

# GazeConduits: Calibration-Free Cross-Device Collaboration through Gaze and Touch

**Simon Voelker**  
RWTH Aachen University  
Aachen, Germany  
voelker@cs.rwth-aachen.de

**Sebastian Hueber**  
RWTH Aachen University  
Aachen, Germany  
hueber@cs.rwth-aachen.de

**Christian Holz**  
ETH Zurich  
Zurich, Switzerland  
christian.holz@inf.ethz.ch

**Christian Remy**  
Aarhus University  
Aarhus, Denmark  
remy@cc.au.dk

**Nicolai Marquardt**  
University College London  
London, United Kingdom  
n.marquardt@ucl.ac.uk

## ABSTRACT

We present GazeConduits, a calibration-free ad-hoc mobile interaction concept that enables users to collaboratively interact with tablets, other users, and content in a cross-device setting using gaze and touch input. GazeConduits leverages recently introduced smartphone capabilities to detect facial features and estimate users' gaze directions. To join a collaborative setting, users place one or more tablets onto a shared table and position their phone in the center, which then tracks users present as well as their gaze direction to determine the tablets they look at. We present a series of techniques using GazeConduits for collaborative interaction across mobile devices for content selection and manipulation. Our evaluation with 20 simultaneous tablets on a table shows that GazeConduits can reliably identify which tablet or collaborator a user is looking at.

## Author Keywords

Cross-device interaction, gaze input, touch input

## CCS Concepts

•Human-centered computing → Ubiquitous and mobile computing systems and tools;

## INTRODUCTION

Cross-device interaction between multiple co-located mobile devices has become an emerging field of research in human-computer interaction [5, 23, 33, 34]. These setups of co-located mobile devices can be used to create *ad-hoc device communities* [13] almost anywhere to allow users to share or collaborate on content across devices. Especially using the spatial relationship between co-located devices has been

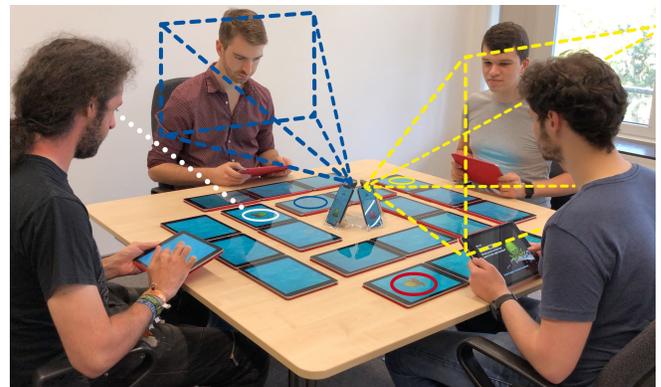


Figure 1. GazeConduits is a mobile and ad-hoc gaze tracking system that allows one or more users to interact across the devices placed in front of them. GazeConduits leverages the eye tracking capabilities of recent commodity devices, allowing them to participate in cross-device interaction scenarios.

shown to be beneficial for applications such as brainstorming [34], collaborative photo sharing [14], and productivity tasks [20]. For seamlessly interacting across devices, researchers have explored a variety of input techniques, such as touch [29] and gestures [6, 30].

One input modality receiving increased attention lately is gaze as a communication channel for recognizing the user's interaction with a device [25]. Research on gaze-based interaction design for mobile cross-device interaction using off-the shelf mobile devices has only just started [4], as the majority of work so far has used dedicated eye trackers for usability testing in controlled lab setups, leaving their use in real-world scenarios largely open for exploration [15]. With the introduction of eye tracking on smartphones like Apple's iPhone X and Samsung's Galaxy S8, researchers expect a new surge of applications and use-cases [15]. We believe that gaze sensing is particularly relevant for *mobile* cross-device interaction, as studies have shown its benefits for interacting with multiple devices [40, 44].

In this paper, we integrate mobile gaze sensing into cross-device interaction to explore the design space of collaborative interaction on everyday mobile devices. Our approach is

guided by the following research question: How can we implement a calibration-free collaborative interaction technique that supports cross-device interaction? We present our gaze tracking system GazeConduits, which leverages the gaze tracking capabilities built into recent smartphones to detect and track users around a set of devices as well as predict their gaze direction towards specific devices for selection and interaction. GazeConduits requires no calibration, which makes it simple to setup in ad-hoc group scenarios. We briefly demonstrate two scenarios that build on GazeConduits in conjunction with touch input to enable dynamic groups of users to collaboratively select and manipulate content across their devices.

We evaluated GazeConduits and found that it can detect which tablet or collaborator a user looks at with 95.58% accuracy.

## RELATED WORK

Our techniques are positioned in the context of three areas of related work: the design of cross-device systems and their requirements for tracking; interaction techniques leveraging a person's gaze as input; and the more recent approaches towards mobile gaze tracking.

### Cross-Device Systems and Tracking

Cross-device systems [1] afford new interactions for sense making [18, 48], curation [2], sharing media [19], and collaborative content editing [16]. Usually, these setups distribute an interface across a number of devices like tablets, smartphones, interactive walls or tabletops, and provide techniques to effectively use the input/output modalities of these multi-device setups. The research has led to the exploration of supporting both individual tasks [7, 20] and collaborative activities [9, 11, 38, 46].

A major research area of cross-device work is investigating effective techniques for a person to interact with this ecology of devices [1]. Inspired by early seminal work such as Rekimoto's *Pick-and-Drop* [29], different techniques have been designed and evaluated for linking devices and transferring digital content between them [5, 6, 8, 24, 35]. Several cross-device systems [10, 21, 33, 49] used overhead-positioned depth-sensing cameras – tracking both people and devices – to enable 3D spatial gestures a person can perform to transfer information across the multi-device ecology. Later, related approaches leveraged screen polarization [32] or tracking a person's face with front-facing RGB cameras [4] to detect device location. Because most of these approaches require complex technical setups, finding new ways towards ad-hoc and flexible cross-device collaboration remains an ongoing research challenge.

### Gaze Interactions

Gaze is our primary sensory channel, as the first interaction with an object is to look at it [50, 51]. In the 1980s, researchers began to explore how gaze can be utilized as an input method [12, 45]. Several obstacles complicate the use of gaze input, such as its inaccuracy, the *double role of eye gaze*, and the *Midas Touch* problem, rendering it ineffective to directly manipulate digital content or control cursors [37]. Instead, gaze has proven to be more useful in applications as a method for

selection of, or focusing on, content, in combination with other input modalities [43], such as touch [28, 36] or the mouse [51]. Multiple studies showed that gaze can be utilized to seamlessly switch between direct and indirect touch input [25, 26, 28, 44]. This is achieved by using direct touch while users are looking at their hands, and warping the touch point to the gaze location when they look at a different part of the display or even at a different display.

An advantage of gaze over touch input is that it does not share the problem of reachability [31], as any objects that are in visible range can be interacted with. Studies have shown that a combination of gaze and touch can be used to extend the touch input on a large surface [41], to transfer objects between multiple devices [40], or to modify objects on a distant screen [39]. Gaze input can also be a particularly useful addition to collaborative settings. Studies by Zhang et al. [52], van Rheden et al. [42], and Pfeuffer et al. [27] used gaze tracking to show that all collaborators recognize at which location a user is looking, thus increasing the awareness of collaborators. All these interaction concepts can be applied in an ad-hoc mobile cross-device setting; however, so far they have required either extensive calibration, used specific hardware, or needed a controlled lab environment to track gaze.

### Mobile Gaze Tracking

One of the first systems that allowed gaze-based interaction on unmodified mobile devices was the EyePhone [22]. It allowed the user to move the phone relative to the user's face such that the user's left eye was in one of 9 possible positions in a 9 element grid, and then used eye blinking to trigger input. Wood and Bulling [47] used the built-in camera of a tablet and employed a model-based approach to detect eye positions while users were looking at the tablet, while Krafka et al. [17] estimated gaze with an end-to-end appearance-based approach using deep learning.

Goswami et al. [3] showed that using depth camera data to geometrically model a user's face and eyes improved gaze tracking accuracy. Recent smartphones use a similar approach to extract the user's facial and eye features, which has brought gaze tracking capabilities to off-the-shelf consumer devices [15]. These devices could hold the key to bring gaze interactions to ad-hoc cross-device settings.

## GAZE TRACKING IN AN AD-HOC SETTING

In ad-hoc cross-device setups, multiple users can create device communities on-the-fly by arranging multiple mobile devices on a surface. With GazeConduits, we follow this ad-hoc approach and design it for easy setup and minimal to no calibration. Our choice of using the smartphone to track gaze instead of a separate eye tracker allows for the system to be set up at any time without specific hardware requirements beyond the ubiquitous smartphone that people carry in their pocket.

GazeConduits builds on iOS running on Apple's iPhone X or later, which tracks the user's face by extracting facial features from a depth map that is calculated using a projected infrared dot pattern. These facial features include the position of the user's nose, mouth, and eyes as well as the position of the pupils, from which iOS estimates gaze directions.

During our investigations, we noticed that Apple’s gaze prediction assumes that the user is holding the device in her hands and looking directly at it; it was not designed for placing the phone on a table and using it to track gaze across surrounding devices. To determine if the built-in capabilities are sufficient for gaze tracking across users and devices around a single iPhone X, we evaluated to what extent the phone may serve as a central hub for gaze interaction, specifically the size of the tracked area in front of the phone in which the user’s face and gaze are detected, and supported tilt angles for placing the device on the table while maintaining solid tracking.

### Determining the Gaze Tracking Area

Since tracking range and accuracy depend on the view frustum and resolution, we conducted a study with five participants to determine the parameters of reliable tracking.

To determine the maximum distance between the user’s face and the phone, we placed the phone on a camera stand and asked users to slowly move their heads away from the phone. At the farthest point at which the phone still detected the user’s head, we measured the distance from the participant’s nose to the phone using a laser distance meter. The results showed that the phone could track the user’s head at a distance between 10 and 88 cm ( $SD = 3.2$  cm). We used the same approach to measure the phone’s frustum angle and how far users could turn their head away from the smartphone while still being tracked by the phone. Our results showed that the maximum supported angle for face tracking is  $30^\circ$  ( $SD = 2.1^\circ$ ) in both the horizontal and vertical direction. We found that users could turn their head by about  $35^\circ$  ( $SD = 3.5^\circ$ ) horizontally and about  $30^\circ$  ( $SD = 2.3^\circ$ ) vertically (Fig.2).

### Determining Optimum Angle and Distance

To support setting down the phone as a tracker onto a table, we determined suitable tilting angles. To provide users with some space to move around in without unintentionally leaving the tracking area, we limited the distance between the phone and the user’s head while sitting directly in front of it to 70 cm. Assuming a user’s typical sitting height of 30–40 cm above the table and close to the table edge, the phone should be placed 60 cm away from that table edge, following basic trigonometry. At this position, we varied the tilting angle of the phone, and found that an angle between 25 and 35 degrees allowed users to move their heads without the phone losing tracking.

### Tracking Gaze on a Table

To track the user’s gaze across multiple devices on the same table, we register the phone with the table using a stand that fixes the phone at a specific angle. We then use the phone’s IMU to map 3D positions obtained from face tracking to positions that correspond to the plane of the table surface. From Apple’s API, we obtain the user’s head location  $h$  and its direction  $\vec{d}$  in the local coordinate system with the phone at its origin. The user’s eyes are represented by  $e_1$  and  $e_2$  in the coordinate system of the user’s head and one directional normal vector for each eye ( $\vec{e}_1, \vec{e}_2$ ). By combining these positions and vectors, we calculate a vector normal  $\vec{g}$  and its origin that represents the user’s gaze in the coordinate system of the phone.

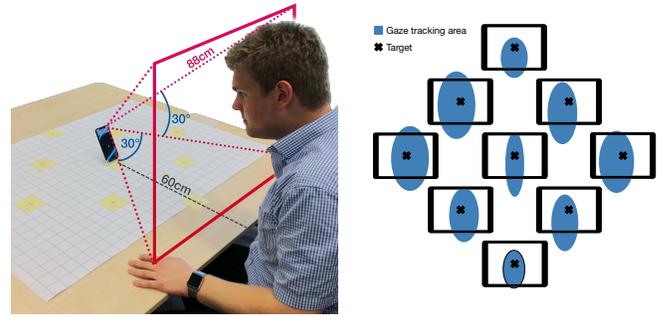


Figure 2. To analyze tracking accuracy, participants were asked to fixate each target for five seconds. The ellipses represent the measured gaze positions on the table in comparison to a tablet.

We can now estimate which location on the table users are looking at, by creating a virtual 3D space that contains the user’s head location  $h$  and the normal gaze vector  $\vec{g}$  as well as the table’s surface, and then calculating the point on the table surface where those vectors intersect.

### Measuring Gaze Tracking Accuracy

To determine the accuracy of the phone’s gaze prediction towards targets on the table, we conducted a study with 10 participants (23–42 years,  $M = 32.73$ ,  $SD = 2.31$ , four female).

#### Apparatus and Task

One iPhone X in a stand was placed on the table with a distance of 60 cm to the table edge. We defined the bottom of the phone as the origin of the coordinate system  $(0,0)$ , and highlighted nine target locations around the phone on the table at  $(0,0)$ ,  $(-40,0)$ ,  $(40,0)$ ,  $(0,-40)$ ,  $(0,40)$ ,  $(-20,-20)$ ,  $(20,20)$ ,  $(-20,20)$ , and  $(20,-20)$  cm (Fig. 2).

Participants were asked to look at each of the targets for five seconds. During this time, they were asked to move their head around while keeping their eyes on the target position. Throughout each trial, the phone recorded the intersection points between the gaze vector and the plane, and the head vector and the plane, 30 times per second. This allowed us to analyze the difference between head and gaze tracking.

#### Results

To evaluate the accuracy of the smartphone’s gaze estimation, we measured the distance between the estimated gaze location on the table and the marked location at which participants were looking. Our results show that the average distance over all participants and all targets was 2.5 cm on the  $x$ -axis and 9 cm on the  $y$ -axis. However, the spread of the data had a large standard deviation of 8.1 cm on the  $x$ -axis and 11.5 cm on the  $y$ -axis. Using this approach allows the smartphone to detect if a user is looking at a specific table area with the size of  $20 \times 25$  cm, which is roughly the size of a tablet. Below, we will show that this simple tracking approach is accurate enough to detect at which particular tablet a user is looking.

### GAZECONDUITS

Based on the parameters elicited from our preliminary investigations, we developed GazeConduits to explore the applications of gaze tracking in an ad-hoc collaborative environment. The goal of GazeConduits is to show that we can achieve a

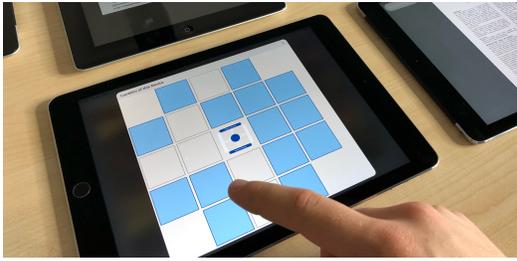


Figure 3. When the user places a new tablet on the table, GazeConduits connects to existing devices and asks the user to select its grid position.

calibration-free collaborative environment that supports a combination of gaze and touch interactions on commodity devices and minimal setup requirements.

To set up the GazeConduits system, the first user places an iPhone on a table and starts our GazeConduits app to act as the central hub for gaze input. To enforce correct smartphone placements on the table at a working tilt angle, we built a simple case with a stand, as shown in Figure 4. Using the measurements of this stand and the IMU of the smartphone we can calculate the exact position on a plain surface that represents the table in the coordinate system of the smartphone. The flexibility of the stand allows users to expand and collapse the setup during operation. Such a case could also be built in a thinner form factor, e.g., by using a 3D printer or laser cutter.

After placing her smartphone on the table, the user can add up to 20 tablets around the smartphone following a rough grid layout (Fig. 1). To be part of the shared device environment, the GazeConduits app is launched on each tablet, which automatically connects it to the smartphone, in our case via Wi-Fi. Each new tablet then displays a message asking the user to select its rough position on the grid (Fig. 3).

The layout and size of our grid, and the limitation to 20 tablets, are based on a number of design decisions. Each grid element has a size of  $25 \times 25$  cm; based on our accuracy tests, this size ensures that the gaze tracking mechanism can always reliably detect whether a user is looking at a target within that grid element. It is also slightly larger than common tablets like the iPad ( $23.8 \times 16.7$  cm), Microsoft Surface GO ( $24.5 \times 17.5$  cm), or Samsung Galaxy Tab ( $24.9 \times 16.4$  cm), making sure that common tablets fit into a grid element without overlap issues, regardless of their orientation. Two rows of tablets are placed in front of each user; with more rows the distance from user to phone would exceed the maximum 88 cm derived in previous tests, especially if the user is leaning back or rocking back and forth in their chair.

Assuming a setup of up to four collaborators around the table, we can support a maximum of a  $5 \times 5$  grid, with the central grid element being reserved for the smartphones. Furthermore, the corners of the grid do not allow for reliable gaze tracking, as users looking at tablets in the corners of the table have to rotate their heads more than 30 degrees. Thus, we limited the number of grid elements in the first row in front of each user to three elements, resulting in the grid used in Figure 1. GazeConduits works on other layouts, too, e.g., in setups with only one or two users or in scenarios where not all tablets



Figure 4. The smartphone case (left side) has an included stand that can be expanded and collapsed. It ensures that the smartphone can easily be placed on the table in the correct tilting angle. The connector widget (right side) allows the users to easily arrange up to four smartphones in a fixed position.

should be selectable via gaze. As soon as one or more tablets are placed on the table and the user selects at which position these tablets are placed, GazeConduits can directly detect and track which tablet or other collaborator the user is looking at.

While GazeConduits displays a static, predefined grid on its tablet UI during this positioning phase, the system does not require tablets to be placed in perfect alignment with the grid elements. It does display a grid to suggest good placement positions, but after tablets have been placed and the users tapped on the closest grid element for each, the system then splits up the entire shared space between the existing tablets, mapping every area to the closest existing device, in a simplified Voronoi grid approach. This makes our system robust against sparsely filled tablet grids and misaligned devices. The grid becomes more important if two tablets are placed directly adjacent to each other. In this case, aligning them to the grid helps disambiguate which tablet is being looked at.

In addition, GazeConduits can also detect if one or more users are present, their locations around the table, and (via face identification) who is located where. We implemented this detection by comparing the geometry (nose, chin, and mouth) of the detected faces, achieving a simple user identification to simulate the use of a more advanced face recognition system in collaborative scenarios.

To support up to four users at once, each user places a smartphone in front of them as described above. To ensure that four smartphones can easily be fixed in a position to each other, we created a small connector widget that uses a magnetic locking mechanism to hold the cases in one of four possible positions. To identify in which position the new smartphone is placed, GazeConduits compares the compass readings of all smartphones. By comparing the compasses of each tablet and smartphone, GazeConduits is also capable of roughly determining the orientation of each tablet. This enables GazeConduits to, e.g., change the display orientation of a tablet dynamically based on which user is looking at it.

## STUDY 1: EVALUATING GAZE-TO-TABLET TRACKING

To understand how well the gaze tracking of the phone and our correction method can identify which tablet a user is looking at, we created a game for a user study with 10 participants (23–35 years,  $M = 28.73$ ,  $SD = 3.31$ , two female). In this study, users had to perform a gaze-and-drop interaction similar

to the content transfer techniques introduced by Turner et al. [40].

### Apparatus and Task

We set up GazeConduits with one iPhone X that tracked the participant’s gaze and 20 shared tablets placed on the grid around it as shown in Fig. 1. Participants were asked to sit in front of the smartphone such that it could track their face. At any time, the system provided feedback by displaying a crosshair on the shared tablet they were looking at. In addition, they held one private tablet that they were instructed to use for touch input. The goal of the game was to feed octopuses on the shared tablets with a shrimp that was being displayed on the private tablet. This task mimics a typical cross-device object movement operation, in which a user wants to transfer an object from one tablet to another.

At the beginning of each trial, all shared tablets only showed a blue water background, and the private tablet displayed a 3 × 3 cm large shrimp at a random location. As soon as participants touched and held the shrimp, an octopus appeared on one of the shared tablets. Participants then had to find the tablet with the octopus on it and, while looking at that tablet, release the touch from the shrimp on their private tablet. As soon as they released their touch, octopus and shrimp disappeared, and a new trial was started. With this study design, we ensured that participants looked at their private screen at the start of each trial. Each octopus appeared four times on each of the 20 tablets in random order, such that each participant had to conduct 80 trials, for a total of 800 recorded gaze selections across 10 participants.

### Variables

Since we were mostly interested in how reliable the system could detect which tablet the participants were looking at and if this differed between tablet positions, we used the TABLET POSITION as independent variable. As dependent variables, we measured SUCCESS [0,1] if the system was able to identify that the user looked at the correct target, and the task completion TIME [s] for each trial.

### Results

The overall success rate was 95.58%, with an average task completion time of 2.13 s (SD: 1.325 s). To analyze the relation between TABLET POSITION and SUCCESS events, we calculated the SUCCESS RATE in percent for each TABLET POSITION and each user, and used a repeated-measures ANOVA to analyze the data. TABLET POSITION had no significant main effect on the SUCCESS RATE ( $F_{8,1336} = 779, p = 0.0758$ ).

For task completion TIME, we used a repeated-measures ANOVA on the log-transformed data. TABLET POSITION had a significant main effect on TIME ( $F_{19,779} = 8.13, p < .001$ ). Tukey HSD post-hoc pairwise comparisons showed that the tablet positions are divided into four groups that are significantly different from each other in terms of task completion time. Figure 5 shows these groups.

### Discussion

Our study results highlight that GazeConduits is capable of reliably identifying which tablet a user is looking at. The



**Figure 5. Task completion times in the Gaze-to-tablet study. Each color represents a significance level. Participants were able to select the green tablets fastest, followed by yellow, orange, and red.**

results for SUCCESS RATE show that the system was able to identify each tablet position in the same way. However, since the system displayed a cursor that indicated the tablet at which participants were looking, participants were able to correct the selection by moving their eyes and head until the system selected the correct tablet. Task completion times suggest that for most tablet positions (group 1 and 2), GazeConduits was able to directly identify the tablet the user was looking at. This is also in line with our observation that for most of these tablet positions, it was sufficient for participants to just naturally look at them.

However, for the tablet positions in group 3 and 4, the system was not always able to identify which tablet users were looking at, and the cursor jumped between multiple tablets. Most users tried to stabilize the cursor by actively moving their heads and eyes towards the tablet position, resulting in longer task completion times. Especially for tablet positions in group 4, we could not find a final explanation why it took participants more time to select them. We hypothesize that, because all of them were surrounded by several other tablets, and due to inaccuracy, the system was not always sure which tablet participants were looking at, thus jumping frequently between different tablets. However, this is also true for other tablets positions like 9 and 10 that were selected significantly faster.

### STUDY 2: EVALUATING GAZE-TO-PERSON TRACKING

To evaluate how GazeConduits tracks the gaze of four users at the same time, and how reliably it can detect if a user is looking at another user, we conducted a second study ( $n=12$ , age 22–33,  $M=26.17, SD=3.02$ , two female).

### Apparatus and Task

Setup and task were similar to the first study, except now with four participants around the table instead of one, all playing the game simultaneously (Fig. 1). In this setup, the game acted merely as a distractor to keep participants busy and engaged. The actual task we focused on in our analysis was that participants had to look at other participants from time to time. This was triggered by a notification on the participant’s

private tablet, displaying the ID of the other participant. The participant then had to look at the person indicated and confirm this selection with a touch on their private tablet. During the study, each participant had to look at every other person eight times, for a total of 24 selections per participant.

### Variables

As independent variable, we used the COLLABORATOR LOCATION (left, front, right) that described at which other person a participant had to look. As dependent variables, we used the same as in the first study: SUCCESS [0,1] and task completion TIME [s] for each trial. We measured the time from when the notification appeared on the private tablet until participants confirmed their selection.

### Results

The overall success rate was 95.14%, with an average task completion time of 2.26 s ( $SD = 0.882$  s) across all trials and participants. As in the first study, we calculated the SUCCESS RATE as a percentage for each COLLABORATOR LOCATION and each user, and used a repeated-measures ANOVA to analyze this data. COLLABORATOR LOCATION had no significant main effect on SUCCESS RATE ( $F_{2,284} = 1.2, p = 0.0858$ ). For TIME, we used ANOVA on the log-transformed data, which did not show a significant difference between the COLLABORATOR LOCATION conditions ( $F_{2,284} = 0.55, p = 0.574$ ).

### Discussion

The results of this study showed that GazeConduits can reliably detect when a user is looking at a particular other user. It also shows that the shared virtual 3D space between the four smartphones is stable enough to support such interactions across users. However, task completion time was relatively large. This was likely due to the game being too much of a distractor, as we often observed participants searching for the next octopus instead of noticing the notification on their private tablet.

## DISCUSSION AND SCENARIOS

GazeConduits takes a step towards gaze as a real-world input modality, by removing the requirement for calibration altogether, while expanding its functionality to include multiple users and devices simultaneously, all with off-the-shelf devices. We believe that the increased availability of gaze tracking has the potential to enable a variety of new cross-device, collaborative interaction scenarios. To illustrate this potential, we present two scenarios that GazeConduits enables, and that make use of its potential for user awareness around a table, gaze-at-device interactions, and gaze-at-user interactions.

### Interactions through GazeConduits' User Awareness

The first scenario makes use of the fact that GazeConduits maintains a map of users' presence around the table, detecting who enters and leaves the tracking space where and when.

In this scenario, *GazePoker* (Fig. 6), each user can only see their own set of cards. When a player temporarily gets up and leaves the table, her cards flip to blanks, such that a neighboring player cannot peek. Only when the player returns does GazeConduits authenticate her and restore her cards. Similarly,



**Figure 6.** GazePoker prevents cheating, by continuously authenticating users in front of their cards based on their faces. When a player leaves in GazePoker, or another peeks in, those cards are hidden.

when a second player attempts to peek while the first person is present, GazePoker detects this and temporarily hides the cards until the second player has left the frame again. This behavior is enabled by GazeConduits' continuous tracking and authentication of surrounding users. Due to the player detection implemented in GazePoker, the users could even switch seats or change devices while maintaining privacy, i.e., players will always see their cards and their cards only.

### Interactions through Gaze-at-Device Tracking

In our second scenario, we leverage the continuous tracking of people's gaze across the tablet devices to allow for a different gaze&touch interaction technique for cross-device group collaboration.

This scenario, *GazeMirror*, suggests a new technique for users to rapidly mirror the content of any other device onto the screen of their own tablet directly in front of them. The mirroring is triggered when a person performs a four-finger multi-touch gesture on their own tablet while targeting their gaze at another tablet device on the table. As long as the person holds the four-finger touch gesture on their screen, they can now view and, using their other hand, interact with the mirrored remote content on their local device. Releasing the touch ends the mirroring. The technique is designed as a light-weight, ad-hoc technique for collaborative settings.

This technique also enables users to transfer content from the remote device to the local device by touching the object that should be moved while releasing the four-finger mirroring gesture. When the mirroring ends, the object stays at the user's finger and is moved to the local device. GazeMirror can also be used to mirror the content of the gaze-tracking smartphone to a tablet. Example applications for this gesture are content transfer from the user's smartphone to a private or public tablet, such as a private message, or accepting a call that a user received during the cross-device session. Since only the user who is tracked by the smartphone can select it using gaze, she is the only one who can mirror content from her phone to a different device, accounting for privacy concerns.

GazeMirror also facilitates cross-device collaboration. In traditional setups, each user maintains their own tablet, and collaboration oftentimes involves invading each others' private space. The form factor of tablets further complicates this collaboration, as the limited screen real estate does not allow for multiple users to comfortably interact with the same device. GazeMirror addresses this issue by creating a working copy of one user's screen on the other user's device, so both can work on the same content together. This allows users to work

on their own device within their own private space, yet to collaborate in real-time with each other.

### Interactions through Gaze-at-Users Detection

GazeConduits maintains the location of each user, and can also detect if a user is looking at another person. We use this feature to detect if two users look at each other. This virtual handshake can trigger actions that both users have agreed upon. If a user wants to edit the personal content of another user, she first asks for permission by looking at the content and then at the owner of the content. If the owner looks back at the user and performs a particular touch gesture, the owner provides the user permission to edit the content. This feature can also be used for synchronized object transfer between two users.

### LIMITATIONS AND FUTURE WORK

GazeConduits is primarily designed to allow users to interact with a set of mobile devices using gaze input in an ad-hoc cross-device setting without the need for gaze calibration. However, GazeConduits was only evaluated with tablets of a certain size, and it can only detect that a user is looking at a tablet, but not at which location on the tablet. To support smaller devices such as smartphones, or to detect the exact gaze location on a tablet, gaze tracking accuracy would need to be improved. This could be achieved by using more sophisticated tracking and calibration methods. However, we also anticipate an increase in gaze tracking accuracy, as it only recently became a feature in commodity devices, and a more widespread adoption may lead to increased demand. While the feature of gaze tracking is only available in the most recent models of smartphones, a more widespread availability will allow for more reliable use of GazeConduits due to expected improvements of the tracking accuracy, with the basic interaction technique remaining the same. For example, the face detection in the iPhone 11 covers a wider area, which allows GazeConduits to track the users face and gaze in a larger area in front of the device.

When devices are within easy reach, using touch can be more useful than gaze interaction. However, in general, using gaze can resolve situations in which a device is out of reach (the distance to the furthest tablet could be over 1 m), and even when within reach, gaze can provide a better solution when reaching into another user's personal space to pick up a tablet would be awkward—and if the other user is actually holding a tablet, then, rather than grabbing that tablet out of their hands, GazeConduits will help to select that tablet using the person as a proxy (see our Study 2).

The maximum table depth is limited by the phone's tracking area and resolution, to ensure that users' faces remain recognizable. GazeConduits is currently limited to four users that are sitting at fixed positions. Depending on the available technology, this can increase to more users in specific scenarios. In some scenarios, these constraints match the actual use case well. For example, GazePoker entails social constraints (staying close to your cards), and user positions tend to be stationary in this scenario.

GazeConduits currently does not support moving tablets on the table, as it cannot directly track the position of tablets.

While the system can detect when devices are moved, it does not detect their location, and very slow movements are currently not detected. Therefore, users have to update positions manually on the device when prompted. In future work, we want to explore methods in which users can use their gaze to specify the location of a tablet, which requires a more accurate gaze tracking algorithm.

### SUMMARY AND CONCLUSION

We presented GazeConduits, a system that uses a combination of gaze and touch input in a collaborative ad-hoc setting. The addition of gaze tracking features to commodity devices such as smartphones has opened up new avenues for research, which we exploited in our application of GazeConduits. Our implementation of the gaze tracking system for a collaborative scenario does not require calibration, special hardware, or time-consuming setup – users can simply sit down at a table, connect their devices to the GazeConduits system, and start using a combination of gaze and touch to interact with multiple devices and users seamlessly. Our evaluation shows error rates of less than 5%, even with up to four users, four phones, four private tablets, and 20 shared tablets in one scenario combined. Not only can users interact with all these devices, but GazeConduits can distinguish between different users, which enables additional new interaction techniques. We highlight scenarios that arise from those new opportunities, to provide an outlook of how gaze input can shape the future design of interactive technologies.

### ACKNOWLEDGMENTS

We thank Kerstin Kreutz for helping with the video. This work was funded in part by the German B-IT Foundation, the Velux Foundations grant: Digital Tools in Collaborative Creativity (grant no. 00013140), and the Aarhus University Research Foundation grant: Creative Tools.

### REFERENCES

- [1] Frederik Brudy, Christian Holz, Roman Rädle, Chi-Jui Wu, Steven Houben, Clemens Klokmoose, and Nicolai Marquardt. 2019. Cross-Device Taxonomy: Survey, Opportunities and Challenges of Interactions Spanning Across Multiple Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. DOI:<http://dx.doi.org/10.1145/3290605.3300792>
- [2] Frederik Brudy, Steven Houben, Nicolai Marquardt, and Yvonne Rogers. 2016. CurationSpace: Cross-Device Content Curation Using Instrumental Interaction. In *Proceedings of the 2016 ACM on Interactive Surfaces and Spaces (ISS '16)*. ACM, New York, NY, USA, 159–168. DOI:  
<http://dx.doi.org/10.1145/2992154.2992175>
- [3] G. Goswami, M. Vatsa, and R. Singh. 2014. RGB-D Face Recognition With Texture and Attribute Features. *IEEE Transactions on Information Forensics and Security* 9, 10 (Oct 2014), 1629–1640. DOI:  
<http://dx.doi.org/10.1109/TIFS.2014.2343913>
- [4] Jens Grubert and Matthias Kranz. 2017. HeadPhones: Ad Hoc Mobile Multi-Display Environments Through

- Head Tracking. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 3966–3971. DOI : <http://dx.doi.org/10.1145/3025453.3025533>
- [5] Peter Hamilton and Daniel J. Wigdor. 2014. Conductor: Enabling and Understanding Cross-device Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 2773–2782. DOI : <http://dx.doi.org/10.1145/2556288.2557170>
- [6] Ken Hinckley. 2003. Synchronous Gestures for Multiple Persons and Computers. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology (UIST '03)*. ACM, New York, NY, USA, 149–158. DOI : <http://dx.doi.org/10.1145/964696.964713>
- [7] Ken Hinckley, Morgan Dixon, Raman Sarin, François Guimbretière, and Ravin Balakrishnan. 2009. Codex: A Dual Screen Tablet Computer. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, New York, NY, USA, 1933–1942. DOI : <http://dx.doi.org/10.1145/1518701.1518996>
- [8] Ken Hinckley, Gonzalo Ramos, François Guimbretière, Patrick Baudisch, and Marc Smith. 2004. Stitching: Pen Gestures That Span Multiple Displays. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI '04)*. ACM, New York, NY, USA, 23–31. DOI : <http://dx.doi.org/10.1145/989863.989866>
- [9] Leila Homaeian, Nippun Goyal, James R. Wallace, and Stacey D. Scott. 2018. Group vs Individual: Impact of TOUCH and TILT Cross-Device Interactions on Mixed-Focus Collaboration. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 73, 13 pages. DOI : <http://dx.doi.org/10.1145/3173574.3173647>
- [10] Gang Hu, Derek Reilly, Mohammed Alnusayri, Ben Swinden, and Qigang Gao. 2014. DT-DT: Top-down Human Activity Analysis for Interactive Surface Applications. In *Proceedings of the Ninth ACM International Conference on Interactive Tabletops and Surfaces (ITS '14)*. ACM, New York, NY, USA, 167–176. DOI : <http://dx.doi.org/10.1145/2669485.2669501>
- [11] Shahram Izadi, Harry Brignull, Tom Rodden, Yvonne Rogers, and Mia Underwood. 2003. Dynamo: A Public Interactive Surface Supporting the Cooperative Sharing and Exchange of Media. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology (UIST '03)*. ACM, New York, NY, USA, 159–168. DOI : <http://dx.doi.org/10.1145/964696.964714>
- [12] Robert J. K. Jacob. 1990. What You Look at is What You Get: Eye Movement-based Interaction Techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '90)*. ACM, New York, NY, USA, 11–18. DOI : <http://dx.doi.org/10.1145/97243.97246>
- [13] Hans-Christian Jetter and Harald Reiterer. 2013. Self-Organizing User Interfaces: Envisioning the Future of Ubicomp UIs. In *Workshop Blended Interaction (in conjunction with CHI '13 ACM SIGCHI Conference on Human Factors in Computing Systems)*. University of Konstanz. <http://nbn-resolving.de/urn:nbn:de:bsz:352-250974>
- [14] Haojian Jin, Christian Holz, and Kasper Hornbæk. 2015. Tracko: Ad-hoc Mobile 3D Tracking Using Bluetooth Low Energy and Inaudible Signals for Cross-Device Interaction. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 147–156. DOI : <http://dx.doi.org/10.1145/2807442.2807475>
- [15] Mohamed Khamis, Florian Alt, and Andreas Bulling. 2018. The Past, Present, and Future of Gaze-enabled Handheld Mobile Devices: Survey and Lessons Learned. In *Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '18)*. ACM, New York, NY, USA, Article 38, 17 pages. DOI : <http://dx.doi.org/10.1145/3229434.3229452>
- [16] Clemens N. Klokrose, James R. Eagan, Siemen Baader, Wendy Mackay, and Michel Beaudouin-Lafon. 2015. Webstrates: Shareable Dynamic Media. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM Press, 280–290. DOI : <http://dx.doi.org/10.1145/2807442.2807446>
- [17] K. Krafska, A. Khosla, P. Kellnhofer, H. Kannan, S. Bhandarkar, W. Matusik, and A. Torralba. 2016. Eye Tracking for Everyone. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2176–2184. DOI : <http://dx.doi.org/10.1109/CVPR.2016.239>
- [18] R. Langner, T. Horak, and R. Dachsel. 2018. VisTiles: Coordinating and Combining Co-located Mobile Devices for Visual Data Exploration. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (Jan. 2018), 626–636. DOI : <http://dx.doi.org/10.1109/TVCG.2017.2744019>
- [19] Andrés Lucero, Jussi Holopainen, and Tero Jokela. 2011. Pass-them-around: Collaborative Use of Mobile Phones for Photo Sharing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 1787–1796. DOI : <http://dx.doi.org/10.1145/1978942.1979201>
- [20] Nicolai Marquardt, Frederik Brudy, Can Liu, Ben Bengler, and Christian Holz. 2018. SurfaceConstellations: A Modular Hardware Platform for Ad-Hoc Reconfigurable Cross-Device Workspaces. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 100, 12 pages. DOI : <http://dx.doi.org/10.1145/3173574.3173647>

- Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, Article Paper 354, 14 pages. DOI: <http://dx.doi.org/10.1145/3173574.3173928>
- [21] Nicolai Marquardt, Ken Hinckley, and Saul Greenberg. 2012. Cross-device Interaction via Micro-mobility and F-formations. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*. ACM, New York, NY, USA, 13–22. DOI: <http://dx.doi.org/10.1145/2380116.2380121>
- [22] Emiliano Miluzzo, Tianyu Wang, and Andrew T. Campbell. 2010. EyePhone: Activating Mobile Phones with Your Eyes. In *Proceedings of the Second ACM SIGCOMM Workshop on Networking, Systems, and Applications on Mobile Handhelds (MobiHeld '10)*. ACM, New York, NY, USA, 15–20. DOI: <http://dx.doi.org/10.1145/1851322.1851328>
- [23] Miguel A. Nacenta, Carl Gutwin, Dzmitry Aliakseyeu, and Sriram Subramanian. 2009. There and Back Again: Cross-Display Object Movement in Multi-Display Environments. *Human-Computer Interaction* 24, 1-2 (2009), 170–229. DOI: <http://dx.doi.org/10.1080/07370020902819882>
- [24] Michael Nebeling. 2016. Cross-Device Interfaces: Existing Research, Current Tools, Outlook. In *Proceedings of the 2016 ACM International Conference on Interactive Surfaces and Spaces (ISS '16)*. ACM, New York, NY, USA, 513–516. DOI: <http://dx.doi.org/10.1145/2992154.2996361>
- [25] Ken Pfeuffer, Jason Alexander, Ming Ki Chong, and Hans Gellersen. 2014. Gaze-Touch: Combining Gaze with Multi-Touch for Interaction on the Same Surface. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*. Association for Computing Machinery, New York, NY, USA, 509–518. DOI: <http://dx.doi.org/10.1145/2642918.2647397>
- [26] Ken Pfeuffer, Jason Alexander, Ming Ki Chong, Yanxia Zhang, and Hans Gellersen. 2015. Gaze-Shifting: Direct-Indirect Input with Pen and Touch Modulated by Gaze. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. Association for Computing Machinery, New York, NY, USA, 373–383. DOI: <http://dx.doi.org/10.1145/2807442.2807460>
- [27] Ken Pfeuffer, Jason Alexander, and Hans Gellersen. 2016. GazeArchers: Playing with Individual and Shared Attention in a Two-Player Look&shoot Tabletop Game. In *Proceedings of the 15th International Conference on Mobile and Ubiquitous Multimedia (MUM '16)*. Association for Computing Machinery, New York, NY, USA, 213–216. DOI: <http://dx.doi.org/10.1145/3012709.3012717>
- [28] Ken Pfeuffer and Hans Gellersen. 2016. Gaze and Touch Interaction on Tablets. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 301–311. DOI: <http://dx.doi.org/10.1145/2984511.2984514>
- [29] Jun Rekimoto. 1997. Pick-and-drop: A Direct Manipulation Technique for Multiple Computer Environments. In *Proceedings of the 10th Annual ACM Symposium on User Interface Software and Technology (UIST '97)*. ACM, New York, NY, USA, 31–39. DOI: <http://dx.doi.org/10.1145/263407.263505>
- [30] Jun Rekimoto, Yuji Ayatsuka, and Michimune Kohno. 2003. SyncTap: An Interaction Technique for Mobile Networking. In *Human-Computer Interaction with Mobile Devices and Services*, Luca Chittaro (Ed.). Springer, Berlin, Heidelberg, 104–115.
- [31] Christian Remy, Malte Weiss, Martina Ziefle, and Jan Borchers. 2010. A Pattern Language for Interactive Tabletops in Collaborative Workspaces. In *Proceedings of the 15th European Conference on Pattern Languages of Programs (EuroPLoP '10)*. Association for Computing Machinery, New York, NY, USA, Article Article 9, 48 pages. DOI: <http://dx.doi.org/10.1145/2328909.2328921>
- [32] Roman Rädle, Hans-Christian Jetter, Jonathan Fischer, Inti Gabriel, Clemens N. Klokmoose, Harald Reiterer, and Christian Holz. 2018. PolarTrack: Optical Outside-In Device Tracking That Exploits Display Polarization. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, 497:1–497:9. DOI: <http://dx.doi.org/10.1145/3173574.3174071>
- [33] Roman Rädle, Hans-Christian Jetter, Nicolai Marquardt, Harald Reiterer, and Yvonne Rogers. 2014. HuddleLamp: Spatially-Aware Mobile Displays for Ad-hoc Around-the-Table Collaboration. ACM Press, 45–54. DOI: <http://dx.doi.org/10.1145/2669485.2669500>
- [34] Roman Rädle, Hans-Christian Jetter, Mario Schreiner, Zhihao Lu, Harald Reiterer, and Yvonne Rogers. 2015. Spatially-aware or Spatially-agnostic?: Elicitation and Evaluation of User-Defined Cross-Device Interactions. ACM Press, 3913–3922. DOI: <http://dx.doi.org/10.1145/2702123.2702287>
- [35] Adalberto L. Simeone, Julian Seifert, Dominik Schmidt, Paul Holleis, Enrico Rukzio, and Hans Gellersen. 2013. A Cross-device Drag-and-drop Technique. In *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia (MUM '13)*. ACM, New York, NY, USA, Article 10, 4 pages. DOI: <http://dx.doi.org/10.1145/2541831.2541848>
- [36] Sophie Stellmach and Raimund Dachselt. 2012. Investigating Gaze-supported Multimodal Pan and Zoom. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '12)*. ACM, New York, NY, USA, 357–360. DOI: <http://dx.doi.org/10.1145/2168556.2168636>

- [37] Sophie Stellmach and Raimund Dachsel. 2013. Still Looking: Investigating Seamless Gaze-supported Selection, Positioning, and Manipulation of Distant Targets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, New York, NY, USA, 285–294. DOI: <http://dx.doi.org/10.1145/2470654.2470695>
- [38] Norbert A. Streitz, Jörg Geißler, Torsten Holmer, Shin'ichi Konomi, Christian Müller-Tomfelde, Wolfgang Reischl, Petra Rexroth, Peter Seitz, and Ralf Steinmetz. 1999. i-LAND: An Interactive Landscape for Creativity and Innovation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99)*. ACM, New York, NY, USA, 120–127. DOI: <http://dx.doi.org/10.1145/302979.303010>
- [39] Jayson Turner, Jason Alexander, Andreas Bulling, and Hans Gellersen. 2015. Gaze+RST: Integrating Gaze and Multitouch for Remote Rotate-Scale-Translate Tasks. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 4179–4188. DOI: <http://dx.doi.org/10.1145/2702123.2702355>
- [40] Jayson Turner, Andreas Bulling, Jason Alexander, and Hans Gellersen. 2014. Cross-device Gaze-supported Point-to-point Content Transfer. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '14)*. ACM, New York, NY, USA, 19–26. DOI: <http://dx.doi.org/10.1145/2578153.2578155>
- [41] Jayson Turner, Andreas Bulling, and Hans Gellersen. 2011. Combining Gaze with Manual Interaction to Extend Physical Reach. In *Proceedings of the 1st International Workshop on Pervasive Eye Tracking & #38; Mobile Eye-based Interaction (PETMEI '11)*. ACM, New York, NY, USA, 33–36. DOI: <http://dx.doi.org/10.1145/2029956.2029966>
- [42] Vincent van Rheden, Bernhard Maurer, Dorothé Smit, Martin Murer, and Manfred Tscheligi. 2017. LaserViz: Shared Gaze in the Co-Located Physical World. In *Proceedings of the Eleventh International Conference on Tangible, Embedded, and Embodied Interaction (TEI '17)*. ACM, New York, NY, USA, 191–196. DOI: <http://dx.doi.org/10.1145/3024969.3025010>
- [43] Roel Vertegaal, Aadil Mamuji, Changuk Sohn, and Daniel Cheng. 2005. Media Eyepliances: Using Eye Tracking for Remote Control Focus Selection of Appliances. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems (CHI EA '05)*. ACM, New York, NY, USA, 1861–1864. DOI: <http://dx.doi.org/10.1145/1056808.1057041>
- [44] Simon Voelker, Andrii Matvienko, Johannes Schöning, and Jan Borchers. 2015. Combining Direct and Indirect Touch Input for Interactive Workspaces Using Gaze Input. In *Proceedings of the 3rd ACM Symposium on Spatial User Interaction (SUI '15)*. Association for Computing Machinery, New York, NY, USA, 79–88. DOI: <http://dx.doi.org/10.1145/2788940.2788949>
- [45] Colin Ware and Harutune H. Mikaelian. 1987. An Evaluation of an Eye Tracker As a Device for Computer Input. In *Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface (CHI '87)*. ACM, New York, NY, USA, 183–188. DOI: <http://dx.doi.org/10.1145/29933.275627>
- [46] Daniel Wigdor, Hao Jiang, Clifton Forlines, Michelle Borkin, and Chia Shen. 2009. WeSpace: The Design Development and Deployment of a Walk-up and Share Multi-surface Visual Collaboration System. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '09)*. ACM, New York, NY, USA, 1237–1246. DOI: <http://dx.doi.org/10.1145/1518701.1518886>
- [47] Erroll Wood and Andreas Bulling. 2014. EyeTab: Model-based Gaze Estimation on Unmodified Tablet Computers. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '14)*. ACM, New York, NY, USA, 207–210. DOI: <http://dx.doi.org/10.1145/2578153.2578185>
- [48] Paweł Wozniak, Nitesh Goyal, Przemysław Kucharski, Lars Lischke, Sven Mayer, and Morten Fjeld. 2016. RAMPARTS: Supporting Sensemaking with Spatially-Aware Mobile Interactions. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 2447–2460. DOI: <http://dx.doi.org/10.1145/2858036.2858491>
- [49] Chi-Jui Wu, Steven Houben, and Nicolai Marquardt. 2017. EagleSense: Tracking People and Devices in Interactive Spaces Using Real-Time Top-View Depth-Sensing. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 3929–3942. DOI: <http://dx.doi.org/10.1145/3025453.3025562>
- [50] Shumin Zhai. 2003. What's in the Eyes for Attentive Input. *Commun. ACM* 46, 3 (March 2003), 34–39. DOI: <http://dx.doi.org/10.1145/636772.636795>
- [51] Shumin Zhai, Carlos Morimoto, and Steven Ihde. 1999. Manual and Gaze Input Cascaded (MAGIC) Pointing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99)*. ACM, New York, NY, USA, 246–253. DOI: <http://dx.doi.org/10.1145/302979.303053>
- [52] Yanxia Zhang, Ken Pfeuffer, Ming Ki Chong, Jason Alexander, Andreas Bulling, and Hans Gellersen. 2017. Look together: using gaze for assisting co-located collaborative search. *Personal and Ubiquitous Computing* 21, 1 (Feb. 2017), 173–186. DOI: <http://dx.doi.org/10.1007/s00779-016-0969-x>