Understanding Users' Touch Behavior on Large Mobile Touch-Screens and Assisted Targeting by Tilting Gesture

Youli Chang **Sehi L'Yi** **Kyle Koh** **Jinwook Seo**
Human-Computer Interaction Laboratory, Department of Computer Science and Engineering
Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 151-744, Korea
{ylchang, shlyi, kyle}@hcil.snu.ac.kr, jseo@snu.ac.kr

ABSTRACT

As large-screen smartphones are trending, they bring a new set of challenges such as acquiring unreachable screen targets using one hand. To understand users' touch behavior on large mobile touchscreens, we conducted an empirical experiment to discover their usage patterns of tilting devices toward their thumbs to touch screen regions. Exploiting this natural tilting behavior, we designed three novel mobile interaction techniques: *TiltSlide*, *TiltReduction*, and *TiltCursor*. We conducted a controlled experiment to compare our methods with other existing methods, and then evaluated them in real mobile phone scenarios such as sending an e-mail and web surfing. We constructed a design space for one-hand targeting interactions and proposed design considerations for one-hand targeting in real mobile phone circumstances.

Author Keywords

Large mobile screen; One-hand interaction; Tilting gesture

ACM Classification Keywords

H.5.2. User Interfaces: Input devices and strategies, Interaction styles, Screen design.

INTRODUCTION

With the ever increasing size of touchscreen mobile devices, a new category of devices called "phablets" (for phone-tablet) is emerging. Apple has also joined the trend by releasing a large iPhone 6/6+ [1]. Such large devices provide enough screen display, delivering a stream of applications that take advantage of the new space. However, it is inevitable that some parts of the screen become unreachable when used with one hand. People often encounter the situation where only one hand is available to control the device while the other hand is occupied with other operations. This brings us the challenge of acquiring unreachable targets in large-screen mobile devices.

There have been several attempts to address the issue. Some investigated a feasible region for users' thumbs [3,13], and others proposed innovative one-hand targeting methods to assist selecting far-targets [1,2,10,11,12,14,15,16,18,19,22, 23,24,25]. However, there has not been any research on understanding users' touch behaviors in terms of tilting devices and exploiting them to design novel mobile interactions to reach uncomfortable regions.

CHI 2015, April 18–23, 2015, Seoul, Republic of Korea.

ACM 978-1-4503-3145-6/15/04…$15.00.
http://dx.doi.org/10.1145/2702123.2702425

In this paper, we presented an empirical analysis of users' tilting behavior when touching large mobile screens with one-hand using their thumbs. We introduced three novel mobile interaction techniques using the tilting gesture to touch targets in uncomfortable regions on large screen smartphones. To validate their performance and preference, we conducted a controlled experiment to compare them with other techniques introduced before.

The new interaction techniques had to blend with existing and more conventional interactions in current mobile applications. For such reason, we conducted a user study on ordinary mobile phone scenarios to verify their feasibilities. Also, we proposed a design space for one-hand targeting interactions and discussed design considerations based on the results of the user study.

RELATED WORK

As an endeavor to assist the user in real mobile phone scenarios, one-hand targeting interactions have to be taken into consideration by their trigger mechanisms as well as their targeting mechanisms (we verified this through our Experiment 3). Thus, we constructed a design space by *Trigger Mechanisms* versus *Targeting Mechanisms* in Table 1. The *Targeting Mechanisms* could be further categorized into *Assisted Touch* and *Cursor*. *Assisted Touch* mechanism can directly touch targets with the thumb by improving the reachability of the target either by transforming the screen space (*Screen Transform*) or by introducing a proxy space for the screen (*Proxy Region*), while *Cursor* mechanism involves a pseudo-fingertip like a remote cursor.

AppLens [12] is one of the methods using the *Screen Transform* mechanism. It is a tabular fisheye interface that is invented to show multiple applications in different levels of details simultaneously. *AppLens* brings an application screen closer to the thumb by allowing users navigate applications by a rectangle cursor (called object cursor) that can be moved by predefined set of gestures or a physical joystick embedded on a PDA. The gestures are designed inside of the thumb's accessible area to facilitate one-handed device use. *ThumbSpace* [14] is one of the methods

Trigger Mechanisms \ Targeting Mechanisms		Assisted Touch (Touch with user's thumb)		Cursor (Called pseudo-fingertip)	
		Screen Transform	Proxy Region	Same Direction	Opposite Direction
Always On		*AppLens [12]*	*ThumbSpace [14]*		
Tapping			*TapTap [18]*	*Offset Cursor [16,19]*	
Dragging	Edge (Bezel)	*ES [15]*		*EC [15] (Called ETEC)*	
			BezelSpace, CornerSpace [25]		
	Anywhere	*LS [15]*		*Adaptive Horizontal Offset [10]*	*MagStick [18], LC [15]*
Mode Change	Tilting Gesture	*TiltReduction, TiltSlide*		*TiltCursor*	
	Physical button	*iPhone6/6+ [1]*	*"enhanced" ThumbSpace[11]*	*A Piece of Butter [22]*	
	Touch Gesture	*One-handed operation mode[2]*			
Back-Side Touch				*Behind-the-Display Cursor [23,24]*	
Long Touch / Double Touch					

Table 1. A design space by *Trigger Mechanisms* versus *Targeting Mechanisms* for one-hand targeting interactions.

using the *Proxy Region* mechanism. It uses an empty semitransparent proxy view where all available targets on the screen are mapped. The proxy view is placed in a comfortable region for the user's thumb. However, both methods are activated at all times, even for targets located in comfortable regions. Thus, Karlson et al. improved their method in "enhanced" *ThumbSpace* [11] triggered only by pressing a button (the center of the DPad) and combined *Shift* [21] to solve the occlusion of targets from a thumb.

Newly released large smartphones are equipped with a *Screen Transform* mechanism as a reachability feature to support one-hand interaction. *iPhone 6/6+* [1] enables users to pan down the screen by half by double-tapping the home button. After an item on the screen is touched, the screen returns back to the original position. However, it cannot address the reachability problem of bottom uncomfortable regions and is not efficient to successively touch multiple far-targets. *One-handed operation mode* [2] of Samsung Galaxy Note3 shrinks the screen to make it comfortable for one-hand targeting by a pre-defined touch gesture, i.e., swiping from the bezel to the center and going back to the bezel at once. It renders the actual screen to help users identify targets, and we took a similar approach in our *TiltReduction* method.

On the other hand, the methods using the *Cursor* mechanism can be further categorized by the direction of cursor movement. *Same Direction* methods shift a cursor in the same direction as the thumb, while *Opposite Direction* methods shift it in the opposite direction.

As the *Same Direction* methods, *Offset Cursor* [16,19] (called take-off paradigm) appears at a fixed distance above the tapping point until the thumb is lifted. For acquiring a target, users tap the screen and drag their finger until the cursor is located on top of the target. However, users cannot guess a distance to a target before tapping. Thus, *Adaptive Horizontal Offset* [10] proposed an improvement by adjusting horizontal distance dynamically. However, either cannot acquire the target that is placed under users' thumbs.

TapTap and *MagStick* [18] are designed to improve one-hand targeting on small touch screens; nonetheless, they can help with far-targets' accessibility. *TapTap* shows a zoomed-in pop-up view at the center of a screen when users touch on a region where targets are densely positioned. It brings far-targets to the center of a screen by touching the area near the target. *MagStick* shows a cursor at the touched position and then moves it in the direction opposite a thumb movement. Such *Opposite Direction* methods can avoid occlusion from a thumb, but reverse direction movements may involve awkwardness. The cursor of *MagStick* also attracts targets, so it sticks to the nearest target that is closer than the pre-defined threshold. The magnetized cursor could speed up target acquisition, but it is not effective for densely populated targets.

Kim et al. [15] presented four targeting methods by combining two trigger techniques (a bezel swipe or a wide area touch by the pad of the thumb) and two support techniques (sliding screen or extendible cursor). *Edge trigger with Sliding screen* (*ES*) and *Large-touch trigger with Sliding screen* (*LS*) slides screen towards a thumb as much as drag movement of the thumb to get far-targets closer, while *Edge Trigger with Extendible Cursor* (*ETEC*) and *Large-touch trigger with extendible Cursor* (*LC*) use an extendible cursor multiplying thumb movements to reach far-targets. Among these methods, *ETEC* showed significant improvements in terms of speed and accuracy. We adopt a similar approach in our *TiltCursor* method by using an extendible cursor in the same direction as a thumb.

BezelSpace and *CornerSpace* [25] utilize the benefits of the *Cursor* mechanism and the *Proxy Region* mechanism. When a user perform bezel swipe, *BezelSpace* generates a cursor and a proxy region under the thumb. The user can move the cursor inside the proxy region. Likewise, by bezel swipe, *CornerSpace* generates four corner buttons (denoting each corner of a screen) and a cancel button at the lift-up position of the thumb. By selecting a corner button, a cursor appears from the corner of the screen and a proxy region instead of the buttons.

A Piece of Butter [22] also uses the C*ursor* mechanism to select an item from a fixed menu. A cursor moves along the direction of the tilt by that much of the tilting angle. Usage of a tilt sensor is similar to our methods, but adjusting the cursor by tilting is hard for accurate targeting. Apart from using built-in sensors, *Behind-the-Display Cursor* [23,24] added sensors such as touch pad and optical sensor mounted on the back side of a mobile device to adjust a cursor to select hard-to-reach items by the forefinger. However, such methods requires additional hardware, making it hard to be adopted in current smartphones as it is.

UNDERSTANDING USERS' BEHAVIOR (EXPERIMENT 1)

First, we were interested in users' behavior of tilting the large screen smartphones towards their thumbs using the palm and the rest of fingers when they had to directly touch far-targets located outside the thumb's naturally reachable region. We presumed that the direction of tilting depended only on the position of the target and not on the distance from the thumb. In relation to this issue, GripSense [9] used rotational movement as a factor to detect which side of the user's hand holds the device. However, there has not been any empirical evidence to our knowledge to determine the direction and amount of tilting to touch far-targets on large mobile screens.

Second, we also noted the degree of uncomfortableness (called uncomfortable degree) of touch gestures for target acquisition. Karlson et al. [13] discussed the issue in various handheld devices. However, the results cannot be directly applied to the devices currently available in the market because of differences, including but not limited to screen size/ratio, thickness, and weight. Moreover, because even the largest screen tested in their study was smaller than the smallest device used for our experiment, we found it necessary to re-evaluate state-of-the-art large devices.

We designed an empirical experiment reflecting our observations. We recorded data from a built-in gyroscope sensor at the precise moment of target acquisition. We also collected task completion time between start action and target acquisition. We examined uncomfortable degree for all areas of a touchscreen through the questionnaire.

Participants

We recruited 11 participants (5 females) through an online reservation system announced on a campus bulletin board. Their age ranged from 20 to 29 years (M = 24.4, SD = 2.9). We screened participants so that all were right-handed and had experience with touchscreen mobile devices for more than three months. We did not require any previous experience with large mobile devices because we wanted to examine the general tendency of users' touch behaviors.

Apparatus

We conducted a controlled experiment using two large touchscreen mobile devices: (1) Samsung Galaxy S4 (136.6 × 69.8 × 7.9 mm, 5" screen) and (2) Samsung Galaxy Note3 (151.2 × 79.2 × 8.3 mm, 5.7" screen). The former is considered large screen smartphones, whereas the latter reaches the new category of "phablets".

Task

The participants were asked to hold the device in their right hands (dominant hands). Since their grip posture could change during the task, we enforced them to go back to

(A)

38.4	33.5	27.8	22.1	19.8	21.1	13.3	11.9
30.0	23.5	25.8	17.6	19.0	9.5	13.8	9.8
26.2	24.8	17.4	17.2	13.9	12.3	7.8	5.4
24.9	24.4	14.1	9.4	7.3	6.5	3.4	1.2
26.8	17.6	11.9	7.6	10.2	2.1	-0.8	-3.4
19.6	11.2	7.2	8.2	0.1	-0.7	1.1	0.3
17.0	9.8	6.0	7.1	0.0	-1.7	-2.4	-2.4
16.9	12.2	6.5	-0.5	1.3	-1.6	-6.0	-7.0
10.6	9.8	4.8	2.8	-2.9	-5.4	-7.6	-10.8
10.3	4.5	2.0	-1.4	-1.6	-5.0	-6.2	-13.9
11.5	8.4	4.6	2.4	-5.5	-5.3	-8.9	-12.0
19.0	7.9	2.7	-0.6	-5.2	-7.3	-12.4	-13.1
20.1	5.9	7.7	-2.1	-5.6	-9.0	-9.5	-18.3
21.3	7.4	4.5	-4.0	-9.8	-13.1	-15.7	-18.9

(B)

12.7	12.5	13.6	14.8	15.7	12.6	15.0	10.1
9.7	14.6	14.8	11.9	13.1	12.3	12.6	13.2
14.0	14.1	13.0	10.1	10.4	10.8	14.0	9.7
10.5	9.3	9.6	14.7	8.2	11.2	8.2	7.2
7.7	9.9	9.8	8.5	6.4	3.6	4.6	0.4
6.8	11.4	5.8	0.7	3.9	-2.1	-2.7	-1.4
0.0	5.3	5.0	-0.5	0.0	2.0	0.5	-0.6
2.7	-2.8	-4.7	4.0	-2.0	-3.7	-1.9	-4.3
-0.9	-12.1	-10.2	-7.9	-1.2	-8.9	-6.3	-10.8
-7.3	-7.3	-7.9	-10.2	-9.8	-12.6	-13.0	-11.4
-13.3	-14.4	-16.3	-16.2	-7.5	-16.1	-16.9	-14.7
-26.0	-17.1	-18.8	-22.1	-16.4	-15.9	-20.3	-16.7
-28.0	-26.0	-31.1	-20.7	-20.8	-23.5	-22.2	-22.7
-35.1	-30.9	-33.4	-33.3	-23.0	-27.6	-29.8	-27.8

Figure 1. Degree of tilting converted from gyroscope data relative to Start button for x-axis (A) and y-axis (B) when acquiring targets on Samsung Galaxy Note3. The red squares denote the position of Start button.

their original state by having them press the Start button before each targeting task in order to prevent the functional area of the thumb from being affected [3]. We chose the Start button to be positioned where the thumb could reach naturally (The position of the Start button was empirically determined through a pilot study. We verified in the Results section that the position was within the comfortable region). They initiated the task by touching the red Start button and tapping on targets that appear in random regions using their thumb. The width of each square target was 7mm resulting in a 7 × 12 (84 cells) grid for Galaxy S4 and an 8 × 14 grid (112 cells) for Galaxy Note3. The target cells were highlighted one at a time in random order.

Procedure

At the beginning, we measured the length of each participant's thumb and palm size since they are important factors to determine the ability of the thumb to reach. The thumb length ranged from 53mm to 73mm (M = 62mm, SD = 6.8mm) and the average palm size was 81.5mm (palm length, SD = 7.1mm) × 105.8mm (palm breadth, SD = 8.6mm).

All participants performed the tasks for all cells of the grid once on Galaxy S4 first and then on Galaxy Note3. The devices' specific order was chosen from the device with mild reachability problem to severe. Before beginning the experiment with each device, they had a training session that consisted of tapping 16 targets, which were equally distributed across all regions of the screen. This allowed them to have a relative perception of comfortableness of reaching various regions of the device. For the main task, after tapping a target, they were asked to verbally report the rating of uncomfortableness of touching the target. The rating ranged from 1 to 3 (1 for <Natural for a thumb>, 2 for <Somewhat uncomfortable because of bending or stretching of the thumb>, and 3 for <Very uncomfortable because of too much bending or stretching of the thumb>).

Results

The larger touchscreen device (5.7" screen)

We aggregated the gyroscope data for touching each region and showed the results in a heat map (Figure 1). We calibrated all values to be aligned with the gyroscope value

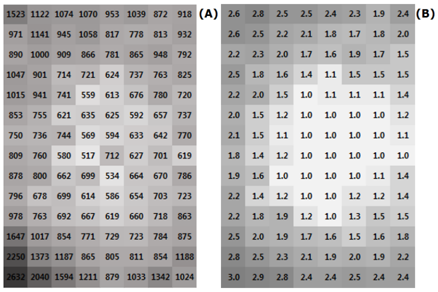

Figure 2. Mean completion time in milliseconds (A) and mean degree of uncomfortableness (B) on Samsung Galaxy Note3.

acquired upon touching the Start button where users could touch naturally when they are holding the device at their front-facing angle.

Figure 1(A) indicates the amount of left-right (or x-axis) tilting. The red color (positive value) means users tilted the devices to the right side, whereas the blue color (negative value) means they tilted the devices to the left side. Figure 1(B) indicates the amount of top-bottom (or y-axis) tilting. The red color (positive value) denotes the devices were tilted to make the top of the devices closer to the user, and the blue color (negative value) means the devices were tilted to make the bottom of devices closer to the user. The darker the color is, the more tilted the device is.

We confirmed that users tilted the devices much more on both axis when they had to touch farther targets by using linear regression to check the correlation between the amount of tilting and the distance of targets from the Start button. The tilting amount correlated significantly with distance for both x-axis (r = 0.584, p = 0.000) and y-axis (r = 0.735, p = 0.000).

Additionally, we confirmed that the tilting direction depended on the location of targets. First, for the targets placed on the left side, the tilting direction of y-axis was the opposite for top-left and bottom-left ones. Also, we verified that users tilted significantly more on x-axis for tapping the top-left than the bottom-left ($F_{1,10}$ = 15.315, p < .005) using RM-ANOVA. Second, for the targets placed on the right side, the tilting direction of x-axis was the opposite for top-right and bottom-right ones. We inferred that the ergonomical limitation of the thumb being rather difficult to bend downward may have resulted in such behavior. Although tilting directions for far-targets varied, they all were attempts to bring the target closer to their thumb.

We aggregated task completion time and uncomfortable degree of participants and represented the results using a heat map (Figure 2). We confirmed that the center of the screen was comfortable for all participants where the Start button was positioned. We also discovered that some targets near the right side edge were uncomfortable. The targets were within reach of a thumb but too close, so they required participants to bend their thumbs. Through these results, we found that not all uncomfortable targets were equally far. We verified task completion time has significant correlation with uncomfortable degree (r = 0.788, p = 0.000) using linear regression. Namely, participansts tilted the devices when they felt uncomfortable at tapping the targets even when they were close.

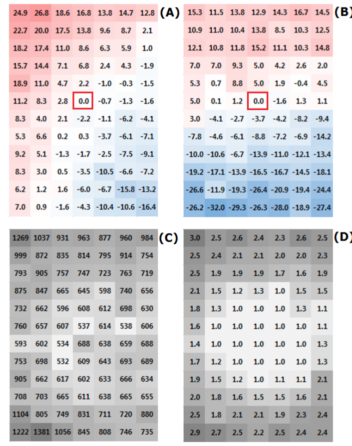

Figure 3. Degree of tilting converted from gyroscope data relative to Start button for x-axis (A) and y-axis (B), mean completion time in milliseconds (C), and mean degree of uncomfortableness from questionnaire (D) on Samsung Galaxy S4. The red squares denote the position of Start button.

Also, we discovered that the targets at the bottom-left region took significantly more time to touch than the targets at the top-left region ($F_{1,10}$ = 6.125, p < .05) using RM-ANOVA. Also, we confirmed that the bottom-left targets was significantly more uncomfortable than the top-left targets (Z = -2.000, p < .05) using Wilcoxon signed-rank test. We discuss this phenomenon by comparing them with the results on smaller devices.

The smaller touchscreen device (5" screen)

We observed similar results on a smaller screen device, the Galaxy S4 (Figure 3). The tilting amount correlated significantly with the distance for both x-axis (r = 0.592, p = 0.000) and y-axis (r = 0.738, p = 0.000), namely, users tilted the devices much more for farther targets regardless of the screen size if they were larger than five inches.

We also found that task completion time has significant correlation with uncomfortable degree (r = 0.823, p = 0.000) using linear regression and the targets near the right edge were uncomfortable because they were too close.

Interestingly, aligning the two results from smaller Galaxy S4 and larger Galaxy Note3 by the Start buttons excluding the non-existent targets, we found that they matched well in terms of both tilting direction and tilting amount for the targets in the same location. We verified strong correlations for both x-axis tilting (Pearson correlation coefficient of 0.958) and y-axis tilting (0.953). We also verified strong correlation for uncomfortable degree of two devices (0.918). It would be interesting to see if this concordance can be generalized into even larger devices.

The most interesting finding was that, in contrast with the result of larger Galaxy Note3, the targets at bottom-left region had no significant difference from the targets at top-

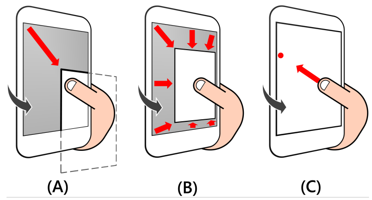

Figure 4. Three mobile interaction techniques using tilting gesture: *TiltSlide* (A), *TiltReduction* (B), and *TiltCursor* (C).

left region in task completion time ($F_{1,10}$ = 0.062, ns) and uncomfortable degree (Z = -1.000, ns). We could suspect that the uncomfortableness of the bottom-left area was due to the larger size of the screen. From this result, we suggest that we need to be careful when putting targets in the bottom-left area on a large mobile screen.

ONE-THUMBED INTERACTION BASED ON TILTING

We verified that users use the strategy of tilting the devices to bring the targets at uncomfortable regions closer to their thumbs. Tilting gesture in mobile devices has been actively evaluated, from exploring the effectiveness of combinations of tilt and drag directions [20] and the effective discretization of tilt angle [17] to evaluating the suitability of tilt to control pointers on large displays [5]. However, there have not been any researches on the tilting gesture for one-hand targeting on large mobile touchscreen. We first designed three novel mobile interaction techniques using tilting gesture: *TiltSlide*, *TiltReduction*, and *TiltCursor*.

TiltSlide (*TS*): When a user tilts a device, the screen slides to the tilting direction (Figure 4A). This method resembles iPhone 6/6+'s Reachability feature [1] although it only brings the screen down to one direction, i.e. *down*, whereas *TS* supports eight different directions depending on the way the device was tilted, i.e., *right-down*, *right*, *right-up*, *up*, *left-up*, *left*, *left-down*, and *down*. The screen slides until any of its edges hits the center of the physical screen.

TiltReduction (*TR*): When a user tilts a device, the screen shrinks to fit into a customized comfort region (Figure 4B). Inspired by *ThumbSpace* [14], we enable users to determine the comfort region as a rectangle that they diagonally draw by dragging their thumbs from top-left to bottom-right. The rectangle's aspect ratio is maintained to that of the screen to avoid distortion. While dragging, users can preview a shrunk-sized screen rendered within the rectangle. They were allowed to draw the rectangle repeatedly until their thumb could comfortably reach the entire rectangle.

TiltCursor (*TC*): A cursor appears when a user starts to drag while a device is being tilted (Figure 4C). At first, the cursor appears under the thumb and moves to the same direction as the dragging gesture. It also moves faster than the thumb movement to reach even far-targets by manipulating within the thumb's comfort region. The cursor does not exceed the boundary of the screen to be helpful especially on reaching four corners, since the width of a target at the corner goes infinite considering Fitts' law [8]. The target underneath the cursor is selected upon lifting.

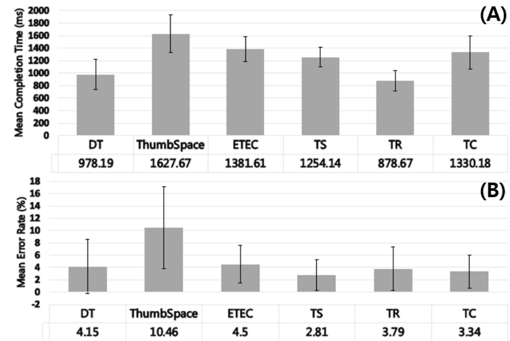

Figure 5. Mean completion time (A) and mean error rate (B) for one targeting. Error bars denote standard deviation.

In our first prototype, the amount of "slide" for *TS* was determined by the degree of x-axis tilting. However, we got the feedback that it was too sensitive, and it was difficult to control the degree of tilting. Thus, we changed *TS* to slide the screen to the point where any of edges hit the center of the screen, when the degree of tilting exceeds the predetermined thresholds of each eight direction. Also, the speed of a cursor was set to be 2.5 times that of a thumb after careful calibration. The tilting thresholds to "trigger" our methods used an average of the gyroscope value from the first experiment and were further adjusted heuristically through pilot study to better fit users' intention; 35 degrees on x-axis for *TR* and *TC*.

EXPERIMENTAL EVALUATION (EXPERIMENT 2)

To validate the effectiveness of our methods, we conducted a controlled and comparative experiment with three other methods: *Direct-Touch* (*DT*) (a normal method without an assistant), *ThumbSpace* [14], and *ETEC* [15]. We chose them since they suggest different methodologies such as a proxy region and an edge cursor. We chose *ETEC* over other cursor methods for the comparison, since it allows access to the entire area of a screen (while *Offset Cursor* [16,19] and *Adaptive Horizontal Offset* [10] do not). Also, *MagStick* [18] was not considered because its weakness for densely located targets may interfere with the result. We also excluded *TapTap* [18] because its main goal is to assist accurate target acquisition.

Hypotheses

We made hypotheses about the results from two different perspectives: method and region.

Comparison of the method

TR and *TS* would be faster than *ThumbSpace* (**H1**). Though they all use *Assisted Touch*, i.e., directly touching the target, *ThumbSpace* needs additional cognitive effort to determine the relative position of a target on an empty proxy view. We inferred that these additional costs would result in extra time. We also hypothesized that *Assisted Touch* (*TR*, *TS*) would be faster and more accurate than *Cursor* (*ETEC*, *TC*) (**H2**), since dragging a cursor until it reaches a target would take more time than directly touching the target with the thumb. Also, when the user lifts his/her thumb for selection,

ETEC							
-27.3	534.8	813	999.3	865.4	784.7	811.8	1491
213	801.9	908.3	820.8	344.6	853.9	910.2	1236
60.17	901.1	754.2	726.6	713.2	684.1	503.2	482.7
706.2	827.5	754.2	720.7	643.3	320.6	348.9	359
752.5	641.6	508	709.3	-135	265	569.5	457.6
823.3	999.1	491	527.3	400.4	220.6	115.7	143.9
1121	596.7	574.4	313.1	259.2	589.8	238.7	172.4
919.5	555.1	444.9	-197	274.3	238.6	155.2	223.6
1030	802.4	414.6	298.1	-170	184.4	-354	288.2
781.5	360.9	427.1	433.3	224	192.9	236.4	190.5
385.9	651.6	502.7	390.9	136	181.1	187.6	228.1
-922	635.7	343.8	486.3	409	211.7	199.4	121.9
-1207	-582	169.4	329	157.9	238.6	266.7	175.8
-1013	-959	-584	351	623.2	129.3	454.6	59.28

ThumbSpace							
-172	-49.1	480.3	106.4	947.4	431.4	748.3	391.3
143.5	723.1	755.9	865.3	462.1	855.6	1046	1368
-104	678.3	747.4	624.5	980.1	892.3	583.9	823.8
609.1	753.1	1004	715.4	774.5	563.1	1160	629.1
685.7	678.8	973.4	918.5	621.9	739.7	1024	846.6
630.1	903.1	712.1	891.9	695.6	1157	1074	983.1
845.3	1064	819.4	1020	900.8	955.3	865.6	889.9
807.2	911.9	1022	623.9	925.2	1050	788.3	1006
726.6	1201	956.8	1196	599.2	954.6	756.3	982.7
869.6	772.5	665.8	996	1039	832.4	1020	1413
194.2	525.6	889.6	837.1	822.1	1030	1257	1199
-1470	671.4	703.1	689.6	747.8	944.1	837.5	854
-1080	-531	192.1	874.2	510.1	963.1	724.9	1019
-1773	-1437	-750	372.3	588.1	354.5	1039	149.6

TiltReduction							
-953	-787	-334	-483	167.8	-209	108.2	-83.7
-718	-379	2.722	-24.8	-322	-106	159.3	578.6
-920	-217	-16.2	-74.7	146.9	195.1	-13.3	231.5
-245	4.611	192.2	136.8	83.92	-26.8	629.6	-119
-69.4	-160	54.03	252.5	-350	109.2	73.22	254.6
-56.1	4.611	-45.4	304.9	-218	147	145.1	177.4
244.8	193.1	-15.9	273.1	39.33	149.4	85.81	308.7
79.56	117.1	160.4	-273	181.4	121.3	63.89	319.4
-184	69.28	77.47	74.5	-308	3	-142	109.7
-89.4	62.17	327.4	144.1	89.69	104.7	139.6	115.9
-566	-151	52.28	130.5	-38.4	-29.9	70.89	476.3
-2180	-351	-231	-0.56	69.22	-7.11	-5.72	59.31
-2347	-1242	-503	453.6	-309	70.89	15.22	154
-2301	-1884	-1331	-210	-369	-149	-249	134.6

TiltCursor							
-110.2	528.2	777.1	576.2	1400	706.8	1179	764.7
121.4	937.4	1112	731.9	316.2	644.8	933.3	718.3
252.6	861.6	639.4	653.4	721.6	867.2	394.4	785.6
591.8	1186	764.3	596.8	525.7	278.2	651.1	418.9
807.4	429.3	525.1	549.3	4.333	434.5	347.1	498.4
574.9	684.3	385.9	566.1	-44.58	198.5	204.3	328.7
1114	578.3	586.9	175.9	182.8	185.3	172.1	79.28
918	294.8	208.6	-394.5	81.28	79.94	222.8	213.3
896.3	296.4	252.1	84.83	-244.5	147.4	-316.4	314.7
1021	459.7	125.7	264.7	95.86	20.67	38.5	270.1
534.7	325.7	179.3	291.1	87.31	-27.17	49.67	101.3
-1074	773.3	270.2	460.3	154.2	38.44	331.6	172.2
-1136	-360.8	106	485.3	77.69	310.6	382.4	437.2
-1053	-534	290.9	640.6	390.6	702.1	754.5	189.2

TiltSlide							
-195	-320	394	338	877.1	1236	464.4	733.5
583.2	684.9	850.6	417.1	689.4	443.8	637.2	851.1
71.94	955.7	602.4	20.67	551.4	589.4	1103	593.5
568.3	193.6	257.2	837.4	648.4	369.3	354.9	488.2
642.4	651.6	923.2	589.5	478.4	171.1	255.2	468.8
584.9	852.2	142	371.5	452.6	477.4	172.4	253.7
463.1	726.3	477.4	246.5	43.44	611.7	340.8	170.4
503.6	351.8	562.8	-36.7	-7	240.7	129.7	47.83
210.8	388.2	173.6	189.7	2.556	228.5	36.89	657.7
364.4	216.9	90.06	209.5	11.69	341.9	62.17	110.4
-274	365.5	249.2	98.72	57.75	82.89	205.4	153.1
-1307	-4.44	-67.2	502.2	149.1	33.78	615.3	215.9
-1737	-497	-271	-131	47.64	391.6	469.4	405.2
-1612	-1425	-518	-13.8	31.58	324.8	606.7	510.2

Table 2. The differences between each method and *Direct Touch* ($\text{Value}_{\text{method}}$ - Value_{DT}) in task completion time. The red color (positive value) denotes the region took more time than *DT*, and the blue color (negative value) denotes the region took less time than *DT*. The white color means there were no differences.

a slight disposition of the cursor could make an inaccurate target acquisition. In *Cursor*, *TC* would be faster than *ETEC* (**H3**) because *TC* enables users to start dragging nearer targets, while *ETEC* has to start dragging only from predetermined bezel locations. Since *TR* has reduced target size from shrinking the screen size to fit into the comfort region, *TR* would be less accurate than *DT* and *TS* (**H4**).

Comparison of the region

ThumbSpace would result in uniform task completion time among most regions (**H5**) since a user has to utilize a proxy view for every target; the targets in comfortable regions as well as in uncomfortable regions.

Participants

We recruited 18 participants (10 females) through an online reservation system announced on a campus bulletin board. Their age ranged from 20 to 30 (M = 24 and SD = 3.2). We screened participants to be right-handed and to have had experience with touchscreen mobile devices for more than one years. They were given $10 for their time.

Apparatus

We used Samsung Galaxy Note3 (151.2 × 79.2 × 8.3 mm, 5.7" screen). We developed an Android application [7] that implemented six methods: *TS*, *TR*, *TC*, *DT*, *ETEC*, and *ThumbSpace*. We collected task completion time and the number of errors through the application.

Task

The participants were asked to acquire a target as fast as and as accurate as possible while holding a device in their right hands. Targets were located in an 8 × 14 grid (112 cells) on screen and the width of each square target was 7mm. Each cell was activated and highlighted to be a target, and we counted the number of errors when non-highlighted cells were touched. We asked participants to sit straight on a chair and not to lean forward to control other external factors such as posture.

Procedure

The participants performed the target acquisition tasks using the six methods by five trials each. One trial contained 30 targets at randomly activated regions while maintaining the same total thumb-travel distance. We used a Latin Square design to eliminate ordering effects. A five-minute training session for each method was performed before they started the first trial to get themselves familiar with each method. They were allowed to have a five-minute break after five trials were completed. Seven questions were given upon completion of the experiment.

Results

We analyzed results (Figure 5) using RM-ANOVA and found significant main effects on task completion time ($F_{5,85}$ = 24.59, p < .01) and error rate ($F_{5,85}$ = 9.65, p < .01). Post hoc analysis using Bonferroni correction allowed us to rank the methods as follows: the fastest *TR* (878.67ms) and *DT* (978.19ms); *TS* (1254.14ms), *TC* (1330.18ms), and *ETEC* (1381.61ms); the slowest *ThumbSpace* (1627.67ms). Namely, only *TR* showed competitive performance with *DT*.

TR and *TS* were significantly faster than *ThumbSpace*; 1.9 and 1.3 times respectively (p < .001 for both), confirming **H1**. Only *TR* was significantly faster than *Cursor* (*ETEC*, *TC*); 1.6 and 1.5 times respectively (p < .001 for both). Also, only *ThumbSpace* (10.46%) had significantly more errors than all other methods; *TS* (2.81%), *TC* (3.34%), *TR* (3.79%), *DT* (4.15%), and *ETEC* (4.5%) (p < .001 for all), thus partially confirming **H2**. We suppose that difficulty in determining a slide direction took more time than expected for *TS*. We could not confirm **H3**; we speculate that the more complex method for the trigger of *TC* may have counter-affected performance in time, since *TC* requires a combination of two gestures (tilt and drag), while *ETEC* requires only one (drag). We could not confirm **H4**; even with its reduced size, the accuracy for *TR* was as good as *DT* and *TS*. We assume that showing a preview of reduced targets for *TR* while defining a customized comfort region might prevent making targets too small to touch accurately.

We show the mean task completion time of each method relative to that of *DT* for all areas of the screen in a heat map (Table 2). *ThumbSpace* generally performed worse than *DT* over most regions, confirming **H5**. Interestingly, all methods outperformed *DT* in the bottom-left region; it is consistent with the previous finding that bottom-left is the hardest region for direct touch and has to use assistive methods to get over a difficulty.

Actually, during the experiment, participants had a hard time and often dropped the device in an attempt to touch a

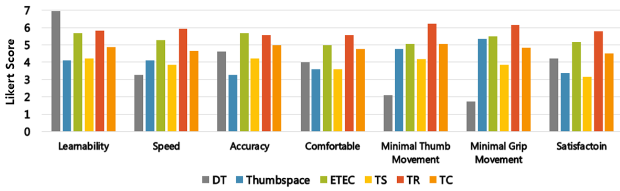

Figure 6. Questionnaire results (means)

bottom-left target with *DT*. One participant had to rotate the device 180 degrees to touch bottom-left and top-left targets. However, while using other assistive methods, none of the participants dropped or rotated the device.

We analyzed questionnaire results (Figure 6) using Friedman test; thumb movement ($\chi^2 = 44.25$, $p < .001$) and learnability ($\chi^2 = 44.85$, $p < .001$). *DT* was significantly less preferred than all other methods in thumb/grip movement (mean = 2.1/1.7 respectively) ($p < .001$ for all); however, *DT* (6.9) was preferred than all other methods except *TR* (5.8) in terms of learnability ($p < .05$ for all). In summary, compared to *DT*, *TR* showed a promising result because it required less thumb/grip movement with comparable performance in speed, accuracy, and learnability.

REAL SCENARIO EVALUATION (EXPERIMENT 3)

We confirmed that our tilting methods had potential for acquiring targets in an uncomfortable region. However, we cannot apply the result directly to real mobile phone circumstances because we excluded several factors in the previous experiment: (1) Mechanisms of our methods could be in conflict with conventional mobile interactions such as long touch, double tap, swipe, scroll, and drag; and (2) Though we ascertained that a reduced screen size had not shown any difference in accuracy for acquiring the target, there could be overlooked factors such as small font size or figures that might be difficult to see the details when reduced in size.

We noted that one-hand targeting interactions used different trigger mechanisms. We speculated that they could show different performance in real mobile phone scenarios because of conflicts with their trigger mechanisms and existing mobile interactions. Also, smartphone usage could require frequent re-triggers to perform a series of real-world tasks (e.g. tapping a far-target in reduced screen size and checking details in original large screen size using the reduced-screen method). Thus, the time required for trigger becomes more crucial in real mobile phone scenarios.

We designed a user study to evaluate one-hand targeting interactions triggered by different mechanisms in real mobile application scenarios. We chose two among our methods: *TR* (triggered by tilting gesture) and *TC* (triggered by both tilting gesture and dragging). We excluded *TS* since it used the same trigger mechanism as *TR* with worse performance than *TR* in our Experiment 2. We compared our methods with others introduced in previous research. *ETEC* (triggered by dragging) was included for comparison with *TC*, namely we wanted to investigate the effect of

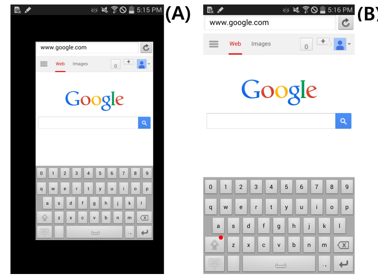

Figure 7. *TiltReduction* (A) and *TiltCursor* (B) on the Android application developed to similar to common web browsers.

trigger mechanisms for the same *Cursor* methods. We also chose *One-handed operation mode* [2] (triggered by touch gesture) for comparison with *TR*, since they used the same reduced-screen method to assist one-hand targeting. In short, we conducted experiments using five methods: *Direct Touch*, *TR*, *TC*, *ETEC*, and *One-handed operation mode*.

We also chose the tasks that could represent real mobile phone usage. Bohmer et al. [4] logged mobile phone usage and uncovered patterns of application usage during a day. Among them, we chose two frequently used tasks in our daily usage of smartphones: typing and search. For typing tasks, we asked participants to compose and send an e-mail. For search tasks, we asked participants to do web surfing.

Participants

We recruited 15 participants (4 females) through an online reservation system announced on a campus bulletin board. We screened participants to be right-handed and to have had experience with touchscreen mobile devices for more than one year. We recruited only those who usually used QWERTY software keyboard as their primary input method on mobile devices and who mainly used Google Search /Gmail as web browser and e-mail to eliminate possible learning effects. Their age ranged from 20 to 29 years (M = 23.8, SD = 2.6). They were compensated with $10 for an hour, total experiment time.

Apparatus

We chose Samsung Galaxy Note3 (5.7" screen) to compare the result with our previous experiments. We developed an Android application whose appearance was similar to common web browsers in smartphones. We used WebView (an Android control) and built a custom software keyboard designed to function similarly to other common smartphone QWERTY keyboards (Figure 7). It should be noted that no correction algorithms were applied to typing. We collected task completion time, the number of "Delete" key presses on the software keyboard, and touch coordinates.

Task

The participants performed two tasks: sending an e-mail and web surfing. For the e-mail task, they navigated the Google homepage upon launching the application, and they were asked to navigate Gmail and read a randomly chosen, pre-designated e-mail out of 15. The e-mail consisted of 12 small letters that could only be recognized on a normal size

Q1	Go to Notice, and find out opening hours for <Name> Café.
Q2	Go to News, and find out the ratio of Asian students in internationalization chart.
Q3	Go to Events, and find out who directed the movie showing on October 16.
Q4	Go to Academic-College of Engineering-Dept. of Computer Science and Engineering, and list the name of a seminar on September 15.
Q5	Go to About-Symbols, and find out School Bird.

Table 3. Questions given for web surfing tasks.

screen (not on a reduced size screen). The 12 letters were compounded with random alphabets (e.g., qUeK xHaP wYcI) to have no meaning, while maintaining the same average amount of uncomfortable degree based on the first experiment (we ensured that all the e-mails had no significant difference in time for typing their letters using direct touch (*DT*) through a pilot study). We asked the participant to reply to an e-mail with the same sequence of the letters by memorizing a chunk of four letters at a time before typing, so that he/she has to consistently change between the normal size and the reduced size for every e-mail. Also he/she was asked to perform the tasks as quickly and as accurately as possible.

For the web surfing task, the participants performed without any restriction in time and action. They were given a question before starting and asked to find an answer by surfing the web. The questions are summarized in Table 3. We utilized our university website because navigating the website required various mobile interactions such as swipe and horizontal and vertical scrolling.

Procedure

The participants were asked to hold the devices in their right hands and use all five methods one at a time. The five methods were assigned to subjects using a Latin Square to eliminate ordering effects. Before performing e-mail and web surfing tasks using each method, the participants had a training session long enough to get used to the method and the e-mail task. And then they were asked to perform a web surfing task using the method as frequently as possible, since we wanted them to have enough experience for insightful feedback. Lastly, they performed an e-mail task three times for the method and were asked to use the method only for touching uncomfortable targets. Upon completion of the experiment using all methods, the participants completed a questionnaire with seven questions for subjective evaluation using a seven point Likert scale [Rating: 1 = strongly disagree, 7 = strongly agree].

Hypotheses

For the e-mail task, we hypothesized that *TR* would be faster than *One-handed operation mode*, since *One-handed operation mode* needs additional time to be triggered (**H1**). Furthermore, *TR* would not have a significant difference with *DT* on task completion time, considering that trigger by tilting gesture requires less time (**H2**). *Cursor* (*ETEC*, *TC*) would be slower than *Assisted Touch* (*TR*, *One-handed operation mode*) since *Cursor* requires a separate trigger

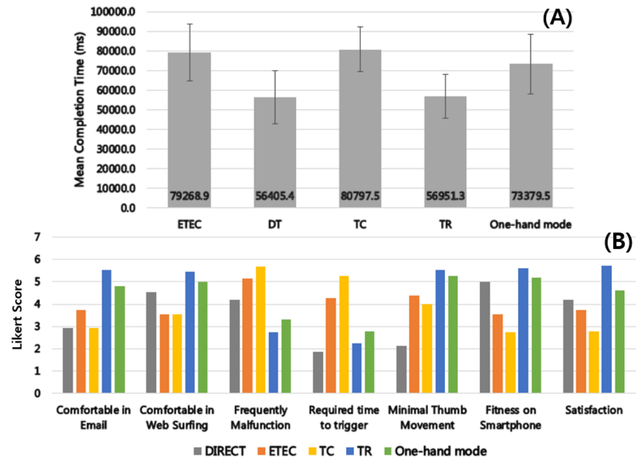

Figure 8. Mean completion time for e-mail tasks (A) and mean results of each question (B).

action for each target acquisition, whereas *Assisted Touch* allows users to select multiple targets once it is triggered (**H3**). *DT* would have more errors than other methods, because directly touching a far target may result in miss-pressing (**H4**). For the web surfing task, *Cursor* (*ETEC*, *TC*) would be preferred less than *Assisted Touch* (*TR*, *One-handed operation mode*) since dragging a cursor conflicts with swiping or scrolling in web browsers (**H5**).

Results

E-mail task

We analyzed results (Figure 8) using RM-ANOVA and found a significant main effect on task completion time ($F_{4, 56} = 27.22$, $p < .001$) and the number of errors ($F_{4, 56} = 4.81$, $p < .01$). We treated the number of pressing the "Delete" key as the number of errors. Post hoc analysis using Bonferroni correction allowed us to rank the methods as follows: *DT* (56405ms) and *TR* (56951ms); *One-handed operation mode* (73379ms), *ETEC* (79268ms), and *TC* (80797ms).

TR was significantly faster than *One-handed operation mode*; almost 1.3 times ($p < .001$), confirming **H1**. From the result, though both *TR* and *One-handed operation mode* used mode-change mechanisms to trigger reduced screen, tilting gesture was significantly faster than touch gesture. We also confirmed **H2**; all other methods except *TR* were significantly slower than *DT* ($p = .001$ for *One-hand operation mode*, otherwise $p < .001$). Though *TR* was frequently re-triggered during the e-mail task, it showed competitive performance in time with *DT*, like the result of Experiment 2. For such reason, we were able to verify tilting gesture requires less time. Only *TR* was significantly faster than *Cursor* (*ETEC*, *TC*); almost 1.4 times ($p < .001$ for both), partially confirming **H3**. In other words, the fewer number of trigger by touch gesture (because of multiple target selection in a trigger) requires as much time as the more number of trigger by dragging.

DT (mean = 1.82 errors) had significantly more errors than *TR* (0.4) ($p < .05$) excluding other methods: *ETEC* (0.69), *TC* (0.91), and *One-handed operation mode* (1.09); partially confirming **H4**. Comparing this result with the accuracy result of Experiment 2, some trigger mechanisms such as dragging and touch gesture could have as many errors as *DT* in real mobile phone circumstances because of conflicts with existing mobile interactions. We assume that touch action for the trigger might be mis-classified as direct touch.

We analyzed questionnaire results on comfortableness for e-mail task ($\chi^2 = 31.45$, $p < .001$) using Friedman test. *TR* (mean = 5.5) and *One-handed operation mode* (4.8) were significantly preferred than all other methods ($p < .05$ for all). The dragging mechanism for *TC* (2.9) or *ETEC* (3.7) was felt as uncomfortable as *DT* (2.9), which does not assist targeting at all for e-mail task on a large smartphone.

Web surfing task

We confirmed **H5** by analyzing questionnaire results on comfortableness for web surfing task ($\chi^2 = 27.87$, $p < .001$). *Cursor* (*ETEC*, *TC*: 3.5/3.5 respectively) was preferred significantly less than *Assisted Touch* (*TR*, *One-handed operation mode*: 5.5/5.0 respectively) and even *DT* (4.5) ($p < .05$ for all). We verified that participants felt that they activated unintended actions significantly more with *Cursor* (5.1/5.7) than *Assisted Touch* (2.7/3.3) ($p < .01$ for all). They rated the appropriateness and preference of *Cursor* (3.5/2.7 and 3.7/2.8) on smartphones to be lower than those of *Assisted Touch* (5.6/5.2 and 5.7/4.6) ($p < .01$ for all).

We got interesting feedbacks from interviews after the user study; most participants (86%) reported the difficulty of dragging gesture with the device tilted. We also verified through the questionnaire results that they felt *TC* (5.3) required significantly more time to trigger than *ETEC* (4.3) ($p < .05$). These are kinds of feedbacks not found in the previous study, implying that they might be confused to deal with three factors: tilting, dragging, and conventional mobile interactions such as scrolling. They also said that the typing using *Cursor* was especially annoying since they wanted to type in the way they were already familiar with.

DISCUSSION

In evaluating one-hand targeting methods in real mobile phone scenarios, we verified that their trigger mechanisms were important factors as assisting techniques on large mobile devices. In Table 1, the *Trigger Mechanisms* can be categorized into *Always On* (always staying as "triggered"), *Tapping*, *Dragging*, *Mode Change* (switching use/non-use of targeting mechanism), and *Back-Side Touch* (using a touch pad attached on the back side). Furthermore, there could be other mechanisms for trigger such as *Long Touch* and *Double Touch*, but they had not been explored in the past to our knowledge. It might be because of conflicts with conventional touch interactions in mobile phones.

We verified that *ThumbSpace* [14] could be helpful for some targets in uncomfortable regions, but generally showed bad performance among most regions including comfortable regions. Therefore, the *Always On* mechanism has to be used for targeting methods designed to support both uncomfortable regions and comfortable regions.

Since the *Tapping* mechanism overlaps with direct touch in mobile phones, targeting methods using this mechanism have to block the direct touch (e.g. *Offset Cursor* [16,19]) or use other means of resolving the ambiguity (e.g., *TapTap* [18] is triggered only by touching the region that has two targets in close proximity).

The *Dragging* mechanism can be divided into dragging *anywhere* and dragging from the *edge*. Targeting methods using *Dragging* a*nywhere* [10,15,18] conflict with dragging interactions generally used in mobile phones. Thus, they have to block the conventional dragging or utilize other conditions, e.g., *TC* provides a cursor by dragging while the device is tilted. However, the results of our Experiment 3 showed that users felt it difficult to satisfy both conditions. Hence, we suggest that combining several mechanisms has to be designed and applied with care.

The *Mode Change* mechanism utilizes various modalities such as pressing a *physical button* [1,11,22], defining a unique *touch gesture* [2], or using a built-in sensor, e.g., *tilting gesture*. Through Experiment 3, we verified that *tilting gesture* was significantly faster than *touch gesture*, which is considered one of the most efficient modes of touch [6]. However, the targeting methods using sensors must consider false positive issues and endeavor to stabilize them. In order to stabilize our tilting mechanism, we mainly relied on the gyroscope sensor to detect triggers (rather than an accelerometer sensor) and used the threshold angles which are beyond the normal shake while typing and walking, except only rare circumstances, e.g., recklessly driven public buses. We verified them through an informal in-lab experiment by constantly using our custom-built web browser from Experiment 3.

Overall, we verified that while using a cursor can assist in acquiring far-targets, it is slower than touching with a thumb. Therefore, targeting methods using *Cursor* will be useful if used with direct touch in comfortable regions.

CONCLUSION AND FUTURE WORK

In this paper, we explored users' touch behavior using tilting devices on large mobile touchscreens. We also investigated the thumb's comfortable region with state-of-the-art large mobile devices and proposed that the bottom-left region is yet to be explored for further optimization. Based on our observations, we designed three novel mobile interaction techniques using tilting gesture to facilitate one-hand targeting. We analyzed them with other methods of related works in a controlled environment and real mobile phone scenarios. We found that the trigger mechanism of one-hand targeting methods was an important factor for time, accuracy, or preference. Based on the analysis, we proposed design considerations as our design space for one-hand targeting interactions.

We could not evaluate our methods with the recently released iPhone6/6+'s method. It may be effective for upper half screen; thus, we may compare it with our methods in the future. Also, we would like to empirically analyze false positive trigger issues. From the first experiment, we found that tilting patterns of a smaller device matched well with those of a larger device for same regions. It will be interesting to expand the experiment to other larger or smaller devices to see if we can generalize such pattern.

ACKNOWLEDGEMENTS

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (NRF-2014R1A2A2A03006998) and also supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (NRF-2013R1A1A3006706).

REFERENCES

1. https://www.apple.com/iphone-6/design/ Retrieved on Sept. 19, 2014.
2. http://www.androidcentral.com/how-shrink-galaxy-note-3-s-display Retrieved on Sept. 20, 2014.
3. Bergstrom-Lehtovirta, J. and Oulasvirta, A. Modeling Functional Area of the Thumb on Mobile Touchscreen Surfaces. *Proc. CHI'14*. 1991-2000. 2014.
4. Bohmer, M., Hecht, B., Schoning, J., Kruger, A., and Bauer, G. Falling Asleep with Angry Birds, Facebook and Kindle: a large scale study on mobile application usage. *Proc. MobileHCI'11*. 47-56. 2011.
5. Boring, S., Jurmu, M., and Butz, A. Scroll, Tilt or Move It: Using Mobile Phones to Continuously Control Pointers on Large Public Displays. *Proc. OzCHI'09.* 161-168. 2009.
6. Bragdon, A., Nelson, E., Li, Y., and Hinckley, K. Experimental Analysis of Touch-Screen Gesture Designs in Mobile Environments. *Proc. CHI'11*. 403-412. 2011.
7. Chang, Y., L'Yi, S., and Seo, J. Reaching Targets on Discomfort Region Using Tilting Gesture. *Proc. UIST'14*. 115-116. 2014.
8. Fitts, P.M. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47, 381-391. 1954.
9. Goel, M., Wobbrock, J.O., and Patel, S.N. GripSense: Using Built-In Sensors to Detect Hand Posture and Pressure on Commodity Mobile Phones. *Proc. UIST'12*. 545-554. 2012.
10. Huot, S. and Lecolinet, E. Focus+Context Visualization Techniques for Displaying Large Lists with Multiple Points of Interest on Small Tactile Screens. *Proc. Interact'07*. 2007.
11. Karlson, A.K. and Bederson, B.B. One-Handed Touchscreen Input for Legacy Applications. *Proc. CHI'08*. 1399-1408. 2008.
12. Karlson, A.K., Bederson, B.B., and SanGiovanni, J. AppLens and LaunchTile: Two Designs for One-Handed Thumb Use on Small Devices. *Proc. CHI'05*. 201-210. 2005.
13. Karlson, A.K., Bederson, B.B., and Contreras-Vidal, J.L. Understanding Single-Handed Mobile Device Interaction. *Tech Report HCIL-2006-02.*
14. Karlson, A.K. and Bederson, B.B. ThumbSpace: Generalized One-Handed Input for Touchscreen-Based Mobile Devices. *Proc. Interact'07*. 324-339. 2007.
15. Kim, S., Yu, J., and Lee, G. Interaction Techniques for Unreachable Objects on the Touchscreen*. Proc. OzCHI'12*. 295-298. 2012.
16. Potter, R.L., Weldon, L.J., and Shneiderman, B. Improving the accuracy of touch screens: an experimental evaluation of three strategies. *Proc. CHI'88*. 27-32. 1988.
17. Rahman, M., Gustafson, S., Irani, P., and Subramanian, S. Tilt Techniques: Investigating the Dexterity of Wrist-Based Input*. Proc. CHI'09*. 2009.
18. Roudaut, A., Huot, S., and Lecolinet, E. TapTap and MagStick: Improving One-Handed Target Acquisition on Small Touchscreens. *Proc. AVI'08*. 146-153. 2008.
19. Sears, A. and Shneiderman, B. High Precision Touch-screens: Design Strategies and Comparison with a Mouse. *International Journal of Man-Machine Studies*. 593-613. 1991.
20. Tsandilas, T., Appert, C., Bezerianos, A., and Bonnet, D. Coordination of Tilt and Touch in One- and Two-Handed Use. *Proc. CHI'14*. 2001-2004. 2014.
21. Vogel, D. and Baudisch, P. Shift: A Technique for Operating Pen-Based Interfaces Using Touch. *Proc. CHI'07*. 657-666. 2007.
22. Weberg, L., Brange, T., and Hansson, A.W. A Piece of Butter on the PDA Display. *CHI'01 Extended Abstracts*. 435-436. 2001.
23. Yang, X.D., Irani, P., Boulanger, P., and Bischof, W.F. One-Handed Behind-the-Display Cursor Input on Mobile Devices. *CHI EA'09.* 4501-4506. 2009.
24. Yang, X.D., Mak, E., Irani, P., and Bischof, W.F. Dual-Surface Input: Augmenting One-Handed Interaction with Coordinated Front and Behind-the-screen Input. *Proc. MobileHCI'09*. 1-10. 2009.
25. Yu, N.-H., Huang, D.-Y., Hsu, J.-J., and Hung, Y.-P. Rapid Selection of Hard-to-Access Targets by Thumb on Mobile Touch-Screens. *Proc. MobileHCI'13*. 400-403. 2013.