Towards Around-Device Interaction using Corneal Imaging

Daniel Schneider
Coburg University
daniel.schneider@hs-coburg.de

Jens Grubert
Coburg University
jg@jensgrubert.de

ABSTRACT

Around-device interaction techniques aim at extending the input space using various sensing modalities on mobile and wearable devices. In this paper, we present our work towards extending the input area of mobile devices using front-facing device-centered cameras that capture reflections in the human eye. As current generation mobile devices lack high resolution front-facing cameras we study the feasibility of around-device interaction using corneal reflective imaging based on a high resolution camera. We present a workflow, a technical prototype and an evaluation, including a migration path from high resolution to low resolution imagers. Our study indicates, that under optimal conditions a spatial sensing resolution of 5 cm in the vicinity of a mobile phone is possible.

Author Keywords

around-device interaction; camera-based interaction; corneal imaging

ACM Classification Keywords

H.5.2. User Interfaces: Input devices and strategies

INTRODUCTION

Since over ten years, smartphones are a popular interaction medium, as they allow us to interact in a multitude of mobile contexts. However, small device sizes, aiming at increased mobility [10], can sacrifice the input space of those devices. If devices shrink, while fingers stay the same, interaction may become inefficient [29]. Hence, there is a need for compensating for the lack of physical interaction area.

One option is to extend the input space of interactive displays using sensors, leading to a decoupling between input and output space [13]. Various research has sparked in the area of around-device interaction, extending the input space to near-by surfaces or to mid-air. So far, most research focused on equipping either mobiles [3], the environment [26] or the user [5, 11, 12] with additional sensors. However, deployment of such hardware modifications is hard. Market size considerations discourage application developers, which limits technology acceptance in the real-world [4].

We envision a future in which *unmodified* mobile and wearable devices that are equipped with standard front-facing cameras allow for ample movements, including the environment around and to the sides of the device, without the need for equipping the device or the environment with additional sensing hardware. This could be achieved by utilizing the combination of built-in camera and the reflection in the human eye as lens-mirror (or catadioptric) system. Specifically, the environment of the user is reflected in the eye, which in turn can be captured by the device camera. However, to date, front-facing cameras typically do not posess the resolution needed for corneal imaging at typical interaction distances of handheld devices. Hence, in this paper, we investigate the feasibility of employing corneal imaging techniques with a mobile device-centred high-resolution camera. We present a pipeline for corneal imaging targeted at interaction with mobile devices and a user study investigating the performance of our approach under laboratory conditions. Besides looking at possible resolutions of future cameras, we also investigate a possible migration path up to resolution available in current mobile device cameras.

RELATED WORK

Corneal Imaging

Nishino and Nayar [21] used the term "corneal imaging system" to describe as catadioptric (mirror + lens) system with combination of the cornea of an eye, as mirror, and the camera capturing the appearance of the eye. Nishino and Nayar give a comprehensive analyses of the information about environment embedded in a single image of the eye. In this context, they describe several characteristics of a catadioptric system with cornea, as mirror, and camera. They also describe how to reconstruct a 3D object using both eyes. In contrast to the reconstruction using both eyes, we will only use a single eye for reconstruction in our evaluation to cover a wider interaction space. Nitschke et al. [24] give an overview of applications, which use corneal imaging in various settings. One uses the reflection to get information about the illumination of an Image [20]. Another possible application is espionage. With corneal imaging an image with the eye can show the content of screen a person looks at [2]. Further, a person's gaze direction and gaze point can be estimated [22]. Nitschke et al. [23] uses multiple images with various gaze directions to reconstruct an object and Schnieders et al. [28] uses the gaze direction of both eyes for the reconstruction. Typically, corneal imaging system focus on head-mounted eyet tracker, but applications for gaze estimation also have been proposed for mobile (e.g., [16]) and public displays (e.g., [15]) as well as smartwatches (e.g., [8]).
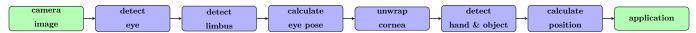
| camera image | → | detect eye | → | detect limbus | → | calculate eye pose | → | unwrap cornea | → | detect hand & object | → | calculate position | → | application |

Figure 1. Pipeline for calculation of positions (purple) with input and output (green).

Around-device Interaction

Along with the reduction of the size and weight of mobile and wearable devices, the need for complementary interaction methods evolved. While the available space on a mobile device is continuously shrinking, research began investigating options for interaction next to [25], above [18, 9], behind [7, 33], across [27, 6], or around [39, 36] the device. The additional modalities are either substituting or complementing the devices' capabilities. These approaches rely on modifying existing devices using a variety of sensing techniques, which can limit their deployment to mass audiences.

Recently, researchers started to investigate the use of unmodified devices. Nandakumar et al. [19] propose to use the internal microphones of mobiles to determine the location of finger movements on surfaces, but can not support mid-air interaction. Song et. al [31] enabled in-air gestures using the front and back facing cameras of unmodified mobile devices. However, their interaction space is limited to the field-of-view of the cameras, constraining the interaction space to two narrow cones in front and behind a device. Much of the interaction space around the mobile device, such as the areas to the sides of the device are not observed by these cameras. In contrast, our work focuses on a larger interaction space, covering positions around the device sides, as well as large hand gestures. With Surround See, Yang et al. [37] modified the front-facing camera of a mobile phone with an omnidirectional lens, extending its field of view to 360° horizontally. They showcased different application areas, including peripheral environment, object and activity detection, including hand gestures and pointing, but did not comment on the recognition accuracy.

The closest work to ours is GlassHands [12]. Grubert et al. demonstrated how the input space around an unmodified device can be extended, by using a built-in front-facing camera of an unmodified handheld device and some reflective glasses, like sunglasses, ski goggles or visors. The approach of Grubert et al. is to locate the hand and the handheld device in the reflection of the glasses and calculate their relative position in a common coordinate frame, hence virtually extending the touch screen of the mobile device. However, the usage of dark glasses in low-light environments may not be appropriated due to perceptual and social reasons. In this work, we propose to modify their original idea to integrate corneal imaging techniques. In our work, the replacement of the mirror (from glasses to the cornea) is the first major difference. Due to different forms of the reflection area the approach to detect these are also different (Grubert et al used a simple thresholding approach for detecting the glass area). For unwrapping Grubert et al. proposed a special calibration target and process for unwrapping images from curved glasses. The cornea model we use makes this calibration step redundant.

CONCEPT AND IMPLEMENTATION

Our approach aims at detecting the position of the user's hand relative to a mobile handheld device such as a smartphone. This way, the hand can be used as input device in the vicinity of the device. To calculate the position of those objects of interest in a reflected area, one needs to be able to detect the reflection area, remove the distortion from the shape of the cornea, detect objects (in our case a handheld device and a hand), and, finally, reconstruct the relative position of those objects in a common coordinate frame. The workflow is summarized in Figure 1. First, an input image is searched for an eye, then the limbus, the contour of the cornea, and, therefore, the contour of the reflected area, is detected and the eye pose and location relative to the camera is calculated. Next, this information is used for unwrapping the the corneal region, and, afterwards, analyze the scene based on the resulting image. In our case, we search for objects of interest (such as a hand or a finger) in the vicinity of a smartphone and determine their relative positions in a common coordinate frame. This data is finally passed to an application which makes use of it.

Figure 2 provides an overview of the process data. The content of the pictures are related to the laboratory study presented below. The input data (original image) is shown in Fig. 2 (a). Fig. 2 (b) shows the eye area for the right eye and is the result of the eye detection. The result of the limbus detection is depicted in Fig. 2 (c). In Fig. 2 (d), the distortion of the cornea is partial compensated during the unwrapping step. Objects of interest Fig. 2 (e) are detected and their relative distance is calculated (Fig. 2 (f)).

The first step of the pipeline is the detection of the eyes. For automatic eye detection the approaches of Kothari and Mitchell [17] or Kawaguchi et al. [14] can be used. Please note, that these approaches assume a full face to be visible in the image.

The next step of the the pipeline is the limbus detection. We are using a RANSAC-based algorithm by Wood and Bulling [35] to detect the limbus as an ellipse in the image. The detected limbus is improved by predefined sample-sets of the RANSAC algorithm.

To calculate the pose of the eye, we use an eye model of two intersecting spheres from Nitschke et al. [24]. For the corneal reflection, the sphere of the model containing the cornea is used. One pole of the sphere lies at the center of the pupil. Alternatively, more advanced eye models can be used (e.g., [34]). As proposed by Nishino and Nayar [21], the position and orientation of the eye can be calculated with the limbus center. First, the distance between the camera center of projection and the center of the limbus has to be determined. Under the assumption of weak perspective projection, the limbus in 3D space is rst orthographically projected onto the average depth plane, which is parallel to the image plane and passes through the center of the limbus. The limbus in the projection plane is an ellipse. With the orientation of the ellipse and

the ratio of major and minor axis, representing the orientation of the pol-axis, the center of the sphere can be reconstructed relative to the camera.

With the eye pose and the location of the eye, the pixel of the reflection in the image can be projected back to the surface of the mirror (cornea). In order to process the surface of the cornea further, the texture is unwrapped. For the unwrapping, we use a equirectangular projection [30]. This type of projection has higher distortions in the polar region than around the equator. To minimize distortions, we rotate our eye model $90°$ around the x-axis, so the cornea is no longer the polar region of the. These rotation must be taken into account and reversed during further processing.

Then, scene analysis algorithms can be applied to this unwrapped image.We detect the smartphone display and the fingers in a similar manner as in GlassHands [12]. For interactive purposes, we aim at reconstruction the relative positions of the display and the hand in a common coordinate frame (the display plance of the mobile device, virtually extended around the device). We assume that the interaction takes place in this virtual plane.

For the reconstruction of the plane with mobile device and hand, it is assumed that the mobile device can be approximated by a rectangular shape and the projection rays from the pinhole of the camera are parallel. The first step in the reconstruction, is the calculation of the left (L), center (C) and right (R) points of the object on the cornea are needed. With the reflection, these points are projected to the points L_E, C_E and R_E on the plane. The width of the real object is the distance between L_E and R_E. Using the distance between L and R, the distance between L_C and R_C can be calculated. The same applies for the point L_R and the distance from this to R. The distance between R_C and L_M is the same as L_E and L_C. With the ratio of the distance between R_C and L_M and the distance between R_C and R_E the distance to the plane can be calculated.

To calculate the orientation of the plane, the ratio of width and height between the image and physical dimensions are used. The center of the mobile device is the origin in the object plane. The coordinates of the hand will be calculated via intersection of the reflection vector and the plane, and, then, transformed into plane coordinates.

If a metric distance is not needed between smartphone and hand (e.g. for spatial gestures), one could, alternatively, determine the relative distances in pixel space, and, potentially, skip the unwrapping and reconstruction steps.

The pipeline was implemented in C++ and OpenCV was used for image processing functions.

EVALUATION

To determine the accuracy of the position estimation of the algorithm, and, thus, to answer the question whether it is suitable for the interaction with the mobile device, a laboratory study was carried out. In this study, a highly controlled environment is selected to present the usage under optimal conditions.
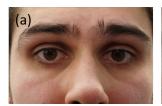
Design and Apparatus

We aimed for descriptive measures of our approach, both under optimal and degraded conditions. To this end, we varied the image resolution and used two detection scenarios (see Figure 4). The focus in the first scenario (condition *RECT*) was the position estimation for reflected objects under optimal conditions. To this end, we opted for easily detectable colored shapes. The objects used in the study are a red rectangle (7 cm by 14 cm), which represents a mobile device and a blue square (edge length of 2 cm) as a replacement for a finger (see Figure 4, left). During the study, the distances between the midpoints of the two objects were measured. The distance was divided into an X and a Y component. The X axis is parallel to the narrow edge (7 cm edge) of the rectangle and the Y axis is parallel to the large edge (14 cm edge). To further optimize object detection, a black background was used (high contrast to red and blue object). For a better lightning the flash of the camera was used.

While still under laboratory conditions, the second scenario (condition *FINGER*) used actual objects that could be used in natural contexts (see Figure 4, right). These were a Sony Z3 (display size of 6.4 cm by 11.4 cm, black case) showing a white image (following the assumption of Grubert et al. that often the mobile device screen can be the brightest part in a scene [12]) and a fist with straightened index finger, which was illuminated by an external light to counterbalance shadows from a chin rest used in the study. During the study, the distance between the smartphone center and the top of the index finger were measured (replicating scenario 1).
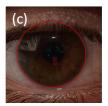
We envision future built-in cameras to have high resolution imagers (or alternatively, dual wide angle and telephoto lenses as in the Samsung Galaxy Note 8), and, hence, did not want to limit ourselves by today's available front-facing smartphone cameras. Instead, for the recordings, a Sony $alpha$ 7r camera with 36 megapixel sensor and macro lens Sony FE 90mm f / 2.8 Macro G OSS was used. The camera was placed such that the nodal point would coincide approximately with the position of a possible built-in camera (slightly above to still allow object detection).

We invited ten people (2 female and 8 male, mean age of 25 years, sd=2.4, variety of iris textures) to participate in the study. After an introduction phase, participants were asked to position their chin on a pedestal at ca. 45 cm distance (mimicking a typical interaction distance for handheld interaction [1]), put on an empty glass frame with a marker and were instructed to look into the camera.
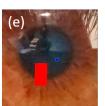
The picture taken with the camera captures parts of the face (see Figure 2 (a)), and, therefore, the exisiting algorithms ([14, 17]) mentioned above can not be used for automatic eye detection in our study setup. Instead, we first use the marker to roughly determine the eye region and then detect and center the real eye region by locating the pupil via thresholding, contour detection and ellipse fitting. During the data acquisition, the subjects were asked to keep still and not to change their position. If the participants moved, the data was discarded and positioning and measurements were repeated. In the first part, the blue square and in the second part the partic-
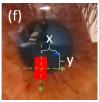
Figure 2. Results of single steps of the pipeline, (a) original camera picture, (b) eye region, (c) eye region with limbus, (d) Unwrapping result, (e) Object detection (red and blue rectangle) (f) distance of objects.

ipants fist and index finger were positioned at nine different locations. The positions are divided into three columns and three lines to the right of the red rectangle. These are parallel to the X and Y axes. The columns are at a distance 10 cm, 20 cm and 30 cm from the center, and the rows are offset 10 cm up, 0 cm and 10 cm downwards. Four images were taken for each position. After completing the recording, the participants were thanked for their participation in the test. The test setup for both condition are shown in Figure 4. With these setup there are a total of 360 samples (10 participants x 9 locations x 4 repetitions) for each scenario.

Currently, front cameras in mobile phones do not provide resolutions of our employed camera system. To provide a migration path between high and low resolution imagers, the algorithm is applied to images of different sizes. For this purpose, the height and width of the images are scaled by a factor (0.5, 0.25, 0.125). The camera produces images with a resolution of 7360x4912 px. In these images the eye region is 1000x1000 px. The eye region for factor 0.5 is 500x500 px, for 0.25 250x250px and for 0.125 125x125 px, a resolution achievable with today's cameras. For example, a 2 megapixel smartphone camera in 20 cm distance or a 8 megapixel smartphone camera in 40 cm distance, with wide angle lenses, produce images with eye regions of ca. 125x125 px.

Results

The scatter plot with the position of all calculated and expected positions is shown in Figure 5 for condition *RECT* and in Figure 6 for condition *FINGER*. For condition *RECT*, the root mean square error (RMS) for the full resolution image over all points was 40.65 mm (sd=28.13) with a 99.53% detection rate. For 500x500 px, it was 42.62mm
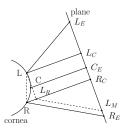

Figure 3. Schematic view from cornea and reconstruction plane. Mobile device on cornea has the left (L), center (C) and right point (R). Mobile device on plane has the left (L_E), center (C_E) and right point (C_E). Projection with $C\vec{C}_E$ from L to L_C and R to R_C on plane. L_R and L_M are mirrored points of L and L_E on line C C_E

.

Figure 4. Setup of the laboratory study. Left: Condition *RECT* with red rectangular and blue marker. Right: Condition *FINGER* with smartphone and finger.

(sd=30.87, 95.28% detection rate), for 250x250 pixel, 42.22 mm (sd=30.60, 72.17% detection rate) and for 125x125 px, 44.53 mm (sd=27.57, 61.08% detection rate), see Figure 5.

As we witnessed a sharp decline in location accuracy at a horizontal distance of 30 cm, we also calculated results for closer interaction distances. The RMS error for the full resolution image with an expected horizontal interaction distance x=100 mm and y=0 mm is 16.38 mm (sd=7.79). For a distance range of y from -100 mm and 100 mm at an x distance of 100 mm the mean location accuracy is 21.51 mm (sd=12.22). For an x range of 100 to 200 mm and y in range of -100 mm and 100 mm is the mean location accuracy is 30.48 mm (sd=19.09).

For condition *FINGER*, the RMS error for the full resolution image over all points is 41.11 mm (sd=24.02, 93.17% detection rate), for 500x500 px, 41.25 mm (sd=23.94, 89.76% detection rate), for 250x250 px, 45.18 mm (sd=26.29, 86.59% detection rate) and for 125x125 px 62.90 mm (sd=30.60, 57.32% detection rate), see Figure 6. As in the previous scenario, we also looked at closer interaction ranges. The RMS error for the full resolution image with expected x=100 mm and y=0 mm is 16.12 mm (sd=9.06), for x=100 mm and y in range of -100 mm and 100 mm is 33.13 mm (sd=20.41). For x in the range of 100 to 200 mm and y in range of -100 mm and 100 mm the mean location accuracy is 35.90 mm (sd=21.48).

It is also possible to explain the behavior over the resolutions (see Figure 7 and Figure 8). At the largest resolution, a pixel corresponds to approximately $0.12°$. The smaller the resolution, the larger the space that is projected onto a single pixel. The smallest resolution (125x125 px) results in a pixel corresponding to approximately $1°$. In addition to the direct distance, the calculation of the projection of the red rectangle (for the construction of the plane) leads to considerable fluctuations (similar for the y-direction).
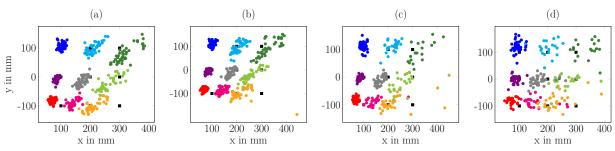
Figure 5. Scatter plot for all resolutions of eye regions of condition *RECT*. Resolution of eye region is (a) 1000 x 1000, (b) 500 x500, (c) 250 x 250, (d) 125 x 125. The color marks the aimed positions (x, y): red (100, -100), magenta (200, -100), orange (300, -100), violet (100, 0), gray (200, 0) light green (300, 0), blue (100, 100), light blue (200, 100) and green (300, 100).
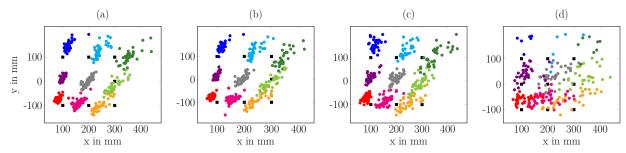

Figure 6. Scatter plot for all resolutions of eye regions of condition *FINGER*. Resolution of eye region is (a) 1000 x 1000, (b) 500 x500, (c) 250 x 250, (d) 125 x 125. The color marks the aimed positions (x, y): red (100, -100), magenta (200, -100), orange (300, -100), violet (100, 0), gray (200, 0) light green (300, 0), blue (100, 100), light blue (200, 100) and green (300, 100).
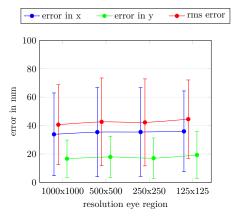

Figure 7. Errors (x, y, RMS) across all resolutions for condition 1.
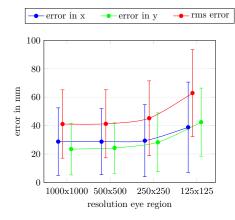

Figure 8. Errors (x, y, RMS) across all resolutions for condition 2.

The algorithm was executed on a ThinkPad S531 with a Intel Core i7-3537U processor (2 cores, 2.0GHz, 4MB cache) and 10 GB DDR3 RAM. We measured the time the prototype needed to calculate the position of the objects for all tested resolutions. For condition *RECT* and an eye region with resolution of 1000x1000 pixel the mean time to complete the task over all tests is 137 milliseconds (sd=41; eye detection 31 ms, limbus detection 35 ms, unwrapping 62 ms, scene analysis 8 ms). For the resolution with 500x500 pixel average processing time is 68 milliseconds (sd=11; eye detection 27 ms, limbus detection 24 ms, unwrapping 13 ms, scene analysis 2 ms), for the resolution of 250x250 pixel it is 61 milliseconds (sd= 13; eye detection 27 ms, limbus detection 30 ms, unwrapping 3 ms, scene analysis 0.7 ms) and for 125x125 pixel the average time is 63 milliseconds (sd=10; eye detection 26

ms, limbus detection 33 ms, unwrapping 3 ms, scene analysis 0.6 ms). For condition *FINGER* the mean processing time is 287 milliseconds (sd=110; eye detection 31 ms, limbus detection 37 ms, unwrapping 63 ms, scene analysis 154±100 ms) for the full resolution, for the eye region of 500x500 pixel it is 140 milliseconds (sd=48; eye detection 31 ms, limbus detection 27 ms, unwrapping 14 ms, scene analysis 66 ms), for 250x250 pixel 92 milliseconds (sd=22; eye detection 29 ms, limbus detection 32 ms, unwrapping 3 ms, scene analysis 26 ms) and for 125x125 pixel it is 111 milliseconds (sd=27; eye detection 33 ms, limbus detection 43 ms, unwrapping 4 ms, scene analysis 31 ms). The scene analysis in condition *FINGER* takes longer due to the more complex algorithm. For the detection of the hand and finger an iterative flood-fill algorithm is used. In each iteration the colors are adapted until

the hand is detected. The resolution of the image processed depends on the individual steps. For eye detection the image resolution is fixed to 960x614 pixel . For the limbus detection the image (eye region) is scaled to 125x125 pixel. The unwrapping and object detection step work with the original resolution, with exception of the smallest. To improve the success rate of the object detection these images are scaled up to a size of 250x250 pixel for the eye region.

DISCUSSION AND CONCLUSION

The evaluation has indicated that, under optimal conditions, in a region of 20 cm next to and 10 cm above or below the center point of a mobile device, positions between two sensed objects (such as a finger) could be distinguished if these are approximately 5 cm apart. Please note, that both conditions took place under optimal laboratory conditions. This was done on purpose to get a best case estimate. Future work should validate these findings in more ecological valid settings. To this end, the scene analysis step would need further adoption to increase robustness to different lighting conditions.

In both scenarios, the error increases with the x-coordinate (distance between the two objects in x-direction). This behavior can be partly explained due to the curvature of the reflection, which results at lower image resolutions for eye surface regions facing away from the view direction. Further, with decreasing resolution of the eye region, the detection rate drops dramatically. For example, the blue marker in condition *RECT* has a size of about 6x6 pixel at full resolution. At other resolutions, the color is already mixed with the surrounding background or iris texture. The second condition has the same problem. Here the width of the index finger of the participants were typically even smaller then the width of the blue marker.

A further deviation comes from the fault-prone object recognition, which, albeit the colored rectangles in condition *RECT*, amplifies errors due to pixel-wise offsets in the position detection. For example, at 5 x 10 pixel ground truth size, by detecting 6 x 10 pixels, deviations of 20% in x-direction and 10% in y-direction result. With more complex objects (hand) or other environments, the detection error might increase. One way to potentially aid object recognition, is the inclusion of iris texture removal algorithms as a pre-processing step before scene analysis (e.g.,[32]). Besides required resolution, a limitation of our pipeline is that the algorithm is not directly optimized for execution on mobile devices.

Another limitation may be the currently assumed optimal gaze direction. With head or eye movements present, the objects may be projected towards outer regions of the cornea or even sclera. This in turn can lead to challenging object detection (due to larger overlap with iris texture) and to higher uncertainties in object localization (due to a stronger oval shape of the cornea in image space, which leads to a lower image space resolution). However, under the assumption that users look at the screen while interacting, the deviations from our setup can be small.

Also, since the detection of the limbus is based on a selection of random values, this results in strong fluctuations of the results. The algorithm we employed [35] has an accuracy of 7° for the gaze direction, other state of the art approaches (c.f., [38]) can achieve an accuracy of 5° and could be applied in the future.

Finally, the use of the rectangular projection leads to distortions in the image. Another projection or better rotation during unwrapping can also provide for improvement.

To summarize, we work towards extending the input area of mobile devices using front-facing device-centered cameras that capture reflections in the cornea. To this end, we adapted corneal imaging techniques for the use in an around-device interaction pipeline. We studied the feasibility of around-device interaction using corneal reflective imaging based on a high resolution camera but also investigated the performance on lower resolution images. Our results indicate, that under optimal conditions around-device sensing could be performed with a positional resolution of ca. 5 cm. In future work, we want to optimize this approach to work on mobile phones with built-in high resolution (4k, 8k) cameras and increase the robustness in real-world conditions.

REFERENCES

1. Bababekova, Y., Rosenfield, M., Hue, J. E., and Huang, R. R. Font size and viewing distance of handheld smart phones. *OaVS 88*, 7 (2011), 795–797.

2. Backes, M., Chen, T., Dürmuth, M., Lensch, H. P., and Welk, M. Tempest in a teapot: Compromising reflections revisited. In *30th IEEE SSaP*, IEEE (2009), 315–327.

3. Butler, A., Izadi, S., and Hodges, S. Sidesight: Multi-"touch" interaction around small devices. In *Proc. UIST '08*, ACM (2008), 201–204.

4. Buxton, B. The long nose of innovation. *Insight 11* (2008), 27.

5. Chan, L., Hsieh, C.-H., Chen, Y.-L., Yang, S., Huang, D.-Y., Liang, R.-H., and Chen, B.-Y. Cyclops: Wearable and single-piece full-body gesture input devices. In *Proc. CHI '15*, ACM (2015), 3001–3009.

6. Chen, X., Grossman, T., Wigdor, D. J., and Fitzmaurice, G. Duet: exploring joint interactions on a smart phone and a smart watch. In *Proc. CHI '14*, ACM (2014), 159–168.

7. De Luca, A., von Zezschwitz, E., Nguyen, N. D. H., Maurer, M.-E., Rubegni, E., Scipioni, M. P., and Langheinrich, M. Back-of-device authentication on smartphones. In *Proc. CHI '13*, ACM (2013), 2389–2398.

8. Esteves, A., Velloso, E., Bulling, A., and Gellersen, H. Orbits: enabling gaze interaction in smart watches using moving targets. In *Proc. UbiComp/ISWC'15 Adj.*, ACM (2015), 419–422.

9. Freeman, E., Brewster, S., and Lantz, V. Towards usable and acceptable above-device interactions. In *Proc. MobileHCI '14*, ACM (2014), 459–464.

10. Gemperle, F., Kasabach, C., Stivoric, J., Bauer, M., and Martin, R. Design for wearability. In *Wearable Computers, 1998. Digest of Papers. Second International Symposium on* (Oct 1998), 116–122.

11. Grubert, J., Heinisch, M., Quigley, A., and Schmalstieg, D. Multifi: Multi fidelity interaction with displays on and around the body. In *Proc. CHI '15*, ACM (2015), 3933–3942.

12. Grubert, J., Ofek, E., Pahud, M., Kranz, M., and Schmalstieg, D. Glasshands: Interaction around unmodified mobile devices using sunglasses. In *Proc. ISS '16*, ACM (2016), 215–224.

13. Harrison, C. Appropriated interaction surfaces. *Computer 43*, 6 (2010), 0086–89.

14. Kawaguchi, T., Rizon, M., and Hidaka, D. Detection of eyes from human faces by hough transform and separability filter. *Electronics and Communications in Japan (Part II: Electronics) 88*, 5 (2005), 29–39.

15. Khamis, M., Alt, F., and Bulling, A. Challenges and design space of gaze-enabled public displays. In *Proc. UbiComp/ISWC'16 Adj.*, ACM (2016), 1736–1745.

16. Khamis, M., Alt, F., Hassib, M., von Zezschwitz, E., Hasholzner, R., and Bulling, A. Gazetouchpass: Multimodal authentication using gaze and touch on mobile devices. In *Proc. CHI EA '16*, ACM (2016), 2156–2164.

17. Kothari, R., and Mitchell, J. L. Detection of eye locations in unconstrained visual images. In *Proc. ICIP '96*, vol. 3, IEEE (1996), 519–522.

18. Kratz, S., and Rohs, M. Hoverflow: exploring around-device interaction with ir distance sensors. In *Proc. MobileHCI '09*, ACM (2009), 42.

19. Nandakumar, R., Iyer, V., Tan, D., and Gollakota, S. Fingerio: Using active sonar for fine-grained finger tracking. In *Proc. CHI '16*, ACM (2016), 1515–1525.

20. Nishino, K., and Nayar, S. K. Eyes for relighting. In *ACM TOG*, vol. 23, ACM (2004), 704–711.

21. Nishino, K., and Nayar, S. K. Corneal imaging system: Environment from eyes. *IJCV 70*, 1 (2006), 23–40.

22. Nitschke, C., Nakazawa, A., and Nishida, T. I see what you see: point of gaze estimation from corneal images. In *Proc. ACPR '13*, IEEE (2013), 298–304.

23. Nitschke, C., Nakazawa, A., and Takemura, H. Display-camera calibration using eye reflections and geometry constraints. *Computer Vision and Image Understanding 115*, 6 (2011), 835–853.

24. Nitschke, C., Nakazawa, A., and Takemura, H. Corneal imaging revisited: An overview of corneal reflection analysis and applications. *IPSJ Transactions on Computer Vision and Applications 5* (2013), 1–18.

25. Oakley, I., and Lee, D. Interaction on the edge: Offset sensing for small devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, ACM (New York, NY, USA, 2014), 169–178.

26. Rädle, R., Jetter, H.-C., Marquardt, N., Reiterer, H., and Rogers, Y. Huddlelamp: Spatially-aware mobile displays for ad-hoc around-the-table collaboration. In *Proc. ITS '14*, ACM (2014), 45–54.

27. Schmidt, D., Seifert, J., Rukzio, E., and Gellersen, H. A cross-device interaction style for mobiles and surfaces. In *Proc. DIS '12*, ACM (2012), 318–327.

28. Schnieders, D., Fu, X., and Wong, K.-Y. K. Reconstruction of display and eyes from a single image. In *Proc. CVPR '10*, IEEE (2010), 1442–1449.

29. Siek, K., Rogers, Y., and Connelly, K. Fat finger worries: How older and younger users physically interact with pdas. In *Human-Computer Interaction - INTERACT 2005*, M. Costabile and F. Patern, Eds., vol. 3585 of *LNCS*. Springer Berlin Heidelberg, 2005, 267–280.

30. Snyder, J. P. *Flattening the earth: two thousand years of map projections*. University of Chicago Press, 1997.

31. Song, J., Sörös, G., Pece, F., Fanello, S. R., Izadi, S., Keskin, C., and Hilliges, O. In-air gestures around unmodified mobile devices. In *Proc. UIST '14*, ACM (2014), 319–329.

32. Wang, H., Lin, S., Ye, X., and Gu, W. Separating corneal reflections for illumination estimation. *Neurocomputing 71*, 10 (2008), 1788–1797.

33. Wigdor, D., Forlines, C., Baudisch, P., Barnwell, J., and Shen, C. Lucid touch: A see-through mobile device. In *Proc. UIST '07*, UIST '07, ACM (New York, NY, USA, 2007), 269–278.

34. Wood, E., Baltrušaitis, T., Morency, L.-P., Robinson, P., and Bulling, A. A 3d morphable eye region model for gaze estimation. In *ECCV*, Springer (2016), 297–313.

35. Wood, E., and Bulling, A. Eyetab: Model-based gaze estimation on unmodified tablet computers. In *Proc. ETRA '14*, ACM (2014), 207–210.

36. Xiao, R., Lew, G., Marsanico, J., Hariharan, D., Hudson, S., and Harrison, C. Toffee: enabling ad hoc, around-device interaction with acoustic time-of-arrival correlation. In *Proc. MobileHCI '14*, ACM (2014), 67–76.

37. Yang, X.-D., Hasan, K., Bruce, N., and Irani, P. Surround-see: enabling peripheral vision on smartphones during active use. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, ACM (2013), 291–300.

38. Zhang, X., Sugano, Y., Fritz, M., and Bulling, A. It's written all over your face: Full-face appearance-based gaze estimation. *arXiv preprint arXiv:1611.08860* (2016).

39. Zhao, C., Chen, K.-Y., Aumi, M. T. I., Patel, S., and Reynolds, M. S. Sideswipe: Detecting in-air gestures around mobile devices using actual gsm signal. In *Proc. UIST '14*, ACM (New York, NY, USA, 2014), 527–534.