How to Position the Cursor? An Exploration of Absolute and Relative Cursor Positioning for Back-of-Device Input

Khalad Hasan[1], Xing-Dong Yang[2], Hai-Ning Liang[3], Pourang Irani[1]

[1]Depart. of Computer Science
University of Manitoba
Winnipeg, Manitoba, Canada
{khalad, irani}@cs.umanitoba.ca

[2]Depart. of Computer Science
University of Alberta
Edmonton, Alberta, Canada
xingdong@cs.ualberta.ca

[3]Department of Computer Science
& Software Engineering
Xian Jiaotong-Liverpool University
Suzhou, China
haining.liang@xjtlu.edu.cn

ABSTRACT

Observational studies indicate that most people use one hand to interact with their mobile devices. Interaction on the back-of-devices (BoD) has been proposed to enhance one-handed input for various tasks, including selection and gesturing. However, we do not possess a good understanding of some fundamental issues related to one-handed BoD input. In this paper, we attempt to fill this gap by conducting three studies. The first study explores suitable selection techniques; the second study investigates the performance and suitability of the two main modes of cursor movement: Relative and Absolute; and the last study examines solutions to the problem of reaching the lower part of the device. Our results indicate that for BoD interaction, relative input is more efficient and accurate for cursor positioning and target selection than absolute input. Based on these findings provide guidelines for designing BoD interactions for mobile devices.

Author Keywords

Back-of-device input, touch input, selection mechanism, relative and absolute cursor positioning.

ACM Classification Keywords

H.5.2. [**Information interfaces and presentation**]: User Interfaces, Input Devices, Interaction Styles.

General Terms

Design, Experimentation, Human Factors.

INTRODUCTION

When interacting with a mobile device such as a smartphone, people often chose to use only one hand [9]. One-handed interaction is convenient and allows users to multi-task by freeing their other hand for tasks such as holding a bus handle. Karlson et al. [9] have found that users prefer using one hand for two handed tasks. Despite some advantages, this form of interaction imposes several limitations, such as occlusion and reachability [8, 20, 27]. Prior studies have shown that these challenges can be overcome by allowing users to interact with the back-side of the device [1, 15, 18], or *back-of-device input*. Back-of-device (BoD) input allows users to control an on-screen cursor from behind the screen, thus reducing occlusion and improving the performance of some routine tasks, such as pointing and steering [26].

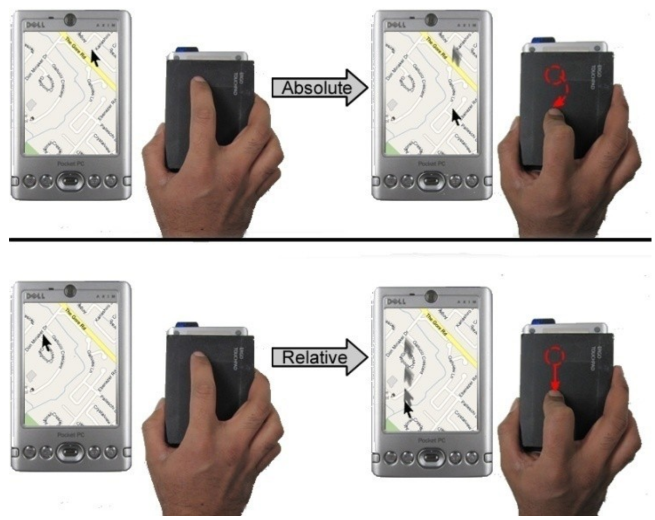

Figure 1. One-handed back-of-device input using the index finger (actual device used in our studies). In this case a touch pad senses input, allowing for both absolute (top) and relative (bottom) input, each of which present different affordances and limitations.

Cursor movement can be performed in either absolute or relative mode, and their hybrid variants [12, 19]. In absolute mode, the input device's input space has a one-to-one mapping to the screen's display space (Figure 1, top). In relative mode, the user's hand motion maps to the motion of the cursor so that the direction of the cursor's movement is consistent with the direction of the user's hand movement. (Figure 1, bottom). Despite the wide array of applications benefitting from BoD input (from pointing to text-entry) [1, 10, 16, 18, 21, 25, 26], very little research has focused on studying *selection mechanisms* and *positioning control* for *one-handed back-of-device* input. This has left some fundamental questions unanswered, such as: (1) What selection mechanism is suitable for the BoD input? (2) Which mode of cursor positioning (absolute or relative) is preferable on

MobileHCI'12, September 21–24, 2012, San Francisco, CA, USA.

the back? (3) How will target size and location affect performance with these two modes?

To answer these questions, we conducted three separate studies and from the results we make the following contributions: (1) identify suitable selection mechanisms for one-handed BoD input; (2) explore the benefits of absolute and relative cursor control for BoD input; (3) examine two variations of absolute pointing mode to resolve the reachability issues discovered in our studies.

RELATED WORK

One-handed interaction

One-handed interaction is a very common method for operating mobile devices. Several in-situ observations demonstrated that at least 74% of mobile users employ only one hand when interacting with their cellular devices [8]. The thumb is thus naturally the main input method in such contexts [8, 13]. However, the thumb is often prone to inaccurate selection [20], and can be too short to reach the entire screen [8]. To alleviate concerns with thumb interaction, Applens and LaunchTile resorted to using thumb gestures as input [9]. With Applens, users could very easily navigate a grid of values using simple thumb gestures. LaunchTile allowed users to access parts of a tabular region (such as a calendar) by pressing on soft buttons associated with an area of the grid. User gestures were correct 87% of the time, implying that thumb gestures are memorable.

A study on the biomechanical limitations of thumb input revealed that users do not interact with all areas of a device with equal facility [8]. User grip, hand size, device ergonomics, and finger dexterity can strongly impact thumb reach. These results led to the development of ThumbSpace [6], which provides a miniature proxy window of the entire touch-screen. The position of the proxy window is user-defined and thus enhances thumb reach. In an extensive study, users performed better at selecting distant targets using Thumbspace than other techniques. Used in conjunction with Shift [20], ThumbSpace can be precise for target selection with the finger [7].

These findings suggest that there is sufficient evidence for the use of one-handed interaction techniques on mobile devices. However, in addition to problems of inaccurate finger pointing, which can be resolved with techniques such as Shift [20], one-handed input is also prone to limited reach. To resolve these primary concerns, researchers have proposed using the space on the back of a mobile device to interact with virtual content.

Back-of-device input

There has been significant recent activity considering the potential of BoD interaction [1, 10, 15, 16, 25]. BlindSight [10], and RearType [16] support text-entry from the back of a mobile phone by placing a key pad behind the screen. HybridTouch has a trackpad mounted on the back of a PDA to enable gesture based commands for tasks such as scrolling and panning. LucidCursor [25] has an optical sensor on the back of a PDA, allowing users to control a cursor from the back-side. NanoTouch [1] demonstrates that BoD input can lead to very small devices by avoiding on-screen finger occlusion.

However, when interacting with the back of the device, the user has limited knowledge of her finger position. LucidTouch [21] uses a rear-mounted camera to provide constant on-screen feedback of back-of-device finger movements and positions. Most other systems simply use an auxiliary sensor (such as a trackpad) on the back-side [1, 18, 22, 26], and the cursor(s) only shows up after the user's finger touches the back.

Cursor positioning through the device's back-side can be controlled with either absolute or relative movement (Figure 1). In absolute mode, a one-to-one mapping between the input and the display space positions the cursor at the touch location. In relative mode, the cursor position is updated with motions of the user's finger. In absolute mode, the cursor jumps discretely from location to location; whereas in relative mode the cursor always moves in continuous trajectories without any jumps and often requires users to perform clutching actions.

Existing systems for BoD input primarily rely on absolute input [1, 2, 21] or relative input [18, 23, 26]. Yang et al. [26] show that pointing and steering tasks on a mobile device can be carried out more precisely and efficiently with relative control than absolute control. However, in their study the two positioning modes were not used on the same side of the device. In their setup, absolute control was used on the front of the device, while relative control was used on the back. To the best of our knowledge, very little is known on which of these two control mechanisms works best for cursor control for BoD input.

Absolute positioning can be faster than relative control because it allows for rapid cursor displacement from one area to another [12]. However, absolute control for BoD can be imprecise and prone to reachability issues akin to on-screen input [8].On the other hand, relative control can be slow, particularly in cases requiring a large amount of clutching [26], but has the added benefit of precise cursor control [19]. Also, relative control does not require a one-to-one mapping between the input space and display space, allowing users to easily reach remote targets. It is unclear which of these two styles allow for more effective BoD input.

Selection mechanisms

There are several mechanisms that allow target selection while interacting from the back-side. Common techniques rely on land-on and lift-off operations. In land-on, selection occurs when the finger first touches the input component of the device, whereas in lift-off the target is selected when the finger is lifted after contact. Double tap, a combination of land-on and lift-off, is another possibility in which a selection would take place on a sequence of land-on, lift-off, and land-on actions. In addition, buttons placed on or around the device can serve to select targets.

There has been little research examining selection mechanisms and interaction from the backside. In one experiment, Baudish and Chu [1] compared target selection on lift-off and on pressing a button placed on the side of a device while users interacted through its back-side. They found that users committed more errors using lift-off across devices smaller than 2.4″. In terms of time spent per task, they found that users did better with lift-off than the side button for the 2.4″ device. In a second experiment, they found that given targets with sizes of 1.4mm, 2.8mm, 5.5mm, and 11mm participants made few errors and took less time in target selection with the side button than lift-off. Lift-off also has the unwanted effect of always having an active selection, i.e. once the user commits to touching the screen an object will get selected the moment the lift-off.

EXPERIMENT 1: SELECTION MECHANISMS

This first study investigated selection mechanisms when interaction is carried out from the back-side. We were mainly concerned with finding out which selection mechanism(s) would be suitable for absolute and relative modes.

Apparatus

The study was conducted on Dell Axim X30 with a 624 MHz processor, 53×71mm (240 × 320) TFT touch screen, and a size of 81×127×18mm. We placed a trackpad (from Ergonomic®) on the back of the device, and adjusted the size and the position of the trackpad so that they were mapped on a one-to-one basis with the touch screen in the front. We could have used other forms of sensors, but this provided the most pragmatic method for obtaining either absolute or relative input on the back of the device. The application was implemented in C#.NET.

Participants

Twelve participants (10 males) between the ages of 21 and 30 were recruited from a local university to participate in this study. Participants were all right-handed.

Task and procedure

We used a target acquisition task in which participants interacted with the device with their right hand. Each trial began when a participant clicked a "Start" button which was placed randomly but at a fixed distance of 100 pixels (1 pixel = 0.22 mm in real world units) away from the center of the screen. Tapping on the region of the trackpad directly underneath the button would start the trial. The button would then be replaced by a cursor pointer which participants would have to position in a square-shaped target to select it. The target was always placed at the center of the display. A trial ended either if a successful selection was made or if the participants failed to do so within 25 secs. Warm-up trials were given to participants prior to starting the experiment. The experiment lasted about 45 minutes.

Experimental conditions

Target selection mechanisms

We considered several existing techniques as potential candidates, e.g. lift-off, dwell, double-tap, hardware button, etc. However, some of them are not suitable for BoD input. For example, we did not include a lift-off selection as it is always active, i.e. the moment the finger rests on the back touch sensor, selection has to be invoked. We also excluded dwell because of its speed overhead. In this experiment, we tested three selection mechanisms: *side-button*, *double-tap*, and *frontside-touch*. *Side-button* allowed participants to make a selection by pressing their thumb on a button placed on the side of the device. Double-tap was implemented in a standard way as how it works on a touchpads. To make a selection using *double-tap*, participants would have to tap the touchpad twice inside the target within a short time frame (300ms). *Frontside-touch* was derived from dual-surface input [26], in which the selection was made by tapping anywhere on the front display using the thumb.

Relative cursor movement

In relative mode, participants were asked to move the cursor anywhere inside the target in the same way as with a trackpad and make a selection with one of the three techniques. A control-display ratio of 1:1 was used, and no cursor enhancements and acceleration were included to avoid confounds. Clutching was needed if a single movement did not bring the cursor inside the target.

Absolute cursor movement

Initially, we wanted participants to land a finger on the area exactly underneath the target. Informal trials indicated a high percentage of errors, often requiring a few attempts before landing on the target. This was mainly due to the discrete nature of movements or lack of continuous visual feedback. We experimented with several modifications, and found the approach used by [1] to be easy to use and provided high accuracy. In their technique, as long as the finger was left on the trackpad, users could drag the cursor and move it to the intended target. Once on the target, they could make a selection. We used this method of absolute cursor movement for all our experiments.

Experimental design

The experiment employed a 2×3×3 within-subject factorial design. The independent variables were *Cursor Movement Mode* (*Relative* and *Absolute*),*Selection Mechanism* (*side-button*, *double-tap*, and *frontside-touch*), and *Target Size* (*14*, *22*, *30* pixels or 3.1, 4.8, 6.6 mms). Each trial was repeated 6 times, with the presentation of *Cursor Movement Mode* and *Selection Mechanism* counterbalanced using a Latin Square design and *Target Size* presented randomly.

Dependent measures include selection time and error rate. Since our main focus was to investigate the performance of selection techniques, pointing time, i.e. the time the cursor traveled from the start button to the target, was excluded from the analysis. Thus selection time was measured from when the cursor entered the target to when the target was selected successfully. An error occurred if participants made the selection outside the target area. The trial did not stop until the target was successfully selected.

Hypotheses

Based on the properties of our selection techniques and cursor positioning modes, we had the following hypotheses:

H1: Side-button will have the lowest selection time as it requires the least amount of hand movement for selection;

H2: Double tap with absolute mode will result in higher errors as participants need to tap twice on the target.

Results

Selection time

For both absolute and relative modes, *side-button* and *front-side-touch* had similar performance times, with *double-tap* being the slowest mechanism (Figure 2 left).

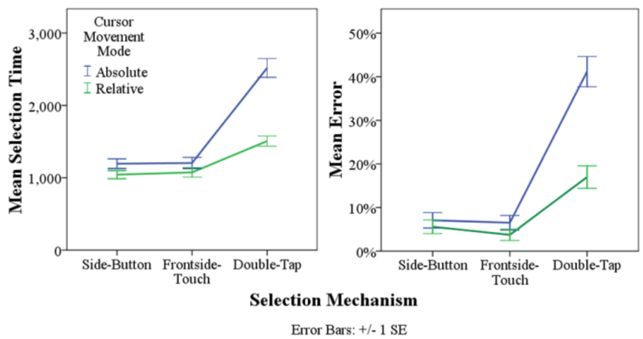

Figure 2. Left: Mean selection time across all techniques. Right: Mean error rates across all techniques.

We further analyzed the data separately for each mode using the Repeated-Measures ANOVA and Bonferroni corrections for post-hoc comparisons.

In *absolute mode*, there were significant effects for both *Selection Mechanism* ($F_{2,22}$=28.72, p<0.001)and *Target Size* ($F_{2,22}$=20.4, p<0.001), and a significant interaction effect between *Selection Mechanism*×*Target Size* ($F_{4,44}$=12.09, p<0.001). Post-hoc comparisons showed the *side-button* (M=1195ms, s.e.=84) and *frontside-touch*(M=1201ms, s.e.=104) were significantly faster than *double-tap* (M=2618ms, s.e.=252; p<0.001). There was no significant difference between *frontside-touch* and *side-button* (p=1). Similarly, post-hoc showed significant differences among all three target sizes (p<0.02) (Figure 3 left).

In *relative mode*, there were significant effects for both *Selection Mechanism* ($F_{2,22}$=6.87, p<0.01) and *Target Size* ($F_{2,22}$=13.16, p<0.001), but no interaction effect was found between *Selection Mechanism*×*Target Size* ($F_{4,44}$=1.00, p=0.41). Post-hoc pair-wise tests indicated a significant effect between *side-button* (M=1044ms, s.e.=96) and *double-tap* (M=1512ms, s.e.=100), and *frontside-touch* (M=1077ms, s.e.=109) and *double-tap* (all p<0.02), but not between *side-button* and *frontside-touch* (p=1).In addition, there was a significant effect between target sizes of 14 and 22 (p<0.02) and 14 and 30 (p<0.002), but no significant difference between 22 and 30 (p=0.45).

Error rate

Participants made fewer errors with *side-button* and *front-side-touch*. *Double-tap* led to the highest number of errors for both absolute and relative modes (Figure 2 right).

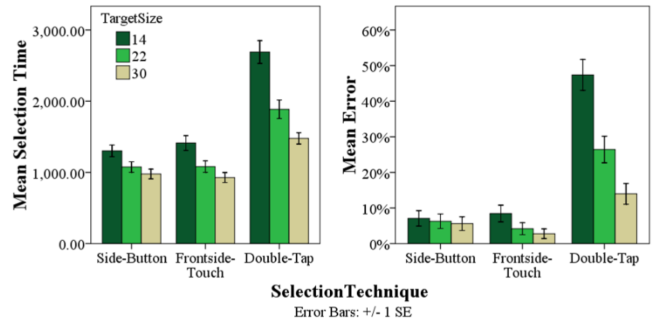

Figure 3. Left: Mean Selection time; Right: Error rate - according to target size and technique.

For *absolute mode*, we found significant effects for *Selection Mechanism* ($F_{2,22}$=67.9) and *Target Size* ($F_{2,22}$=17.72,) as well as a significant interaction effect between *Selection Mechanism*× *Target Size* ($F_{4,44}$=20.09; all p<0.001). Post-hoc comparisons showed that there were significant effects between double tap (43%, s.e. 3%) and the other two mechanisms (both p<0.001), but not between frontside-touch (7%, s.e. 2%) and side-button (7%, s.e. 2%; p=1). There were significant differences among the three target sizes (p<0.05) (Figure 3, right).

For *relative mode*, we found significant effects for both *Selection Mechanism* ($F_{2,22}$=12.63, p<0.001) and *Target Size* ($F_{2,22}$=3.716, p<0.05) as well as a significant interaction effects for *Selection Mechanism*×*Target Size* ($F_{4,44}$=4.45, p<0.005). Post-hoc pair-wise tests showed that there were significant differences between side-button (6%, s.e. 2%) and double-tap (17%, s.e. 2%; p<0.01), frontside-touch (3%, s.e. 1%) and double-tap (p<0.01); no significant differences were found between frontside-touch and side-button (p=1). In terms of target size, post-hoc tests indicated significant differences for target sizes of 14 and 22 (p<0.02), but not for target size of 14 and 30 (p=0.37) and of 22 and 30 (p=1).

User preference

Data collected from the post-experiment questionnaire indicated that participants preferred *side-button* (M=3.8 on a 5-point Likert scale, as compared to *frontside-touch* (M=2.9) and *double-tap* (M=2.3)) with both absolute and relative modes. They also expressed that both side-button (M=3.8) and frontside-touch (M=3.4) were relatively easier to use than double-tap (M=2.1).

Discussion

The results show that *double-tap* in its standard implementation is not a suitable selection mechanism for BoD input. It has a higher selection time and is more error prone, especially with absolute pointing mode, thus supporting H2. This is mainly due to the lack of feedback of the finger's contact location, which leaves space for future exploration and improvements for double-tap for BoD input. *Side-button* and *frontside-touch* have led to similar task completion times and error rates (partially rejects our H1). This seems to suggest that assigning cursor movement and target selection to a different finger may be beneficial.

Participants appear to prefer the side-button over front-side touch. One reason may be because of ergonomics: activation with the thumb was relatively comfortable. Also, unlike frontside-touch, there is no occlusion of the display. In addition, with frontside-touch, although activation was performed by touching anywhere on the display, several participants felt that this did not match their model of how it should work because they linked the area they would touch with selecting a component or target located in that area.

The results also show that target size has a significant impact on performance and number of errors for both modes, especially when dealing with small targets (e.g., 14 pixels). However, relative mode seems to be affected by big targets (e.g., 22 pixels and up).

Overall, these observations on selection mechanism suggest that side-button is the preferred technique for BoD input. Frontside-touch has several limitations (e.g., occlusions, low user rating) and double-tap has a higher selection time and error rate. Therefore, we selected *side-button* as the selection technique for the next two experiments.

EXPERIMENT 2: ABSOLUTE VS. RELATIVE CURSOR MOVEMENT

While the previous experiment focused on selection mechanisms, this study investigatesusers' ability to move the cursor with either absolute or relative control. We were particularly interested in exploring which mode is most effective for one-handed BoD input.

Apparatus

We used the same apparatus as in the previous experiment.

Participants

Twelve participants (11 males) between the ages of 21 and 35 were recruited from a local university to participate in this study. All participants were right-handed, and had at least 1 year of experience with mobile devices.

Task and procedure

We used a target acquisition task, where we placed a target in one of 9 cells in a 3×3 grid (Figure 4). Upon clicking the Start button, a square target was placed in the center of a randomly picked cell. The distance between the Start button and all the targets was fixed (80 pixels or 17.6 mm).The Start button was located at the center cell. However when a target was placed in that cell, the Start button was moved to another randomly-chosen cell with the same distance.

Participants were asked to perform the task using either absolute or relative cursor control. Informed by the first study, selection was carried out with *side-button*.

Figure 4.The 3×3 grid where targets were placed randomly at the center of cells.

A trial started after the Start button was clicked, and ended if a successful selection was made or if the participants failed to do so within 25 seconds. Participants were allowed to press the side button multiple times to make the correct selection. Warm-up trials were given to the participants prior to the experiment. The entire experiment lasted about 45 minutes. At the end of the experiment, the participants filled out a questionnaire.

Experimental design

The experiment employed a 2×3×9 within-subject factorial design. The independent variables were *Cursor Movement Mode* (*Relative* and *Absolute*), *Target Size* (*14*, *22*, *30* pixels), and *Location* (*1-9*). Each trial was repeated 3 times, with the presentation of *Cursor Movement Mode* counterbalanced using a Latin Square design, while *Target Size* and *Location* were selected randomly.

Dependent measures included task completion time and number of attempts. Completion time was measured from the time when the Start button was clicked to when the target was successfully selected. Number of attempts was measured by the number of times a participant pressed the side button to complete the task.

Hypothesis

The purpose of the experiment was to evaluate the performance (task time and number of attempts) of each mode, target size and location. We hypothesized the following:

- **H3**: Selection of targets located in the lower regions of the device will take longer than those located in the middle and top portions.

Results

We analyzed the data using the Repeated-Measures ANOVA and Bonferroni corrections for post-hoc comparisons. Trials with timeouts were removed (less than 1%) from the analysis. For simplicity of result analysis, we grouped the locations into three different *Target Positions*: Top (Locations 1-3), Middle (Locations 4-6) and Bottom (Locations 7-9). This division was also used in [8] for studying one-handed input selection.

Task completion time

Figure 5 summarizes the completion time for both cursor movement modes by target size (Left) and target position

(Right). Overall participants were significantly faster working with *Relative* mode (M=3,043ms; s.e.=155) than with *Absolute* mode (M=3,801ms; s.e.=153; $p<0.001$).

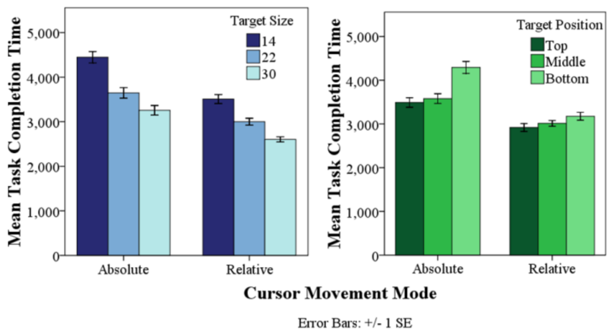

Figure 5. Left: Mean task completion time according to cursor movement mode and target size; Right: Mean task completion time according to target position.

We found significant effects for all three independent variables (*Cursor Movement Mode*: $F_{1,11}$= 32.37;*Target Size*: $F_{2,22}$=45.35; and *Target Position*:$F_{2,22}$=17.72, all $p<0.001$).There was also a significant interaction effect for *Target Size* × *Target Position* ($F_{4,44}$=2.72; $p<0.05$).

In both cursor modes, it took the participants longer to complete the task as the target size decreased. Similarly, the lower the location of the targets, the longer it took the participants to complete the task. Post-hoc tests revealed significant effects among all three target sizes ($p<0.01$). Also, there was a significant effect of targets located in *Bottom* (M=3,751ms; s.e.=163) with other two positions: *Top* (M=3,212ms; s.e.=147) and *Middle* (M=3,304ms; s.e.=139); $p<0.005$). However, there was no significant effect of target located in *Top* and *Middle* (p=0.90).

Number of attempts

Recall that users were allowed multiple selections before completing a trial, resulting in multiple attempts. As Figure 6 shows, in either mode the number of attempts increases when target sizes decrease (left figure). Interestingly, in relative mode, the number of attempts increases if the targets were located in the *Bottom* position.

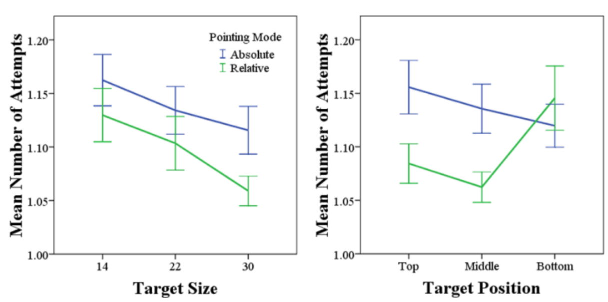

Figure 6. Left: Average number of attempts according to cursor movement mode and target size; Right: Average number of attempts according to target position.

User preference

Participants indicated a stronger preference for *Relative* mode over *Absolute* mode (M=3.92 vs. M=2.92 based on a 5-point Liker scale). They felt that *Relative* mode was relatively easier to use than *Absolute* (M=3.92 vs. M=2.83). In addition, participants noted that *Relative* allowed greater control and precision and also let them reach remote targets with ease. Figure 7 shows how participants rated the regions in terms of 'ease-of-use' (1=easy; 9=difficult). Regions on the bottom were rated as the most difficult to use, followed by the regions on the right side of the screen.

Absolute

4.00	2.75	4.67
3.42	2.33	5.67
6.50	6.83	8.83

Relative

3.67	3.00	4.92
3.00	2.25	5.83
6.75	6.67	8.92

Figure 7. The participants' ratings of ease-of-use according to each region (1=easy; 9=difficult).

Discussion

The analysis confirmed **H3**. Targets located on the bottom row were more difficult to select, particularly in *Absolute* mode. In *Relative* mode, it was not as difficult because participants were able to clutch their index finger on the upper part of the trackpad to move the cursor toward the target. In *Absolute* mode, however, clutching was not available and because of the biomechanical characteristics of the index finger, reaching targets in the bottom row was difficult. One reason for this difference might have been the fixed distance we selected for this experiment, requiring a minimal number of clutches in *Relative* mode (we observed 1.9 clutches on average).

Regardless of the mode, target size had a significant effect on selection time, reinforcing the findings from the first experiment. In addition, Relative mode led to faster selections and fewer attempts based on the target sizes we tested. These sizes were similar to those tested in [1] and it's possible that we would observe cross-over effects between Absolute and Relative modes for larger targets.

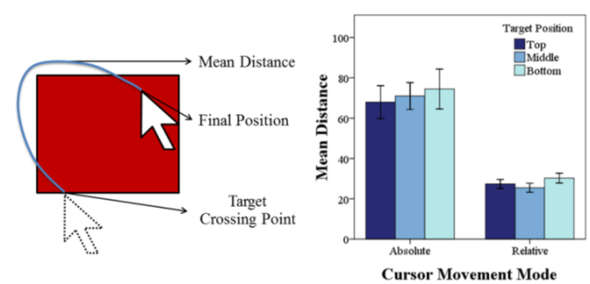

Figure 8. Left: Mean distance calculation; Right: Mean distance in Cursor movement (in pixels) and target position.

We also recorded mean distance traveled by the cursor before completing a selection. This distance was computed by looking at the distance the cursor traveled once it intersected the target to its final position at selection (Figure 8 left). Absolute cursor movement resulted in twice as much movement across all the positions (68 to 74 pixels) compared to relative pointing mode (25 to 30 pixels). This suggests that participants had better control in placing the cursor on top of the target with relative pointing mode. This also suggests that Relative input is relatively easier for precise pointing with BoD input.

EXPERIMENT 3: EXPLORING ACCURACY AND REACHABILITY WITH ABSOLUTE POSITIONING

Overall, our results indicate that relative mode provides better precision than absolute mode for BoD input. Absolute mode suffered from input in the lower regions of the trackpad. This concern with reachability has also been noted in one-handed thumb input [8]. Based on this result, we chose to explore ways to improve precise selection with absolute mode to resolve concerns of reachability. To this end, we designed and evaluated two variations of absolute pointing and compared them against relative pointing.

Two new variations of absolute cursor positioning

Inspired by ThumbSpace [6], we implemented *Physical ThumbSpace*. We mapped a region of the trackpad to the entire screen of the device (Figure 9).

Since all areas of the mapped region of the trackpad would be within reach of the index finger, users would be able to reach targets with ease. As in the previous two experiments, we used a modified version of absolute pointing, which upon landing allowed users to move the cursor to attain greater precision.

Apparatus

The apparatus was similar to that in study 1 and 2.

Participants

Twelve participants (11 males) between the ages of 21 and 35 were recruited from a local university to participate in this study. All participants were right-handed, and had at least 1 year of experience with mobile devices.

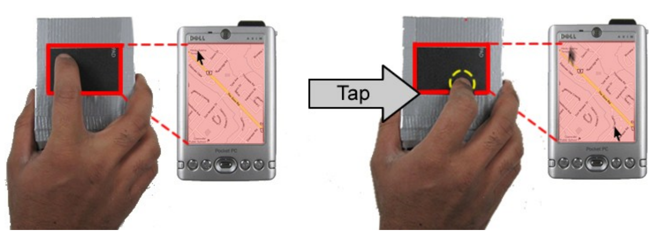

Figure 9. Left: The highlighted trackpad input region maps to the entire screen; Right: Tapping near the lower end of the input region moves the cursor to the bottom of the screen.

Our second enhancement, *SlidingWindow* (Figure 10), was inspired by cursor displacement techniques [14, 17]. *SlidingWindow* divided the display area in two regions of equal sizes, each of which had a 1:1 mapping to the top part of the trackpad. Users could move from one region to the other (by double tapping), and this would allow them to reach targets located in the lower regions with ease.

Tasks and procedure

Participants were asked to perform target selection tasks using Physical Thumbspace, SlidingWindow, Absolute (same as in the previous experiments), and Relative pointing (same as in the previous experiments).

In this experiment, we placed the Start button at the top-middle position on the screen. Targets were placed in two different positions relative to the Start button, *Near* (70 to 90 pxls or15.4 to 19.8 mm) and *Far* (250 to 270 pxls or 55 to 59.4 mm). A trial started after the Start button was selected, and ended if a target was successfully selected or if participants failed to make the selection within 25 seconds.

Figure 10. Left: The highlighted input region is mapped to the upper portion of the screen; Right: Double tapping toggles the mapping.

Experimental design

The experiment used a 4×2 within-subject factorial design. The independent variables were *Technique* (*Physical-Thumbspace*, *SlidingWindow*, *Relative* and *Absolute*) and *Target Position* (*Near* and *Far*). Each trial represented a *Techniques* × *Target Position* combination, and was repeated 12 times by each participant. The order of presentation of *Technique* was counterbalanced using a Latin square design. The size of the target was fixed at 30 pixels (6.6 mm), and the selection mechanism was side-button.

Measures

Dependent measures include task completion time and the number of attempts. The task completion time was recorded as the time from when the user selected the Start button to when they successfully selected the targets. The number of attempts was measured by the number of times a participant presses the side button for selection.

Results

Task completion time

We removed outliers, defined by 3 s.d. from the mean. This resulted in 2% of trials being excluded.

We found a significant effect of *Technique* ($F_{3,33}$=5.92, p<0.005) and *Target Position*($F_{1,11}$=88.71, p<0.0001). However, there were no significant interaction effects of *Technique*×*Target Position* ($F_{3,33}$=0.61, p=0.61).

Post-hoc pair-wise comparisons showed no significant differences between *Relative* (M=2977ms, s.e.=167) and *Absolute* pointing (M=3556ms, s.e.=261), but found a trend of Relative outperforming Absolute pointing (p=0.08). However, in Relative pointing, participants were significantly faster than *Physical Thumbspace* (M=3589ms, s.e.=240;

p<0.05) and *SlidingWindow* (M=3686ms, s.e.=211; p<0.001). There were no significant differences between the Absolute pointing and its two enhancements (p=1).

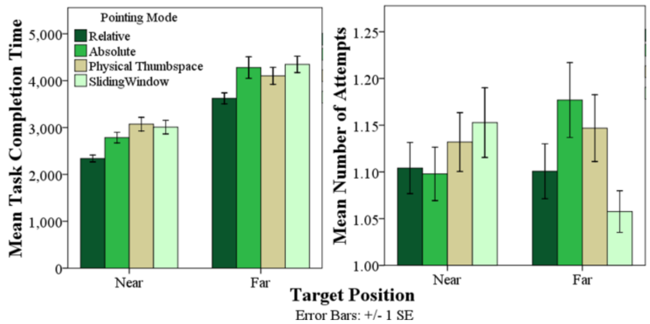

Figure 11. Mean completion time and number of attempts of the 4 techniques based on the target distance.

Post-hoc pair-wise comparisons also revealed significant differences between *Near* (M=2800ms, s.e.=178) and *Far* (M=4105ms, s.e.=227; p< 0.001). Participants were fastest with *Relative* (M=2,339ms) than with *Absolute* (M=2786ms), followed by *PhysicalThumbspace* (M=3010ms) and *SlidingWindow* (M=3072ms), when targets were located in *Near* position. For *Far* targets, participants were also faster with *Relative* (M=3,622ms), followed by *PhysicalThumbspace* (M=4,120ms), then *Absolute* (M=4,279ms) and *SlidingWindow* (M=4,347ms).

Number of Attempts

ANOVA tests yielded no significant effects of *Technique* ($F_{3,33}$=0.64, p=0.6), *Target Position* ($F_{1,11}$=0.002, p=0.96) and *Techniques×Target Position* ($F_{3,33}$=2.83, p=0.05) on the number of attempts. For Near targets, Absolute led to fewest attempts, whereas for Far targets it was SlidingWindow.

User preference

In an exit survey, participants ranked the four techniques using a 1-5 Likert scale. They preferred *Relative* the most (M=4.17), followed by *PhysicalThumbSpace* (M=3.75), then *SlidingWindow* (M=3.08), and finally *Absolute* (M=2.33). A similar pattern was observed in terms of ease of control. Participants were also asked to rate the techniques based on how well they supported selecting *Near* and *Far* targets (5=really well; 1=not well at all). Table 1 summarizes the results. *Absolute* was rated the second best, after *Relative*, for *Near* targets, but the worst for *Far* targets. *ThumbSpace* was rated the best for reaching *Far* targets.

	Absolute	Relative	SlidingWindow	Thumbspace
Near	3.92	4.75	3.50	3.58
Far	1.83	3.33	3.33	3.92

Table 1. Participants' rating for selecting targets in different positions and techniques.

DISCUSSION

We first discuss the results of Experiment 3 and then present a general discussion.

Discussion of experiment 3

We found a trend that Relative pointing outperforms Absolute even though with relative pointing, users were required to travel larger distances in this study, requiring additional clutching. Number of clutching can be alleviated by increasing the control-display ratio of the cursor as we had only provided a 1:1 gain. Other techniques are also possible, such as cursor acceleration to increase CD gain as the cursor speed increases.

Figure 12 (right) illustrates the amount of overshooting taking place with the various pointing styles; a lower percentage indicates that participants had more control with a technique. Relative allowed better control. As expected SlidingWindow has less overshooting (thus more control) than Absolute pointing.

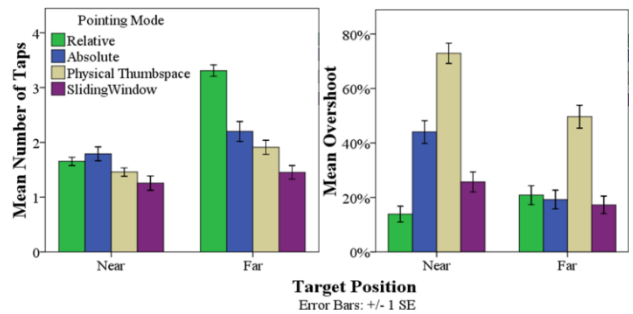

Figure 12. Left: Mean number of taps according to target distance;Right: Average vertical distance ('overshooting') according to target distance.

Physical ThumbSpace used a distorted the mapping between input and screen (a non-1:1 mapping) so that remote objects could be reached with ease. Although this technique did not do as well as we had thought, it allowed us to observe that users were capable of developing motor memory and perform well using a distorted mapping (we only provided minimal training in our study). Physical ThumbSpace allowed participants to rapidly move to a distance location. The flip side of this technique was that it could lead to imprecise movements when reaching a closer region (see Figure 12, right).We could resolve this by introducing nonlinear mappings. For example, we could use a 1:1 mapping for nearby regions; and, as we move further away the mapping can gradually scale up. The imprecise movements are also due to the lack of visual feedback—an inherent problem with Absolute pointing. For techniques that distort the 1:1 mapping, it may be necessary to have additional feedback to reduce trial-and-error actions.

SlidingWindow tried to resolve the reachability issue with a different approach. The need to move the window up and down introduced additional cognitive load and taps. We found that participants sometimes accidentally double tapped for some targets located in the Near locations (17%); for targets in Far locations, some participants double tapped more than once (13%). This led to longer completion times. This explicit mode switch made it less efficient for Far targets. However, for Near targets, *SlidingWindow* performed relatively well (slightly better than Physical ThumbSpace) and was less affected by overshooting problems.

General discussion

Summary of findings

The findings from the first experiment show that, among the three tested selection techniques, *Side-button* was most preferred by users and also had the best performance. This is because we designed *Side-button* to take advantage of the thumb position so that selection of targets requires minimal additional movement.

Results from our second experiment indicate that Relative input can be more efficient and accurate for cursor positioning and target selection (see also *Limitations* below). Our results also show that Absolute positioning can be as efficient in tasks that require less accuracy (e.g., positioning inside large targets) and that take place in a reachable region (e.g., top and middle parts of the back). In addition, our results suggest that both Absolute and Relative positioning required almost the same number of attempts (about 1) to select a target. Our third study was mainly exploratory in nature and aimed at finding ways to improve Absolute positioning, especially for targets located in the lower regions. Of the two new techniques, *SlidingWindow* showed better performance, especially in terms of the number of attempts to select Far targets. The other technique, *Physical Thumbspace*, did not do as well due to the additional cognitive effort to use the distorted mapping between the input space and the display. This is apparent in the number of attempts and amount of "overshooting", problems that are exacerbated because of the lack of feedback of where the finger lands in Absolute mode. Overall, we found a trend that Relative input improves performance in comparison to variations of Absolute pointing.

Limitations

Our last two studies show that Relative cursor positioning allows for better overall performance. However, our results are heavily influenced by the target sizes we chose. As target sizes increase, Absolute positioning will have comparable performance to Relative pointing, if the control-display gain is 1:1. As is, our results show that for targets of 30 pixels (6.6mm) or less, Relative positioning is more precise.

Our last two studies also suggest that Relative positioning seem to be a better choice for BoD input. However, in some situations, e.g. with a multi-touch input device, Absolute input can be more natural and intuitive. Similarly, results from the third study suggest that distorting the mapping between the input space and the display does not improve the Absolute input. This is true of the two techniques we tested. However, we have not studied a variety of mapping factors, which could enhance the performance of Relative and/or Absolute positioning.

Lessons for designers

Based on the findings of our experiments, we make the following guidelines when designing for one-handed BoD interaction:

- Regardless of the cursor positioning method, separating selection and cursor control so that one finger is assigned to each function leads to increased performance (Experiment 1);
- With Absolute positioning, targets should be at least 30 pixels in size (6.6 mm) and preferably placed in the top index-reachable part of the display (Experiment 2);
- When using Relative positioning, targets can be as small as 22 pixels (4.8 mm) (Experiment 2);
- For tasks requiring precision, Relative positioning should be provided (all three experiments);
- In Absolute mode, when distorting the 1:1 mapping between the input space and display, consider providing additional visual feedback (Experiment 3).
- Whereas most work on BoD input has considered cursor positioning in absolute mode, our results suggest that designers could consider relative input mode as being effective in a number of different conditions.

Advantages of Relative input over Absolute input

We do not strongly advocate for either type of pointing style as both have their merits. Our studies show a trend toward integrating Relative input for BoD pointing, a finding that has not been previously reported. We have further shown that Relative input allows for higher precision. Therefore relative input is a compelling choice for tasks requiring small targets. Absolute input alleviates the need to clutch and can works well with targets that are at least 6.6 mm in size. Unfortunately, Absolute input is affected by concerns of reachability, as the index finger is limited in its range of movement behind the device.

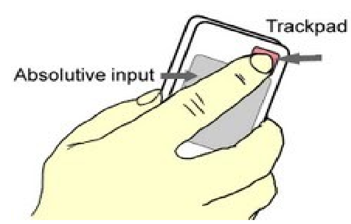

Figure 13. An illustration of the design of relative + absolute input on the back of the device

We did not explore varying the control-display gain for Relative input. This style of interaction can allow for more variety of control and could also benefit from improvements to selection derived for mouse-cursor input on the desktop. Such augmentations can potentially make Relative input a choice technique for BoD interaction. One option might be to place a relative input controller on the back, along with a touch-pad for absolute interaction, such as text-entry. Other combined alternatives could also be designed such that both modes co-exist (Figure 13).

CONCLUSIONS

In this paper, we present the results of three experiments which explore several aspects of one-handed back-of-device interaction. In the first experiment, we examine which selection mechanisms are suitable, with the results suggesting that it may be best to assign selection to one finger and cursor control to another. In the second experiment, we compare the performance of Absolute and Relative cursor positioning modes, and found that Relative is better for tasks requiring high precision, while Absolute may be suitable for tasks for targets located in the top 2/3 of the display. Finally, we used our third study to explore two approaches to solving the reachability issue with Absolute cursor control so that targets located in the lower end of the screen could accessed with ease. We found that distorting the mapping between the input space and the display (e.g., having a smaller input space to control a cursor in a larger display) can be difficult to work with. In addition, enhancements to Absolute control could be based on providing additional visual feedback to users.

REFERENCES

1. Baudisch, P. and Chu, G. (2009). Back-of-device Interaction allows creating very small touch devices. CHI'09. pp. 1923 - 1932.
2. Forlines, C., Wigdor, D., Shen, C., and Balakrishnan, R. (2007). Direct-touch vs. mouse input for tabletop displays. CHI'07. pp. 647-656.
3. Grossman, T. and Balakrishnan, R. (2005). The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. CHI'05. pp. 281-290.
4. Irani, P., Gutwin, C., and Yang, X. D. (2006). Improving selection of off-screen targets with hopping. CHI'06. pp. 299-308
5. Kabbash, P. and Buxton, W. (1995). The "Prince" technique: Fitts' law and selection using area cursors. CHI'95. pp. 273-279.
6. Karlson, A. and Bederson, B. (2007). Thumbspace: Generalized one-handed input for touchscreen-based mobile devices. Interact'07. pp. 201-210.
7. Karlson, A. and Bederson, B. (2008). One-handed touchscreen input for legacy applications. CHI'08. pp. 1399-1408.
8. Karlson, A., Bederson, B., and Contreras-Vidal, J. (2006). Understanding Single-Handed Mobile Device Interaction. TechReport HCIL.
9. Karlson, A., Bederson, B., and SanGiovanni, J. (2005). AppLens and launchTile: two designs for one-handed thumb use on small devices. CHI'05. pp. 201-210.
10. Li, K., Baudisch, P., and Hinckley, K. (2008). BlindSight: eyes-free access to mobile phones CHI'08. pp. 1389-1398.
11. MacKenzie, S. (1992). Fitts' Law as a Research and Designtool in Human-computer Interaction. Human-Computer Interaction. v7 (1), pp. 91-139.
12. McCallum, D. C. and Irani, P. (2009). ARC-Pad: absolute+relative cursor positioning for large displays with a mobile touchscreen. UIST'09. pp. 153-156.
13. Parhi, P., Karlson, A. K., and Bederson, B. B. (2006). Target size study for one-handed thumb use on small touch-screen devices. MobileHci'06. pp. 203-210.
14. Potter, R., Weldon, L., and Shneiderman, B. (1988). Improving the accuracy of touch screens: an experimental evaluation of three strategies. CHI'88. pp. 27-32.
15. Schwesig, C., I. Poupyrev, and E. Mori. Gummi: a bendable computer. Proceedings of CHI'2004. 2004: ACM: pp. 263-270
16. Scott, J., Izadi, S., Rezai, L. S., Ruszkowski, D., Bi, X., and Balakrishnan, R. (2010). RearType: Text Entry Using Keys on the Back of a Device. MobileHCI'10. pp. 171-180.
17. Sears, A. and Shneiderman, B. (1991). High precision touchscreens: Design strategies and comparisons with a mouse. IJMMS. v34 (4), pp. 593-613.
18. Sugimoto, M. and Hiroki, K. (2006). HybridTouch: an intuitive manipulation technique for pdas using their front and rear surfaces. Extended Abstracts of MobileHCI'06. pp. 137-140.
19. Vogel, D. and Balakrishnan, R. (2005). Distant Freehand Pointing and Clicking on Very Large, High Resolution Displays. UIST'05. pp. 33-42.
20. Vogel, D. and Baudisch, P. (2007). Shift: a technique for operating pen-based interfaces using touch. CHI'07. pp. 657-666.
21. Wigdor, D., Forlines, C., Baudisch, P., Barnwell, J., and Shen, C. (2007). LucidTouch: A See-through Mobile Device. UIST'07. pp. 269-278.
22. Wigdor, D., Leigh, D., Forlines, C., Shipman, S., Barnwell, J., Balakrishnan, R., and Shen, C. (2006). Under the table interaction. UIST'06. pp. 259-268.
23. Wobbrock, J. O., Myers, B. A., and Aung, H. H. (2008). The performance of hand postures in front- and back-of-device interaction for mobile computing. IJHCS. v66 (12), pp. 857-875.
24. Worden, A., Walker, N., Bharat, K., and Hudson, S. (1997). Making computers easier for older adults to use: area cursors and sticky icons. CHI'97. pp. 266-271.
25. Yang, X. D., Irani, P., Boulanger, P., and Bischof, W. F. (2009). One-handed behind-the-display cursor input on mobile devices. CHI EA '09. pp. 4501-4506.
26. Yang, X. D., Mak, E., Irani, P., and Bischof, W. F. (2009). Dual-Surface Input: Augmenting One-Handed Interaction with Coordinated Front and Behind-the-Screen Input. MobileHCI'09. pp. 1-10
27. Yang, X. D., Mak, E., McCallum, D. C., Irani, P., Cao, X., and Izadi, S. (2010). LensMouse: Augmenting the Mouse with an Interactive Touch Display. CHI '10. pp. 2431-2440.