HaptiCase: Back-of-Device Tactile Landmarks for Eyes-Free Absolute Indirect Touch

Christian Corsten Christian Cherek Thorsten Karrer Jan Borchers
RWTH Aachen University
52074 Aachen, Germany
{corsten, cherek, karrer, borchers}@cs.rwth-aachen.de

ABSTRACT
Using a smartphone for touch input to control apps and games mirrored to a distant screen is difficult, as the user cannot see where she is touching while looking at the distant display. We present *HaptiCase*, an interaction technique that provides back-of-device tactile landmarks that the user senses with her fingers to estimate the location of her finger in relation to the touchscreen. By pinching the thumb resting above the touchscreen to a finger at the back, the finger position is transferred to the front as the thumb touches the screen. In a study, we compared touch performance of different landmark layouts with a regular landmark-free mobile device. Using a landmark design of dots on a 3×5 grid significantly improves eyes-free tapping accuracy and allows targets to be as small as 17.5 mm—a 14% reduction in target size—to cover 99% of all touches. When users can look at the touchscreen, landmarks have no significant effect on performance. HaptiCase is low-cost, requires no electronics, and works with unmodified software.

Author Keywords
Eyes-free touch; back-of-device interaction; tactile feedback

ACM Classification Keywords
H.5.2. User Interfaces: Input Devices and Strategies

INTRODUCTION
User interfaces (UIs) for smartphone applications rely on a touchscreen, which collocates a touch sensor for touch input with a display for graphical output. The mapping between the touch input and the graphical output is *absolute*, i.e., the absolute position of a touch on the touch input layer maps to an absolute position of the graphical output layer. Furthermore, since these layers are collocated, the user is given the illusion of *directly* touching what she is seeing: To hit a touch target, the user looks at the display to locate the target and then moves her finger towards it and touches the display.

There are scenarios, for which a mirroring of the graphical output to a shared screen, such as a TV, is desired. One example is multi-player games. The players follow the game on the big shared screen, while using their smartphone's touch input to control the game. A similar use case are interactive presentations where slides are mirrored from a smartphone to a projector. While presenting, the presenter uses the smartphone's touch input to interact with the content, such as playing a particular video on the current slide while looking at the projection or the audience.

CHI 2015, April 18–23 2015, Seoul, Republic of Korea
Copyright 2015 ACM 978-1-4503-3145-6/15/04...$15.00
http://dx.doi.org/10.1145/2702123.2702277

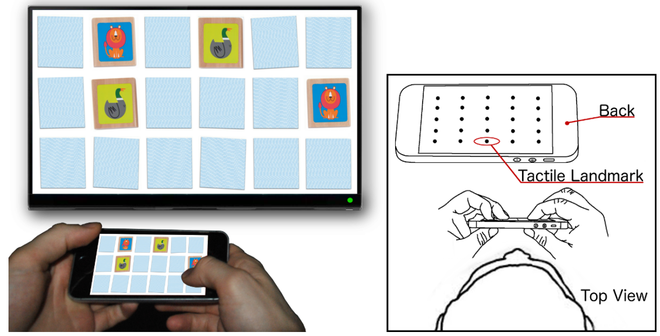

Figure 1. HaptiCase brings tactile landmarks to the back of mobile touchscreens to allow eyes-free touch interaction by proprioceptive pinching of finger and thumb. This enables, e.g., to play tapping games, mirrored to a big screen without the need to look at the touchscreen.

When the user focuses at the graphical output on the shared screen, she must be able to hit touch targets eyes-free, i.e., without looking at the smartphone display. However, this is difficult [10]: The user does not receive any feedback about her finger position relative to the touchscreen. Consequently, a target might be missed or—even worse—a different target might be selected accidentally.

To alleviate this problem, the user has to glance back and forth between the touchscreen and the shared screen, which interrupts the flow and quality of the interaction. Alternatively, we could use other senses than the visual sense to guide the user to where her finger will land on the touchscreen, before actually touching it. HaptiCase follows this approach by providing a generic grid of tactile landmarks on the back of the device that help the user estimate where her finger is located in relation to the dimensions of the touchscreen: The user holds her smartphone in her hand and spots a target of interest on the shared screen. First, she mentally translates the target position to the corresponding position on the touchscreen. Then, she moves her finger across the tactile landmarks on the back to estimate the target position. When the right position is found, the user transfers her finger position at the back to the touchscreen at the front by simply pinching the thumb resting above the touchscreen to the finger. This causes the thumb to touch the screen at the corresponding location (Fig. 1). The pinching is based on human proprioception and works eyes-free.

HaptiCase is a lightweight solution to eyes-free absolute touch: It is low-cost, as it only requires generic landmarks on the back of smartphone. These can be either provided by the phone manufacturer, or the user can buy a landmark-equipped protection case and clip it to the phone. Since only the back of the phone is modified, normal use of the touchscreen is still possible. Additionally, HaptiCase does not require any software modification. Using built-in screen mirroring, the smartphone can mirror the graphical output of any application to a shared screen.

HaptiCase could also bring eyes-free touch input for touch-based UIs where the touch input layer differs from the graphical output on the shared screen, such as remote control applications. For frequently used applications with a simple UI, the user tends to build up spatial memory of target positions on the touch input layer, as shown in [12]. Even further, this could allow users to control smartphone applications without the need of referring to a graphical output: For example, a car driver could use HaptiCase to find the “answer phone call”-button with HaptiCase while keeping her eyes on the road.

This paper makes two contributions: (1) We present the HaptiCase interaction technique, that enables eyes-free absolute touch on mobile touchscreens via back-of device tactile landmarks. (2) We showed that HaptiCase significantly improves the accuracy of hitting targets eyes-free compared to when using a default landmark-free smartphone.

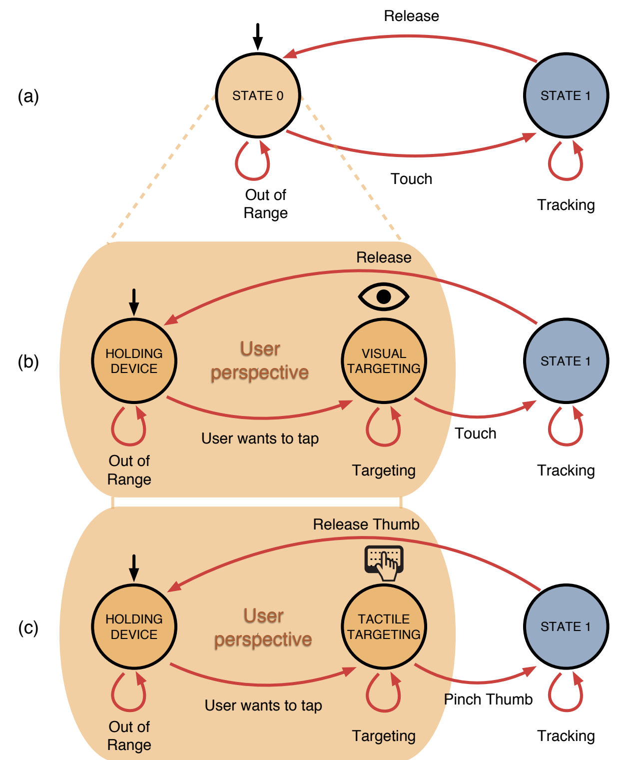

Figure 2. (a) Buxton’s Two-State Model for touch input devices. (b) State 0 broken down into user’s individual interaction steps: To hit a target, the user visually acquires the tapping position first. (c) For eyes-free touch, HaptiCase introduces *Tactile Targeting*, which informs the user via landmarks on the back of the device about where her touch will land prior to tapping the screen.

THE HAPTICASE INTERACTION TECHNIQUE

Interaction with a touchscreen follows Buxton’s Two-State Model [5] (Fig. 2a). As long as the user’s finger is not in contact with the touchscreen, *State 0* is maintained as the finger is *out of range*. When the finger touches the screen, *State 1* is entered. As long as the the finger is on the screen it is *tracked* and *State 1* is maintained. When the finger is *released*, *State 0* is re-entered. Typically, to hit a target, the user touches the screen and immediately lifts her finger again, called *tapping*, which instantly fires an associated event. Tapping requires the user to hit a target on her first attempt; touching an adjacent target could fire an undesired event. Corrections before lift-off are not possible, e.g., moving the finger away from the target could execute a different gesture. To hit an on-screen object correctly, the user first hovers her finger above the touchscreen at the location of the target (*State 0*) and then taps the screen (*State 1*). While targeting, the user watches her fingertip move to obtain feedback.

As Buxton’s model only models the interaction from the perspective of the input device, Fig. 2b shows an extension of Buxton’s Two-State Model that breaks down *State 0* into two distinct steps in the interaction from the perspective of the user: She starts in *Holding Device* and then finds the position for her finger to touch the screen by *Visual Targeting*. This state is maintained as long as the user moves her finger above the touchscreen towards the target. Touching the screen enters *State 1*, like in Buxton’s Two-State Model. This model applies to tapping tasks, as long as the user can watch her finger above the device. However, when the user cannot look at the input device, the visual feedback gained from this *Visual Targeting* state is lost, and tapping may fail as hitting the wrong target is likely.

HaptiCase introduces a *Tactile Targeting* state to absolute touch instead, for situations in which the user’s locus of attention is not the touchscreen. HaptiCase provides tactile landmarks on the back of the smartphone that correspond to the dimensions of the touchscreen at the front. Starting again in *Holding Device*, this time the user can obtain *tactile* position feedback by moving her fingers over the tactile landmarks (*Tactile Targeting*). When the user senses a landmark close to the position of the target of interest, she simply *pinches* her thumb resting above the touchscreen to the finger on the back. Since the touchscreen is located between finger and thumb, this pinch causes the thumb to touch the screen at the position equivalent to that of the landmark, and *State 1* is entered. This back-to-front location transformation is based on human proprioception and can be performed without looking.

RELATED WORK

HaptiCase combines (1) absolute indirect touch with (2) a-priori tactile feedback through (3) back-of-device interaction. Tactile patterns for eyes-free discrimination and guidance is prevalent in industrial design. We will highlight related research from each of these fields in turn and contrast them to HaptiCase.

Absolute Indirect Touch

Gilliot et al. [10] compared different input conditions when tapping a mobile touchscreen using an absolute indirect mapping. They found that using the same aspect ratio for input and output surface, and being able to look at the input device even when nothing is displayed on it, enables users to acquire smaller targets ($\geq$ 16.8 mm) compared to when being blindfolded ($\geq$ 23.0 mm). However, the influence of tactile landmarks on eyes-free touch was not investigated.

ARC-Pad [20] is a smartphone-based trackpad that combines absolute indirect touch with relative pointing. To select a remote target, the user taps the touchscreen to roughly position a virtual cursor on the distant screen based on an absolute mapping. When sliding the finger, ARC-Pad switches implicitly to relative pointing for fine adjustment. ARC-Pad reduced clutching by half compared to native cursor acceleration. Nancel et al. [21] extended ARC-Pad with a two-finger interaction technique for coarse positioning of the cursor on large wall displays. Alternatively, head-tracking was used to determine the initial cursor position. Although users were able to look at the input device, ARC-pad should also work eyes-free, since the user receives visual feedback through the cursor. However, for immediate tapping without a cursor, as used in typical smartphone apps, ARC-pad will not work.

Pietroszek and Lank [23] investigated absolute indirect touch on a 48$\times$36 mm^2 touchscreen connected to a 2.5$\times$2.0 m^2 projection screen. In one condition, the touchscreen was blank, in the other condition, the projection was mirrored to the touchscreen. The targeting error for tapping was twice as big for the non-mirrored condition compared to when visual feedback was shown on the touchscreen (62 px vs. 34 px resp. 3.07 mm vs. 1.68 mm). However, the results are means over different target sizes, ranging from 2.4–12.0 mm, and users did look at the input surface in both conditions.

Imaginary Interfaces [12] makes use of absolute (indirect) touch by using the human palm as input surface mapped to an invisible GUI based upon the user's spatial memory of frequently used UIs, such as the home screen of a smartphone. Absolute touch positions of the index finger on the palm of the non-dominant hand are transferred and scaled to the corresponding position in the imaginary interface. Typically, the user moves the finger over the palm and the system reads out the targets aloud first, before the user taps to engage. Despite not being able to see the referred-to GUI, users can reliably acquire 17.7 mm diameter targets. The authors state that targeting accuracy is most influenced by the user ability to watch their hands interact. However, tactile cues sensed by the palm and the index finger [12] also contribute to targeting accuracy. Imaginary Interfaces work best for frequently used interfaces, as the user must be able to memorize the spatial layout of a GUI. However, combining this ability to memorize spatial UI layout with our HaptiCase technique could enable low-cost and technically simple in-pocket smartphone interaction.

Feed-Forward for Eyes-Free Touch Interaction

Directly touching a target without being able to look at the input surface is difficult [10]. A typical approach to compensate, aimed at making touchscreens accessible to the visually impaired, are *touch-and-explore* interfaces, e.g., [15]. The user explores available targets by sliding her finger across the touchscreen. When a target is crossed, audio output, announces the underlying target. To engage, the user has lifts her finger and taps again at the same location. This technique, however, introduces a mode that prohibits immediate target selection by tapping, and target reading might interfere with background sound, e.g., in games.

To haptically explore touch targets, tactile overlays can be added on top of the touchscreen. The Tactile Talking Tablet [18] uses relief paper to enable blind people to spot locations on a map. When touched, the tablet reads out the target. Transparent overlays such as [8, 17, 18], provide cutouts to make both the location and boundaries of on-screen targets tactile. These are, however, application-specific, i.e., each UI requires a different overlay. TouchPlates [16] are more generic, but not designed for mobile touchscreens.

Guerreiro et al. [11] attached tactile grids to smartphone and a tablet touchscreens. Blind users orient themselves along the grid and tap the adjacent area to perform a touch. Users stated that "the border helps with localization, and it is also a safe place to rest the fingers.". Also, participants performed significantly better in corner areas. However, only up to twelve static touch targets are supported. A drawback of tactile cues on the touchscreen is that gestures, such as flicking or dragging, cannot be executed smoothly anymore since the cues act as barriers to the finger. HaptiCase also makes use of tactile cues, but they are located on the back of the device, so that the touchscreen remains unobstructed for gestures.

BrailleTouch [9] provides a tactile grid of seven buttons on the back of a smartphone to type Braille. The user holds the device with both hands in landscape mode, but with the back facing towards her. Three virtual buttons on the left and right of the touchscreen and one center button are used to type braille with three fingers per hand. The grid on the back corresponds to virtual buttons on the screen, but it was rather used for support of the thumbs since the device is ergonomically difficult to hold. Compared to HaptiCase, BrailleTouch is held in reverse and the user uses the thumb to feel tactile landmarks. HaptiCase extends the idea of tactile landmarks to a wider field of applications. Unfortunately, BrailleTouch did not investigate the impact of back-of-device landmarks.

Apart from tactile cues on the front or back of the touchscreen, physical landmarks can also be placed on the side to guide eyes-free touch input. PocketMenu [22] exploits the screen edge of a smartphone as a tactile landmark. All targets are aligned vertically to the screen edge, such that the user only needs to move her finger up and down the border. However, this touch-and-explore approach reduces a UI to 1D, limiting the number of touch targets. Buzzi et al. [6] present a concept based on tactile spheres on the left side of a mobile phone screen to physically mark the logical UI segments, such as a navigation or title bar. Moving the finger to the right reads out all targets within a segment. However, this concept requires dynamic rendering of tactile cues to be compatible with different UIs.

Tactile feedback can also be rendered dynamically. SemFeel [31], renders tactile feedback on the back of the device using vibration motors. Vibration patterns indicate different touch targets, but only if the user knows the vibration vocabulary.

TeslaTouch [1], UltraHaptics [7], VacuumTouch [13], MudPad [14], Programmable Friction [19], and FingerFlux [26] render tactile feedback on top or above a touch surface using focused ultrasonic, air suction, electric vibration, or electro magnets. VacuumTouch and FingerFlux also allow to attract a user's finger to a touch target. The latter also uses repelling force to push the finger away. MudPad uses a magnetic fluid pouch whose viscosity is selectively controlled by electromagnets to render a relief of the UI. Yet, all systems require a complex hardware setup and do not have a mobile form factor. The effects emitted by TeslaTouch and Programmable Friction are more suitable for dragging than tapping.

Like FingerFlux, PreSense [24] informs the user that she is about to engage with a target before actually doing so, also called feedforward. Each physical key of a 4×5 keypad has an embedded binary capacitive sensor. This way, PreSense acts as a low-resolution touchpad. When the user touches a key, she receives visual feedback from a GUI regarding which key she is about to press. If the right key is touched, she can then press the key down to engage. PreSense is compact, provides tactile feedback through physical keys, and uses an absolute indirect mapping to a distant UI. However, it has low resolution, and unlike typical touchscreens, the "touchpad" can be pressed, which allows for an easy distinction between "about to engage" and actually engaging with the device.

Back-of-Device Interaction

So far, the work presented investigated touch at the front of a touchscreen. Wigdor et al. [28] extended traditional input on direct-touch tabletops with a touch-sensitive underside, which can be used for absolute indirect touch input to a distant screen. A study on the accuracy of under-the-table tapping revealed that the missed target rate and first touch error were significantly higher compared to targeting on the top surface, as users cannot see their hands when interacting under the table. No tactile landmarks were used for guidance.

Wobbrock et al. [30] investigated human performance of hand postures in front- and back-of-device interaction with mobile touchscreens/pads. Users had to drag targets to the end of a virtual or horizontal line using either the index finger or thumb while holding the device with one or two hands. The results showed that using the index finger for touch input works well for both the front and the back, but performance dropped when using the thumb for input on either side. Yet, the study did not investigate tapping. Furthermore, users could look at the input device and watch their fingers.

Motivated by the problem that a finger occludes an on-screen target when touching it, Wigdor et al. [27] extended a mobile resistive touchscreen with a camera that captured finger input on the back of the device. The captured shape of the fingers is displayed on the touchscreen as a transparent overlay, giving the user the illusion of seeing through the device. A follow-up prototype by Baudisch and Chu [2] investigated back-of-device interaction for small screens to address the fat finger problem. Their 2.4" prototype uses a capacitive trackpad on the back with absolute mapping to the frontal screen, which visualizes touch interaction with a virtual finger behind on-screen targets. According to their study, targets must be at least 12.2×12.2 mm^2 to obtain a targeting error rate below 10% (without user calibration). Despite a separation of input and output surface, both devices still use direct touch as the user sees her "finger" through the screen.

Using a-priori feedback, absolute tapping is still possible for eyes-free absolute indirect touch. HaptiCase demonstrates that this can be achieved with little technical complexity based on tactile landmarks and human proprioception.

Industrial Design

The idea of using tactile patterns for haptic discrimination and guidance is already used for industrial design. For example, Bradley et al. [3] investigated different tactile patterns for both discriminability and manipulability of cylindrical knobs. Users were able to discriminate between smooth, fluted, or knurled rims of the knobs, but also differences in diameter ($\geq$ 0.5 inches) and thickness ($\geq$ 0.375 inches) helped for discrimination. Burnett et al. [4] claim an increased need for haptic cues for control interfaces for future cars. The authors state that tactile guidance for dashboard controls minimize the need for vision, which will play an important role for screen-based interfaces used in modern cars. Similarly, HaptiCase aims at a compensation for vision by the use of tactile guidance. El Saddik et al. [25] report that the just noticeable difference for two tactile stimuli is 2.5 mm. HaptiCase therefore uses landmarks that are at least 7.5 mm apart, as mentioned in the subsequent section.

DESIGNS FOR HAPTICASE

We developed five different tactile landmark designs for HaptiCase. Fig. 3 shows all designs alongside the landmark-free baseline, which we called *Base*. Each of the designs spans the exact size of the back of the device. The simplest landmark design, *Frame*, has a raised rectangle that has the absolute position and dimension of the touchscreen at the front. In contrast to *Base*, it allows users to feel where the touchscreen is at the front and what its size is. Because we consider this an important landmark, all HaptiCase designs include this reference frame. To provide more tactile landmarks, we created a design with small raised dots that were distributed on a 3×5 grid, centered inside the reference frame (*DotsL*). The adjacent dots are all slightly gradient such that the user's finger slides smoothly over them. We also added a second version of the same design with a higher density of dots *DotsH*. Here, the dots are laid out on a 5×9 grid, with the dots half as far away from each other as on the 3×5 grid. As an alternative to the discrete grid of dots, we created two designs, that used continuous lines, (*LinesL* and *LinesH*).

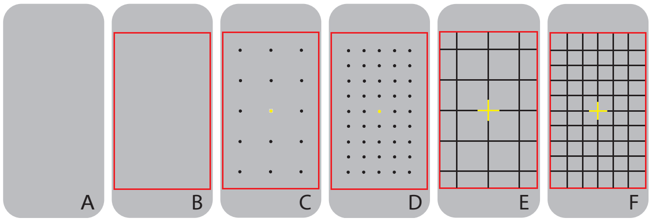

Figure 3. HaptiCase designs: A = *Base*, B = *Frame*, C = *DotsL*, D = *DotsH*, E = *LineL*, F = *LineH*. B–F provide tactile landmarks, such as the touchscreen position (red frame). Center highlighted in yellow.

Preliminary Study on Ergonomics

We originally envisioned HaptiCase to be used in two orientations: A one-handed *portrait* orientation and a two-handed *landscape* orientation. For one-handed use, the thumb rests above the touchscreen, while the remaining fingers hold the back. When using two hands, both thumbs rest above the touchscreen and the remaining fingers of each hand cover the the left and the right half of the device back. Holding the device in these postures allows the user to move the fingers on the back to explore the tactile landmarks.

To check these assumptions, we investigated whether holding the device in either orientation while moving the finger on the back and performing touches on the front was feasible. While we envision the technique to be largely independent of the device form factor as long as the whole back of the device can be comfortably reached, we chose to use a smartphone for the study; this form factor is in widespread use and can be used one-handed as well as two-handed.

We gave an iPhone 5s equipped with a *LinesH* HaptiCase to 10 users and let them perform absolute indirect touch tasks for 10 mm targets—slightly bigger then the minimum size for buttons in iOS—displayed at random positions on a distant screen. Users alternated between holding the phone in portrait and landscape orientation and were instructed to explore the tactile landmarks and touch the targets as accurate as possible without looking at the touchscreen. A customized bezel was attached to the distant screen to match the portrait and landscape orientation of the phone. We observed the users' hand posture and recorded whether a target was hit or failed. We also collected qualitative feedback afterwards.

Results

Landscape orientation worked fine with regards to ergonomics: Using two hands gave enough support to hold the phone firmly enough to explore the landmarks and perform corresponding touches. One-handed portrait mode, however, was difficult. Especially for targets located at the corners, users complained that it was hard to hold the device in a firm grasp while simultaneously sensing the cues and performing touches. In fact, some users involuntarily dropped the phone. Furthermore, users complained that they often had to the re-grasp the phone, as the range of their fingers did not fully cover the back of the device.

Based on these results, we conclude that with smartphones the technique works best for two-handed interaction, which is also the prevalent mode for larger mobile devices like phablets. For one-handed use, we would have to reduce how often users need to reposition their hand, e.g., by limiting the interaction to a smaller screen region. For the further studies, we therefore focused on the two-handed case.

Manufacturing the Designs

We manufactured prototypes for all five designs and the baseline by gluing acrylic sheets (123.8×58.6 mm^2, 2.0 mm thick) to the back of off-the self iPhone 5s compatible clip-on cases. We then engraved the designs with a lasercutter, such that the haptic cues were raised by 0.45 mm. The raised frame measured 90.25×51.6 mm^2, the dots were 2.5 mm in diameter and spaced 17.6 mm apart for *DotsL* (8.8 mm for *DotsH*), and the lines were 2.0 mm wide and used the same spacing. To smoothen the engraved areas, we laser-cut thin plastic sheets and glued them into the abraded areas.

STUDY 1: EYES-FREE TAPPING ACCURACY

The goal of the first study was to investigate users' eyes-free tapping accuracy for our HaptiCases compared to the landmark-free *Base*. We hypothesized that (1) tapping an on-screen target with any HaptiCase is more accurate than with *Base*, (2) users are more accurate with high resolution HaptiCase designs compared to low resolutions, and (3) the users are less precise in tapping targets towards the center as opposed to targets close to the screen edges.

Apparatus

To perform the absolute indirect touch task, users used an iPhone 5s with a 4" screen (1136×640 px^2). The touchscreen was blank throughout the study. The phone was wirelessly connected to a notebook to transmit the collected touch data. The task was presented on 30" display (2560×1600 px^2) to the user. The aspect ratio of the UI on the display matched that of the phone, as recommended for absolute indirect touch by [10]. A white cardboard bezel with a 284×160 mm^2 cutout at the center was attached to the screen so that the UI was standing out from the rest of the display. The display was put on a desk (height 74 cm), with users sitting 120 cm away. The chair had a height of 50 cm, and the upper visible line on the screen was at 105 cm. The screen was orthogonally aligned to the table. To prevent our participants from looking at their fingers and the touchscreen bezel, we built a $45 \times 32 \times 23$ cm^3 box with a cutout such that the users held the device in their hands beneath the box (Fig. 4).

Task

During our study, participants had to perform multiple sessions of an absolute indirect touch tapping task. Each individual trial began with the presentation of a new target. On the distant display, users were asked to hit this target by tapping the touchscreen of the phone held with two hands in landscape mode with either the left or right thumb. To estimate where the target was located, a user could use the tactile landmarks provided by the different HaptiCase designs.

The targets were colored gray with a white cross hair, the background of the screen was black. When a user touched the input device she received visual feedback as to where the finger hit the touchscreen: red if the user missed the target,

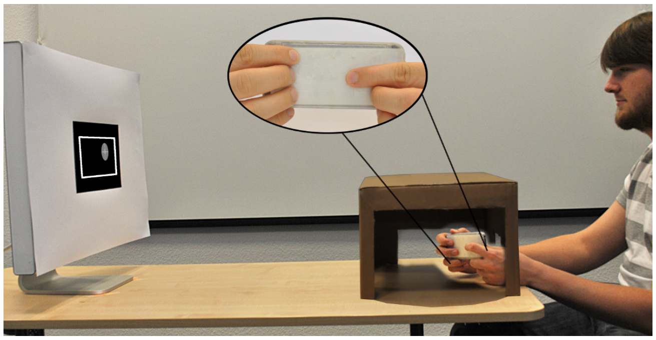

Figure 4. Study Setup. The user uses the HaptiCase smartphone to tap targets displayed at the big screen without looking at the input device.

green if they hit. After lifting the finger, the feedback faded out. As typical buttons are rectangularly shaped, the user received a hit feedback every time the center of the touch point was located within a 15 mm square around the target center. While the feedback was visible on screen, no new touch event was processed to prevent accidental double touches. The 15×15 mm^2 target size on the input device corresponded to a target size of 34.4×34.4 mm^2 on the distant screen. If the user hit the target, the next one was presented; if not, they had to repeat the task for the same target until it was hit.

For each touch we logged the exact position of the target and the touch, as well as whether the user hit or missed the target. As the user probably assumes all landmarks to be equidistant from each other, which is, however, not true for the outermost landmarks with respect to the frame, we displayed a visual outline on the screen matching the position of the dots resp. crossings closest to the frame (Fig. 4). The participants were asked to be as accurate as possible, and not to try to be as fast as possible. For completeness, we also logged the time users needed to tap each target. After each set of trials, the users were encouraged to take a break.

Participants

In total, 24 users (aged 21–33, M = 24.95, 5 left-handed, 8 females) participated in our first study. All users regularly use a smartphone (display size: 3.5–5.0").

Study Design

The factors were CASE (6 levels: *Base*, *Frame*, *DotsL*, *DotsH*, *LinesL*, and *LinesH*) and TARGET (28 levels, Fig. 5). Each user tested each case for all 28 targets (within subjects design). The targets were chosen to cover most of the the screen. With respect to the visual outline, targets 18–27 were located completely within, and targets 0–17 were at least partially on the outline. With respect to the landmarks, targets 0–4, 6, 8, 10, 11, and 22 were located exactly on a dot or crossing for *Dots/LinesL*, and targets 5, 7, 9, 18, 20, 21, 23, 24, 26, 27 were located on landmarks of *DotsH* and *LinesH*. Targets 12–17 are located directly at the border of the screen, and targets 19 and 25 are not located on a any landmark for all designs. For our evaluation, we categorized these targets into three TARGET GROUPS: *Key* targets (0–3, and 22) cover all corners and the center and are potentially easy to locate because of their outstanding position. *Border* targets (4–17) are located close to the edge of the touchscreen but could be more difficult to hit accurately than *Key* targets. *Middle* targets (18–27, w/o 22) are potentially the hardest to hit correctly (cf. [10]).

The sequence of CASEs for each user was balanced using a Latin square. The sequence of TARGETs was pseudo-randomized. To make themselves familiar with the HaptiCase design, the users performed a set of ten training trials before each session started. They were allowed to inspect the current CASE before starting each trial. Each target was presented exactly once for each case. Since we had 28 different targets, we got a sufficient amount of trials for each TARGET GROUP. For each user we recorded 6×28 touches. After each session, users were asked to fill out a questionnaire on their strategy used to hit targets for the currently tested CASE.

Results

Whether successful or not, we only included the users' first touch attempt for each target, since HaptiCase is meant to be used for tapping tasks. For each CASE, touches that were off more than three SDs from the respective mean were marked as outliers, which resulted in 44 removals (1.1%).

We defined OFFSET as the euclidean distance from the center of each target to the center of each corresponding touch for the first attempt. Since OFFSET by CASE was neither normally distributed, nor log-normally distributed, we applied an aligned rank transform (ART) [29] on the data for all analyses. We then conducted a two-way repeated-measures ANOVA on the aligned ranks.

Target Offset by Case. There was a significant main effect of CASE on OFFSET ($F_{5,3797}$ = 8.06, $p < .0001$). Tukey HSD post-hoc pairwise comparison revealed that both *Dots* designs were significantly different from *Base* ($p < .0001$, each), *Frame* ($p < .001$, each), and *LinesH* ($p < .05$, each). Other pairwise comparisons were not significantly different. Table 1 lists mean, SD, and the 95% CI for OFFSET by CASE. Using *Frame* (7.46 mm), targeting was slightly more accurate than *Base* (7.63 mm), but performed almost the same as *LinesH* (7.48 mm), which could indicate that the high density of lines was rather perceived as a consistent surface with a frame around it. Again, overall, OFFSET for *Base* is worst, and for *DotsH* (6.70 mm) and *DotsL* (6.75 mm), users performed the most accurately. Since both *Lines* designs and the *Frame* did not perform significantly better than *Base*, we remove these designs from further analyses.

Target Offset by Target Group. We compared *Base* against *DotsL/HighL* regarding the users' target offset for our TARGET GROUPs. Fig. 6 illustrates OFFSET by TARGET GROUP. OFFSET for *DotsH* was significantly smaller for all groups compared to *Base* ($p < .05$, each). For *DotsL*, only for *Key* and *Border* targets, users' OFFSET was significantly smaller compared to *Base* ($p < .01$, each). Yet, for *Middle* targets, a trend can be reported ($F_{1,403.1}$ = 3.47, p = .0631).

Hence, we can conclude that using HaptiCase (*Dots*), users can hit all target categories more precisely compared to having no landmarks at the back. *Key* targets lead to the smallest offset ($\leq$ 5.61 mm), followed by *Border* targets ($\leq$ 6.49 mm).

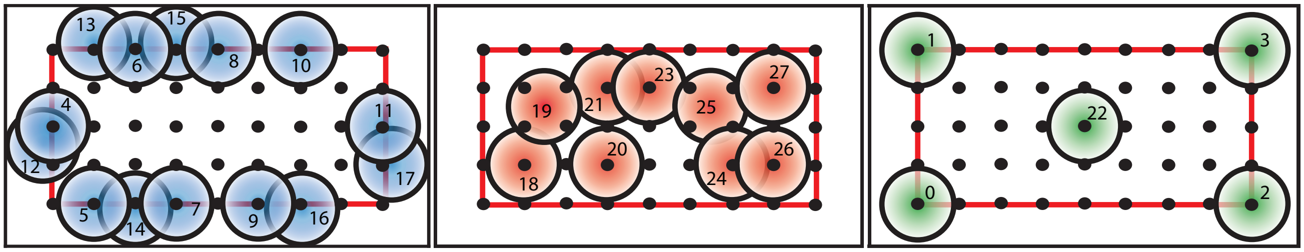

Figure 5. TARGET GROUPS for study 1. Left: *Border*, center: *Middle*, and right: *Key* targets. The visual outline is highlighted in red.

Condition	Offset [mm]				Time [ms]	
			95% CI			
	Mean	SD	lower	upper	Mean	SD
Base	7.63	4.06	7.32	7.94	1503.34	572.48
Frame	7.46	3.81	7.17	7.75	1996.50	1663.50
DotsL	6.75	3.88	6.46	7.04	1944.86	911.66
DotsH	6.70	3.60	6.43	6.98	2200.36	1268.99
LinesL	7.16	3.91	6.86	7.46	2093.10	1178.25
LinesH	7.48	4.07	7.17	7.79	2159.70	1220.37

Table 1. Summary of the results for OFFSET for study 1.

Middle targets lead to the highest offset ($\leq$ 7.99 mm), exceeding the set target width of 15 mm (7.5 mm to both sides).

Against our expectations, targets at the center of the top border (*Top*, targets 6, 8, 10, 13, 15) were significantly more difficult to hit accurately (7.08–7.87 mm) compared to the rest of the border targets (*Base*, *Top*: 8.54 mm, *Border*: 6.56 mm, $p < .0001$; *DotsH*, *Top*: 7.19 mm, *Border*: 6.11 mm, $p < .01$; *DotsL*, *Top*: 7.12 mm, *Border*: 5.94 mm, $p < .05$). One possible reason for this observation is that these targets are farthest away from the thumbs; when using the fingers to feel the tactile cues on the back, the thumbs need to be fully stretched to reach targets at this location. Still, users' targeting accuracy for *Top* targets was significantly better for *DotsH* (*DotsL*) than for *Base*: 7.19 mm (7.12 mm) vs. 8.54 mm ($p < .001$). Hence, despite a more relaxed stretching of the thumbs for the base case since the user does not need to use the fingers on the back to sense landmarks, users still performed more accurate for *Top* targets using HaptiCase.

Dot Resolution. Since there was no significant difference for the target offset for *DotsL* and *DotsH* over all targets, we looked closer at specific targets that were located on a dot for *DotsH*, but located at the center of four dots for *DotsL* (targets 5, 7, 9, 18, 20, 21, 23, 14, 26, and 27, see Fig. 5). A repeated measures ANOVA revealed no significant difference for OFFSET for both resolutions ($F_{1,435.2} = 0.69$, $p = .4078$). On average, the target offset for these targets was 7.51 mm (*SD* = 4.25 mm) for low resolution and 7.19 mm (*SD* = 3.72 mm) for high resolution. A higher resolution with dots placed at the center of four dots did not improve users' targeting accuracy.

Learning Effects. For each CASE, we contrasted the OFFSET from users who tested a particular case design at the beginning with the OFFSET from users who tested that case design at the end. Learning effects were significant for *DotsL* (–2.31 mm), *DotsH* (–1.16 mm), and Frame (–1.40 mm) (ANOVA, $p < .01$, each), which is counterbalanced by the Latin square design. Regarding learning effects within using one case design, we contrasted the first 14 appearing targets with the remaining half for the cases that were tested at the beginning. An ANOVA showed significance only for *Base* ($F_{1,107} = 2.23$, $p < .05$).

Time. Table. 1 lists the tapping time per case. CASE had a significant main effect on time (ANOVA on log-transformed data: $F_{5,3967} = 76.13$, $p < .0001$). *Base* was significantly faster compared to all other cases (Tukey HSD, $p < .0001$). Using *DotsL*, however, is only 442 ms slower and still significantly faster than *Dots/LinesH* (Tukey HSD, $p < .01$, each).

User Feedback
Users were asked to give insight into which strategy they used to hit the targets. For *Base*, 15 users reported to have used pure guessing. Some users mentioned that they tried to set their fingers as constraints to orientate themselves in future trials. For *DotsL*, 15 users moved their finger over the tactile landmarks to find the correct position on the screen, and afterwards used a proprioceptive pinch to hit the target. By contrast, six users fixed their index-, middle-, and ring finger on the tactile landmarks and hit targets by approaching these fingers with the thumb instead of moving a finger around on the back of the device. For both *DotsH* and *LinesH*, some users commented that these denser patterns felt more like a texture than distinct landmarks; also, they could derive the same information by finding the center between two cues in the sparser patterns. Two users mentioned that they did not use the landmarks frequently, since they felt uncomfortable with the device posture or more accurate without landmarks.

Summary
From Study 1, we could confirm Hypothesis 1 and 3: Users were more accurate for eyes-free indirect tapping using HaptiCase compared to having no tactile landmarks at the back (best: *DotsL*; OFFSET: 6.75 mm, on average). The center target and targets and the corner were most accurately hit, followed by targets at the border, and targets in the middle. We could not confirm that a higher density of landmarks performs better than a lower one (Hypothesis 2), as there was no difference in accuracy for both *Dots* designs. Therefore, we concentrated on the *DotsL* design for our second study.

STUDY 2: TACTILE LANDMARKS VS. VISION
The goal of study 2 was to investigate the impact of tactile targeting on visual targeting, when both targeting strategies

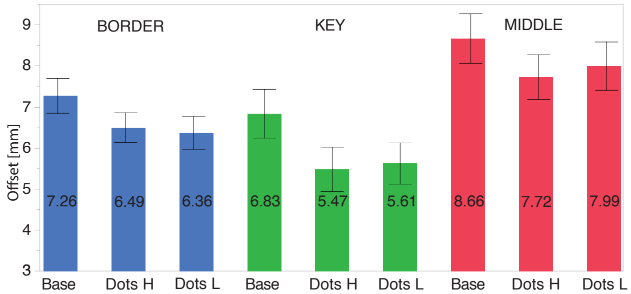

Figure 6. Mean OFFSET [mm] and 95% CI by TARGET GROUP.

are combined. We hypothesized that, when visual targeting and tactile targeting are combined, there is no significant difference in targeting performance or execution time compared to pure visual targeting. For this, we measured TARGET OFFSET and the time users need to hit targets for pure visual targeting, pure tactile targeting, and visual and tactile targeting combined. For tactile targeting, we used the *DotsL* design. Visual targeting allowed the users to look at the blank touchscreen. For combined targeting, we used the *DotsL* design and allowed the users to look at the blank input device.

Apparatus and Task

The second study used almost the same study setup as used in study 1. The only difference was that, for conditions in which the users were allowed to look at the input device, the cardboard box was removed. We also reused the task from the first study. In addition, we logged the time the user needed to perform a tap after the target was shown on the distant screen. We also removed the visual outline to see whether it had influenced OFFSET from the previous study.

Participants

Twelve users (aged 20–36, M = 24.74, one left-hander, five females) participated in the second study. All but one of the participants regularly use a smartphone (display size: 3.5–5.5”). None of these users participated in the first study.

Study Design

One factor was CONDITION with three levels: *Base* with vision (*Base Vis*), *DotsL* with vision (*DotsL Vis*), and *DotsL* without being able to look at the device (*DotsL NoVis*). The other factor, TARGET, was same as in study 1. CONDITION was counterbalanced (balanced Latin square), TARGET was pseudo-randomized. At the end of the study, our users were asked to rank their preference for each CONDITION.

Results

Condition and Target on Target Offset

As in study 1, we analyzed the effect of CONDITION and TARGET on OFFSET. Since OFFSET was neither normally distributed, nor log-normally distributed, we applied an ART [29]. Yet, there was a significant main effect of CONDITION on OFFSET ($F_{2,913}$ = 26.50, $p < .0001$). Tukey HSD post-hoc comparison showed a significant difference for OFFSET between *DotsL NoVis* and *Base Vis* and between *DotsL NoVis* and *DotsL Vis* (both $p < .0001$). Table 2 (left) shows mean

Condition	Offset [mm]				Time [ms]	
	Mean	SD	95% CI lower	95% CI upper	Mean	SD
Base Vis	5.36	3.42	4.99	5.72	1510.30	386.11
DotsL NoVis	6.99	3.62	6.60	7.38	1845.18	881.74
DotsL Vis	5.66	3.64	5.27	6.05	1636.88	636.44

Table 2. Summary of the results for OFFSET, and for TIME for study 2.

and SD for OFFSET per CONDITION. On average, users were 1.63 mm (1.33 mm) more precise in targeting when using *Base Vis* (*DotsL Vis*) compared to *DotsL NoVis*. There was no significant difference between the two vision conditions.

In comparison to the results from study 1, the targeting offset for eyes-free targeting with *DotsL* increased by 0.24 mm in study 2 from 6.75 mm to 6.99 mm, which fits within the upper bound of the 95% confidence interval from study 1. Furthermore, a one-way independent measured ANOVA comparing target offset (aligned rank-transformed) for this condition between Studies 1 and 2 was not significant ($F_{1,1004}$ = 1.63, p = .2018), confirming that the target accuracy was almost the same for both studies for *DotsL*. Since we omitted the visual outline in this study, but the results are similar to study 1, the visual outline did not have much influence for targeting.

Condition and Target on Time

TIME on CONDITION was neither normally distributed, nor log-normally distributed. Thus, we applied the ART for TIME [29]. We included the timings for both, successfully acquired and failed targets, since we did only analyze the first attempt. We conducted a repeated measures ANOVA on the transformed data. Interestingly, there was no significant main effect of TARGET on TIME, and there was no CONDITION×TARGET interaction effect. However, there was a significant main effect of CONDITION on TIME ($F_{2,994}$ = 20.98, $p < .0001$). Tukey HSD post-hoc comparison showed a significant difference between *Base Vis* and *DotsL NoVis* and between *DotsL Vis* and *DotsL NoVis* (both $p < .0001$). Table 2 (right) shows mean and standard deviation for TIME on CONDITION. As hypothesized, users were slowest when using HaptiCase without looking (+334.88 ms compared to *Base Vis*). Comparing both vision conditions, users were slower using HaptiCase compared to using *Base* (+126.58 ms). Yet, this difference was not significant.

Summary

In an absolute indirect touch task, users were both faster and had a lower offset to the target when being able to look at the input device (*Base Vis*, *DotsL Vis*) compared to when they had to perform the task eyes-free (*DotsL NoVis*); this was independent of the presence of tactile landmarks. Although we observed a positive effect of the tactile landmarks on the users’ accuracy for the *eyes-free setting* in study 1 (*Base* vs. *DotsL*), we could not observe the same positive effect in study 2 when users were able to look at the device during the interaction (*Base Vis* vs. *DotsL Vis*).

DISCUSSION

From our first study, we have learned that eyes-free *Tactile Targeting* outperforms eyes-free targeting without tactile landmarks for absolute touch tasks on distant screens. With respect to the landmark designs we presented, users' targeting performance is best for dots laid out on a static 3×5 grid. Increasing the landmark density of the dots does not further improve targeting accuracy. For the 3×5 dots design, 99% of all touches would hit targets successfully for an average target width of 14.3 mm (upper 99% CI). Some targets are easier to hit accurately than others, depending on where they are located: Targets that are located at the corners or at the center are hit most reliably and should be at least 12.6 mm wide. Targets located at the border are also hit reliably if they are at least 13.8 mm in size, and are not located along the center of the top edge. However, targets located in this special location are more difficult to hit accurately and should be at least 16.2 mm wide. Targets close to the center are most difficult to hit and must be at least 17.5 mm wide (all target widths based on upper 99% CI).

Interestingly, these target sizes are a lot smaller than the minimal target width of 23 mm reported by Gilliot et al. [10] for eyes-free indirect touch tasks without tactile landmarks. We assume that this difference is due to two aspects: (1) As we have shown, tactile landmarks significantly improve eyes-free targeting for absolute touch. (2) In contrast to the study design used by Gilliot et al., our users held the touchscreen with both hands while performing the tasks. Being able to feel the device dimensions may have contributed to the improved targeting accuracy, since the determined target size of 18.9 mm (upper 99% CI for most difficult to tap *Middle* targets) for our landmark-free condition is still better than in [10].

From our second study, we have learned that *Tactile Targeting* does not have a significant impact on targeting accuracy when combined with *Visual Targeting*. Users hit targets more accurately when they can look at the blank touchscreen compared to performing tapping tasks eyes-free when the targets are displayed at a distant screen (Table 2). Thus, the choice of not having to look at the input device, which is desirable for scenarios motivated at the beginning of the paper comes at a cost. For touch-based UIs that use targets larger than 17.5 mm wide, HaptiCase might be a good option to provide high targeting accuracy without having to look back and forth between distant screen and touchscreen. The memory game shown in Fig. 1, or the presentation scenario might be good candidates for this. For applications that typically have targets positioned towards the center, UI designers could design a customized UI that is used for the touch input layer when mirroring is activated (cf. Braille Touch [9]): Such UIs can provide smaller targets at the border and increased targets towards the center and the top. This would compensate the trade-off mentioned above. Another solution could be to adapt the layout of the tactile landmarks to the UI: The phone example described in the introduction is a candidate for this. The UI designer could position the targets, such as the "answer call" button, at the border of the screen, such that they can be easily reached eyes-free by *Tactile Targeting*.

LIMITATIONS

(1) As we have seen from our studies, users' tapping performance is significantly slower for *Tactile Targeting* compared to *Visual Targeting*. For games that build upon fast tapping at random locations pure *Tactile Targeting* will possibly not be satisfactory, at least when the UI is not optimized for HaptiCase. (2) Additionally, our study used only one device size. While using devices of a similar size will probably lead to similar performance, targeting accuracy for HaptiCase on larger devices, could be worse, as the thumbs might not easily reach the center of the screen in landscape mode. (3) Some users reported that they observed a strong offset from time to time between finger and thumb when they pinched. This offset tends to be stronger the more the device is tilted. However, in our studies, we did not fix the angle between the users' hands and the device. (4) We compared HaptiCase only with indirect absolute input without tactile landmarks and not with established indirect techniques like relative pointing with a cursor on the distant screen. While these techniques do not quite fit the envisioned use case, as they require the software to be modified, such a comparison would still be insightful.

FUTURE WORK

Apart from analyzing other device form factors as well, our next goal is to further improve targeting accuracy for HaptiCase. One approach is to study the angle-offset problem deeper. If this can be modeled in software, the touch offset could be canceled out. Alternatively, to stick with HaptiCase's benefit of unmodified software, a non-symmetric alignment of the tactile landmarks could be considered to counteract the offset for common angles at which a smartphone is held. We will also investigate additional design alternatives, such as mixtures of different landmark types or irregular haptic features and arrangements. Finally, we would also like to experiment with dynamic designs for HaptiCase that are based upon a braille display attached to the back of the phone. This way, HaptiCase could adapt to the touch UI and only provide selective landmarks only.

CONCLUSION

With this paper, we present *HaptiCase* as an initial foray into a new interaction technique that improves touch performance for eyes-free tapping to GUIs mirrored to a distant screen. HaptiCase provides tactile landmarks on the back of a smartphone that the user senses with her fingers while holding the device. Based on human proprioception, the user pinches her thumb resting above the touchscreen to a finger on a landmark at the back to transfer it into a touch at the front. We conducted an study to compare the users' touch performance for five different landmark designs against a baseline for eyes-free indirect touch. Using a landmark design of dots on a 3×5 grid significantly improves eyes-free tapping accuracy and allows targets to be as small as 17.5 mm—a 14% reduction in target size—to cover 99% of all touches. HaptiCase can also guide the user to eyes-free touch especially well for touch UIs whose targets are located near the border of the touchscreen and on the center. A second study showed that looking at the input device even though nothing is displayed on it outperforms tactile landmarks regarding accuracy and time.

ACKNOWLEDGMENTS

This work was funded by the German Research Foundation (DFG), grant BO 3093/1–1. We also thank Marcel Lahaye.

REFERENCES

1. Bau, O., Poupyrev, I., Israr, A., and Harrison, C. TeslaTouch: Electrovibration for Touch Surfaces. In *Proc. UIST* (2010), 283–292.
2. Baudisch, P., and Chu, G. Back-of-Device Interaction Allows Creating Very Small Touch Devices. In *Proc. CHI* (2009), 1923–1932.
3. Bradley, J. V. Tactual Coding of Cylindrical Knobs. *Human Factors: The Journal of the Human Factors and Ergonomics Society 9*, 5 (Oct. 1967), 483–496.
4. Burnett, G. E., and Porter, J. M. Ubiquitous Computing within Cars: Designing Controls for Non-Visual Use. *International Journal of Human-Computer Studies 55*, 4 (Oct. 2001), 521–531.
5. Buxton, W. A Three-State Model of Graphical Input. In *INTERACT* (1990).
6. Buzzi, M. C., Buzzi, M., Donini, F., Leporini, B., and Paratore, M. T. Haptic Reference Cues to Support the Exploration of Touchscreen Mobile Devices by Blind Users. In *Proc. CHItaly* (2013), 28:1–28:8.
7. Carter, T., Seah, S. A., Long, B., Drinkwater, B., and Subramanian, S. UltraHaptics: Multi-Point Mid-Air Haptic Feedback for Touch Surfaces. In *Proc. UIST* (2013), 505–514.
8. El-Glaly, Y. N., Quek, F., Smith-Jackson, T., and Dhillon, G. Touch-Screens are Not Tangible: Fusing Tangible Interaction with Touch Glass in Readers for the Blind. In *Proc. TEI* (2013), 245–252.
9. Frey, B., Southern, C., and Romero, M. BrailleTouch: Mobile Texting for the Visually Impaired. *HCI 6767*, Chapter 3 (2011), 19–25.
10. Gilliot, J., Casiez, G., and Roussel, N. Impact of Form Factors and Input Conditions on Absolute Indirect-Touch Pointing Tasks. In *Proc. CHI* (2014), 723–732.
11. Guerreiro, T., Jorge, J., and Gonçalves, D. Exploring the Non-Visual Acquisition of Targets on Touch Phones and Tablets. In *2nd Workshop on Mobile Accessibility, MobileHCI* (2012).
12. Gustafson, S. G., Rabe, B., and Baudisch, P. M. Understanding Palm-Based Imaginary Interfaces: The Role of Visual and Tactile Cues when Browsing. In *Proc. CHI* (2013), 889–898.
13. Hachisu, T., and Fukumoto, M. VacuumTouch: Attractive Force Feedback Interface for Haptic Interactive Surface Using Air Suction. In *Proc. CHI* (2014), 411–420.
14. Jansen, Y., Karrer, T., and Borchers, J. MudPad: Tactile Feedback and Haptic Texture Overlay for Touch Surfaces. In *Proc. ITS* (2010), 11–14.
15. Kane, S. K., Bigham, J. P., and Wobbrock, J. O. Slide Rule: Making Mobile Touch Screens Accessible to Blind People Using Multi-Touch Interaction techniques. In *Proc. ASSETS* (2008), 73–80.
16. Kane, S. K., Morris, M. R., and Wobbrock, J. O. Touchplates: Low-Cost Tactile Overlays for Visually Impaired Touch Screen Users. In *Proc. ASSETS* (2013), 22:1–22:8.
17. Kincaid, R. Tactile Guides for Touch Screen Controls. In *Proc. BCS HCI* (2012), 339–344.
18. Landau, S., and Wells, L. Merging Tactile Sensory Input and Audio Data by Means of the Talking Tactile Tablet. In *Proc. EuroHaptics* (2003), 414–418.
19. Levesque, V., Oram, L., MacLean, K., Cockburn, A., Marchuk, N. D., Johnson, D., Colgate, J. E., and Peshkin, M. A. Enhancing Physicality in Touch Interaction with Programmable Friction. In *Proc. CHI* (2011), 2481–2490.
20. McCallum, D. C., and Irani, P. ARC-Pad: Absolute+Relative Cursor Positioning for Large Displays with a Mobile Touchscreen. In *Proc. UIST* (2009), 153–156.
21. Nancel, M., Chapuis, O., Pietriga, E., Yang, X.-D., Irani, P. P., and Beaudouin-Lafon, M. High-Precision Pointing on Large Wall Displays Using Small Handheld Devices. In *Proc. CHI* (2013), 831–840.
22. Pielot, M., Kazakova, A., Hesselmann, T., Heuten, W., and Boll, S. PocketMenu: Non-Visual Menus for Touch Screen Devices. In *Proc. MobileHCI* (2012), 327–330.
23. Pietroszek, K., and Lank, E. Clicking Blindly: Using Spatial Correspondence to Select Targets in Multi-Device Environments. In *Proc. MobileHCI* (2012), 331–334.
24. Rekimoto, J., Ishizawa, T., Schwesig, C., and Oba, H. PreSense: Interaction Techniques for Finger Sensing Input Devices. In *Proc. UIST* (2003), 203–212.
25. Saddik, A. E., Orozco, M., Eid, M., and Cha, J. *Haptics Technologies: Bringing Touch to Multimedia*. Springer, Sept. 2011.
26. Weiss, M., Wacharamanotham, C., Voelker, S., and Borchers, J. FingerFlux: Near-Surface Haptic Feedback on Tabletops. In *Proc. UIST* (2011), 615–620.
27. Wigdor, D., Forlines, C., Baudisch, P., Barnwell, J., and Shen, C. Lucid Touch: A See-Through Mobile Device. In *Proc. UIST* (2007), 269–278.
28. Wigdor, D., Leigh, D., Forlines, C., Shipman, S., Barnwell, J., Balakrishnan, R., and Shen, C. Under the Table Interaction. In *Proc. UIST* (2006), 259–268.
29. Wobbrock, J. O., Findlater, L., Gergle, D., and Higgins, J. J. The Aligned Rank Transform for Nonparametric Factorial Analyses Using Only Anova Procedures. In *Proc. CHI* (2011), 143–146.
30. Wobbrock, J. O., Myers, B. A., and Aung, H. H. The Performance of Hand Postures in Front- and Back-of-Device Interaction for Mobile Computing. *Human-Computer Studies 66*, 12 (Dec. 2008), 857–875.
31. Yatani, K., and Truong, K. N. SemFeel: A User Interface with Semantic Tactile Feedback for Mobile Touch-Screen Devices. In *Proc. UIST* (2009), 111–120.