Gaze and Touch Interaction on Tablets

Ken Pfeuffer, Hans Gellersen
Lancaster University
Lancaster, United Kingdom
k.pfeuffer@lancaster.ac.uk, hwg@comp.lancs.ac.uk

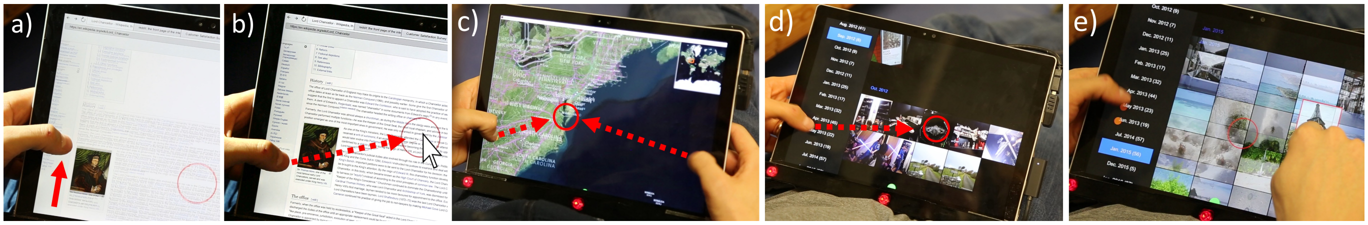

Figure 1. We explore gaze and touch input for tablet interaction during grip, e.g. to enable the user to scroll a webpage with direct-touch (a), and temporally use a cursor to click hyperlinks (b) with the same hand. A two-handed grip allows a comfortable two-thumb based zoom, while the zoom location is specified with gaze (c). Our gallery application allows users to open an image simply by looking at it and a tap from the gripping hand (d). Or, to drag images through direct-touch, while users indirectly scroll the view they look at (e). The red circle indicates the user's gaze in all figures.

ABSTRACT

We explore how gaze can support touch interaction on tablets. When holding the device, the free thumb is normally limited in reach, but can provide an opportunity for indirect touch input. Here we propose gaze and touch input, where touches redirect to the gaze target. This provides whole-screen reachability while only using a single hand for both holding and input. We present a user study comparing this technique to direct-touch, showing that users are slightly slower but can utilise one-handed use with less physical effort. To enable interaction with small targets, we introduce CursorShift, a method that uses gaze to provide users temporal control over cursors during direct-touch interactions. Taken together, users can employ three techniques on tablets: direct-touch, gaze and touch, and cursor input. In three applications, we explore how these techniques can coexist in the same UI and demonstrate how tablet tasks can be performed with thumb-only input of the holding hand, and with it describe novel interaction techniques for gaze based tablet interaction.

Author Keywords

Gaze; tablet; touch; cursor; indirect input; eye tracking.

ACM Classification Keywords

H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
UIST '16, October 16-19, 2016, Tokyo, Japan
© 2016 ACM. ISBN 978-1-4503-4189-9/16/10...$15.00
DOI: http://dx.doi.org/10.1145/2984511.2984514

INTRODUCTION

Previous work showed that gaze can provide interaction benefits in the context of interactive multi-touch surfaces [19, 20, 25, 30, 31]. A more popular and established multi-touch surface is a tablet, which provides a different challenge because of its portability. The multi-touch capabilities are affected by how users naturally hold the device. While holding, the thumb can theoretically add touch input, but is limited in reach [18, 29, 34]. With eye tracker integration on mobile devices becoming increasingly feasible [3, 23, 28], we explore how gaze can enhance touch interaction of the hand that holds the device.

We propose combined gaze and touch input on tablets, where touch input of the thumb is redirected to the gaze target. This provides whole-screen reachability with utilising a single hand for both holding and input. For instance, a direct-touch selection task can involve both hands: one hand holds the device, the other hand selects a target. Using gaze and touch, users can simply point their gaze at the target, and use the thumb of the holding hand to select the target. We conducted a user study comparing direct-touch to a previously introduced gaze and touch technique [19] in a homescreen based tapping task. Results show that gaze and touch was only slightly slower, but easily allowed one-handed single-thumb use on the whole screen with less physical effort.

In other applications, however, small targets can be difficult to select as our study also indicated that gaze has lower accuracy than direct-touch input. We aimed at support of precise touchpad-like cursor input, which is difficult because cursor dragging and direct-touch scrolling gestures conflict with each other. We therefore developed *CursorShift*, a method to temporally activate a cursor. The cursor activates at the user's gaze position when issuing a tap from the grip position, and deactivates after another tap which is also used to perform a 'click' on the cursor's target. This technique is particularly useful in browsers. Users can comfortably scroll a webpage using the thumb with the hand that holds the device (Figure 1a). Then, if the user wants to select a hyperlink, the user can utilise the same finger to instance a cursor, drag it precisely, and click the link (b).

With the three techniques direct-touch, gaze and touch, and cursor input potentially available, we explore how these techniques combine and their potential utility in three applications. In our browser application, we show that, in addition to interleaved scrolling and clicking, users can utilise gaze and touch to rapidly interact with the browser menu and the virtual keyboard. In our maps application, we show gaze-enhanced navigation techniques that, for example, allow users to choose the zooming target with their gaze when using a comfortable two-handed grip (Figure 1c). In our gallery application, we show how opening and closing images is simplified (d), gaze based scrolling, and how these techniques complement image dragging actions (e). Collectively, the interaction examples demonstrate how tablet interactions can be performed with single-finger, single-hand, and single-grip input, indicating that that the introduction of gaze on tablets can make tablet devices easier to use.

Our contributions are as follows. First, we present a user study comparing direct-touch to a previously introduced gaze and touch based method in a tablet tapping task. Second, we introduce CursorShift as a method that enables switching between direct-touch and an indirect cursor. Third, we present three applications that describe how input of the direct-touch, gaze and touch, and cursor combine in the same UI, demonstrate the potential utility of gaze, and with it introduce novel interaction techniques for tablet devices. Our work also shows that by introducing gaze we can reduce physical effort, and combine them for novel bimanual techniques.

RELATED WORK

Gaze and Touch Interaction

Gaze was initially used to support mouse cursor interaction, such as Zhai et al's MAGIC technique [37] that warps the cursor to the user's gaze position in order to eliminate large portions of cursor dragging. Researchers extended MAGIC to the touch modality, which allows refining inaccurate gaze cursors with precise touch dragging [6, 20, 26]. Our proposed CursorShift technique is based on the MAGIC principle: at tap, a cursor appears at the user's gaze. It is different as it does not improve cursor dragging per se — CursorShift is a method that enables temporal activation of a cursor for integration with direct-touch.

Researchers extensively studied combined gaze and touch input on interactive surfaces [19, 21, 26, 30]. On the same user interface, systems can integrate intelligent methods to enable both direct-touch and gaze and touch input [19, 20, 25, 31]. Studies on large displays indicate that users perform comparably whether using direct or gaze and touch [21, 22, 30], and found qualitative benefits such as remote, occlusion-free input or avoidance of the fat-finger issue [21, 22, 27, 30]. This motivated us to investigate gaze and touch in a tablet context. In particular, we leverage two previously introduced techniques in our work:

- *Gaze-touch* [19], a technique based on the division of labour 'gaze selects, touch manipulates'. Users look at a target, touch down from anywhere to select and manipulate it.
- *Gaze-shifting* [20], a technique to switch touch input between direct and indirect input. When users touch down where they look, it is processed as direct-touch. When users touch offset from their gaze, it is processed as indirect touch on the visual target (i.e., a gaze-touch).

Tablet Interaction

Researchers analysed the grip and reach issue of mobile devices [2, 18, 29, 34, 35]. For tablets in particular, users often employ diverse sets of grip, whether one- or two handed [11, 32]. During grip, the thumb is theoretically freely available for input — however quite limited in reach [18, 34].

To support interactions while holding the device, the user interface can be adapted to the finger's (or grip's) position by bringing remote UI elements to their vicinity. Relevant UI elements can be placed near common grip locations, to ease reach [32]. Grip sensing approaches detect position and posture of the grip on the device, and enable automatic placement of UI elements to it [4, 9, 10, 17]. For example, the GripSense system [9] uses motion sensing to detect handedness on a smartphone, and responds with appropriate thumb-only interaction techniques. Overall, these approaches enhance unimanual input and alleviate thumb-reach issues, but are limited to a fixed amount of UI elements.

As another approach, users can employ indirect input techniques from the grip position to interact with remote objects on the screen. A simple indirect method is controlling a cursor with an on-screen touchpad. More advanced techniques are inverse cursors (e.g., MagStick [24]) or miniature interaction areas (e.g., ThumbSpace [14]), but normally these techniques conflict with the direct-touch enabled display. Indirect touch techniques are usually slower than direct-touch, but allow reaching the whole screen [34]. To avoid conflicts with direct-touch input, indirect touch needs some kind of splitting interaction to direct/indirect input. E.g. by using special gestures such edge-swipe or 'large touch' [15], or placing indirect input areas on the back of the device [1, 33]. However, while resolving the conflict, these methods constrain to specific gestures and fixed areas. Instead, we investigate indirect input by gaze as it can be used complementary to direct-touch in the same area [19, 20].

Eye tracking is becoming increasingly feasible on mobile devices [3, 12, 28, 36]. The potential of *gaze-only* input was explored in mobile user interfaces for watches [8], phones [5, 7], and tablets [23]. We continue this direction with a focus on *combined gaze and touch* interaction on tablets for its rich expressive possibilities.

USER STUDY: HOMESCREEN TAPPING TASK

We conducted a controlled study on gaze and touch interaction to investigate the feasibility of the previously introduced gaze-touch technique [19] in the new tablet context. We chose a homescreen based interface for our study design, as they are commonly used in tablets, and as targets are relatively large alleviating potential gaze inaccuracy [16, 26]. We compare two techniques: touch-only versus gaze-touch. We chose a fixed setup, although unusual for tablet devices, to avoid potential effects on eye tracking accuracy from user movement. The tablet is placed on a table (its integrated stand fixes a comfortable position), and users sit on a chair in front of it (≈ 55cm) during the study (Figure 2a).

Study Design and Method

System: we use a Microsoft Surface Pro 4 tablet with an attached Tobii EyeX eye tracker. The tablet has a display size of 2736x1824 pixel (27x18 cm), runs on a i5-6300U CPU (2.4GHz) at 8GB RAM on Windows 10, and supports up to 10 finger touch at 60Hz. The eye tracker provides gaze data at 30 Hz. A 16-point gaze accuracy test for each user, after each study, showed an average accuracy of M=1.23° (≈43 px) of visual angle (SD=.88°). The software is implemented with Java and Processing, logging temporal and spatial data of the used modalities at 30 Hz.

Task Design: 24 targets were evenly distributed in a 6 x 4 grid on the UI, and designed to resemble a typical homescreen layout with app-like icon targets (Figure 2b). All targets were grey, the current target to be selected is red and becomes green when successfully selected. When the wrong target was selected, it briefly appeared blue. Each target has a size of 160x160 px, i.e. 1.57x1.57 cm, and the target's centres are 360 px offset from each other. After an error, the next target was displayed and the missed target was repeated in the same block to ensure equal number of successful trials. Users performed 7 blocks per technique, overall resulting in 2 techniques × 7 blocks × 24 targets = 336 successful trials per user.

Techniques: With *direct-touch*, users selected a target by directly tapping on it (using their dominant hand). If the user's touch point was within the target's boundaries the task was successful. Users were allowed to use the non-dominant hand to hold the tablet while they interacted with it. With *gaze-touch* [19], users selected a target by looking at it and tapping anywhere on the screen. Here users only used their dominant hand to hold the device, while they performed tap actions with the free thumb of this hand. Due to the inaccuracy of eye trackers we used a target snapping mechanism based on previous work [19, 31]. The target which center was closest to the system's gaze estimate is highlighted with a yellow border, and was selected when users touch the screen. This effectively increases the target's size to 360px (≈3.6cm).

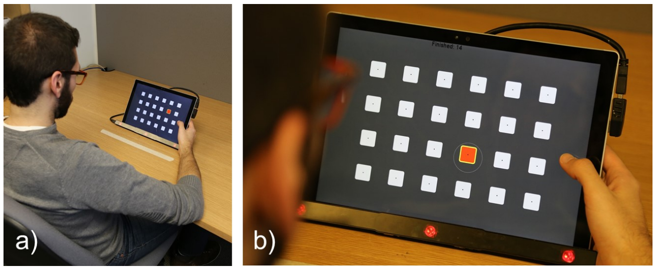

Figure 2. User study: users interacted with the tablet placed on a table (a), and performed a task that resembled homescreens of tablet UIs (b).

Procedure: After an initial briefing, the eye tracker was calibrated to the user with the standard Tobii EyeX tool. Users then conducted the study tasks with each technique (order counterbalanced). Before each technique, users performed a few training trials to become used to the technique (≈ 3-5 trials) and were instructed to be as fast as possible, while still comfortable. Then, in each technique session, users performed 7 blocks. Targets appeared in random order. An additional target was used at the beginning of the block not included in the data analysis. After each block, the user can have a small break (but normally users continued). After each technique, users filled out a questionnaire with six Likert-scale statements from 1 (strongly disagree) to 5 (strongly agree): *'The task with this technique was [easy to use | fast | precise | easy to learn | comfortable for my eyes | physically comfortable (hand, arm, shoulder, or neck)]'*. After the study, users ranked the techniques and gave qualitative feedback on the techniques. Overall, the study lasted 15-20 minutes per user.

Participants: 12 volunteers of the local university from 24 to 35 years (M=27.3, SD=4, 4 female) participated in the study. Three users wore glasses, two were left handed. On a scale between 1 (no experience) to 5 (Expert), they rated themselves as well experienced with multi-touch (M=4.1, SD=1.1), and less experienced with eye gaze interaction (M=2.8, SD=1.2).

Results

We employed a T-Test for the statistical analysis of the following performance results. For time and accuracy measures, only successful trials were chosen (when the correct target was selected), for non-successful trials see error rate.

Time (Fig. 3a): The users were significantly faster with touch (M=725ms, SD=48ms) than with the gaze-touch (M=911ms, SD=227ms) condition (t(12)=2.8, p=.017).

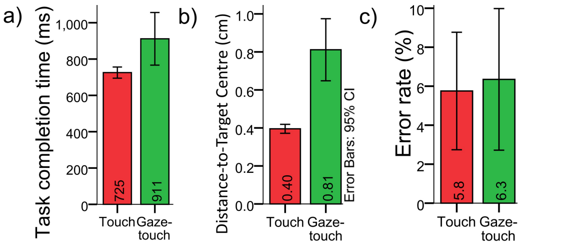

Figure 3. Quantitative study results

To get further insights into temporal performance, we measured how long it took users to select the target after they saw it. Our system detects 'seen' when the target closest to the user's gaze equals the task's selection target. We found that direct-touch users needed 260 ms, and with gaze-touch

388 ms to finish the task after they saw it (significantly different at t(13.7)=2.6, p=.02). The major performance decrease comes from the point after users visually acquired the target.

We then plotted the spatial relationship between gaze and target position across time in Figure 4. The Y-axis shows the distance between gaze and target point, and the X-axis shows the time relative to the point where users touch down (t=0). We found users look closer to the target during the task, until they reach a minimum at approximately the moment of touch down. The results indicate that when using gaze-touch users are more visually focused on the target after touch down, whereas they already started to look away with direct-touch. The added attention could be a factor for the increased completion time for gaze-touch.

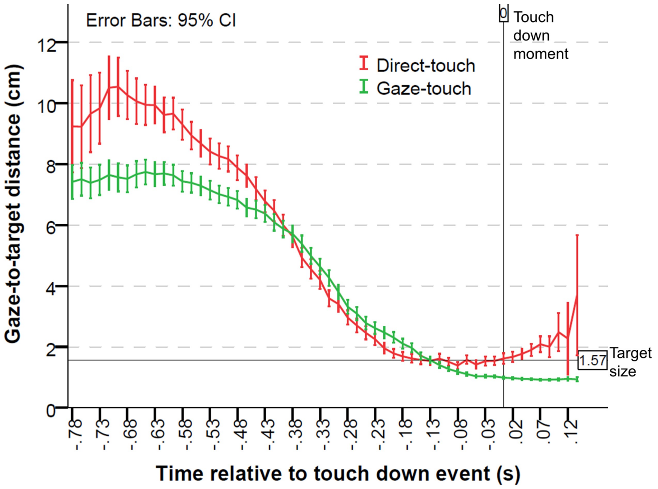

Figure 4. Gaze-to-target distance over time, averaged across all tapping tasks. As expected, users users initially look far away, then focus the target. With gaze-touch, users focus on the target for longer.

Distance-to-Target Centre (Fig. 3b): Although users were not instructed to precisely select the centre of the target as it is the norm in homescreens, users selected the target significantly closer to the centre with touch (M=40px / .39cm, SD=4px / .04cm) than with the gaze-touch (M=82px / .81cm, SD=26px / .26cm) condition (t(11.5)=5.5, p<.001).

Error rate (Fig. 3c): No significant difference was found between the touch (M=5.6%, SD=4.7%) and the gaze-touch (M=6.3%, SD=5.7%) condition (t(22)=.28, p = .79). The error rate describes the probability that users did not select the correct target (i.e. error trials / number of trials).

Feedback (Fig. 5): We ran a Friedman Test on the Likert-scale categories, and found no significant differences in the categories except for physical comfort ($\chi^2(1)$=11, p=.001). Users perceived gaze-touch (M=4.2, SD=1) as more physically comfortable than direct-touch (M=2.1, SD=.9). This result correlates with the rankings, where 8 of 12 users favoured gaze-touch instead of direct-touch. While only 2 users stated they dislike gaze-touch for inaccuracy (*e.g., 'the gaze and touch was not always 100% accurate'*), 8 users preferred gaze-touch for less physical effort and no screen occlusion (e.g. *'touch was tiring and putting my hand over the screen obscured'*).

	Direct-touch		Gaze-touch	
	M	SD	M	SD
Tap time	53.2ms	14.3ms	116.8ms	79.3ms
Tap path	14.8px .14cm	6.8px .06cm	4.6px .04cm	2.4px .02cm

Table 1. Tap gestural differences between the techniques.

Gestural differences: We looked at tap path and duration differences between index finger (direct-touch) and thumb (gaze-touch). Results are listed in Table 2. For tap path, we found that direct-touch users significantly involve more movement than with gaze-touch (t(11.7)=2.7, p=.019). This difference is potentially due to the directional use of the index finger on the slightly diagonally oriented tablet. This potentially induces a minor sliding touch effect. For tap duration, we found that gaze-touch users significantly tapped for a longer time than direct-touch users (t(13.7)=4.9, p<.001). Possibly the thumb provides a larger touchspace and thus could be sensed as a longer touch; second, because of above mentioned synchronisation issue, users might touch for longer to ensure reliable selection.

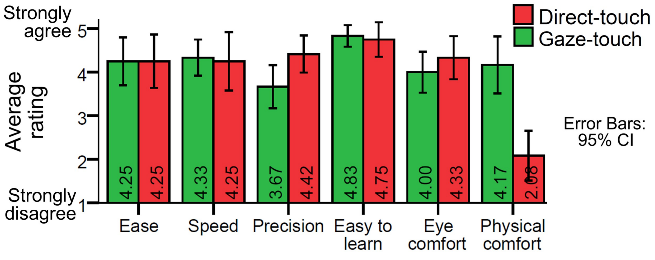

Figure 5. Likert-scale ratings

Discussion

This study shows that users can easily select targets with just a single hand and use of gaze and touch modalities while holding the tablet. Most users preferred gaze-touch and found it less physically fatiguing than direct-touch selection. These results empirically show the potential of gaze on tablets, and

Direct-touch users were about 20% faster than with gaze-touch. Yet, users did not perceive this as a hard problem. This result aligns with previous work that compared direct-touch to other indirect techniques (e.g., 35% faster than miniature interaction areas [34], or 55% faster than gaze-only techniques [23]). Lastly, we tested with a basic gaze selection in this study as a baseline measure, but in future we want to consider more intelligent selection mechanisms [16, 31].

This study focused on a homescreen task that by default comes with large targets, avoiding issues of gaze inaccuracy. This demonstrates the potential of gaze on tablets, but also that further study is needed to get more insights into its general applicability on tablets. There are numerous applications, e.g. browsing or ports of desktop applications, that require more precise input than eye gaze can offer. In the remainder of the paper, we will focus on how to cope with interfaces that require finer control, and further how gaze input integrates into existing direct-touch user interfaces.

PROPERTY	DIRECT-TOUCH	GAZE-TOUCH	CURSOR
Reach and grip	**Low** (grip constraint, needs two hands)	**High** (Indirect input; with any grip position)	
Accuracy	**Moderate** (fat-finger issue)	**Low** (gaze imprecision)	**High** (pixel precise)
Speed	**High** (move hand)	**Moderate** (look and touch)	**Low** (Drag cursor before tap)
Fatigue/movement	**High** (move hand)	**Low** (direct gaze or drag cursor)	
Occlusion	**High** (finger/hand/arm)	**None** (gaze)	**Minimal** (cursor)
Gesture possibilities	**High** (e.g. tap, drag, long touch, double tap)		**Moderate** (drag occupied by cursor control)

Table 2. Summary of the interaction techniques' properties.

CURSORSHIFT

Our study showed that thumb-only input on the whole tablet screen is possible with gaze-touch, but also that gaze selection is potentially inaccurate. We want to support a cursor technique on tablets for precise input. The cursor interaction can be based on a laptop's touchpad. Its core operation is as follows: touch dragging moves the cursor, and a tap selects the object under the position of the cursor. However, the cursor dragging gesture conflicts with the direct-touch scrolling gesture for which we developed the following technique.

CursorShift is a method to enable switching between direct-touch dragging and cursor interaction. It uses the gaze modality and distinction between tap and drag gestures to accomplish this. The idea is to use an indirect tap to toggle to cursor mode. At tap, the system determines the distance between the gaze and tap position (Figure 6a-b). If the distance is lower than a threshold of 5 cm, it is processed as direct-touch. If it is higher, cursor mode starts. The cursor then appears at the user's gaze position (b). Subsequent drag gestures then drag the cursor (c). A tap gesture performs a 'click' at the cursor position, and also toggles back to direct-touch mode (d).

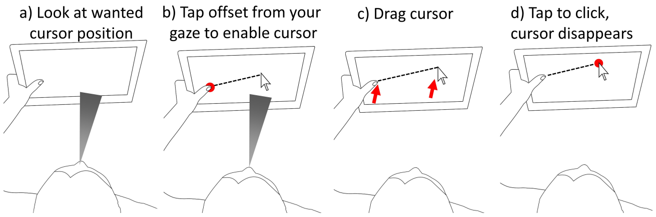
a) Look at wanted cursor position
b) Tap offset from your gaze to enable cursor
c) Drag cursor
d) Tap to click, cursor disappears

Figure 6. CursorShift is a method to temporally enable a cursor, without interfering with default direct-touch actions.

In cursor mode, there are multiple ways to return to direct-touch (Figure 7). First, users can tap and perform a click to finish the action; whether the target is an object or blank space. Then, users can use the second hand to touch in the area of the cursor. This is processed as direct-touch input conflicting with the cursor, and the cursor is cancelled.

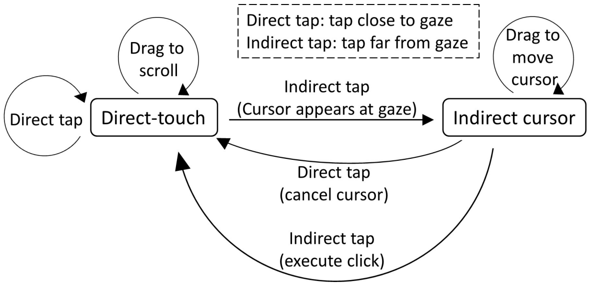

Figure 7. State model of CursorShift.

CursorShift is suitable for applications that require both direct-touch input and cursor input. We illustrate its utility in our next section, particularly in the browser application.

COMBINING TOUCH, GAZE-TOUCH, AND CURSOR

In summary, we can support three techniques for input on tablets: direct-touch, gaze-touch, and cursor (Figure 8). We now explore combinations of the three techniques in three applications: browser, maps, and image gallery. In these applications, CursorShift enables switching between direct-touch and cursor, and gaze-shifting [20] enables switching between direct-touch and gaze-touch.

A short characterisation of the three techniques is given in Table 2. Direct-touch allows natural, fast and precise input, but has limited reach during grip, and is affected by occlusion of content and the fat-finger issue [13]. Our study showed that gaze-touch is slightly slower, less accurate, but since it requires less movement it also had less physical effort. Complementary, a cursor allows for high-precision input but it is slower as it requires cursor dragging. Also both indirect techniques enable reaching the whole screen from grip.

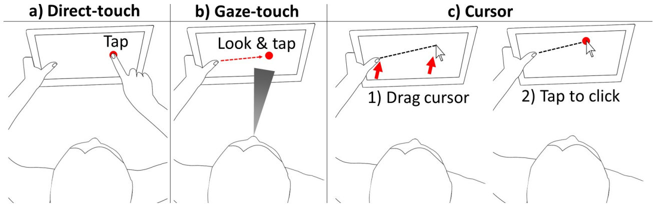

Figure 8. Target selection with the three investigated techniques.

BROWSER

Our browser prototype integrates techniques for scrolling and clicking, menu navigation, and virtual keyboards.

Scrolling and Clicking

Clicking hyperlinks can be tedious because they are relatively small. With direct-touch, users perform a well-targeted touch, or zoom-in before doing so. With CursorShift, users can temporally instance a cursor and click the link. For example, Figure 9 shows how users do the same task with either cursor or direct-touch. A second available hand allows to physically reach and touch the desired target (a). Alternatively, the holding hand's thumb can select the target as well (b). As the target in this example is a relatively large area, users can just look at this area and perform a quick double-tap that combines cursor activation, click, and cursor deactivation.

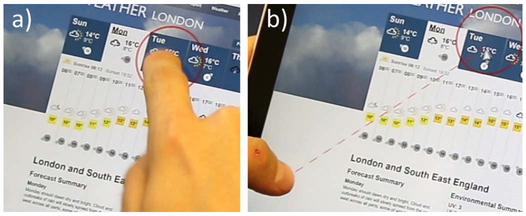

Figure 9. Click an element with direct-touch (a) or with a cursor (b).

Figure 10 shows how webpage scrolling and hyperlink clicking are interleaved into one flow of interaction. The user is browsing a wiki, and scrolling the information on the website (a). Suddenly, a new item becomes interesting, and the user utilises CursorShift to conveniently click this link (b). Immediately, the user continues scrolling the new site (c).

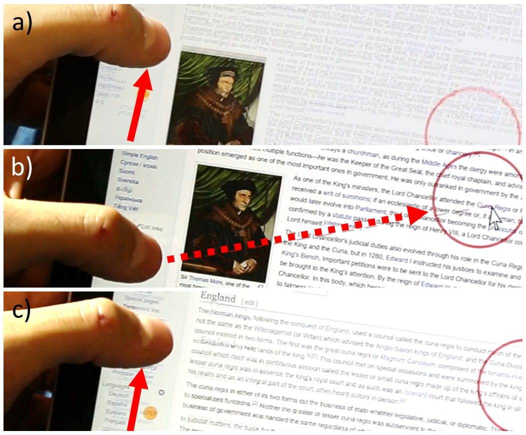

Figure 10. A user scrolls the page (a), toggles the cursor to click a link (b), then continues scrolling the next page (c).

Figure 11 shows how this technique also allows users to perform both tasks bimanually. Users acquire a two-handed grip, and divide the labour between both hands. For example, to scroll the website with the right hand, and issue cursor clicks with the left hand.

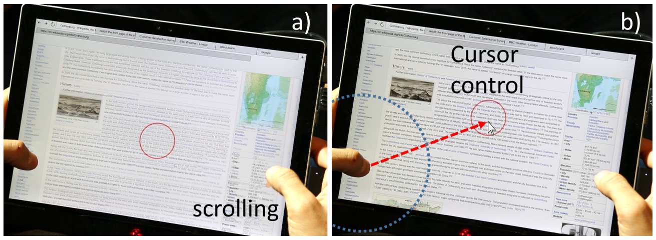

Figure 11. In a two-handed grip, users can divide cursor control and website scrolling. The right hand scrolls (a), the left hand clicks a hyperlink (b). The blue circle indicates the invisible touchpad area.

This is enabled by including an invisible touchpad area. When users activate the cursor through CursorShift, an invisible area around the correponding touch position is created. It is essentially a touchpad area, where users drag the cursor and click. The area is 7 cm in radius around the initial touch position, and updates its center point with the initial touch position of subsequently arriving drag gestures. It disappears when the cursor disappears. The benefit is that another hand that issues touch input outside this radius can interact with direct-touch, or even instance a second cursor for potential multi-cursor input.

Menu Navigation

We also integrated gaze for interaction with the browser menu. We found cursor control for basic menu interactions too slow, and therefore designed a gaze-touch based browser menu. The menu supports default operations, e.g. users can navigate tabs, click 'back' and 'forward' buttons, and open/close tabs. The menu is direct-touch enabled. If users physically touch on one of the browser elements, it is directly triggered. When users look at any point in the whole menu area, and follow up with touch input anywhere else, users acquire a menu pointer, i.e. an indirect handle toward the menu (Figure 12a).

This menu pointer is based on a mechanism that provides rapid selection of large targets, and in case of small targets the option to refine selections. At touch down, the current gaze target becomes indicated as the potential selection by a menu pointer. Users can now optionally perform drag gestures to change the selection to another target (Fig. 12b), and then touch up to confirm (c). Or, users can also just touch up after touch down, if confident that the right target is selected at first sight.

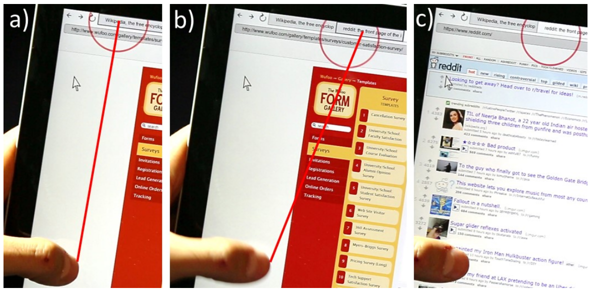

Figure 12. Menu navigation: look at the menu and touch down to acquire a menu pointer (a), drag to refine pointer position (b), and touch up to select the tab (c).

This menu technique extends prior gaze based menus [19, 20]. At touch down the menu pointer's x position is equal to the gaze, and the y position lies in the middle of the tabs' height. The upper row with the tabs and navigation elements is therefore the first that users can access, as it is expected to be more frequently used. Overall, the menu integrates the following drag semantic:

- Drag left-right: changes the menu pointer to another tab, or to one of the other buttons in the vertical row.
- Drag up: closes the tab that is selected by the menu pointer.
- Drag down: allows access to the address bar.

The 'back' and 'add tab' buttons are placed at the left and right end of the first row; this allows users to rapidly reach these positions for successive actions.

Keyboard

The menu allows rapid access to the address bar, at which the virtual keyboard is triggered as well. This keyboard is then direct-touch enabled.

In addition, users can utilise gaze-touch to interact with the keyboard from a remote touch position. This is particularly useful when users have a different grip, e.g. at the top of the tablet, and just want to type a few words. Gaze-touch to the keyboard activates when the user's gaze is within the keyboard, and the user's touch is not on any of the keys. This also allows users to additionally use the empty borders of the keyboard as gaze-touch area to any of the keys.

The keys are relatively large by design, as in typical virtual keyboard designs, but in other contexts (e.g. tablet in portrait

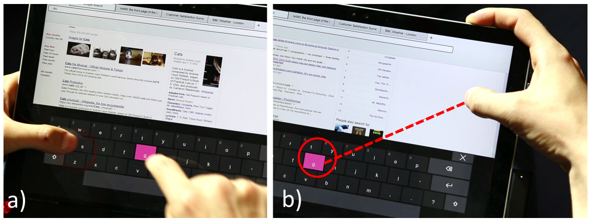

Figure 13. The virtual keyboard allows to use one hand to hold the device, and a second for direct-touch key input (a). Or, users hold the device and with the same hand issue gaze-touch input to select keys (b).

orientation) can be smaller. Thus, the interaction technique involves a similar precision enhancing technique as with the browser menu. Users look at the respective key, and touch down to acquire a handle. Drag gestures move the handle's 2D position, to correct potential erroneous initial gaze selections. Advanced key input, such as long-touch for numbers, is similarly supported: users keep the indirect handle over the key to simulate a long-touch on it.

MAPS

Map navigation applications come with the tablet, allowing users to explore locations, or to see how far one place is from another. To support users in these tasks with tablet-specific grips in mind, we built the following techniques.

Zooming

In addition to standard direct pinch-to-zoom (Figure 14a), we designed two zooming techniques that are specifically useful during grip. They activate when all used touch points were detected as indirect using the gaze-shifting principle (5 cm gaze-to-touch distance); otherwise direct-touch zooming is enabled. Both indirect techniques enable zoom-in at the gaze position, however we fixed the zoom-out position to the screen's center as we found it unnatural to zoom-out at gaze.

2F-grip-zoom (Fig. 14b): This technique is used during a two handed grip, where one hand is at each side of the tablet making only two thumb fingers available. By drawing the thumbs apart, users zoom in. In principle users can do the same with direct-touch already — however the distinct benefit is that users can direct their gaze to zoom where they want (instead of only in the center of the thumbs as in direct-touch). We also modified the zooming gain because the two thumbs disallow large movement. The gain depends on the distance between both fingers: with increasing distance, the gain increases and thus allows fast zoom in this grip.

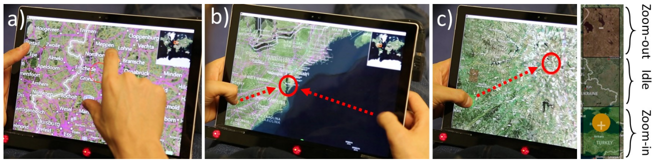

Figure 14. Our maps application supports three zoom variants: direct pinch-to-zoom (a), two-thumb gaze based pinch-to-zoom (b), and one-thumb relative gaze-based zoom (c).

1F-grip-zoom (Fig. 14c): Single-finger zoom is particularly useful when users hold the tablet and intend to navigate maps with the same hand. As single-finger drag is used for panning, users perform a double-tap-drag gesture to activate zoom mode. Then a drag up zooms out, and a drag down zooms in. This technique is similar to the current single-finger zoom technique of phones and tablets, but extends it with a *relative, continous zoom*.

The visual design is shown at the right side of Figure 14c. After touch down, a three-element based widget appears. Initially, the finger locates at the center of this widget. When users move their finger upwards, the map continuously zooms in; downwards continuously zooms out. Returning to the initial touch down point stops zoom ('idle' mode). It is relative because the further users move their finger away from the initial point, the higher is the zooming gain. To accomplish this, we mapped the distance between both fingers to a zoom gain transfer function; which keeps the zoom gain at about a similar level as when performing a unimanual pinch gesture. In sum, after double-tap-hold, users can zoom in, out, or remain idle without needing to touch up.

Minimap

Some map instances provide users with an additional overview window, e.g. using a minimap of the world to rapidly change locations, while the main window provides standard zooming behaviour. Direct-touch interaction allows control of both by alternating between both available windows.

With the use of gaze-touch, users can easily reach both windows without relocating the physical touch position. When a user looks at the minimap, users can indirectly control the viewport of the minimap. The user touches down with a finger, which renders an indirect handle toward the viewport (Figure 15a). Then, indirect touch movement directly translates to movement of the viewport (b). After touch down, users can look elsewhere, as gaze is only used to initially acquire the indirect handle.

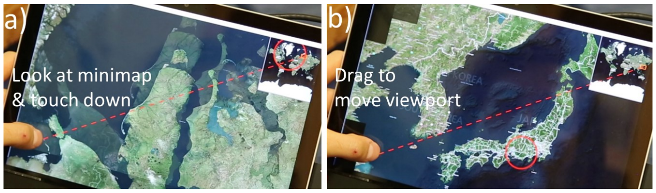

Figure 15. By gaze-touch on a minimap, users pan long distances with a single drag gesture from remote.

Bimanual Combinations with Markers

Users can also add markers at locations of the map, as reminders of places, or for measuring the distance between two markers. This can be done with both direct-touch and indirect cursor controls utilising CursorShift. For direct-touch, users reach to a position. For cursor control, users look at the desired position, and tap indirectly to activate a cursor. Then users can perform three actions with the techniques:

- Create marker: long-touch on marker-free space.
- Drag marker: double-tap-drag on an existing marker.
- Delete marker: long-touch on existing marker.

These actions combine with the other map techniques when using bimanual input (Figure 16):

- Bimanual direct-touch + gaze-touch (a): users can drag a marker with direct-touch (1), while they use gaze-touch on the minimap to quickly travel a distance (2), and then drop the marker (3). This allows to drag markers to long-distance locations.
- Bimanual direct-touch + cursor (b): users can acquire and hold a marker with the cursor (1). While they do this, they can pan the map with the free hand to find the location the marker should be dropped at (2).
- Bimanual gaze-touch + indirect cursor (c): lastly, users can create a marker with the cursor as in the previous example (1). At the same time, gaze-touch is used for single-finger zooming (2).

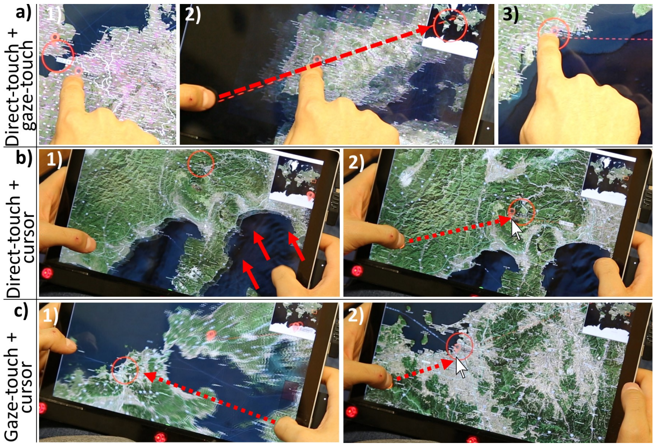

Figure 16. Three bimanual combinations to perform three compound tasks: direct marker dragging with indirect minimap control (a), direct panning with cursor based marker control (b), and indirect zooming with cursor based marker input (c).

GALLERY

An image gallery allows users to browse the images stored in their tablet. The design of our prototype is based on typical gallery applications, enabling users to scroll through image folders and view them in different scales. We describe scrolling, image selection, and image dragging.

Area Scrolling

The gallery is based on two user interfaces. The first UI consists of two areas (Fig. 17a-b). The first area is a grid of images, and the second area is a smaller panel at the left that lists folders ordered by month. After selecting an image in the first UI, users get to the second UI (Fig. 17c-d). This shows the corresponding image enlarged, and in addition provides a horizontal quick list at the bottom with small images.

The drag gestures scroll the area of the UI that users are looking at. At touch events, the system determines which area the user looks at. Drag gestures are then forwarded to the corresponding area. This provides users with the capability to scroll any area in the given UI with a single finger, without the need to leave their current grip. In our first UI, users can look at the list of month folders and scroll it (Figure 17a), or look at the image grid to scroll it (b). In the second UI, users can look at the quick list to scroll it (c), or at the enlarged image to flick to the previous/next image (d).

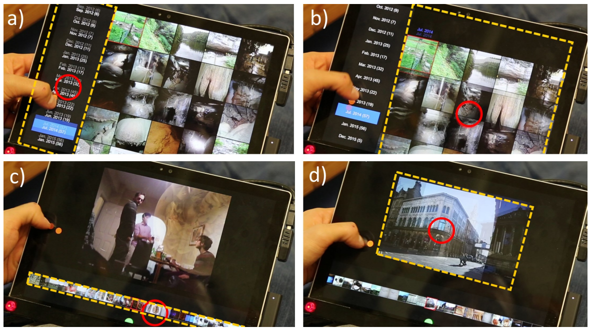

Figure 17. In both UIs, scroll gestures affect the view the user is looking at. The scrolled views are highlighted with an orange border.

Image and Folder Selection

In addition to the scrolling, the first UI enables users to rapidly select images or folders by a single tap. The images are displayed relatively large in the gallery in order to make them visible to the user. This makes it sufficient to use gaze selection instead of more precise cursor input.

Figure 18 shows selection examples with both techniques. Users can directly select the image with the free hand when looking at it (a). The image view shows up, and by touching the 'back' button, users can return to the image grid (b). In case the second hand is not available, users can perform the same actions with the tablet-holding hand. To accomplish this, users look at the image and tap with the free thumb (c). This opens the image. To return to the grid view, users look at the back button, and issue a tap from remote (d). Both direct-touch and gaze-touch variant are useful: e.g., direct input clearly shows others which image is selected, while gaze-touch allows users to frequently open/close images with minimal hand movement.

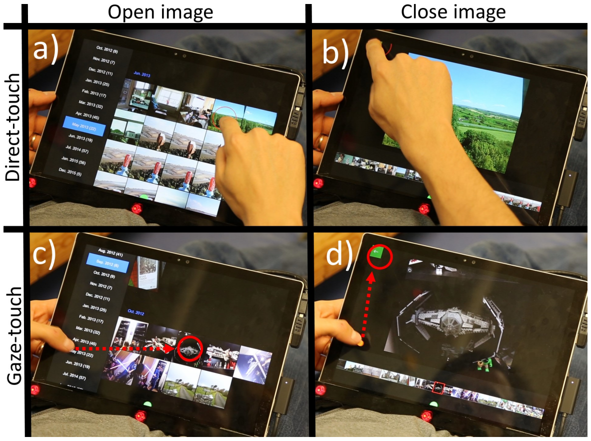

Figure 18. A dynamic switching mechanism allows direct- and gaze-touch image navigation. With direct-touch, users physically touch the image and the 'back' button (a-b). With gaze-touch, users can comfortably use the same finger to do the same actions without using a second hand (c-d).

Bimanual Combinations with Image Dragging

In the following examples, we show how users bimanually use image dragging with the scrolling techniques. A long-touch on an image acquires it, and then users can drag to a new location.

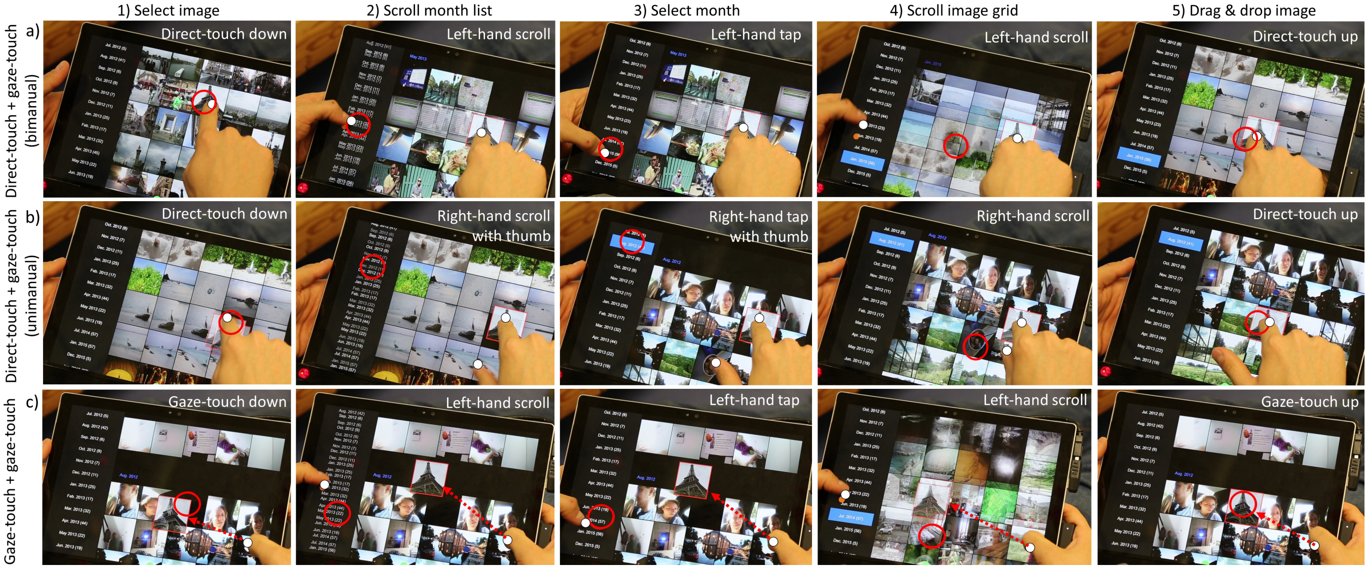

Figure 19. The same task can be performed with different bimanual technique combinations. Touches highlighted in white, gaze as red circles.

Image dragging is possible with both direct-touch and gaze-touch. Users can directly touch on an image, while looking close to it; or indirectly grab a target by looking at the image, and touching down from a grip position. Notably, with gaze-touch, after users selected the image, gaze has no effect — and touch takes over to manipulate the target.

The image dragging technique fluidly combines with the scrolling techniques. For example, users can use direct-touch to grab an image, while they use their grip-hand to scroll the two views (19a). When the desired location is found, users return to direct-touch and drag the image to the location.

In addition, users can perform these actions with unimanual input of the free hand (Figure 19b). As in the previous example, the index finger directly grabs an image. However, then the thumb of the same hand is used to scroll the month view, and then the grid view. When a desired place is found, users drop the image by releasing the index finger's touch input.

Lastly, users can perform all these actions with gaze-touch in a two-handed grip of the device (Figure 19c). At look at the desired image, users touch down with the right hand's thumb. While keeping dragged on the image, the other hand's thumb scrolls the month view, and then the grid view. Finally, users drag the image to a position, and release the image. The indirect dragging is implemented with a higher control-display gain, compensating for the small movement that the thumb can perform.

DISCUSSION

In this paper we investigated how gaze integrates into multi-touch based tablet UIs, and presented techniques that specifically enable users to interact with the whole screen using the same hand that grips the tablet. We described three applications that demonstrate the utility of these techniques on tablet computers, and describe how gaze, touch, and cursor input can be designed for combined use on tablet UIs.

Our work shows that tablet interactions can be simplified through the introduction of gaze onto tablets. Our study of a gaze-enabled homescreen demonstrates this best: the simple task of selecting an application that required two hands, is now reduced to simple unimanual input. We also show that diverse interactions from precise (select hyperlink) to coarse (change tab) ranges can be supported with our browser application; in essence, a single touch position can interact with the whole browser. In addition to the examples in the maps and image gallery application, we find that there are numerous basic interactions on tablets that, with gaze and touch input, can become simpler to use with one hand, and require less physical effort.

Our applications highlight novel bimanual interactions that users can employ, whether in a two-handed grip or in a one-handed grip with a second freely available hand. Combinations include use of direct-touch and gaze-touch, direct-touch and cursor, or gaze-touch and cursor inputs. This is enabled by applying previously developed input shifting methods to tablet contexts [19, 20], as well as our CursorShift method that provides users with easy switching between cursor and direct-touch input. For each hand, users can switch between input modes, and their combination yields the ability to perform advanced compound tasks, such as image dragging while scrolling the gallery, zooming while holding a marker, or moving a cursor while scrolling a website.

Gaze can change interaction with existing manual interfaces, as our applications demonstrated how interaction with default direct-touch UIs could transform into unimanual, single-grip, thumb-only interaction. It is unclear whether transforming a direct UI to an indirect UI on the tablet is a step forward, or a step back. While in past, users mainly interacted with indirect input on desktop systems, now direct-touch mobile devices are in focus; although desktop computers are still used in productive environments. Rather than increasing the tablet's capabilities with indirect input which can be considered as counterintuitive, we strive for a balance between supporting gaze-enhanced indirect input and direct-touch input through novel techniques and modality combinations for tablets.

Our work comes with limitations regarding technical feasibility and user evaluation. The use of multiple interaction styles and techniques enables a rich extended input vocabulary on tablets. In our initial study, we found that users can successfully employ basic gaze and touch based interaction on a homescreen task in a fixed setting. However, our applications go beyond single-technique approaches and with it can increase complexity, and future evaluation is needed to better understand relevant factors such as performance, affordance, and learnability. Additionally, our techniques are currently limited to stationary tablet usage. Here issues include occlusion of the eye tracker's camera when users reach in with their hand, no support of tablet orientation change, and gaze inaccuracy. Eye trackers are becoming smaller and versatile, which made our dynamic interactions possible. Our study indicated that users can utilise gaze for large targets in a stationary setup; for more precision users can employ our cursor techniques. Further study of the techniques are needed in order to tease out the right parameters, and to understand their feasibility in more natural tablet scenarios.

CONCLUSION

We presented an exploration of gaze and touch interactions on tablet devices to increase interaction possibilities with the hand that holds the device. We evaluated this interaction techniques against the baseline of direct-touch, and found it is slightly slower but allows unimanual use of the device with less physical effort. We developed CursorShift, a technique that allows users to seamlessly switch between direct-touch input, the current status-quo interaction technique on tablets, and cursor input, the current status-quo interaction technique on desktop systems. We presented three applications that describe gaze and touch based interaction techniques and combined use of touch, gaze, and cursor inputs. Taken together, we show that through the design of compound techniques, we enable users to leverage single-grip, single-hand, and single-finger input for many of the current tablet interactions that require two hands.

REFERENCES

1. Baudisch, P., and Chu, G. Back-of-device interaction allows creating very small touch devices. In *CHI '09*, ACM (2009), 1923–1932.
2. Bergstrom-Lehtovirta, J., and Oulasvirta, A. Modeling the functional area of the thumb on mobile touchscreen surfaces. In *CHI '14*, ACM (2014), 1991–2000.
3. Bulling, A., and Gellersen, H. Toward mobile eye-based human-computer interaction. *IEEE Pervasive Computing 9*, 4 (Oct. 2010), 8–12.
4. Cheng, L.-P., Liang, H.-S., Wu, C.-Y., and Chen, M. Y. igrasp: Grasp-based adaptive keyboard for mobile devices. In *CHI '13*, ACM (2013), 3037–3046.
5. Drewes, H., De Luca, A., and Schmidt, A. Eye-gaze interaction for mobile phones. In *Mobility '07*, ACM (2007), 364–371.
6. Drewes, H., and Schmidt, A. The magic touch: Combining magic-pointing with a touch-sensitive mouse. In *INTERACT '09*, Springer-Verlag (Berlin, Heidelberg, 2009), 415–428.
7. Dybdal, M. L., Agustin, J. S., and Hansen, J. P. Gaze input for mobile devices by dwell and gestures. In *ETRA '12*, ACM (2012), 225–228.
8. Esteves, A., Velloso, E., Bulling, A., and Gellersen, H. Orbits: Gaze interaction for smart watches using smooth pursuit eye movements. In *UIST '15*, ACM (2015), 457–466.
9. Goel, M., Wobbrock, J., and Patel, S. Gripsense: Using built-in sensors to detect hand posture and pressure on commodity mobile phones. In *UIST '12*, ACM (New York, NY, USA, 2012), 545–554.
10. Hinckley, K., Heo, S., Pahud, M., Holz, C., Benko, H., Sellen, A., Banks, R., OHara, K., Smyth, G., and Buxton, B. Pre-touch sensing for mobile interaction. In *CHI '16*, ACM (2016), to appear.
11. Hinckley, K., Pahud, M., Benko, H., Irani, P., Guimbretière, F., Gavriliu, M., Chen, X. A., Matulic, F., Buxton, W., and Wilson, A. Sensing techniques for tablet+stylus interaction. In *UIST '14*, ACM (2014), 605–614.
12. Hohlfeld, O., Pomp, A., Link, J. A. B., and Guse, D. On the applicability of computer vision based gaze tracking in mobile scenarios. In *MobileHCI '15*, ACM (2015), 427–434.
13. Holz, C., and Baudisch, P. The generalized perceived input point model and how to double touch accuracy by extracting fingerprints. In *CHI'10*, ACM (2010), 581–590.
14. Karlson, A. K., and Bederson, B. B. Thumbspace: Generalized one-handed input for touchscreen-based mobile devices. In *INTERACT'07*, Springer-Verlag (Berlin, Heidelberg, 2007), 324–338.
15. Kim, S., Yu, J., and Lee, G. Interaction techniques for unreachable objects on the touchscreen. In *OzCHI '12*, ACM (2012), 295–298.
16. Kumar, M., Klingner, J., Puranik, R., Winograd, T., and Paepcke, A. Improving the accuracy of gaze input for interaction. In *ETRA '08*, ACM (2008), 65–68.
17. Negulescu, M., and McGrenere, J. Grip change as an information side channel for mobile touch interaction. In *CHI '15*, ACM (2015), 1519–1522.
18. Odell, D., and Chandrasekaran, V. Enabling comfortable thumb interaction in tablet computers: A windows 8 case study. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 56, SAGE Publications (2012), 1907–1911.
19. Pfeuffer, K., Alexander, J., Chong, M. K., and Gellersen, H. Gaze-touch: Combining gaze with multi-touch for interaction on the same surface. In *UIST '14*, ACM (2014), 509–518.

20. Pfeuffer, K., Alexander, J., Chong, M. K., Zhang, Y., and Gellersen, H. Gaze-shifting: Direct-indirect input with pen and touch modulated by gaze. In *UIST '15*, ACM (2015), 373–383.

21. Pfeuffer, K., Alexander, J., and Gellersen, H. Gaze+touch vs. touch: Whats the trade-off when using gaze to extend touch to remote displays? In *INTERACT'15*, vol. 9297, Springer (2015), 349–367.

22. Pfeuffer, K., Alexander, J., and Gellersen, H. Partially-indirect bimanul input with gaze, pen, and touch for pan, zoom, and ink interaction. In *CHI '16*, ACM (2016), 373–383.

23. Roetting, M., and Zukunft, O. Don't touch that tablet: An evaluation of gaze–based interfaces for tablet computers. IEEE (2014).

24. Roudaut, A., Huot, S., and Lecolinet, E. Taptap and magstick: Improving one-handed target acquisition on small touch-screens. In *AVI '08*, ACM (2008), 146–153.

25. Serim, B., and Jacucci, G. Pointing while looking elsewhere: Designing for varying degrees of visual guidance during manual input. In *CHI '16*, ACM (2016), to appear.

26. Stellmach, S., and Dachselt, R. Look & touch: Gaze-supported target acquisition. In *CHI '12*, ACM (2012), 2981–2990.

27. Stellmach, S., and Dachselt, R. Still looking: Investigating seamless gaze-supported selection, positioning, and manipulation of distant targets. In *CHI'13*, ACM (2013), 285–294.

28. The Eye Tribe. The eye tribe tracker. In *https://theeyetribe.com/* (04/2016).

29. Trudeau, M. B., Catalano, P. J., Jindrich, D. L., and Dennerlein, J. T. Tablet keyboard configuration affects performance, discomfort and task difficulty for thumb typing in a two-handed grip. *PloS one 8*, 6 (2013), e67525.

30. Turner, J., Alexander, J., Bulling, A., and Gellersen, H. Gaze+rst: Integrating gaze and multitouch for remote rotate-scale-translate tasks. In *CHI '15*, ACM (2015), 4179–4188.

31. Voelker, S., Matviienko, A., Schöning, J., and Borchers, J. Combining direct and indirect touch input for interactive workspaces using gaze input. In *SUI '15*, ACM (2015), 79–88.

32. Wagner, J., Huot, S., and Mackay, W. Bitouch and bipad: Designing bimanual interaction for hand-held tablets. In *CHI '12*, ACM (2012), 2317–2326.

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

34. Wolf, K., and Henze, N. Comparing pointing techniques for grasping hands on tablets. In *MobileHCI '14*, ACM (2014), 53–62.

35. Wolf, K., Schneider, M., Mercouris, J., and Hrabia, C.-E. Biomechanics of front and back-of-tablet pointing with grasping hands. *Int. J. Mob. Hum. Comput. Interact. 7*, 2 (Apr. 2015), 43–64.

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

37. Zhai, S., Morimoto, C., and Ihde, S. Manual and gaze input cascaded (magic) pointing. In *CHI'99*, ACM (1999), 246–253.