GlassHands: Interaction Around Unmodified Mobile Devices Using Sunglasses

Jens Grubert[1,2]**, Eyal Ofek**[3]**, Michel Pahud**[3]**, Matthias Kranz**[2]**, Dieter Schmalstieg**[4]
[1]Coburg University [2]University of Passau [3]Microsoft Research [4]Graz University of Technology
jg@jensgrubert.de, eyalofek@microsoft.com, mpahud@microsoft.com, matthias.kranz@uni-passau.de,
schmalstieg@tugraz.at

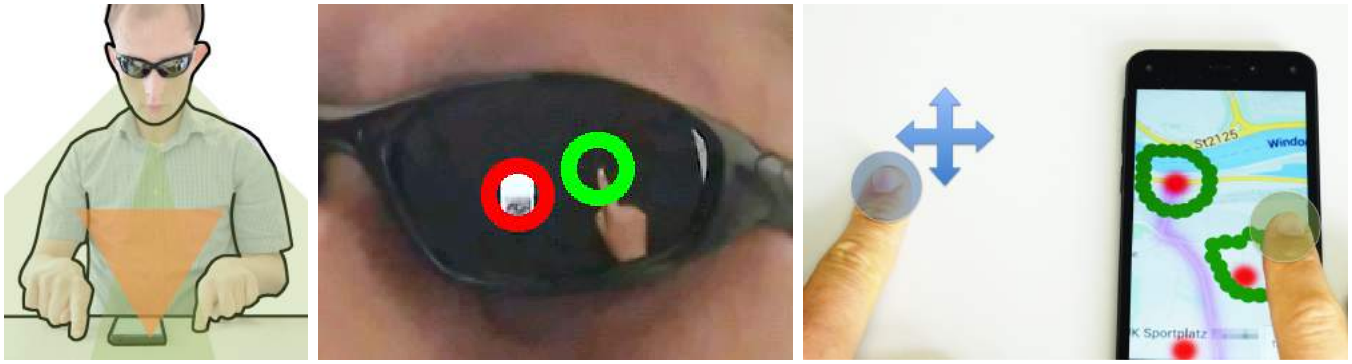


Figure 1. GlassHands extends the input space around mobile devices. Left: The narrow field-of-view of front facing cameras (orange) is extended through sunglasses reflections (green). Middle: Detected mobile phone (red) and finger tip (green) in glass area. Right: Users can continuously pan outside the display while simultaneously tracing over items on the device screen ©Jens Grubert.

ABSTRACT

We present a novel approach for extending the input space around unmodified mobile devices. Using built-in front-facing cameras of unmodified handheld devices, GlassHands estimates hand poses and gestures through reflections in sunglasses, ski goggles or visors. Thereby, GlassHands creates an enlarged input space, rivaling input reach on large touch displays. We introduce the idea along with its technical concept and implementation. We demonstrate the feasibility and potential of our proposed approach in several application scenarios, such as map browsing or drawing using a set of interaction techniques previously possible only with modified mobile devices or on large touch displays. Our research is backed up with a user study.

ACM Classification Keywords

H.5.2 [Information interfaces and presentation]: User Interfaces - Graphical user interfaces

INTRODUCTION

Handheld and wearable touch displays allow us to interact in a multitude of mobile contexts. However, shrinking device sizes, aiming at increased mobility [13], often sacrifice the interactive surface area. If devices shrink, while fingers stay the same, interaction may become inefficient. Hence, there is a need for compensating for the lack of physical interaction area.

One option is to decouple input and output area of interactive displays, using sensors to increase the input area around devices [16], extending it to near-by surfaces or to mid-air. Numerous research has sparked in the area of around-device interaction. So far, most research focused on equipping either mobiles [4], the environment [33] or the user [6, 14] 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 [5].

We present an approach that extends the input area of *unmodified* mobile devices to allow ample movements, including the environment around and to the sides of the device, without any additional sensing hardware. To do so, we propose to enrich the sensing capabilities of unmodified mobiles by everyday common apparels such as sunglasses or common reflective visors.

Our interactive system, called *GlassHands* (Figure 1), utilizes reflective glasses or visors to extend the field-of-view of front-facing cameras built into mobile devices, mimicking effects of catadioptric panorama cameras. Other reflective surfaces, such as ski goggles, diving masks or helmet visors, may enable gesture interaction with phones as well. This is of interest, when fine interaction with small screen is dangerous or impossible, for example, when the hands are covered with gloves. In fact, for some scenarios, such as skiing, reflective visors are so common, that a software-only deployment may also be economically feasible.

We hope that the suggested technology, enabling off-device sensing with a large interaction space without the need for any hardware changes, will open up the opportunity to deploy software-only applications to millions of existing mobile devices. All that is required from users is access to common apparel.

RELATED WORK

GlassHands is inspired by previous works on around-device interaction, around-the-body interaction and corneal imaging. In this section, we give an overview of these topics. In contrast to research extending the input and output of stationary displays [1, 26, 43], we focus on related work in *mobile scenarios*.

Around Device Interaction

Along with the reduction of the size and weight of mobile and wearable devices, the need for complementary interaction methods evolved. Research began investigating options for interaction next to [30], above [12, 23], behind [10, 42], across [9, 36], or around [45, 48] 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 severely limits their deployment to mass audiences.

Surface Interaction

Appropriating *surfaces* around mobiles was investigated in several works. Butler et al. [4] showed compelling multi-touch input around a mobile phone lying on a flat surface, by equipping them with additional side-facing infrared sensors. In contrast, the proposed concept does not require to modify the mobile device and enables a larger input space. Avrahami et al. [3] extended the input space around tablets by using downward facing cameras on a stand. Their system was specifically designed to be portable, not mobile, as it requires a certain setup effort. Our approach can be used without prior setup time.

Harrison and Hudson [16] presented several approaches to appropriate surfaces for extending the input (and output) of mobiles. Recently, it was shown how acoustic sensing could be used to detect touch events on a hard surface around a device, using additionally mounted piezo sensors [45]. Several works also investigated how to extend touch input from touch screens to the human skin [25, 31, 41].

Free Space Interaction

Mobiles and wearables have been equipped with a variety of sensors to extend their input to *free space* gestures. Depth-sensors have been used to enhance interaction in front [23], at the side [37] or at the back [24] of mobile devices, enabling combined touch and free space interaction [7].

Magnetic sensing was used to extend the input space of mobile devices [2, 17]. Those approaches inspired us, as they allow the use of unmodified devices with built-in magnetometers, and instead rely on user worn magnets. However, these approaches mostly focus on in-air gestures or very coarse pointing in close proximity of the device [17].

Other approaches include equipping devices with additional cameras to extend the interaction space [3, 4].

Song et. al [38] 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 (see figure 1, left). In contrast, our work focuses on a larger interaction space, covering positions around the device sides, as well as large hand gestures.

The closest work to ours is Surround See by Yang et al. [46]. They 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.

Their approach, just like ours, supports a large interactive space around the device. The need to add non-standard hardware to the phone limits the deployment of this technology. Furthermore, the 360° lenses used by Yang et al. increase the size of the device thickness, making it hard to access and store a mobile device. In contrast, our approach only requires access to common and widely available apparels, which can result in a software only deployment to enable around-device interaction on millions of existing mobile devices.

Some approaches use stationary tracking systems to explore around-device interactions. Hasan et al. [19] presented AD-Binning, a technique for off-loading mobile content to the space around a device using finger movements. Jones et al. [22] explored free space interaction techniques for multi-scale navigation on mobile devices.

Around the Body Interaction

Similar to extending the interaction space around a device, researchers have investigated the interaction space around the human body. Wagner et al. [40] presented a body-centric design space for classifying multi-surface interaction techniques.

Chen et al. [8] studied how to appropriate body parts and free space around the users in a number of prototypes, including built-in sensors on a mobile phone. Grubert et al. [14] explored how to complement the input and output of multiple body-proximate wearable and mobile displays such as smartwatches and head-mounted displays. A number of works investigated re-purposing everyday objects and surfaces for input and output, turning them into opportunistic tangible user interfaces around the user [18, 20].

Corneal Reflective Imaging

Our approach to utilize reflections in the environment for camera-based sensing has been inspired by corneal imaging techniques [29]. The main idea thereof is to capture, unwarp and analyze the reflections in a human eye using catadioptric camera models [39].

We speculate that, in the future, using 4K or higher resolution front cameras, we may use corneal reflection instead of reflective glasses. Unfortunately, today's mobile phone sensors still do not deliver enough resolution for the practical use of those techniques, however, given the progress of camera technologies, it is not unlikely that such phones will be available in the future.

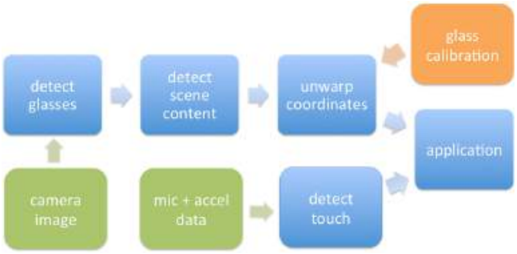


Figure 2. GlassHands processes image, microphone and accelerometer (green) to determine the glasses region, scene content and touch events. Calibration data (orange) is used to transform phone and hand coordinates from the camera's image space to the display space of the phone.

METHOD OVERVIEW

Off-phone interaction requires sensors that can observe the interaction, in our case, touch events on a surface and in-air events around the phone. However, the unmodified phone does not have any such sensing ability. GlassHands mimics a virtual external point-of-view (POV) that captures large areas of the workspace, by using the existing camera to observe a reflection from surface that lies in front of the camera. The reflection will contain the phone itself, the surface around the phone, and the user's hands.

Such a reflectors may be part of the environment, mounted on a ceiling or above a workstation. In this work, we look at *wearable* reflectors such as glasses or visors. Being worn in front of the user eyes, these reflectors are naturally positioned to face the phone. In many useful scenarios where normal touch interaction is difficult, for example, when the user has to wear gloves, or dangerous, such as skiing, bike riding, diving, manufacturing and more, such eye-wear is already common and can be leveraged.

To use the reflection image as an input modality, we need to be able to detect the reflected image in the camera image and extract the relative location of objects that need to be sensed. Specifically, we must detect the position of the user's hands next to the phone and, finally, map this information to world coordinates.

In the following sections, we will describe the components of the system, starting with detection of the user head and the area of the user's glasses where the reflection is visible, detection of the phone and the user hands in the reflection, and translation of all these locations to metric world coordinates around the phone. Finally, the implementation of a touch interface requires the ability to detect when the user's hands touch the surface, which will be described in the last section.

We have implemented the system on an Amazon fire phone using OpenCV for image processing and HTML5 and JavaScript for application development. We note, that while our system was implemented on an Amazon fire phone, it can be employed on other commodity smartphones as well. The system workflow is summarized in Figure 2.

Detecting the Glasses Area

The reflection of the workspace is seen in a relatively small part of the camera image. In a typical 2.1 megapixel image taken with an Amazon fire phone, the glasses cover less than five percent. It is important to limit the analysis of the video exclusively to this area, both for efficiency as well as to avoid false detections.

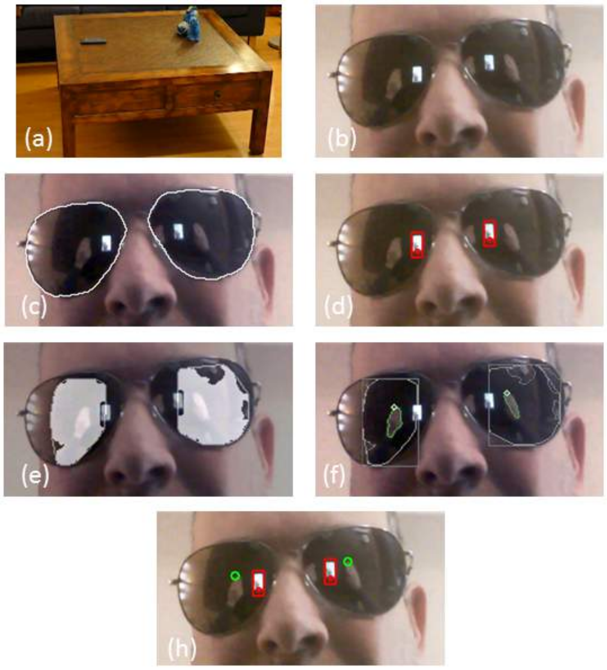


Figure 3. Detection of phones, interaction area (the table) and the hands. (a) A phone is lying on a table, facing the user. (b) A part of an original frame taken by the front facing camera. (c) Head is detected and glasses lenses, which are of different color than skin, are highlighted (d) Detected phones are highlighted. (e) Colors of the area around the phone are used to grow the table area (highlighted). This area is the interaction area. (f) Hands are detected as hand color areas (see explanation in the text), inside the table area. The curved tip of the hand define the hand position (marked by a green diamond). (h) Phones and hand locations as seen in the video.

Amazon fire phone SDK supports recovery of the user's head position using its dedicated special cameras, offering a 5 degrees of freedom head pose. Estimation of the head position in the frame helps to constrain the search area for the glasses.

Specifically, we inspect a region of interest (ROI) around the expected eyes position, masking out human skin color helps to determine the glasses area (Figure 3 (c)).

Alternatively, when using different phones (some videos for this paper were captured by a Microsoft Lumia 550, and an iPhone 6s Plus), a software face detector (such as [28, 34]) can be used to recover head position and orientation. Recent developments ([34]) were able to display impressive performances of 300 fps face tracking on mobile phones, on par with the fire phone accuracy.

Scene content Detection

The scene content is detected with three steps: phone screen, background and hands detection, which are described next.

Phone Screen Detection

The position of the user's hands relative to the position of the phone determines the location of touch or in-air events. We are using simple and efficient computer vision techniques to

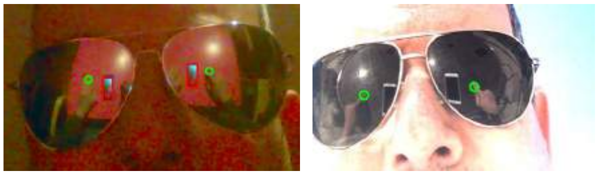


Figure 4. Detection of hands in different environments. Left: A dark room illuminated by a lamp. Right: Outdoor in full sunshine.

detect both hands and the phone screen in the image of the reflections on the glasses.

The mobile device screen can be detected using a visual marker detection techniques [47]. In many environments, in particular, indoors (such as in Figure 3 (a)), the phone is easily detected as the brightest object on the surface, enabling arbitrary applications to run unmodified on the phone. A further verification of the phone size and rectangular shape can remove most false detections. In very bright environments (e.g., outdoors), we can detect the phone's shape, or may display markers such as a unique colored areas or codes, embedded in homogeneous areas of an application [44]. Figure 3 (b) shows a part of a single frame capture by a phone front camera, and Figure 3 (d) highlights detected phones in each of the glasses lenses.

Background Detection

Once the phone is detected, the system samples the environment around the phone and builds a model of the color histogram using a Gaussian mixture model [27]. Using these colors, we grow an area around the phone of similar colors which defines the work area for the interaction, for example, a surface that the phone is positioned on, the snow, when skiiing, or asphalt, when biking. Figure 3 (e) highlights the detected table area. Any holes inside this area are regarded as part of the work area as well. Note, that for simplicity, we detect the right hand in the right lens of the glasses, and right of the phone, and the left hand in the left lens of the glasses and left of the phone. As result, the detected work area is bounded by the phone location in each lens. We store the work area by an approximating polygon that bounds it's area (seen around the highlighted area in 3 (f)).

Hand Detection

Next, the user is asked to put the hands to the sides of the phones. The system samples the colors of the observed hands, and stores it again as a Gaussian mixture model. A hand is detected as a connected area of pixels of minimal size, with hand colors, that is located inside the work area polygon. Figure 3 (f) shows hands candidates as bright polygons, of which only the ones that lie inside the work area are regarded as valid hand candidates. The tip of the finger, which is a high point of large curvature along the hand boundary is regarded as the point of touch [11] (marked in Figure 3 (f) by green diamonds).

Further tracking of the hands over time can be used to eliminate sporadic false detection of hands, and focus on the true hands.

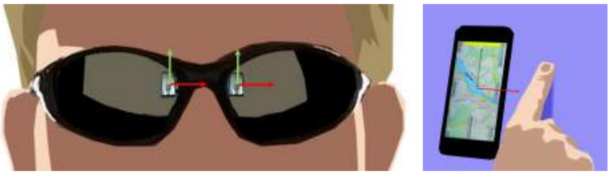


Figure 5. Origin of coordinate systems. Left: The coordinate system of each reflection originates at the center of the phone reflection, with axes parallel to the image axes. Right: The coordinate system of the surface originates at the center of the phone, and the axes are defined by the orientation of the calibration target ©Eyal Ofek.

The above sampling of the environment and hand colors enhance robustness to the color of the environment lighting or to skin or gloves color (as long as they have a minimal difference from the environment color distribution), and gives the system flexibility to work both in dark environment (as seen in Figure 4, left) or a bright outdoor environment (Figure 4, right).

Note that, during the user study, we used a blue work surface, which is the furthest from skin tone (see Figure 10), to ensure maximum robustness of detection during the study. Other methods could also be used, such as hand motion or machine learning of hands models [11], which enable hand detection in a large range of environments.

Unwarping of Image Points

Our goal is to use hand gestures over the planar surface around the phone, as an extension of the touch sensitive display of the phone. To do so, we need to map the position of any hand detected in the reflection in the glasses to the table surface. Let M be a mapping $s = M(p)$ from the position of a point in the camera frame, p, to a position of the surface, s. The coordinate system of the surface should have the same origin and same axes as the phone display (see Figure 5).

In some cases, the model of the reflector is known and the mapping is simple. For example, if the reflecting surface is planar, this mapping is a homography, defined by four points, such as the display corners of the phone. In the general case of curved glasses, the mapping is non-linear.

To estimate the reflection mapping, we place a checkerboard pattern on the working surface and position the phone at its center. Given an image of the reflection in the glasses, the transformation can be measured at the corners of the reflected pattern (see Figure 6) and mapping them to the actual dimensions of the target.

The mapping of other points is linearly interpolated between the known points. A point lying inside a checkerboard tile in the reflected image will be mapped to the corresponding surface point using a homography defined by the tile's corners. If fewer than four corners have a defined mapping, a simpler transformation is used (affine for three corners, and rotation plus scale for two).

The above mapping is depended on the head position and orientation relative to the phone. Let M_i be the mapping from the image of the lens to the surface, when the phone center in the lens is visible at position q_i. Capturing the checkerboard

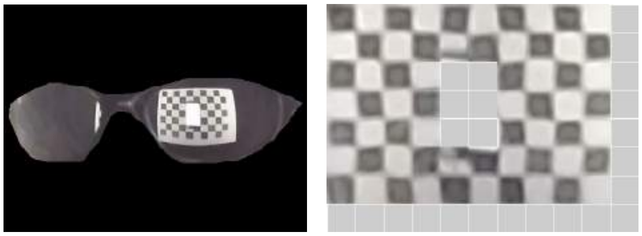


Figure 6. Left: A checkerboard is reflected in the glasses. Right: Area around the phone is unwrapped using recovered mapping for this glasses position.

and the phone images at k different position and orientation of the head enables the recovery of k mappings $\{M_i | i \in 1...k\}$. The mapping for a new position is interpolated from those known mappings using the current phone center location and radial basis functions.

However, in our trials, we found that the user face is constantly aimed at the phone display and the hands tend to be reflected by the central part of the lens (see Figure 3 (f)). The differences between the mapping of different frames, using our curved glasses, were quite small. As a result, we were able to use a fixed mapping, independent of the head position, for all our demos.

Detecting touch events

As a mobile device is lying on a surface, surface touch events may be detected by the peak they generate in the audio or accelerometer signals. Our touch detection algorithm is looking for a spike on the microphone data, similar to the way Hinckley et al. [21] detect spikes on the accelerometer or gyroscope of a grip sensing stylus. Touch-release events can not easily be detected this way. We used a gesture to signal it, such as opening the hand at the time of the release. The change in size of the hand relative to the mobile screen and a change in the number of detected convexity defects is used for the detection of touch-release events.

APPLICATIONS

We will demonstrate the potential of our concept by implementing four application prototypes: three uses bi-manual interaction around a supported phone, and one in-air application, where one hand is holding the phone. Furthermore, we have conceptualized new usage scenarios that are now feasible using the proposed technology.

The HTML5-based applications are integrated using an Android Web-View. Touch data is injected into the applications through JavaScript calls using the standard Android API.

Application 1: Map Navigation

Touch-based map navigation on mobiles is limited by the small screen space: Panning to distant objects or zooming large distances typically involves repeated drag and pinch gestures. Wearing gloves, for example while skying or biking, may prevent using touch to interact with the phone. Using GlassHands, one can pan and zoom utilizing the surface, or the air around the device (see Figure 7) or simultaneously pan and trace over an item of interest (see Figure 1, right). For the latter, the system detects if one touch is registered on the screen and one outside of the screen, classifying the touch on the display as trace action, whereas the touch (and subsequent movements) around the display trigger a pan action. This enables simultaneous panning and tracing compared to explicitly switching between panning and tracing on the device alone.

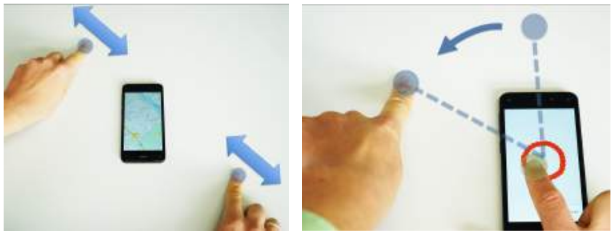


Figure 7. Left: Panning and zooming a map requires fewer repetitive gestures, compared to touch-screen only interaction and avoids screen occlusions. Right: The dominant hand draws on the touch display. The non-dominant hand rotates the virtual canvas around the screen center.

Moreover, navigating a map by outside device gestures avoids occlusion by the fingers, which is an advantage for small displays. Touching solely inside or outside the display causes the map to be paned and zoomed normally.

Application 2: Drawing

Drawing on paper is a task that involves both hands. Task execution is clearly separated between the dominant and non-dominant hand. Mimicking natural roles of hands, the non-dominant hand will be in charge of positioning the workspace, while the dominant hand is responsible for the precision task [15], such as writing or tracing.

One limitation of touch-based drawing applications on mobile devices are frequent switches of modes between workspace repositioning and drawing. The small size of the screen makes large workspace repositioning operations cumbersome, involving switching zooming, panning, and multiple flicks or ratcheting to cover large distances. Furthermore, there is no way to simultaneously draw and position the workspace.

We implemented a drawing application, in which transformations of the drawing canvas can happen simultaneously to drawing. For example, the canvas can be rotated outside the screen at the same time as a drawing action occurs on the screen, mimicking drawing on physical paper.

Application 3: Task Switching

Switching tasks on mobiles involves typically at least two touches and an additional pan action, searching for the requested application among a list of prior used applications. Furthermore, if task switching includes cutting and pasting items via a pop-up menu, at least six consecutive touch actions are needed.

We envision task switching utilizing the surface area around the device. Applications can be placed and retrieved through tapping on the locations, utilizing proprioceptive memory. The user can switch directly to a requested application by pointing on the location associated with that application (see Figure 8, left).

We allow browsing application using a linear ribbon metaphor (see Figure 8, right), and enable fast cutting and pasting: The

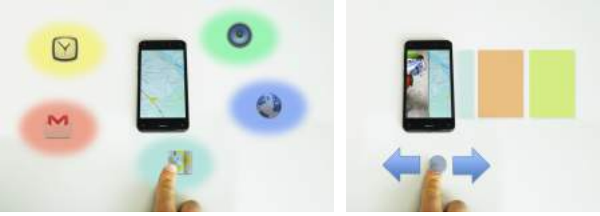


Figure 8. Left: Applications outside the visible screen can be activated through tapping. Right: A virtual ribbon allows browsing open applications, by panning outside the screen ©Eyal Ofek.

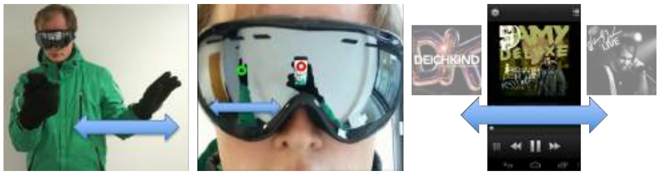


Figure 9. Left: Mid-air sliding gesture. GlassHands support large hand gestures, that enables continues scrolling or selection from a large list of options. Middle: Close-up with detected phone (red circle) and glove (green circle). Right: The album selection of a music player is operated with left and right sliding gestures.

user selects an item to be moved, and holds it. Next, the user switches to the target application, either directly through tapping on locations as mentioned above or by panning the ribbon with a finger drag outside the phone. Upon reaching the target application, the user releases the item to paste it into the target. Prior to releasing, the user can place the item at the desired location within the target application by dragging it on the phone's display. This operation can also be done in mid-air, where the holding hand thumb is used for item selection.

Application 4: Music Player

Working with touch screens of mobile devices while wearing gloves, such as while skiing, viewing a map on a motorbike (at a traffic stop) or answering a phone while using work gloves, can be cumbersome. Users have to take off their gloves to operate common capacitive touch screens, e.g., when browsing through a music collection or when unlocking the screen. While touch screen capable gloves exist, they are definitely rarer than ski goggles. Furthermore, touching mobile screens with gloves can lead to an amplified fat finger problem. We have implemented a music player application that allows browsing music collections using mid-air gestures. The user can initiate a gesture by holding the hand next to the phone. Then, hand movements to either side of the phone are mapped to a scrolling list, as can be seen in Figure 9. The large available interaction space allows fine accuracy of selection. The same application can also be used with hand gestures on surfaces.

TECHNICAL EVALUATION

We carried out technical evaluations on the input accuracy of GlassHands.

Location Accuracy

As GlassHands relies on multiple image transformations, including cropping and warping of images, it is likely that the achievable input resolution is low compared to direct sensing of the scene using wearable cameras. To determine how accurate our approach could deliver touch input, we measured the input accuracy of GlassHands on a touch display. Six users participated in the study (4 male, 2 female).

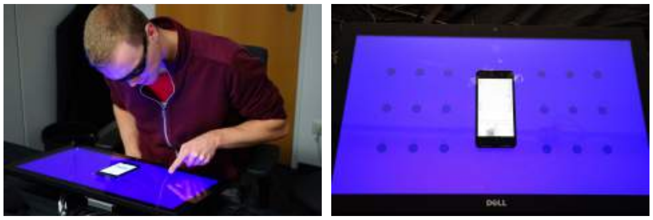


Figure 10. Left: A participant performing the accuracy evaluation. Right: Close up view on the fire phone, touch-enabled monitor and target points.

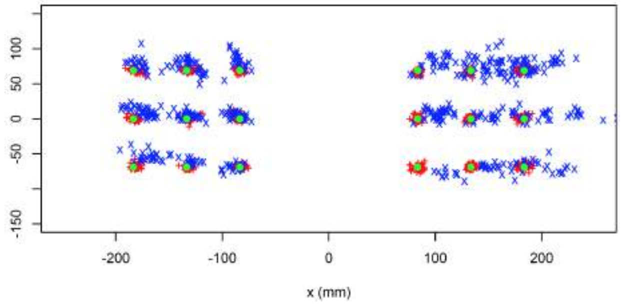
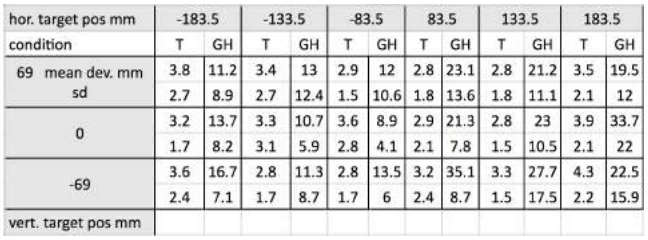


Figure 11. Plot showing a detected touch points on the Dell display. Blue crosses indicate the position of GlassHands touch input, red crosses indicate the detected baseline touches and green disks the target points.

We used a horizontally mounted Dell S2340T 23" multi-touch monitor as direct touch display, to associate detected touch points with target points. An Amazon fire phone was placed in the display center. The image resolution of the front camera was 1280x720 px. Users were placed in a chair in front of the touch display and wore reflective glasses, as seen in Figure 10. They were asked to position themselves so that the monitor would be completely visible in one of the lenses, to ensure a best-case estimate of the touch accuracy. To this end, they saw a video of the reflected area in the phone.

The users were asked to tap 18 fixed locations 10 times. The locations were indicated with blue circles distributed in a grid pattern, with three rows (on the top edge, middle and bottom edge of the phone) and six columns (three to the left at a distance of 5, 10 and 15 cm, same on the right side).

The relative offsets between touch data on the large touch display and the ones detected by glasshands were determined in a common coordinate system, centered at the phone.



hor. target pos mm	-183.5		-133.5		-83.5		83.5		133.5		183.5	
condition	T	GH	T	GH	T	GH	T	GH	T	GH	T	GH
69 mean dev. mm	3.8	11.2	3.4	13	2.9	12	2.8	23.1	2.8	21.2	3.5	19.5
sd	2.7	8.9	2.7	12.4	1.5	10.6	1.8	13.6	1.8	11.1	2.1	12
0	3.2	13.7	3.3	10.7	3.6	8.9	2.9	21.3	2.8	23	3.9	33.7
	1.7	8.2	3.1	5.9	2.8	4.1	2.1	7.8	1.5	10.5	2.1	22
-69	3.6	16.7	2.8	11.3	2.8	13.5	3.2	35.1	3.3	27.7	4.3	22.5
	2.4	7.1	1.7	8.7	1.7	6	2.4	8.7	1.5	17.5	2.2	15.9
vert. target pos mm												

Table 1. Mean errors and standard deviations for each target point (in mm) for the GlassHands (GH) detected fingers and the ground truth touch screen positions (T). Target positions are relative to the fire phone center.

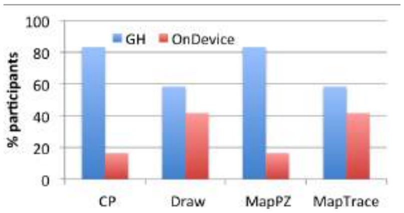


Figure 12. Percent of participants preferring GlassHands (GH) or touch-screen only (OnDevice) interaction for the same tasks as in Figure 14.

Figure 11 shows a plot with the individual touch down events detected using GlassHands (blue crosses), the ground truth touch data (red crosses) and the target points (green disks). Table 1 quantifies the deviations per target location compared to the ground truth touch data. The mean deviation over all target points was 18.3 mm (SD=13.2) using GlassHands; for ground truth touch data the mean error was 3.3 mm (SD=2.2).

To assess the accuracy in other environments, we measured accuracy for a regular room (shown in accompanying video) with manually labeling the fingertip (as ground truth data) for a single user. The hand was detected in 82% of frames with an average accuracy of 14mm (SD: 7.8). In all but one case tracking gaps were <4 frames, which can be addressed using dynamic tracking models (e.g., Kalman filter).

USER FEEDBACK

We conducted a user study, to learn from users' reactions to the GlassHands concept. Our goal was to examine the value proposition of GlassHands and discover potential social concerns. Twelve users (5 male, 7 female, mean age: 24 years, SD: 4 years, 10 with a social science background, 2 with an engineering background) participated in the study and were compensated with a small gratuity for their time. The same apparatus as in the technical evaluation was used.

Procedure

After a short introduction, users were asked to try out four prototypical applications in randomized order: map navigation using pan and zoom (MapPZ), combined map panning and tracing (MT), drawing (Draw), cut-paste (CP). Users tried both using GlassHands (GH) and a touch-screen only (OnDevice) interaction.

For the map applications, participants tried out navigation alone (pan, zoom) and a compound navigation task [32]. The latter task consisted of tracing six target regions and panning along a path. As simultaneous panning and tracing is not possible in the OnDevice condition, users could switch between both modes through a button. For the drawing application, participants were asked to trace a circle on screen. In the GH condition, users could rotate a virtual sheet of paper around the screen center as depicted in Figure 7. For the cut-paste scenario, users were asked to cut an image, embedded it in a word processor application, and paste it into a PowerPoint presentation and back again. For this scenario, we used the Android task manager, with Microsoft Word, Excel and PowerPoint applications opened in the touch-screen only condition. For each application, participants tried out the OnDevice and the GH version (counterbalanced).

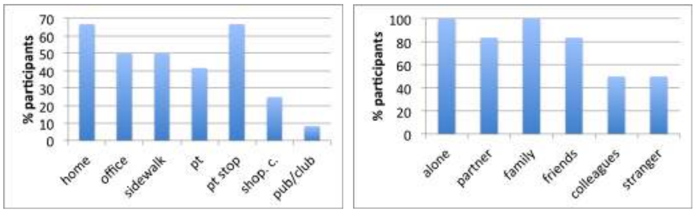


Figure 13. Percent of participants who could imagine using GlassHands at various locations (left) and in front of various audiences (right).

They rated the ease of use and usefulness of the applications through 2-item questionnaires with 5-items Likert scales and selected the preferred way of interacting. After trying both applications, they were asked about the challenges and merits of GlassHands in semi-structured interviews. In a second part of the study, users were presented with the two mid-air concepts using verbal descriptions and pictures. They were then also asked to comment on the potential usefulness of GlassHands interaction and about applications they would like to use with GlassHands in those scenarios.

Findings

Ease of use and usefulness ratings are shown in Figure 14. For ease of use, two-sided Wilcoxon signed-rank tests indicated significant differences for CP (W=45, Z=2.87, p<0.01, Cohen's d=1.45) and MT (W=15, Z=2.23, p=0.026, Cohen's d=1.02). For usefulness, two-sided Wilcoxon signed-rank tests indicated significant differences with large effect sizes for cut-paste (W=28, Z=2.60, p=0.017, Cohen's d=1.26) and MT (W=15, Z=2.23, p=0.026, Cohen's d=1.02, same results as for ease of use). Preferences are depicted in Figure 12.

Most participants preferred GlassHands for CP and and MT. For CP, users mentioned time savings and fewer number of required interactions compared to on-device usage. One user said "it is convenient to just use the hand and swipe to switch applications." Another mentioned that GlassHands interaction is "pretty easy, like on your desktop computer" and that this technique seems specifically useful, if one needs to switch applications often. Similarly, for MT, users mentioned that they preferred the modeless interaction enabled by GlassHands, with one stating "there is no need to switch" the interaction mode, another one mentioning "it is very annoying, if I have to switch between tracking and panning" in the on-device condition.

For the drawing application, users agreed that GlassHands enables more precise tracing of template figures. One highlighted the fact that he was able to draw "more accurate, clean and beautiful." However, while some users liked this features, other users were feeling that this approach hinders artistic expression. One participant mentioned that "for beginners [GlassHands] might be better due to its constraints, while professionals would probably prefer the freedom of on-screen drawing." Users also saw potential for GlassHands for more complex drawing scenarios than tracing a circle. One mentioned "I could concentrate on drawing complex shapes at the top of the screen without worrying about other parts of the canvas."

For map navigation using pan and zoom, participants preferred GlassHands, but did not rate the ease of use or usefulness higher compared to on-device usage. Several users mentioned that the on-device interaction is "already quite easy" and "familiar." Four users mentioned that they felt more confident using GlassHands, as there is a smaller chance for unexpected effects of finger movements (specifically, pinch gestures) due to the larger space. Occlusion free navigation of the map was mentioned as an advantage of GlassHands by five participants. One user also mentioned that it is more fun to use the GlassHands approach, as it "feels like StarTrek."

With a questionnaire on social acceptability [35], we asked participants about suitable locations and audiences (Figure 13) for using GlassHands.

Social issues were primarily mentioned by participants who would not wear sunglasses for fashion purposes (three do not wear sunglasses, seven only for sun protection, two also inside buildings for fashion purposes). Five users mentioned that (in the words of one participant) it would "feel awkward to wear sunglasses, when the sun is not shining," but also mentioned that it would be "inappropriate not to look others in the eyes while interacting with them." Interestingly, the target audience for this inappropriate behavior differed between participants, with some mentioning strangers, while others, partners and friends, as individuals in front of whom they would not wear sunglasses. Three users explicitly mentioned spatial constraints for using GlassHands in public space as "I would be worried to bump into another person." One user also mentioned privacy concerns as "it would be more visible what I am doing with my phone." One participant mentioned to be a trend follower and said that "if everyone would wear sunglasses to interact with their phones, I would do it, too."

We asked participants about their opinion on using mid-air GlassHands while wearing gloves or while biking. Users mentioned that this kind of interaction would be ideal for situations in which they would have to concentrate on a demanding primary task, to trigger secondary tasks. Examples included operating a music player, answering incoming calls or coarse map navigation. However, several participants also mentioned that they would not use GlassHands (or an ordinary phone) while biking at all, due to safety reasons. Finally, two users mentioned that they really like the concepts, but, ultimately, would like to use use the space around an unmodified phone without sunglasses.

DISCUSSION AND CONCLUSION

We have presented GlassHands, an interactive system for around-device interaction on unmodified mobile devices. We demonstrated that interaction at the periphery of a mobile device is feasible given a front-facing RGB camera and a user wearing sunglasses, or any other reflector. In contrast to previous work [38], we significantly broaden the interaction space around an unmodified mobile device. In contrast to Surround-See [46], we enable large interaction space, without the need for special hardware, just everyday common apparel, and with keeping the phone thin and small. Removing the need for special hardware enable us to easily deploy this solution, such as through a store application.

We demonstrated interaction techniques and applications for GlassHands, among them, mode-less panning and tracing on maps, cut and paste by dragging between applications, and drawing with the non-dominant hand controlling the canvas. A group of users have tried these applications, and found their interaction preferable over their common interaction. The consensus was that the higher potential savings in effort, the more participants were willing to use it.

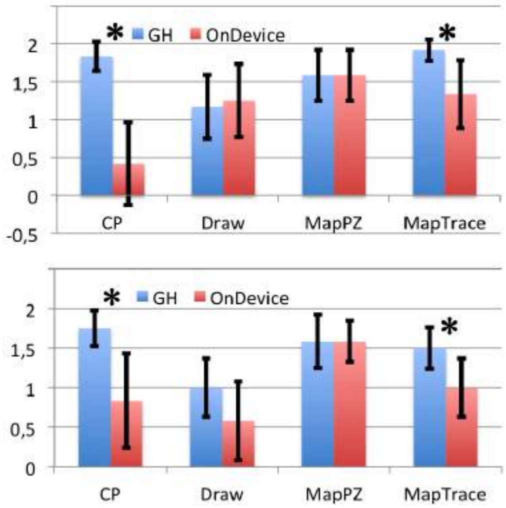


Figure 14. Ease of use (top) and usefulness (bottom) for GlassHands (GH) and Touch-screen only (OnDevice) interaction for cut and paste (CP), drawing (Draw), map pan and zoom (MapPZ) and map tracing (MT) on a 5-point Likert scale (-2 very difficult ... 2 very easy). Statistically significant differences at α = .05 indicated with a *.

The proposed system has several limitations, some which may be addressed in future work:

An inherent limitation of our approach is that users have to wear reflective glasses, which have to be visible in the field of view of the front camera (typically 70-80 degrees field of view). Dark glasses allows the camera to view a clear reflection without the view of the user eyes behind them. If the user face is relatively dark, such as in the case of a desk lamp illuminating the table, while the user's head stays in the dark, then regular glasses can be used just as well.

In general, the usage of dark glasses in low-light environments, like indoor offices, might not be appropriate, both due to perceptual and social reasons. However, there are many situations, both indoors and outdoors, where wearing reflective eye-wear is common: workers wearing safety goggles, skiers, divers, motorcyclists and so on.

On-surface interactions as described above enable comfortable work, where the users hands are supported by the surface, and the tap of a finger on the surface can be used to detect touch. However, the technology is not limited to surface-based interaction. The same 2D interaction may be used in mid-air around the phone with ample gestures. Such in-air interaction with Glasshands could help skiers operate their mobiles without the need to take off their gloves.

Also, GlassHands could be used by bikers, who have mounted their phone on the handle bar. Sliding the hands along the handle bar, relatively far from the phone, could be utilized to steer on-screen applications (see Figure 9, right). A possible application may sense the direction of a pointed hand, reflected in the helmet visor, and use the phone GPS and orientation to announce the street number of the house the biker is pointing at. Divers could use GlassHands to interact with their phone, stored inside a watertight casing. In a similar fashion, workers who wear protective glasses and gloves may interact using large gestures around the phones.

Measuring the hand positioning accuracy shows that a minimal distance of at least 5 cm is needed to reliably separate two individual touch down events on the surface, when a single frame is used for measurement. It is possible to increase the measurement accuracy, for example, using temporal filtering.

Furthermore, we deliberately employed simple and efficient computer vision techniques throughout our pipeline to demonstrate the feasibility of the GlassHands concept. These simple algorithms can not cope well with complex environments, typically found in real-world situations. For practical implementations in commercial applications, more robust algorithms should be used (see also below).

FUTURE WORK

At the camera resolution we used (0.9 megapixel), the image of each lens is about 130 by 170 pixels, which limits the maximum accuracy of the hands location to about 0.5 cm. Better front cameras are already available and should improve the quality of detection and location estimation.

In the future, we want to support a wider variety of glasses models with different reflection and curvature properties. Moreover, the use of reflections directly of the user's eye using corneal imaging [29] could overcome the need for eyewear.

Any method that uses a camera depends on sufficient lighting. We could use the flashlight LED on the phone itself to illuminate the glasses. An interesting alternative would be the reflection of a time-of-flight sensor to determine the 3D position of objects above the surface.

Furthermore, by using both lenses of the glasses, one could model two catadioptric camera systems and apply stereo techniques to recover depth values. We plan to investigate how to estimate such camera models on the fly. Depending on the quality of the stereo reconstruction, usage may range from determining the height of the user's hands above the surface to possible replacement of body-worn depth sensors.

GlassHands allows to sense an ample area in a phone's vicinity. There are options to recognize objects in space and react accordingly. The interaction may involve everyday objects, such as toys on a table, a board-game, or ingredients on a kitchen counter.

We believe that new sensing capabilities for phones will help spreading spatially aware applications. In many cases, the development of such applications is hampered by the limited distribution of required hardware. Hardware manufacturers, on the other hand, may hesitate to include new technology without proven value for applications [5]. We hope that an approach such as GlassHands may break this circle by enabling applications aimed at a specific scenario, such as Skiing, to be commercially successful using existing hardware. Ultimately, this may encourage the development of new dedicated sensors integrated in consumer phones.

REFERENCES

1. Annett, M., Grossman, T., Wigdor, D., and Fitzmaurice, G. Medusa: a proximity-aware multi-touch tabletop. *Proc. UIST '11*, 337–346.
2. Ashbrook, D., Baudisch, P., and White, S. Nenya: subtle and eyes-free mobile input with a magnetically-tracked finger ring. *Proc. CHI '11*, 2043–2046.
3. Avrahami, D., Wobbrock, J. O., and Izadi, S. Portico: tangible interaction on and around a tablet. *Proc. UIST '11*, 347–356.
4. Butler, A., Izadi, S., and Hodges, S. Sidesight: multi-"touch" interaction around small devices. *Proc. UIST '08*, 201–204.
5. Buxton, B. The long nose of innovation. *Insight*, 11, 2008: 27.
6. 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. *Proc. CHI '15*, 3001–3009.
7. Chen, X. A., Schwarz, J., Harrison, C., Mankoff, J., and Hudson, S. E. Air+touch: interweaving touch and in-air gestures. *Proc. UIST '14*, 519–525.
8. Chen, X. A., Schwarz, J., Harrison, C., Mankoff, J., and Hudson, S. Around-body interaction: sensing and interaction techniques for proprioception-enhanced input with mobile devices. *Proc. MobileHCI '14*, 287–290.
9. Chen, X. A., Grossman, T., Wigdor, D. J., and Fitzmaurice, G. Duet: exploring joint interactions on a smart phone and a smart watch. *Proc. CHI '14*, 159–168.
10. De Luca, A., Zezschwitz, E. von, Nguyen, N. D. H., Maurer, M.-E., Rubegni, E., Scipioni, M. P., and Langheinrich, M. Back-of-device authentication on smartphones. *Proc. CHI '13*, 2389–2398.
11. Erol, A., Bebis, G., Nicolescu, M., Boyle, R. D., and Twombly, X. Vision-based hand pose estimation: a review. *Comput. Vis. Image Underst.* 108(1-2), Oct. 2007: 52–73.
12. Freeman, E., Brewster, S., and Lantz, V. Towards usable and acceptable above-device interactions. *Proc. MobileHCI '14*, 459–464.
13. Gemperle, F., Kasabach, C., Stivoric, J., Bauer, M., and Martin, R. Design for wearability. *Proc. ISCW'98*. 1998, 116–122.
14. Grubert, J., Heinisch, M., Quigley, A., and Schmalstieg, D. Multifi: multi fidelity interaction with displays on and around the body. *Proc. CHI '15*, 3933–3942.
15. Guiard, Y. Asymmetric division of labor in human skilled bimanual action. *Journal of Motor Behavior*, 19(4), 1987: 486–517.

16. Harrison, C. Appropriated interaction surfaces. *Computer*, 43(6), June 2010: 86–89.

17. Harrison, C., and Hudson, S. E. Abracadabra: wireless, high-precision, and unpowered finger input for very small mobile devices. *Proc. UIST '09*, 121–124.

18. Harrison, C., Benko, H., and Wilson, A. D. Omnitouch: wearable multitouch interaction everywhere. *Proc. UIST '11*, 441–450.

19. Hasan, K., Ahlström, D., and Irani, P. Ad-binning: leveraging around device space for storing, browsing and retrieving mobile device content. *Proc. CHI '13*, 899–908.

20. Henderson, S. J., and Feiner, S. Opportunistic controls: leveraging natural affordances as tangible user interfaces for augmented reality. *Proc. VRST '08*, 211–218.

21. Hinckley, K., Pahud, M., Benko, H., Irani, P., Guimbretière, F., Gavriliu, M., Chen, X. A., Matulic, F., Buxton, W., and Wilson, A. Sensing techniques for tablet+stylus interaction. *Proc. UIST '14*, 605–614.

22. Jones, B., Sodhi, R., Forsyth, D., Bailey, B., and Maciocci, G. Around device interaction for multiscale navigation. *Proc. MobileHCI '12*, 83–92.

23. Kratz, S., and Rohs, M. Hoverflow: exploring around-device interaction with ir distance sensors. *Proc. MobileHCI '09*, 42:1–42:4.

24. Kratz, S., Rohs, M., Guse, D., Müller, J., Bailly, G., and Nischt, M. Palmspace: continuous around-device gestures vs. multitouch for 3d rotation tasks on mobile devices. *Proc. AVI '12*, 181–188.

25. Laput, G., Xiao, R., Chen, X. A., Hudson, S. E., and Harrison, C. Skin buttons: cheap, small, low-powered and clickable fixed-icon laser projectors. *Proc. UIST '14*, 389–394.

26. Marquardt, N., Jota, R., Greenberg, S., and Jorge, J. The continuous interaction space: interaction techniques unifying touch and gesture on and above a digital surface. In: *Proc. INTERACT '11*. Vol. 6948. 2011. ISBN: 978-3-642-23764-5.

27. McLachlan, G. J., and Basford, K. E. Mixture models : inference and applications to clustering. M. Dekker, 1998.

28. Murphy-Chutorian, E., and Trivedi, M. Head pose estimation in computer vision: a survey. *PAMI*, 31(4), 2009: 607–626.

29. Nitschke, C., Nakazawa, A., and Takemura, H. Corneal imaging revisited: an overview of corneal reflection analysis and applications. *Information and Media Technologies*, 8(2), 2013: 389–406.

30. Oakley, I., and Lee, D. Interaction on the edge: offset sensing for small devices. *Proc. CHI '14*, 169–178.

31. Ogata, M., and Imai, M. Skinwatch: skin gesture interaction for smart watch. *Proc. AH '15*, 21–24.

32. Pahud, M., Hinckley, K., Iqbal, S., Sellen, A., and Buxton, B. Toward compound navigation tasks on mobiles via spatial manipulation. *Proc. MobileHCI '13*, 113–122.

33. Rädle, R., Jetter, H.-C., Marquardt, N., Reiterer, H., and Rogers, Y. Huddlelamp: spatially-aware mobile displays for ad-hoc around-the-table collaboration. *Proc. ITS '14*, 45–54.

34. Ren, S., Cao, X., Wei, Y., and Sun, J. Face alignment at 3000 fps via regressing local binary features. *Proc. CVPR 2014*. 2014.

35. Rico, J., and Brewster, S. Gestures all around us: user differences in social acceptability perceptions of gesture based interfaces. *MobileHCI 09*. 2009, 64:1–64:2.

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

37. Sodhi, R. S., Jones, B. R., Forsyth, D., Bailey, B. P., and Maciocci, G. Bethere: 3d mobile collaboration with spatial input. *Proc. CHI '13*, 179–188.

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

39. Sturm, P., Ramalingam, S., Tardif, J.-P., Gasparini, S., and Barreto, J. a. Camera models and fundamental concepts used in geometric computer vision. *FTCGV*, 6(1-211;2), Jan. 2011: 1–183.

40. Wagner, J., Nancel, M., Gustafson, S. G., Huot, S., and Mackay, W. E. Body-centric design space for multi-surface interaction. *Proc. CHI '13*, 1299–1308.

41. Weigel, M., Lu, T., Bailly, G., Oulasvirta, A., Majidi, C., and Steimle, J. Iskin: flexible, stretchable and visually customizable on-body touch sensors for mobile computing. *Proc. CHI '15*, 2991–3000.

42. Wigdor, D., Forlines, C., Baudisch, P., Barnwell, J., and Shen, C. Lucid touch: a see-through mobile device. *Proc. UIST '07*, 269–278.

43. Wilson, A. D., and Benko, H. Combining multiple depth cameras and projectors for interactions on, above and between surfaces. *Proc. UIST '10*, 273–282.

44. Woo, G., Lippman, A., and Raskar, R. Vrcodes: unobtrusive and active visual codes for interaction by exploiting rolling shutter. *Proc. ISMAR '12*, 59–64.

45. 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. *Proc. MobileHCI '14*, 67–76.

46. Yang, X.-D., Hasan, K., Bruce, N., and Irani, P. Surround-see: enabling peripheral vision on smartphones during active use. *Proc. UIST '13*, 291–300.

47. Zhang, X., Fronz, S., and Navab, N. Visual marker detection and decoding in ar systems: a comparative study. *Proc. ISMAR '02*. 2002, 97–106.

48. 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. *Proc. UIST '14*, 527–534.