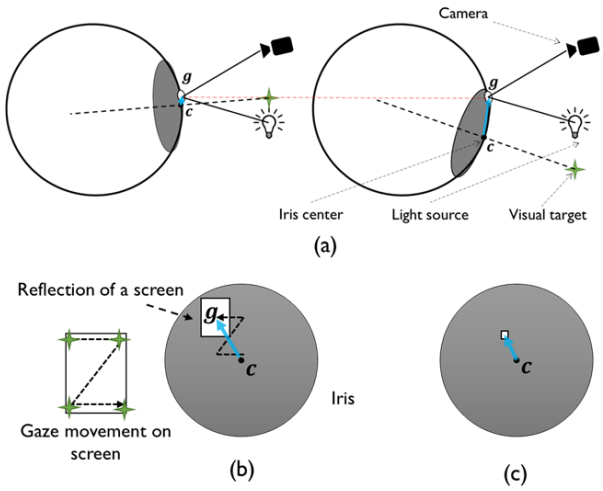

Figure 2. Gaze estimation based on corneal reflection and iris center. The gray area shows the iris (a: side view; b, c: front view); the black dot shows the iris center; the green star indicates the visual target of gaze; the blue arrow shows the glint-iris vector. The location of the glint on the cornea is relatively stable; therefore, the relative location between the glint and iris center can help to indicate the gaze point.

This paper uses the screen glint as a cue to gaze learning. We propose a robust technique for feature extraction from the unmodified camera and conduct an extensive performance evaluation against head pose variations.

SCREENGLINT FOR GAZE ESTIMATION

Prior studies pointed out that gaze estimation based on single glint may not be robust to head pose variations. However, in situations where users face the screen straight-on and the screen is relatively close to the face, as is often the case during smartphone use, gaze estimation based on the glint of the screen can be feasible.

Figure 2-a illustrates the corneal reflection and its relation to the gaze point. Assuming the existence of a fixed light source and a fixed camera, there is an offset from the center of the iris $\boldsymbol{c}$ to the corneal reflection from the light source $\boldsymbol{g}$. The direction and magnitude of the vector from the iris center to the corneal reflection changes as the eye rotates to fixate on different locations. For instance, if the gaze point moves in a zigzag pattern along the screen corners, the screen reflection on the cornea will move inversely to the relative location of the iris as shown in Figure 2-b. In other words, the gaze movement can be measured by the *glint-iris vector*, which points from the iris center to the centroid of the reflection $\boldsymbol{g}$. When the glint is from a weak light source such as a smartphone screen, the edges of the reflection are often

blurred, and the intensity of glint appears to vary in a Gaussian-like manner. To compensate for this, we consider only the centroid of the reflection for gaze estimation.

The effectiveness of this technique can be affected by the face-to-screen distance. A large distance may lead to a screen reflection that is so small that the glint-iris vector is likely to be inaccurate and sensitive to visual noise (see Figure 2-c). Thus, the question is whether, under normal ranges of face-to-screen distances, the glint feature can yield an acceptable prediction of gaze.

GLINTS AND COMMON SMARTPHONE USE POSTURES

To understand the gaze learning based on the screen glint and the related human behaviors, we first conduct a study on common postures of smartphone use. We hypothesize that the face-to-screen distance is the critical issue, and we intend to identify the normal range of face-to-screen distance in daily smartphone use.

Collecting Common Smartphone Use Postures

In previous work, Huang et al. [11] suggested four representative postures on the tablet platform, including standing, sitting, slouching, and lying down. We further subdivide these postures to give a total of 8 categories: sitting at a table, sitting on a chair, slouching, sitting on the floor/bed, prone, supine, standing, and walking.

We recruited 6 subjects (3 females, ages 24-32 yrs, mean 27, standard deviation 2.7) for this study. Subjects were asked to perform simple searching tasks, such as looking for the temperature and the next public holiday, while assuming one of the 8 possible postures. An icon illustrating the requested posture was shown to the subjects for reference (Figure 3);

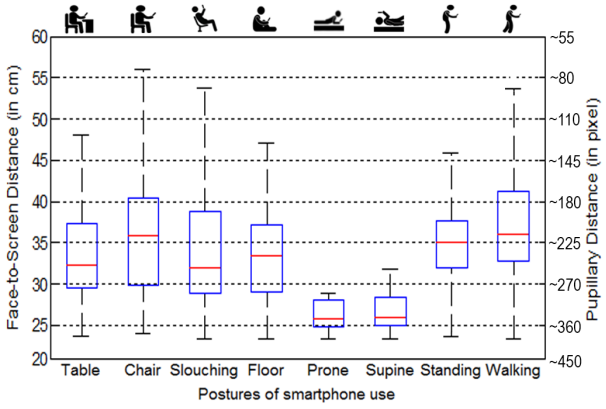

Figure 3. The range of the approximated face-to-screen distances (y-axis, left) and the pupillary distances measured in the images (y-axis, right) against different postures of smartphone use. The red lines show the median; the blue boxes denote the 25th and 75th percentiles; the black whiskers extend to the range without considering outliers. The postures from left to right are: sitting by a table, sitting on a chair, slouching, sitting on the floor/bed, prone, supine, standing and walking.

all subjects completed at least one searching task for each possible posture.

The tasks were performed on an iPhone 6 (with a 4.7-inch screen) using the built-in Safari browser. Simultaneously, the frontal camera of the iPhone recorded video of the user in a 720x1280 resolution and at 30 fps. In total, we collected 117 videos with 26,564 frames across different time slots over 5 days. All the videos were captured indoors. To account for environmental illumination variations, around half of the videos (61) were captured under good lighting conditions (around 150 lx), and the other half under poor conditions (around 45 lx).

Face-to-Screen Distances in Common Postures

We first study the range of the face-to-screen distance across the 8 postures of smartphone use. We measure (1) the pupillary distance and (2) the approximated face-to-screen distance derived from the pupillary distance via regression. It is not difficult to see that the pupillary distance, or the distance between the centers of the irises of the eyes as measured in the image, increases as the user's face approaches the screen, and vice versa. Hence, the pupillary distance can be a good indicator of the face-to-screen distance, especially when adults are concerned. We extract the pupillary distance using an automated method, which will be described in a later section.

Since the pupillary distance varies from individual to individual, we also construct a user-specific regression model to approximate the face-to-screen distance according to the pupillary distance in the image. The user-specific training data for these models were collected in a post-experiment session. Specifically, for each subject we used a ruler to identify 6 positions with different face-to-screen distances (20, 25, 30, 35, 40, 45 cm), and we captured 5 images (with slight head pose differences) in each of these positions. We then used the pupillary distances extracted from these images and the corresponding actual face-to-screen distances to build a 2nd order polynomial regression model.

Figure 3 presents the range of face-to-screen distance and pupillary distances measured in the images corresponding to different postures. The red lines show the median of all frames across subjects; the blue boxes denote the range within $[p_{25}, p_{75}]$, i.e. the 25th and 75th percentiles; the black whiskers indicate the range without outliers, i.e. within $[2.5p_{25} - 1.5p_{75}, 2.5p_{75} - 1.5p_{25}]$.

It can be seen that when people are prone and supine, the overall face-to-screen distance is much lower (26.9 cm on average) than with other postures (34.3 cm). It is intuitive that people prefer to assume a relaxed posture and therefore tend to hold the phone closer to their face when lying on their back. For the same reason, people prefer to seek a support for their arm and elbow, such as a table or a chair arm. When sitting on the floor/bed and slouching forward, people often put their elbows on their laps. When slouching backwards,

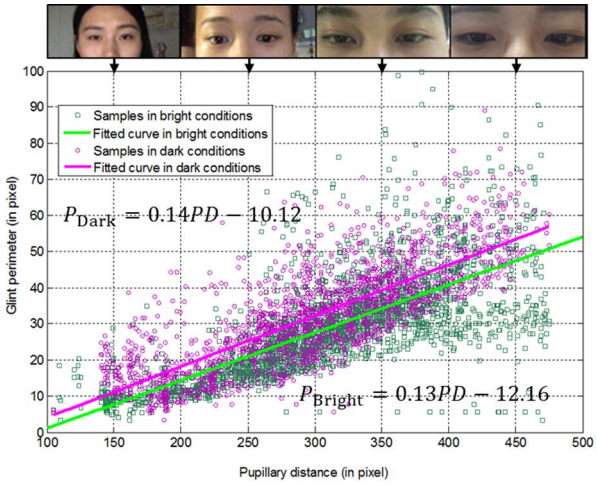

Figure 4. The relation between the glint perimeter and the pupillary distance captured in the image. The green circles show the samples in the bright; the purple squares in the dark. The green and purple lines indicate the linear fitted results of samples in the bright and dark conditions, respectively. The photos on top show the examples with corresponding pupillary distances of 150, 250, 350, and 450 pixels. It can be seen that the change of pupillary distance has a stronger impact on the glint perimeter than that of the illumination.

they tend to keep their arms close to their torso. Hence, sitting at a table (33.4 cm), slouching (33.7 cm) and sitting on the floor/bed (33.0 cm) generally result in low face-to-screen distances. As expected, sitting on a chair (35.7 cm) without a table leads to a large variation in face-to-screen distance. In cases where there is no obvious support for the arm, people hold the phone farther away from their face, such as when standing (34.2 cm) and walking (36.4 cm).

To conclude, the normal range of face-to-screen distance is around 30~40 cm for sitting, standing and walking, and 25~30 cm when lying down. In general, the common postures of smartphone use produce a face-to-screen distance of 25~40 cm, which corresponds to a pupillary distance of approximately 180~360 pixels. It is worth noting that this range is much shorter than that of taking selfies (around 45 cm).

Understanding the Impacts of Illumination and Distance

The size of the glint is an important factor for ScreenGlint. Inspecting the data, we observed that there are two critical factors that affect the perimeter of the glint: the face-to-screen distance, and the environmental illumination. The perimeter of the glint dwindles as the pupillary distance decreases. The glint also generally appears smaller in a brighter environment, which may be because the texture of the iris appears light and clear in a bright environment, which results in a weaker contrast around the edge of the glint. Understanding these factors can facilitate the correct localization of the glint, which is key to our technique.

As the pupillary distance is observable, we examine the relationship this distance and the perimeter of the glint. Figure 4 shows the relation between the pupillary distance PD and the perimeter of the glint contour P. P is measured by the number of pixels on the contour. Samples collected in bright conditions are denoted by green circles, while purple squares show samples collected in dark conditions. The green and purple lines indicate the fitted linear regression lines between pupillary distance and glint perimeter under the two lighting conditions. The photos on the top of the figure show examples corresponding to pupillary distances of 150, 250, 350, and 450 pixels.

As expected, the perimeter of the glint grows as the increase of the pupillary distance, i.e. the decrease of the face-to-screen distance. In addition, glints in dark environments generally appear larger than those in bright conditions, and the glint perimeter correlates better to PD under dark conditions (R-squared: 0.62) than bright conditions (R-squared: 0.50). Therefore, a darker environment seems more preferable to ScreenGlint.

However, according to the fitted results ($P_{\text{Dark}} = 0.14PD - 10.12$; $P_{\text{Bright}} = 0.13PD - 12.16$), it is not difficult to see that the impact of the illumination on the glint perimeter is rather stable along the change of the pupillary distance. In other words, given the same pupillary distance, the perimeter of the glint in dark conditions contains 5 more pixels on average than that in bright illumination.

More interestingly, the increment in glint perimeter resulting from a change in face-to-screen distance is much more significant. For instance, the glint perimeter increases by more than 15 pixels for a 100-pixel increase in pupillary distance.

Our finding suggests that of the two factors, face-to-screen distance has the more significant impact on ScreenGlint. It also provides an acceptable range of perimeter values within which the correct glint can be found, even in the presence of noisy reflections.

Our preliminary experiments give us the range of face-to-screen distance of 25~40 cm, or pupillary distance of 180~360 pixels in common postures, and the corresponding range of the perimeter of the glint (around 10~40 pixels). Our next step is to evaluate whether the glint can be used for accurate gaze estimation under these conditions.

EXTRACTING FEATURES FOR GAZE LEARNING

ScreenGlint estimates the gaze point on the smartphone screen based on the pupillary distance and the displacement from the iris center to the glint. We now describe the feature extraction and the prediction model.

Detection of the Rough Iris Region

Eye feature extraction requires accurate facial landmark alignment. ScreenGlint applies the Supervised Descent Method (SDM) [31] to align 48 facial landmarks as shown in Figure 5-a. SDM uses the descent maps learned from the

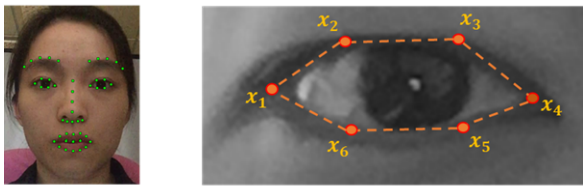

Figure 5. The landmarks tracked by the face alignment model. (a) shows the 48 face landmarks, which are tracked by the SDM face alignment model. (b) shows an eye image and the eye related landmarks $x_1, \dots, x_6$. The dashed polygon indicates the eye contour V, which is a closed shape defined by $x_1, \dots, x_6$.

training data to minimize the non-linear least squares objective without having to compute the Jacobian or Hessian matrices. It has been demonstrated to achieve robust performance in face landmarks alignment in the wild [10][31].

To identify the iris, we segment the eye region into rectangular areas and assume that the iris corresponds to the darkest area. We define the *eye image* as an enlarged region of the bounding area that encloses the eye contour. The eye contour $\boldsymbol{V}$ is a closed shape defined by the landmarks $\boldsymbol{V} = \langle \boldsymbol{x}_1, \dots, \boldsymbol{x}_6 \rangle$ as shown in Figure 5-b. In practice, the precision of the landmark alignment can be affected by peripheral noise, such as that from eyelashes or shadows. The aligned landmarks may not be located on the actual eye contour. Therefore, we use the distance of each point from $\boldsymbol{V}$ to represent its probability of being outside the eye and adjust the intensity of the pixels by this probability. Specifically, the intensity of point $\boldsymbol{u}$ outside $\boldsymbol{V}$ is updated by

$$I_{\boldsymbol{u}} \coloneqq I_{\boldsymbol{u}} + \lambda \cdot \operatorname{argmin}_{\boldsymbol{v} \in \boldsymbol{V}} \|\boldsymbol{u} - \boldsymbol{v}\|$$

where λ (=5) controls the impact of probability and $\boldsymbol{v}$ is any point on the eye contour $\boldsymbol{V}$.

Figure 6 shows two example eye images before and after this intensity adjustment. We perform min-max normalization to adjust the pixels in the eye image. The result is that the region inside the eye contour is darkened, and the region outside is brightened, depending on the distance of the pixel from the eye contour $\boldsymbol{V}$.

We then search for the darkest rectangle from the integral image [17] of the normalized eye image. The rectangle is determined by the shape of the eye, with its width defined as $0.4\|\boldsymbol{x}_1 - \boldsymbol{x}_4\|$ and the height as $0.5\|\boldsymbol{x}_2 - \boldsymbol{x}_6\| + 0.5\|\boldsymbol{x}_3 - \boldsymbol{x}_5\|$.

Localization of the Iris Center

Conventional glint-based methods use the pupil center for gaze estimation, and therefore, a precise localization of the pupil center is essential for accuracy. However, for people with dark eyes, the shape of the pupil is often invisible to RGB cameras. Since the pupil center is close to the iris center, we use the iris center as an approximation.

Figure 6. Two examples of the eye images and their modified versions for iris detection. In the upper row, (a) shows an example captured in the dark and (b) in bright light. After the modification based on the probability of being outside the eye contour, the iris in the eye images (in the lower row) are easier to detect based on intensity.

Our approach assumes that the darkest rectangular area in the eye region corresponds to the iris, and pilot tests show that this assumption is generally valid. However, in real-use situations, it is not uncommon for the iris to be partially occluded by eyelids and eyelashes. Therefore, the center of the resulting rectangle may not correspond to the center of the iris. Therefore, although this approximation is a robust and efficient method for obtaining a rough estimate of the iris region, it is not adequate for accurate location of the iris center.

The estimation of iris center generally relies on the ellipse fitting to the iris contour [6]. ScreenGlint uses the GrabCut [18] algorithm to segment the iris from the eyelids and sclera. It uses Gaussian mixture models to estimate the color space parameters of the foreground and background. The iris is segmented based on the energy minimization, considering both the edge information of the neighboring pixels and the probability of the pixel being the foreground. Therefore, GrabCut generally performs a more accurate and robust segmentation of the iris than methods that use only the gray-scale, local textural information.

In accordance with the shape of the iris, we use 3 concentric circles with the same center $\boldsymbol{o}$ as the iris rectangle to define the initial foreground and background for GrabCut (Figure 7-a). Setting the eye corner distance to be $CD = \|\boldsymbol{x}_1 - \boldsymbol{x}_4\|$ (Figure 7-b), we define the foreground as the circular area $\operatorname{circ}(\boldsymbol{o}, r_1)$, the possible foreground as the annular area $\operatorname{annu}(\boldsymbol{o}, r_1, r_2)$, the possible background as $\operatorname{annu}(\boldsymbol{o}, r_2, r_3)$, and the remaining region in the eye image as the background, where r_1(=0.1CD), r_2(=0.2CD), r_3(=0.3CD) are the radii of the concentric circles and $\operatorname{circ}(,)$, $\operatorname{annu}(,,)$ define the region of a circle and a annulus, respectively. In our experiment, we use 5 iterations in GrabCut to constrain the computational expense, and it generally produces an accurate contour of the iris.

Due to occlusion from eyelids and eyelashes, some of the points on the iris contour do not belong to the limbus. Therefore, only the points on two sides of the iris contour

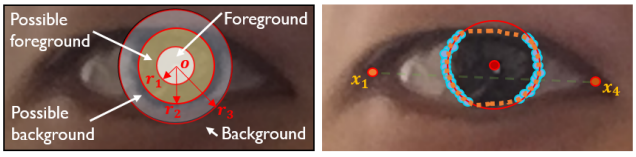

Figure 7. The procedure of the gaze feature vector extraction based on GrabCut. (a) shows the initial regions of the foreground and background for the GrabCut algorithm. (b) shows the localization of the iris center. The dotted orange shape is the segmentation output of GrabCut; the red circle is fitted to the selected points (marked in blue) on the left and right sides of the limbus contour; and the red dot in the middle is the center of the fitted circle, i.e. the iris center.

(Figure 7-b) are used for circle fitting to identify the accurate shape of the limbus. Let $\boldsymbol{L}$ denote the points on the detected iris contour. We consider the points that satisfy the following condition as the points that belong to the actual limbus:

$$\theta_1 < |\angle(\boldsymbol{l}_i - \boldsymbol{l}'_i, \boldsymbol{x}_4 - \boldsymbol{x}_1)|_{\boldsymbol{l}_i, \boldsymbol{l}'_i \in \boldsymbol{L}} < \theta_2$$

where $\angle(,)$ calculates the angle between two vectors, and θ_1(=π/4) and θ_2(=3π/4) are the angular thresholds depicting the points on two sides of the contour; $\boldsymbol{l}_i$ and $\boldsymbol{l}'_i$ are consecutive points on the iris contour $\boldsymbol{L}$ that lie within a small distance (5 pixels) to each other. We then use RANSAC [3] to fit an circle [4] to the selected points. The center of the fitted circle is considered to be the iris center.

Identification of the Glint

During smartphone use, the glint of the screen is located near the center of the iris. However, occlusion and shadow caused by eyelashes sometimes affect the iris region identified by GrabCut. Although the fitted circle can represent the limbus in a more precise manner, it generally contains part of the eyelids. We therefore look for the bright regions within the intersection between the GrabCut result and the fitted circle of the limbus.

To locate the glint despite the presence of reflections from other light sources, we take into consideration the reflection intensity and size as well as the relative location of the iris center. Specifically, we assume that the brightest reflection closest to the iris center with a perimeter less than 60 pixels (supported by Figure 4) is most likely to be the glint of the screen.

Since the brightness of the glint seems to fade out gradually on the edge, the location of a reflection is represented by its centroid. The centroid is measured by the weighted average location, according to the pixel intensity that it encloses.

Gaze Feature Vector for Gaze Prediction

Given the locations of the iris and the glint for both eyes, ScreenGlint calculates five gaze features, which include the information of the pupillary distance, and the glint-iris vectors of both eyes. The gaze feature vector $\boldsymbol{f}$ is defined as

$$\boldsymbol{f} = \langle \alpha PD, \frac{\beta \Delta x_L}{r_L}, \frac{\beta \Delta y_L}{r_L}, \frac{\beta \Delta x_R}{r_R}, \frac{\beta \Delta y_R}{r_R} \rangle$$

where PD denotes the pupillary distance; Δx and Δy indicate the projections of the glint-iris vector onto the x-axis and the y-axis, respectively; r_L, r_R are respectively the iris radii from the left and right eye, as measured in the image; and α(=1/250), β(=2) are the constant scalars that ensure the ranges of the gaze features are in a comparable scale.

Following previous practices [22][24], we use Gaussian Processes [21] for regression, which produces 2 separate mappings from the gaze feature vector to obtain predictions of the x- and y- coordinates of the gaze point on the screen. A simple dot product covariance function (1st-order polynomial kernel) is used in the Gaussian Processes.

EVALUATING SCREENGLINT FOR GAZE ESTIMATION

We evaluate ScreenGlint through an in-depth evaluation of the accuracy of the estimated gaze location. In particular, we intend to investigate the impact of face-to-screen distance, which we believe significantly affects the effectiveness of the ScreenGlint algorithm.

Our gaze learning process follows that of most common glint-based methods. We calibrate our model with fixations on a few pre-determined points, which are then used to construct a user-specific gaze model.

Data Collection for the Evaluation of Gaze Estimation

We recruited 18 subjects (8 females, ages 21-32 yrs, mean 27, standard deviation 3.2). 15 are undergraduate and postgraduate students, the rest are university staff. None of the subjects wore glasses in the experiment.

Our experiment contains 3 sessions, with each session corresponding to a different face-to-screen position (Figure 8-a). Subjects were instructed to sit in the same position on a chair and a smartphone (iPhone 6) is placed in front of them on a table using a dock with a fixed angle φ(=60°) to the table. Varying the position of the phone generates data with

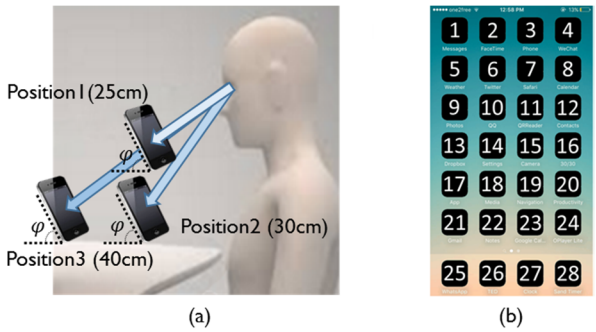

Figure 8. The experiment setting of ScreenGlint. (a) In each session, a smartphone is placed in a dock with a fixed angle φ(=60°) in a different position (with the face-to-screen distance of 25, 30, and 40 cm) on the table. (b) Subjects were prompted to fixate on each point in the predefined 28-point array sequentially.

different face-to-screen distances of 25, 30, and 40 cm and at different angles. These positions are in line with our findings from common smartphone use which give a normal range of face-to-screen distance of around 25~40 cm. This variation is designed to facilitate our understanding of the impacts of the distance and angle through the performance evaluation.

To collect the data points, we superimpose a 4x7 grid of squares on the smartphone screen. The placement and size of these squares are designed to follow the default placement of icons on the iOS home screen (Figure 8-b). The resolution of the screen is 750x1334 pixels, and the squares are 120x120 pixels in size, which gives a 180-pixel horizontal and vertical displacement between centers of neighboring squares.

Following practice from previous work [8], the data collection process requires experiment subjects to gaze at each point in turn, while the unmodified frontal camera on the smartphone records video of the subjects' faces at a resolution of 720x1280 and 30 fps. Subjects were guided by an audio track instruction, which prompted them when they were supposed to move to the next point. To ensure that sufficient good data is collected, the fixation on each point is set to 3 seconds, which gives us 2,520 data points for each subject for each session.

Due to eye blinks and saccadic movements, the collected data contains noise that correspond to periods when subjects are not fixating on the predefined points. We removed the blinks based on the eyelid openness in the video and removed the saccades based on the iris movement. This leaves us with an average of 4,827 data points per subject, and an average of 57 data points of a subject fixating on a particular predefined location.

Gaze Estimation without Head Pose Variations

Our first evaluation is designed to investigate the gaze estimation performance when head pose variations are not present. This is potentially the upper bound, the "best possible" performance that we can achieve with this approach. In this test, both the training and test data come from the same session – i.e. that the head pose is basically the same for both training and test. We use the data points corresponding to fixations on the 4 corners of the screen (i.e. Points 1, 4, 25, 28) as training data (i.e. 4-point calibration), and the rest of the data from the same position session is used for testing. This gives us 3 sessions per subject. The overall performance is measured by the average results across all 3 head positions and 18 subjects.

Figure 9 shows the mean absolute error (MAE) in pixels and the correlation coefficient between estimate and ground truth, which is defined as the point that the user was instructed to fixate upon. The double solid line shows the 2D MAE measured by the Euclidean distance. The orange dotted line indicates the MAE for the *x*-coordinate and the green dotted line for the y-coordinate. Similarly, the orange and green solid lines show the correlation of *x*- and *y*-coordinates, respectively. The error bars show standard

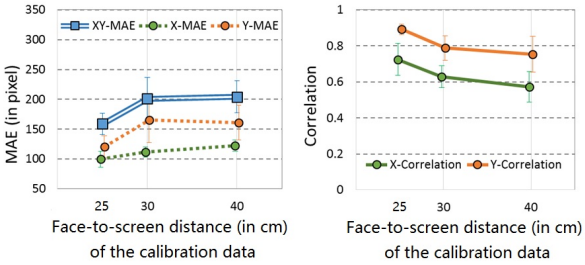

Figure 9. Performance of ScreenGlint without considering head pose variations. The figure on the left shows the mean absolute error (MAE) in pixels. The figure on the right presents the *x*- and *y*- correlation between the ScreenGlint prediction and the ground truth. The *x*-axis indicates the face-to-screen distance in cm. The error bar shows standard deviation. It can be seen that the closer the distance, the better the performance.

deviation. It is important to note that a smaller error in visual angle may correspond to a larger MAE in pixels, due to the impact of distance. Therefore, the MAE of the prediction displacement, or the distance between the predicted gaze point and the ground truth, on the screen reflects performance in a more intuitive manner. We also consider the angular error, $\arctan(E/D)$ [10][11], where E is the on-screen distance error and D is the face-to-screen distance.

Inspecting the results, we can conclude that a closer face-to-screen distance results in a more accurate gaze estimation. As shown in Figure 9, MAE rises and correlations drop as face-to-screen distance increases. The best performance occurs when the phone is in the closest position (25 cm), yielding an average MAE of 129 pixels (i.e. 2.56º visual error) and *x*- and *y*- correlations of 0.72 and 0.89. As the distance increases to 30cm and 40cm, performance deteriorates but stays stable. There is only a marginal difference between MAE at 30 cm (201 pixels; 2.70º) and 40 cm (204 pixels; 2.06º). This is an acceptable performance that is comparable to state-of-the-art (Table 1).

It is, however, a bit counter-intuitive that the overall *y*-correlation is higher than the *x*-correlation, although the MAE for the *y*-coordinate is generally larger than that of the *x*-coordinate. This may be because of the difference between the screen width (750 pixels) and height (1334 pixels). Because of this difference, an equivalent correlation would produce a larger MAE along the *y*-coordinate, since the *y*-axis is longer.

The average MAEs for the *x*- and *y*- coordinates across three positions are 111 and 148 pixels, respectively. Even the maximum MAE (165 pixels in y-coordinate at 30 cm) is smaller than the distance between the centers of two neighboring predefined points (180 pixels). The overall 2D MAE is 188 pixels (≈ 1.47 cm; 2.44º). This suggests that when head pose variation is not a factor, ScreenGlint's performance is acceptable. It is not hard to imagine that with

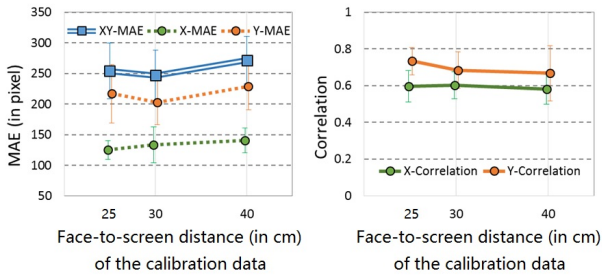

Figure 10. Performance of ScreenGlint under head pose variations. The *x*-axis indicates the face-to-screen distance in cm. Each point indicates the performance of the gaze model trained on the fixations on the 4 corners in the corresponding position, and tested on the rest of the data across all 3 positions. The overall 2D MAE is around 70 pixels higher than models without head pose bias.

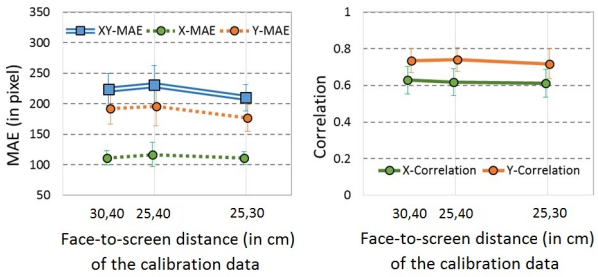

Figure 11. Performance of ScreenGlint with calibration data in 2 different positions. The *x*-axis indicates the face-to-screen distance. Each point in a particular distance indicates the performance of the gaze model trained on the other 2 positions, and tested on the rest of the data across all 3 positions. Compared with the models with single calibration position, the overall 2D MAE of models with 2 calibration positons reduces by 36 pixels on average.

proper post-filtering, ScreenGlint can be useful for gaze-aware applications, such as icon selection by fixation.

Gaze Estimation with Head Pose Variations

This section further investigates the gaze estimation performance under more realistic conditions, e.g. when the head pose variations exist. We use the points from the fixations on the 4 corners in one face-to-head position session for training, and the remaining data points (e.g. the rest of the fixation points with the same position session and those from the other 2 sessions) for testing. In other words, each time we calibrate with fixation points on 4 corners with the same head pose, and test on the rest of the data with both seen and unseen head poses. Repeating this with calibration points from each of the three face-to-screen positions gives us 3 models per experiment subject. The final performance result is the average of the 3 models across all subjects.

Figure 10 shows the MAE and the *x*- and *y*- correlations. As expected, the overall performance is slightly worse than that without head pose bias. The average 2D MAE (257 pixels; 3.38º) is around 70 pixels larger. Interestingly, in contrast to the finding presented in Figure 9 which showed that performance was better under closer face-to-screen conditions, the model trained on the data from the medium distance context yields the lowest MAE (246 pixels; 3.32º) when testing on data from all head positions. This may indicate that calibrating at a medium face-to-screen distance may be able to best compensate for the impact from variation in head poses. However, judging from the large standard deviations from all models, this strategy of calibration appears to make only a modest improvement.

We therefore investigate calibrating with multiple face-to-screen distances, which explicitly takes into consideration the impact of distance. We choose corner fixations from 2 different face-to-screen position sessions (8 corners in total, i.e. 4-point calibration in 2 positions) for training, and the rest of the data for testing.

Figure 11 presents the corresponding result. It is encouraging to see that the overall performance is improved by calibrating with 2 face-to-screen positions. Compared with the models calibrated in a single position, models calibrated in 2 positions produce stable accuracy across different positions, with smaller standard deviations. Furthermore, the average 2D MAE (221 pixels ≈ 1.72 cm; 2.94º) reduces by 36 pixels. More interestingly, there is a marked performance gain at the 40 cm point, where the MAE (209 pixels) is rather close to the lower bound MAE (204 pixels), i.e. when head pose variation is not considered. Since this data point corresponds to calibration data that is taken from the two sessions with closer face-to-screen distance, it corroborates our previous result, which suggests that fixation points taken in contexts with a smaller face-to-screen distance are more useful for gaze learning. This provides a useful guideline for ScreenGlint in real-use situations: the calibration should be conducted with the user's face at relatively close distances to the screen, but with two distinctly different head poses.

Table 1 shows the ScreenGlint performance, as compared against similar state-of-the-art gaze estimation methods that use the RGB camera as an input device. Although the other studies are based on different data sets, this still provides a useful reference with which to compare the performance across platforms. It can be seen that ScreenGlint is competitive with state-of-the-art, even on the desktop platforms. More importantly, ScreenGlint achieves this performance with only 4-point calibration in 2 positions. This is a promising result that bodes well for future integration into real-life applications.

Comparison with deep-learning approaches

Recent deep-learning-based approaches [13][35] have achieved impressive performances and offer an alternative approach. iTracker [13], trained on the GazeCapture dataset [13], achieves an error of 1.71cm~2.58cm in within-(GazeCapture) and cross-dataset (TabletGaze [11])

Methods	Platform	Error
ScreenGlint (ours)	Smartphone	2.94º
Wood et al. [30]	Tablet	6.88º
Holland et al. [9]	Tablet	3.95º
Huang et al. [11]	Tablet	2.86º-4.76º
Sugano et al. [23]	PC	2.9º
Huang et al. [10]	PC	2.56º
Lu et al. [14]	PC	2.49º-3.03º

Table 1. The performance of ScreenGlint, as compared against state-of-the-art gaze estimation techniques based on RGB camera across different platforms. ScreenGlint achieves competitive performance with best-performing techniques on other platforms.

evaluations. Including 4-point calibration improves the accuracy of iTracker to 1.65cm on the GazeCapture dataset. Although GazeCapture encourages subjects to change head pose during data collection, there is no systematic control that ensures head pose differences between the calibration and test dataset for each subject. In comparison, ScreenGlint yields around 1.47cm and 1.72cm distance error when calibrated with and without the same head pose as in the test set. In other words, even with limited user-specific calibrations, ScreenGlint achieves comparable accuracy with iTracker, which learns from large-scale data.

Deep learning provides a powerful tool for representation learning; however, recent CNN-based gaze learning work show conflicting information. Zhang et al. [35] suggest the use of only face image for gaze learning, while Krafka et al. [13] point out that eye image and face location in addition to the face image can be useful. Our study shows that the screen reflection can be indicative of gaze, in particular on smartphones. We hope that this finding can facilitate the model design and interpretation of future research.

A limitation of deep learning is the large amount of data required. A deep model that can literally work for everyone may require an infinite amount of diverse data and a large-scale architecture, which might be highly computational demanding, energy hungry and therefore unsuitable for mobile devices. ScreenGlint provides a user- specific method that is easily understood, reproduced, and can be improved by other researchers in the community, requiring much smaller datasets. We believe that our regression-based study as a competing technique offers a complementary understanding to CNN for optimal gaze representation and helps to maintain the diversity of the field.

CONCLUSION

This paper describes ScreenGlint, an approach that exploits the reflection of the screen for gaze estimation on smartphones. To understand the conditions that ScreenGlint would be expected to handle, we conducted a study investigating the face-to-screen distance that involves subjects performing a searching task on a smartphone, taking on 8 postures and under different illumination conditions. We find that the face-to-screen distance affects the size of the glint more significantly than environmental illumination, and the normal use of smartphone produces a face-to-screen distance of 25~40 cm.

In performance evaluation, we demonstrate that the accuracy of ScreenGlint can reach 2.44º (188 pixels ≈ 1.47 cm) without head pose variations, and 2.94º (221 pixels ≈ 1.72 cm) with head pose variations. This performance is competitive with the state-of-the-art. Our results also inform suggestions for the integration of ScreenGlint into real-use applications: the eye tracker should be calibrated in multiple positions to facilitate gaze learning and compensate for the face-to-screen variations.

There are some limitations to our work. First, our experiments were performed indoors, where the illumination conditions are more manageable. In outdoor use conditions, the extraction of the glint can be an issue, especially when there are multiple overlapping reflections. Second, the iris segmentation of ScreenGlint is computationally expensive and glint detection in the condition of low screen brightness may be problematic. A potential solution is to use some predefined flicker patterns of screen brightness to address the glint detection problem. Tracking techniques of computer vision can be also used to reduce the computational expense of iris and glint localization.

This paper opens avenues up for the gaze estimation on the smartphone platform. We see a variety of potential future work and applications based on ScreenGlint and its associated techniques.

ACKNOWLEDGMENTS

We sincerely thank the anonymous reviewers and area chairs for their time and effort reviewing this paper. This work is partially supported by grants PolyU 5222/13E and PolyU 152126/14E from the Hong Kong Research Grants Council and Hong Kong Polytechnic University respectively.

REFERENCES

1. Duchowski, A.T. *Eye Tracking Methodology: Theory and Practice.* Springer London, London, 2003. https://doi.org/10.1007/978-1-84628-609-4
2. Esteves, A., Velloso, E., Bulling, A., and Gellersen, H. Orbits: Gaze Interaction for Smart Watches using Smooth Pursuit Eye Movements. *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology - UIST '15*, (2015), 457–466. http://doi.acm.org/10.1145/2807442.2807499
3. Fischler, M. a. and Bolles, R.C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM 24*, (1981), 381–395. http://doi.acm.org/10.1145/358669.358692

4. Fitzgibbon, A.W. and Fisher, R.B. A Buyer ' s Guide to Conic Fitting. *British Machine Vision Conference*, (1995), 513–522. https://doi.org/10.5244/C.9.51
5. Guestrin, E.D. and Eizenman, M. General theory of remote gaze estimation using the pupil center and corneal reflections. *IEEE Transactions on Biomedical Engineering 53*, (2006), 1124–1133. http://doi.acm.org/10.1109/TBME.2005.863952
6. Hansen, D.W. and Ji, Q. In the eye of the beholder: a survey of models for eyes and gaze. *IEEE transactions on pattern analysis and machine intelligence 32*, 3 (2010), 478–500. http://doi.acm.org/10.1109/TPAMI.2009.30
7. Hennessey, C., Noureddin, B., and Lawrence, P. A single camera eye-gaze tracking system with free head motion. *Proceedings of the 2006 symposium on Eye tracking research & applications*, (2006), 27–29. http://doi.acm.org/10.1145/1117309.1117349
8. Hohlfeld, O., Pomp, A., Bitsch Link, J.Á., and Guse, D. On the Applicability of Computer Vision based Gaze Tracking in Mobile Scenarios. *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services - MobileHCI'15*, (2015), 427–434. http://dx.doi.org/10.1145/2785830.2785869
9. Holland, C., Garza, A., Kurtova, E., Cruz, J., and Komogortsev, O. Usability evaluation of eye tracking on an unmodified common tablet. *CHI "13 Extended Abstracts on Human Factors in Computing Systems on - CHI EA"13*, (2013), 295–300. http://doi.acm.org/10.1145/2468356.2468409
10. Huang, M.X., Kwok, T.C.K., Ngai, G., Chan, S.C.F., and Leong, H.V. Building a Personalized, Auto-Calibrating Eye Tracker from User Interactions. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, (2016), 5169–5179. http://doi.acm.org/10.1145/2858036.2858404
11. Huang, Q., Veeraraghavan, A., and Sabharwal, A. TabletGaze: Unconstrained Appearance-based Gaze Estimation in Mobile Tablets. *arXiv*, (2015). https://arxiv.org/abs/1508.01244
12. Iqbal, N. and Lee, S. A Study on Human Gaze Estimation Using Screen. *Proceedings of 9th International Conference Intelligent Data Engineering and Automated Learning – IDEAL 2008*, (2008), 104–111. https://doi.org/10.1007/978-3-540-88906-9_14
13. Krafka, K., Khosla, A., Kellnhofer, P., and Kannan, H. Eye Tracking for Everyone. *IEEE Conference on Computer Vision and Pattern Recognition*, (2016), 2176–2184. https://doi.org/10.1109/CVPR.2016.239
14. Lu, F., Sugano, Y., Okabe, T., and Sato, Y. Gaze Estimation From Eye Appearance: A Head Pose-Free Method via Eye Image Synthesis. *IEEE Transactions on Image Processing 24*, (2015), 3680–3693. https://doi.org/10.1109/TIP.2015.2445295
15. Nitschke, C., Nakazawa, A., and Takemura, H. Display-camera calibration using eye reflections and geometry constraints. *Computer Vision and Image Understanding 115*, (2011), 835–853. https://doi.org/10.1016/j.cviu.2011.02.008
16. Papoutsaki, A., Sangkloy, P., Laskey, J., Daskalova, N., Huang, J., and Hays, J. WebGazer: Scalable Webcam Eye Tracking Using User Interactions. *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, (2016), to appear.
17. Paul Viola, M.J. Robust Real-time Object Detection. *International Journal of Computer Vision*, (2001). https://doi.org/10.1023/B:VISI.0000013087.49260.fb
18. Rother, C., Kolmogorov, V., and Blake, A. "GrabCut": interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics 23*, (2004), 309–314. https://doi.org/10.1145/1015706.1015720
19. Rozado, D., Moreno, T., San Agustin, J., Rodriguez, F.B., and Varona, P. Controlling a Smartphone Using Gaze Gestures as the Input Mechanism. *Human–Computer Interaction 30*, (2015), 34–63. https://doi.org/10.1080/07370024.2013.870385
20. Schnieders, D., Fu, X., and Wong, K.Y.K. Reconstruction of display and eyes from a single image. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (2010), 1442–1449. https://doi.org/10.1109/CVPR.2010.5539799
21. Seeger, M. Gaussian Processes for Machine Learning. *International Journal of Neural Systems 14*, (2004), 69–106. https://doi.org/10.1142/S0129065704001899
22. Sesma-Sanchez, L., Zhang, Y., Bulling, A., and Gellersen, H. Gaussian processes as an alternative to polynomial gaze estimation functions. *Proceedings of the 2016 Symposium on Eye Tracking Research & Applications*, (2016), 229–232. https://doi.org/10.1145/2857491.2857509
23. Sugano, Y., Matsushita, Y., Sato, Y., and Koike, H. Appearance-Based Gaze Estimation With Online Calibration From Mouse Operations. *IEEE Transactions on Human-Machine Systems 45*, 6 (2015), 750 – 760. https://doi.org/10.1109/THMS.2015.2400434
24. Sugano, Y., Matsushita, Y., and Sato, Y. Appearance-based gaze estimation using visual saliency. *IEEE transactions on pattern analysis and machine intelligence 35*, 2 (2013), 329–341. https://doi.org/10.1109/TPAMI.2012.101
25. Swirski, L., Bulling, A., and Dodgson, N. Robust real-time pupil tracking in highly off-axis images. *Proceedings of the Symposium on Eye Tracking*

Research and Applications ETRA'12, (2012), 1–4. https://doi.org/10.1145/2168556.2168585

26. Vidal, M., Bulling, A., and Gellersen, H. Pursuits: spontaneous interaction with displays based on smooth pursuit eye movement and moving targets. *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing - UbiComp '13*, (2013), 439–448. https://doi.org/10.1145/2493432.2493477
27. Villanueva, A. and Cabeza, R. Models for gaze tracking systems. *Eurasip Journal on Image and Video Processing 2007*, (2007), 2:1–2:16. https://doi.org/10.1155/2007/23570
28. Wang, K., Wang, S., and Ji, Q. Deep eye fixation map learning for calibration-free eye gaze tracking. *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications - ETRA '16*, (2016), 47–55. https://doi.org/10.1145/2857491.2857515
29. Wood, E., Baltrušaitis, T., Morency, L.-P., Robinson, P., and Bulling, A. Learning an appearance-based gaze estimator from one million synthesised images. *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications - ETRA '16*, (2016), 131–138. https://doi.org/10.1145/2857491.2857492
30. Wood, E. and Bulling, A. EyeTab: Model-based gaze estimation on unmodified tablet computers. *Proceedings of the Symposium on Eye Tracking Research and Applications - ETRA '14*, ACM Press (2014), 207–210. https://doi.org/10.1145/2578153.2578185
31. Xiong, X. and De la Torre, F. Supervised Descent Method and Its Applications to Face Alignment. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE (2013), 532–539. https://doi.org/10.1109/CVPR.2013.75
32. Xu, P., Ehinger, K., and Zhang, Y. TurkerGaze: Crowdsourcing Saliency with Webcam based Eye Tracking. *arXiv preprint arXiv:*, (2015). https://arxiv.org/abs/arXiv:1504.06755
33. Zhai, S., Morimoto, C., and Ihde, S. Manual and gaze input cascaded (MAGIC) pointing. *Proceedings of the SIGCHI conference on Human factors in computing systems the CHI is the limit CHI 99*, (1999), 246–253. https://doi.org/10.1145/302979.303053
34. Zhang, X., Sugano, Y., Fritz, M., and Bulling, A. Appearance-Based Gaze Estimation in the Wild. *28th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015)*, (2015), 4511-4520. https://doi.org/10.1109/CVPR.2015.7299081
35. Zhang, X., Sugano, Y., Fritz, M., and Bulling, A. It's Written All Over Your Face: Full-Face Appearance-Based Gaze Estimation. (2016). https://arxiv.org/abs/1611.08860