# Dual-Surface Input: Augmenting One-Handed Interaction with Coordinated Front and Behind-the-Screen Input

Xing-Dong Yang[1], Edward Mak[2], Pourang Irani[2], Walter F. Bischof[1]
[1]University of Alberta, Canada; [2]University of Manitoba, Canada
xingdong@cs.ualberta.ca; ed.mak.is@gmail.com; irani@cs.umanitoba.ca;
wfb@cs.ualberta.ca

## ABSTRACT

Interaction patterns with handheld mobile devices are constantly evolving. Researchers observed that users prefer to interact with mobile device using one hand. However, only few interaction techniques support this mode of operation. We show that one-handed operations can be enhanced with coordinated interaction using for input the front and back of a mobile device, which we term as *Dual-Surface* interaction. We present some of the design rationale for introducing coordinated Dual-Surface interactions. We demonstrate that several tasks, including target selection, benefit from Dual-Surface input which allows users to rapidly select small targets in locations that are less accessible when interacting using the thumb with one-handed input. Furthermore, we demonstrate the benefits of virtual enhancements that are possible with behind-the-display relative input to perform complex tasks, such as steering. Our results show that Dual-Surface interactions offer numerous benefits that are not available with input on the front or the back alone.

## Categories and Subject Descriptors

H5.2 [**Information interfaces and presentation**]: User Interfaces, Input devices, Interaction styles.

## General Terms

Human Factors, interaction techniques, novel interactions.

## Keywords

Front input, behind-the-screen input, Dual-Surface interaction.

## 1. INTRODUCTION

Handheld devices equipped with touch-sensitive displays are ubiquitous and considered by some as a natural extension to our cognitive resources. Studies show that one-handed use is the preferred method for operating a handheld device [12]. With interactive tasks performed by one hand, users can free their other hand for tasks that are commonly carried out in mobile contexts, such as carrying shopping bags or holding a bus handle. In this mode, the user grips onto the device and interacts using the thumb or other auxiliary fingers. This works well for small devices (e.g. cell-phones) equipped with physical interfaces, such as a physical keyboard. This is because most of the interactions happen in places that may be easily accessed by the thumb. For several reasons, interacting with one hand is still very difficult on many devices. Usually, the distance covered by the thumb is not sufficient to manipulate objects at extreme and opposite corners of the device (see Figure 2). Furthermore, the "foot-print" of a thumb is significantly larger than that of other fingers, resulting in larger occlusions than with a stylus used with both hands [24]. Studies have also reported inaccuracies in selecting targets with a finger on a touch-display, specifically when the targets are small or in proximity to one another or small [23]. Finally, trajectory-based interactions such as scrolling a long document or highlighting a line of a sentence are difficult to operate using the thumb with one hand.

To resolve some of the problems with direct-touch input on handheld devices researchers have proposed a number of techniques. Behind-the-surface interaction [4, 23, 24], resolves some of the complexities associated with finger occlusion. With this type of interaction, the user can interact with objects on the screen that would normally be occluded. Techniques such as Shift [23] allow the use of large finger footprints for selecting small targets on a PDA. It does this at the cost of an offset window that magnifies the target. However, recent studies have shown that Shift can also suffer from reach problems with the thumb [13]. With a few exceptions [13], most of the previously proposed techniques require both hands for operating the device or additional timeouts for invoking visual filters for selecting enlarged targets. Furthermore most of the previously developed techniques employ absolute direct input for object selection, thereby not taking advantage of what might be possible with relative input.

Indirect relative behind-the-screen input was introduced earlier with systems such as HyridTouch [22]. Relative input can assist in solving some of the intricate problems associated with one-handed interactions on a handheld device. A pixel-size cursor tip provides users with an easy and precise mechanism for pointing and selection and avoids occlusion. In addition, cursor-based enhancements [1, 2, 5, 9, 15, 20], which are often seen on desktop PCs, can be applied on mobile devices to facilitate easier interactions. However, relative positioning introduces clutching, which can make interactions inefficient. Clutching can be alleviated by means of direct input on the front of the device. We hypothesize that one-handed interaction on mobile devices can be significantly improved by taking advantage of the best of both worlds: direct thumb input on the front to be in proximity of targets, with relative cursor input for finer control. We refer to this mode of operation as *Dual-Surface* operations whereby the user can coordinate input from the front and back to perform a number of tasks.

*MobileHCI'09*, September 15 - 18, 2009, Bonn, Germany.
ACM 978-1-60558-281-8.

We demonstrate the value of Dual-Surface interaction with a prototype similar to that proposed in [22]. It embeds a touchpad onto the rear surface of a PDA and allows users to interact with the PDA with the index finger that is free from gripping the device with one hand. The device supports *Front* (via thumb), *Back* (via cursor), and *Dual-Surface* (via both thumb and cursor) interaction. The results of the first experiment reveal a performance benefit for targets with *Dual-Surface* input. A second experiment shows that Dual-Surface input can take advantage of virtual enhancements that are possible with relative input to perform more complex tasks such as tunneling.

# 2. RELATED WORK

Several developments are related to our work. Most of them are aimed at addressing the issues concerning touch and single-handed input.

## 2.1 Finger-based Touch Input

Finger based interactions suffer from occlusion. To address this concern, researchers have proposed cursor replacement techniques. The idea is to statically re-locate the cursor to a non-occluded position. Offset cursor [20] displays the cursor 0.5" above the finger. Fluid DTMouse [7] displays the cursor in the middle of two finger touch points so that it is not occluded by either finger. The major concern of such designs is that selection depends highly on the location of the targets. On mobile devices, edges and corners are problematic for static relocation techniques.

Shift [24] addresses some of the problems of the above techniques as it dynamically places a copy of the area occluded under the finger in a "shifted" callout at a non-occluded location. The callout contains a crosshair representing the contact location of the user's finger tip. By sliding the finger contact on the display, the user can fine tune the pointer position and select the target of interest using a take-off gesture. Shift's callout was designed to be triggered only when necessary. When selecting large targets, a callout is not triggered, and selection takes place by simply tapping the screen. Similar to the static cursor relocation techniques, performance with Shift drops when targets are on an edge or at a corner of the mobile device. One drawback includes the use of a callout that can result in a costly occlusion on small displays.

Olwal et al. [18] introduced Rubbing and Tapping to facilitate precise and rapid selection on touch screens. The rubbing technique activates a zoom-in action using a finger-tip gesture. The tapping technique enlarges the target by having the user touch near the target with one hand and tapping a distance away on the screen with the other hand. Studies reveal that rubbing and tapping techniques can provide precise selection of target size ranging from 0.3mm to 4.8mm within 2 seconds [18]. Besides Tapping, other two-handed techniques, such as those presented in [3] have also been proven helpful for selecting small targets using bare fingers. However, bimanual input is not a preferred operation mode for interacting with mobile devices [13].

To circumvent occlusion problems altogether, researchers have proposed interactions on the back of the display surface. LucidTouch [25] uses a back-mounted video camera to capture the user's finger motion. As a result, the system allows users to interact with the device using all their fingers. A circular cursor is assigned to each finger tip, allowing for precise interaction at the pixel level. The system was demonstrated with a set of tasks including object selection and dragging. The authors also explored the coordinated use of front of the display and its back to perform basic navigation tasks. However, they did not provide any empirical support or report on any advantages of using both surfaces in a coordinated manner. Inspired by LucidTouch, NanoTouch [4] facilitated the creation of very small devices by offloading input to the back of the display. Back input was shown to make target selection in unreachable areas such as corners and edges possible and less error prone than enhanced front techniques such as Shift. By having the users interact on the back of the device, the device can be reduced to a size as small as 0.3", without any impact on performance of selection tasks [4]. However, the authors of NanoTouch did not describe the use of coordinated front and back input. Additionally, back input was primarily absolute and not relative.

Although the discussed techniques provide solutions to occlusion from finger touch, the majority of the solutions were not devised or evaluated with one-handed input. Since one-handed input is popular and the preferred method for using a mobile device for many tasks [13], little is known about how such techniques work with one handed input.

## 2.2 One-handed Mobile Device Interactions

The study of one-handed interactions can be traced back to the 1960's [6]. Recent studies stem from formal and informal observations about usage patterns with mobile devices. Karlson and Berderson, conducted several in-situ observations which led to the conclusion that 74% of mobile users use one hand when interacting with their cellular devices [13]. The same observations and web surveys suggested that PDA users were inclined to use two hands, but expressed significant interest in using one hand if possible, suggesting the need for better tools to support one-handed interactions.

Applens and LaunchTile [12] were two of the earliest systems to support one-handed thumb use on PDAs and cell-phones. With AppLens users were provided with a simple gesture set, mimicking the up-down-left-right keys, to navigate with a grid of data values. LaunchTile allows access into the tabular data view by allowing users to press on soft buttons associated with an area of the grid. In a user study, users were found to perform correct gestures 87% of the time, suggesting that simple thumb gestures are memorable. However, they also found that users were reluctant to use gestures and preferred tapping. Their results also showed that error rates were influenced by the direction of the gestures and the number of gestures available, suggesting a limit on the number of different gestures one should design.

Further investigations by Karlson et al [13] on the biomechanical limitations of one-handed thumb input have revealed that users do not equally easy interact with all areas of a device. Instead, user grip, hand size, device ergonomics, can influence dexterity and reach of the thumb. For example, right handed users have more limited thumb movement in the NorthWest-SouthEast direction than in the other directions. Additionally, regions of the device away from the right edge are more difficult to reach. These findings supported the development of Thumbspace [14]. ThumbSpace is based on users' natural inclination to touch the interface with their fingers when a stylus is not available. To facilitate thumb reach, users customize the workspace and shrink the entire workspace into a box. In an extensive study, users performed better at selecting targets further away using Thumbspace than other techniques. Used in conjunction with Shift [15], a technique for high-precision target selection with the finger, Thumbspace resulted in higher accuracy than using either technique alone.

Wobbrock et al. [26] conducted a series of experiments to study gestures of one-handed interaction with mobile devices. In their study, a USB touchpad was used to simulate a PDA. Finger position was mapped in absolute mode and was displayed via a cursor on a computer monitor. Their results suggest that, for a 1-D target selection task, input using the index finger on the back of a device had similar performance to input using the thumb on the front of the device in terms of task completion time. In their study, the target was always reachable by the fingers which may not reflect real cases when using one-hand. Furthermore, the study did not consider occlusion problems, as selection using the thumb was on a touchpad so occlusion did not occur regardless of target size.

Escape [27] facilitates one-handed selection with selection using gestures. Each target on the screen is assigned a unique directional gesture for selection. A selection happens when the user presses an area near the target, and performs a motion in the direction indicated by the target. Escape improves the selection of small targets by not requiring tapping on a target. An experimental study showed that Escape could perform 30% faster than Shift on targets between 6 to 13 pixels wide while still maintaining a similar error rate. However, Escape's gesture-based selection limits its usage in applications where a large number of onscreen targets are selectable. Furthermore, Escape's selection action may confound with other gesture-based interactions such as panning a map.

Overall, the results of these studies suggest that there is sufficient evidence of one-handed use of mobile devices, but we do not have a broad range of techniques to support this form of interaction. Furthermore, the design of one-handed interactions needs to be concerned with the size of the object, the place and position of control items, and the range of allowable gestures with the thumb. We utilized the recommendation offered in previous work [12,13,19] to guide the design of the Dual-Surface interactions proposed in this paper.

## 2.3 Direct vs. Indirect Input

In this paper, direct input is achieved on the display of the device (i.e. touch screen), and indirect input through an intermediate device (i.e. touchpad or jog-dial). Studies have shown that, when using a stylus, direct and indirect inputs were essentially equivalent for target selection tasks [9]. Direct input outperformed indirect input only when selecting a target by crossing, a selection technique not commonly used.

Although direct finger input is widely used in interactions with touch screens, studies show that people prefer using a mouse in some contexts [8, 17, 22]. This may be due to some of the problems, such as occlusion associated with direct input. Sears and Shneiderman [22] compared the performance of mouse input to finger input on a target selection task. Their results showed that finger input outperformed mouse input on targets of size $\geq$ 3.2 mm. Their results also showed that using a mouse does not necessarily lead to lower error rates. But Meyer et al. [17] demonstrated, with a series of goal-directed tasks, that users performed better with indirect input than with direct input. They compared user performance between finger input, stylus, mouse, trackball, and mousepen and found that mouse resulted in the best performance while finger input resulted in the worst performance. Forlines et al. [8] stated that the mouse is the more appropriate for tasks like pointing and dragging. They compared mouse input to finger input on a selection and docking task and found that selection time to be higher using the mouse than with the finger. They also found that using the mouse lead to faster docking time, which makes the overall trial time being almost identical for both devices. Overall, absolute direct and relative indirect input offer numerous benefits. We propose taking advantage of both input types by means of Dual-Surface interactions.

# 3. DUAL-SURFACE INTERACTION

*Dual-surface interaction* involves sequentially coordinating input from the front (typically with the thumb) with an input channel attached to the back of the device. The design of Dual-Surface input is motivated by some of the common problems with one-handed interactions, such as occlusion and out-of-thumb-reach. Unlike prior techniques for back-of-the-screen input, our design considers the use of relative input on the back. This allows users to take advantage of the best of both worlds: direct absolute input and indirect relative input.

Direct absolute input facilitates access to large targets very quickly and allows users to interact with a significant portion of the device. The drawback with absolute input with one-handed interaction is the lack of reachability based on the form factor of the device, occlusion of smaller targets, and difficulties selecting items on the edges. We hypothesize that augmentation of a device with relative input in the back (as in [23]), will reduce or eliminate problems with occlusion. Relative input is generally unaffected by the location of targets and can itself be augmented with virtual enhancements [1, 2, 5, 10, 16, 21], thereby facilitating a number of tasks. However a drawback of this input is its dependence on clutching.

We considered the pros and cons of direct input and relative input in the design of our Dual-Surface interaction. We propose that, with a sequentially coordinated interaction, users can leverage off the benefits from both input types to perform a variety of tasks. Our proposed coordination is very similar to, and inspired by, the kinematic chain model proposed for bi-manual interaction [11]. While users are not constrained to Dual-Surface interaction alone, we propose the following coordination to take advantage of this type of input. The front input leads the back input, sets the frame of reference for the back cursor to operate in, and is used for performing coarse movements. The back relative input follows the front, operates within the frame-of-reference established by the front, and is able to perform finer movements.

# 4. APPARATUS

We developed a prototype similar to that in [23]. We used a Dell Axim X30 PDA with a 624 MHz processor and 64MB memory. We attached an Ergonomic USB touchpad on the rear side of the PDA. The touchpad was oriented along the long side of the PDA (see Figure 1). We placed the touchpad as close as possible to the top of the PDA to make it confortable for users to manipulate the cursor with the index finger when holding the device with one hand. All gestures moved the cursor using a relative mapping. The software was implemented in C#.NET.

**Figure 1 In the experiment, we asked participants to sit in a chair, and perform the target selection task using one hand. The touchpad on the back is operated using the index finger.**

## 5. EXPERIMENT 1 – SELECTION

The goal of this experiment was to compare the performance of target selection with *Dual-Surface* (thumb+cursor) input against the *Front* (thumb only) and *Back* (cursor only). Participants were required to complete a series of target selection tasks using the three techniques with their dominant hand. In this interaction, the index finger and thumb were not used for gripping onto the device.

We implemented the Shift technique [24] for the *Front* in order to address the fat-finger problem introduced by interacting with a touch screen using bare finger. When selecting a target via the *Front*, participants tapped near the target to invoke a callout. Since our targets were always either placed on the top right or bottom left, the callout was placed in a position that it was not occluded by any part of the hand. To perform a selection, participants moved the Shift crosshair cursor onto a target and lifted the finger. The *Back* technique allowed the user to control an arrow cursor with their index finger using the touchpad mounted on the back of the PDA. When moving the cursor with the touchpad, the cursor was always read as a mouse-hover until the user quickly tapped the touchpad. Tapping of the touchpad registered a full mouse click. Initially *Dual-Surface* technique forced the user to tap the screen in the general location of the target with thumb to give an absolute cursor position near the target. After front input was performed, we disabled it for the purposes of experiment (to ensure that no more front inputs were performed) and the back touchpad was enabled, where the input was exactly the same as the *Back* technique. All feedback was provided visually on the PDA. A target was highlighted if the cursor was inside it.

In real world applications, selections often take place at the corners of the screen (see Figure 2). This makes target acquisition with one hand difficult. In order to measure the performance of the three techniques in situations close to real-world applications the targets were placed at varying distances away from the corner of the device. In our study, the targets were located at the top-right and bottom-left corners since, with right-handed users (all our participants were right-handed) movement of the thumb is difficult in the top-left to bottom-left direction [13]. We used an offset distance to place the target at varying spots away from the corner. The smaller the offset, the closer the target was to the corner, and vice versa (see Figure 3).

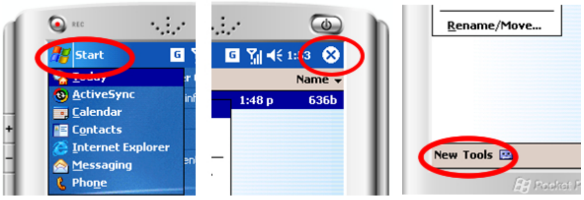


**Figure 2 Mobile applications are mostly replicas of their desktop versions in which targetable items can occur in unreachable or difficult to reach areas of the device when using one hand.**

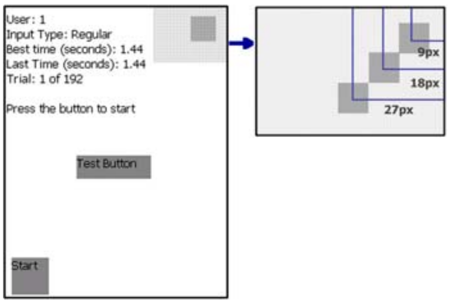
In order to remove bias against the Shift technique in selecting targets being close to the corners, a small pilot study was conducted to measure a reasonable offset. The results showed that 9 pixels (3.6mm) away from the edges of the corner was the closest reasonable offset that would allow the user to select the target with the Shift technique. Furthermore, we found that a square target of size 5 pixels (2mm) was the smallest reasonable size to acquire targets using any of the three techniques.

### 5.1 Participants

Eight participants (6 males and 2 females) between the ages of 20 and 35 recruited from a local university participated in this study. We screened participants so that they were all right-handed, and had previous experience with graphical interfaces.

### 5.2 Procedure

To start a trial, participants clicked the "Start" button (see Figure 3) with either their thumb (when interacting via the *Front* and *Dual-Surface*) or the cursor (when interacting via the *Back*). Then, participants had to use the techniques described above to acquire the target in view. A trial finished either when participants successfully selected the target or when an error occurred. A trial was counted as an error if participants missed a target or failed to make a selection in a timeout of 25 seconds. Participants were instructed to complete the task as fast and as accurate as possible. To promote better performance of all techniques, we showed the time of the last attempt and the overall best time of the current input technique. Users were encouraged to beat their best time.



User: 1
Input Type: Regular
Best time (seconds): 1.44
Last Time (seconds): 1.44
Trial: 1 of 192

Press the button to start

**Figure 3 "Start" button is at the opposite corner of the target.**

A warm-up session was given to participants at the start of each new input technique. Participants were instructed on how to control the cursor in a test trial. Once participants felt comfortable with the technique, they were given two practice trials with the input technique before starting the real experiment. The entire experiment lasted 30 minutes. Participants were encouraged to take breaks during the experiment. Participants filled out a post-experiment questionnaire upon completing the experiment.

## 5.3 Experimental Design

The experiment employed a 3 × 2 × 3 × 3 within-subject factorial design. The independent variables were input *Technique* (*Back*, *Front*, and *Dual-Surface*), *Location* (*Top-Right* and *Bottom-Left*), *Offset* (9px, 18px, and 27px away from corner), and target *Size* (5px, 10px, and 15px). Each trial of the experiment represented a *Technique×Location×Offset×Size* combination, and was repeated 3 times by each participant. The order of presentation of the trials was randomly chosen. Input techniques were counter balanced across participants.

The experimental design can be summarized as:

3 *Techniques*: (*Front*, *Back*, and *Dual-Surface*) ×
3 *Offsets*: (9px, 18px, and 27px) ×
3 *Sizes*: (5px, 10px, and 15px) ×
2 *Locations*: (*Top-Right* and *Bottom-Left*) ×
3 R*epetitions* ×
8 *Participants*
= 1296 data points in total

## 5.4 Results

For all our analyses, we used the univariate ANOVAs and Tamhane post-hoc pair-wise tests (unequal variances) with subjects as random factors.

### *5.4.1 Completion Time*

A total of 51 trials out of 1296 incurred a timeout (3.9%) and the average trial completion time over all trials without timeouts was 4378.7ms (s.e. = 91.3ms). We excluded errors and timeouts from our analysis. We found no significant effect for *Location* and therefore we collapsed our data across this variable, and the following analyses were performed only on the other three variables. The results are shown in Figure 4 (left).

There is a significant effect of *Technique* ($F_{2,14}$ = 16.268, p<0.001), of *Size* ($F_{2,14}$ = 84.578, p<0.001) and of *Offset* ($F_{2,14}$ = 15.94, p<0.001) on completion time. Figure 4 (left) shows average completion time for each technique, by target *Size* and O*ffset*. We found interaction effects for *Technique×Size* ($F_{4,28}$ = 29.395, p<0.001), for *Technique×Offset* ($F_{4,28}$ = 18.630, p<0.001) and for *Size×Offset* ($F_{4,28}$ = 12.817, p<0.001).

Post-hoc pair-wise comparisons (unequal variance assumed) show significant differences across each of the three pairs for all three techniques (p<0.001). The performance with Dual-Surface (3604ms, s.e. 168) was significantly faster than either Shift (7107ms, s.e. 169) or the Back (5106ms, s.e. 168).

Similarly, post-hoc pair-wise comparisons revealed significant differences across all three pairs of target size (p<0.001). Performance was fastest with targets of 15 pixels or 4 mm (3096ms), then with 10 pixel or 4 mm (4680ms) and slowest with 5 pixel or 2 mm (8041ms).

Post-hoc comparisons also show significant differences for the following pairs of offsets: 9 vs. 18 pixels (p< 0.001), and 9 vs. 27 pixels (p<0.001), but not for 18 vs. 27 pixels, with users taking approximately 1.5 times longer with the 9 pixel offset than with the 18 or 27 pixel offsets.

### *5.4.2 Failures*

In addition to recording the completion time, we also recorded the number of failures users had before accurately selecting the target. A failure with Shift consists of releasing the thumb and pressing again. With the Back and the Dual-Surface techniques, a failure consists of clicking outside the target. In general the number of failures captures the number of misses that have occurred prior to properly selecting the target. The results are shown in Figure 4 (right).

There was a significant effect of *Technique* ($F_{2,14}$ = 28.575, p<0.001), of S*ize* ($F_{2,14}$ = 34.044, p<0.001) and of *Offset* ($F_{2,14}$ = 13.038, p<0.001) on failures. We found no significant interaction between these two factors ($F_{6,5}$ =2.2, p=0.203). We found interaction effects for *Technique×Size* ($F_{4,28}$ = 27.501, p<0.001), for *Technique×Offset* ($F_{4,28}$ = 12.97, p<0.001) and for *Size×Offset* ($F_{4,28}$ = 12.504, p<0.001).

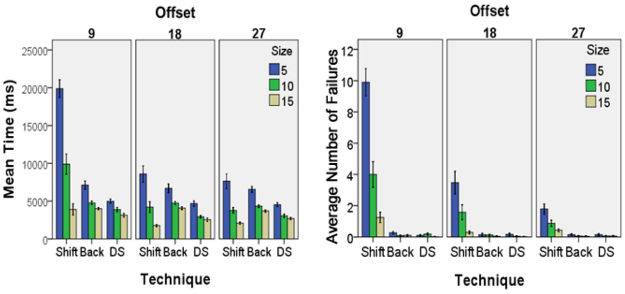


**Figure 4 (left) Average completion time. (right) Average number of failures, for each technique, offset and target size (bars represent +/-1 standard error).**

Post-hoc pair-wise comparisons (unequal variance assumed) show significant differences across the pairs Dual-Surface and Shift, and Back and Shift (p<0.001). The number of failures with Dual-Surface (0.09) and Back (0.113) were significantly smaller than the number of failures with Shift (2.62). There was no significant difference in number of failures between Dual-Surface and Back.

Post-hoc pair-wise comparisons revealed significant differences across all three pairs of target size (p<0.001). There were significantly fewer failures with 15 pixel targets (0.252) than with 10 pixel (0.78) or 5 pixel (1.794) targets.

Finally, post-hoc comparisons also show significant differences for the following pairs of offsets: 9 vs. 18 pixels (p< 0.001), and 9 vs. 27 pixels (p<0.001), but not 18 vs. 27 pixels. The average number of failures with 27 pixel offset was 0.403, with 18 pixel offset was 0.655, and with 9 pixel offset was 1.769.

### *5.4.3 Failure rate*

Of the total 51 timeouts that occurred in the experiment, 50 occurred when using Shift, one with Back and none with Dual-Surface. Of the 50 timeouts with Shift, 88% resulted from selected targets that were offset by 9 pixels from the corner of the display.

### *5.4.4 Subjective Preference*

Of the eight participants, all showed a high preference for Dual-Surface followed by Shift. Participants reported frustration with Dual-Surface as they were required to readjust their grip to perform the task properly. Frustration with Shift resulted from small target sizes but particularly when the targets were in the corner. Participants also commented on the difficulty in selecting targets with the Back alone as users were required to clutch frequently to acquire the target.

## 5.5 Discussion

As expected, target size had a significant effect on task completion time. Shift outperformed Back and Dual-Surface on the larger targets when these were placed away from the edges and closer to the center of the screen. This reveals the advantage of direct input over relative cursor input in one-handed target selection when occlusion is not an issue and when targets are reachable. In contrast, both Back and Dual-Surface outperformed Shift in selection time for small targets (< 10 pixels or 4mm), and when the target was placed closer to the center of the screen.

Back and Dual-Surface led to a relatively consistent performance across targets of different sizes and at different locations. Participants finished the task faster with Dual-Surface than with Back. Note that, with Dual-Surface, participants had to select the target using the Back even if the target was large enough for the thumb. It is possible that without this restriction the performance of Dual-Surface can be improved further. Overall, the results support our hypothesis that one-handed interaction can benefit from the effective coordination of absolute thumb input in the front and relative cursor input in the back.

# 6. EXPERIMENT 2 – TUNNELING

The results of the first experiment revealed that target selection with *Dual-Surface* was more efficient than with input via the *Front* or the *Back* alone. We also wanted to evaluate the possibility of using Dual-Surface with more complex tasks. Note that none of the prior work provided any empirical evidence of the effectiveness of behind-the-screen input for complex tasks. Other tasks for one-handed input are also common, for example, scrolling through a list of contacts with one-hand is [13]. This type of task is commonly categorized as a steering task. Steering is also routinely carried out within a variety of contexts, such as highlighting a piece of text or navigating through file menus [1]. The purpose of this experiment was to evaluate the effectiveness of the Dual-Surface interaction in steering tasks.

## 6.1 Virtual Enhancement

Note that one of the obvious benefits of having a relative cursor is that it can be augmented with virtual enhancements [1, 2, 5, 10, 16, 21] to facilitate a variety of interactions. For instance, menu navigation is more effective with the inclusion of a pseudo-haptic enhancement that allows the cursor to stay within the menu 'tunnel' [1]. In this study, we leverage upon the ability to attach virtual enhancements to the relative cursor in the Dual-Surface interaction. In our task, the virtual enhancement consisted of a pseudo-haptic effect that aids in the tunneling. While this is specific to tunneling, we propose the use of Dual-Surface widgets (see Discussion section) for which the enhancements vary dynamically, based on the control the user is operating on the front. For example, a menu or scrollbar could employ an enhancement similar to the one we describe here. However toolbar buttons or other controls may adopt other enhancements such as those proposed for expanding targets [16] or for dynamically varying the cursor size [10]. Such enhancements are not possible with absolute input and therefore the attachment is associated only with the relative input behind-the-display and not with the front.

Similar to [1], the pseudo-haptic enhancement was simulated by software, and is described as follows:

1. The cursor initially exists within a rectangle (square in this case and the dimensions were defined as the tunnel width)
2. When the cursor touches a side of the square, the cursor's position is centered on the center of that side (see Figure 5).
3. A new square is then centered on the new cursor position and we go back to step 1.

It is possible to fail the task by exiting from the middle of the tunnel.

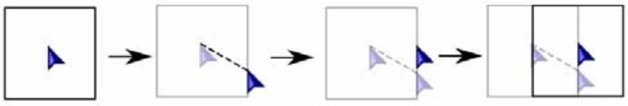
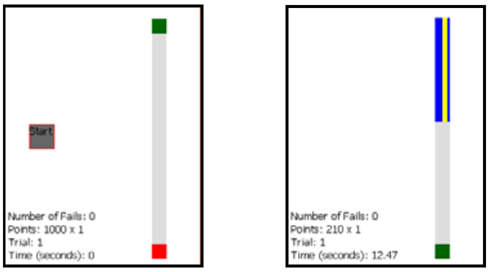


**Figure 5 Once the cursor hits an edge, it is brought back to the center of the edge. The square itself was not visible to the user.**

## 6.2 Task

The task was carried out by first clicking a "Start" button, and then quickly moving to the start of the tunnel to carry out the tunneling. The task simulated a canonical situation, in which a user moves the cursor on the display to a widget before steering in the tunnel (i.e. to acquire the thumb on the scrollbar and then to move it). In the steering component, the participants click the start of the tunnel and then steer within the tunnel until they exit from the end. The tunnel was placed vertically and to the right side, for easy access with the thumb. The start and end of the tunnel were rendered in green and red. The "Start" button was placed on the left-hand side of the screen (see Figure 6 left).



Number of Fails: 0
Points: 1000 x 1
Trial: 1
Time (seconds): 0

Number of Fails: 0
Points: 210 x 1
Trial: 1
Time (seconds): 12.47

**Figure 6 (left) "Start" button and the tunnel. The start and the end of the tunnel were rendered in green and red. (right). The "Start" button disappeared after a trial started. The yellow band indicates the offset of the user's thumb from the center of the tunnel.**

## 6.3 Conditions

We included visual feedback for all techniques. Visual feedback displayed the magnitude of deviation of the thumb from the center of the tunnel. The deviation was displayed using a yellow band. The width of the yellow band gave feedback as to how much the user deviated from the center of the tunnel (see Figure 6 right). As with the first experiment, we evaluated three one-handed techniques: *Front* (via thumb), *Back with virtual enhancement* (via cursor), and *Dual-Surface* (via both). In the *Front* technique, participants completed the entire task using only their thumb. When steering in the tunnel, they slid their thumb on the screen from the top of the tunnel to the bottom. For the enhanced *Back* technique, participants completed the entire task using the cursor, whose movement was controlled by the index finger. When steering within the tunnel, pseudo-haptic enhancement was provided to restrict the cursor movement within the tunnel. For the *Dual-Surface* technique, participants were asked to click the “Start” button and the start of the tunnel using their thumb, but to use the enhanced *Back* technique for steering. As with the Back technique, the Dual-Surface technique also used a virtual enhancement of the cursor.

## 6.4 Participants

Eleven participants (10 males, 1 female) between the ages of 20 and 35 were recruited. Three of the 11 participants had participated in the first experiment. All participants were right-handed.

## 6.5 Procedure

A trial started after participants clicked the “Start” button, and ended either after participants successfully completed the trial or after the trial failed. A trial was marked a failure if participants failed to complete the tunneling task. Participants were given 5 attempts to complete the tunneling task once the green start button of the tunnel was clicked. The tunneling task was marked as successful when participants exited the tunnel at the red end. Failure to do so counted as a failed attempt. A trial failed after 5 failed attempts. Participants were instructed to complete the task as fast and as accurately as possible. To promote better performance with the various input techniques, we implemented a reward system such that when the user performed better more points were awarded.

A warm-up session was provided to the participants at the start of each technique. Participants were given 5 practice trials per condition before starting the experiment. The entire experiment lasted 40 minutes. Participants were encouraged to take breaks during the experiment. Participants completed a post-experiment questionnaire to rank the techniques.

## 6.6 Experimental Design

The experiment used a 3×2×3 within subject factorial design. The independent variables were input *Technique*, (*Front*, *Back with virtual enhancement*, and *Dual-Surface*), tunnel *Length* (160px and 260px), and tunnel *Width* (12px, 15px, 18px). The width of the 15px tunnel (6mm) was selected as the intermediate level as it represents the standard width of a scroll bar on the PDA. Furthermore, our pilot study showed that 12px (4.8mm) was the smallest width of the tunnel that allowed participants to perform the tunneling task using the thumb.

Each trial represented a Technique×*Length*×*Width* combination, and each combination was repeated 10 times by each participant. The order of presentation of the trials was randomly chosen. Input techniques were counter balanced among participants.

The experimental design can be summarized as:

3 *Techniques*: (*Front*, *Back with virtual enhancement*, and *Dual-Surface with virtual enhancement*) ×
2 *Length*: (160px and 260px) ×
3 *Width*: (12px, 15px, and 18px) ×
10 R*epetitions* ×
11 *Participants*
= 1980 data points in total

## 6.7 Results

For all our analyses we used the univariate ANOVA test and Tamhane post-hoc pair-wise tests (unequal variances) with subjects as random factors. For the tunneling task we performed the analysis on several dependent variables separately. The last attempt time, was the time taken to steer within the tunnel for the last successful steering time. This time represents the performance of the participant after attempting 1 or more times to steering within the tunnel. Total time consists of the total time taken by the participants to complete the tasks, after repeated trials or attempts. This time include also the time it took the participants to clutch to return to the top of the tunnel to begin the task. The number of attempts represents the number of times it took the participants to complete the trial.

### *6.7.1 Last Attempt Time*

There is a significant effect of *Technique* ($F_{2,20} = 30.019$, $p<0.001$) and of tunnel *Length* ($F_{1,10} = 48.425$, $p<0.001$) on completion time. There was no main effect of tunnel *Width* ($F_{2,20} = 2.639$, $p=0.096$) on completion time. Figure 7 (left) shows average completion time for each technique, by tunnel width and length. We found interaction effects for *Technique×Length* ($F_{2,20} = 14.664$, $p<0.001$). There was no interaction effect for T*echnique×Width* or for *Width×Length*.

Post-hoc pair-wise comparisons (unequal variance assumed) show significant differences across each of the three pairs of the three techniques ($p<0.001$). Performance with Back (877.4ms, s.e. 29.5) alone was significantly faster than either Dual-Surface (1144.6ms, s.e. 29.5) or just the Front alone (2492.5ms, 29.5).

Post-hoc pair-wise comparisons did not reveal significant differences across all three pairs of tunnel width, suggesting that users performed equally well with the 12 pixel tunnel width as with the 18 pixel tunnel.

### *6.7.2 Total Time*

There is a significant effect of *Technique* ($F_{2,20} = 40.511$, $p<0.001$), of tunnel *Width* ($F_{2,20} = 77.022$, $p<0.001$), and of tunnel *Length* ($F_{1,10} = 43.443$, $p<0.001$) on completion time. Figure 7 (right) shows average total time for each technique, by tunnel width and length. We found interaction effects for *Technique×Width* ($F_{4,40} = 58.107$, $p<0.001$), for *Technique×Length* ($F_{2,20} = 17.801$, $p<0.001$), and for *Width×Length* ($F_{2,20} = 7.336$, $p<0.001$).

Post-hoc pair-wise comparisons (unequal variance assumed) show significant differences across each of the three pairs of the three techniques ($p<0.001$). Performance with Dual-Surface (1974.78ms, s.e. 89) was significantly faster than either Front (6224.6ms, s.e. 89) or Back alone (3147.1, s.e. 89). Note that it took users approximately 1.5 times longer to complete all attempts with Dual-Surface, approximately 4 times longer with Back alone, and 3 times longer with Front, compared to the last attempt time. In our discussion in Section 7, we explain some of the reasons for the differences in these times.

Post-hoc pair-wise comparisons did not reveal significant differences across all three pairs of tunnel width, suggesting that users performed equally well with the 12 pixel tunnel width as with the 18 pixel tunnel.

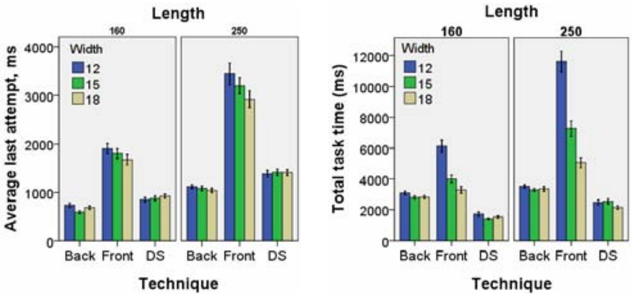


**Figure 7 (left) Average last attempt time; (right) Average task time, for each technique, by tunnel length and width (bars represent +/-1 standard error).**

### 6.7.3 *Number of Attempts*

There is a significant effect of *Technique* ($F_{2,20} = 53.695$, $p<0.001$) and of tunnel *Width* ($F_{2,20} = 53.136$, $p<0.001$) but no main effect of tunnel *Length* ($F_{1,10} = 1.997$, $p=.188$) on number of attempts. Figure 8 (left) shows average number of attempts for each technique, by tunnel width and length. We found interaction effects for *Technique*×*Width* ($F_{4,40} = 53.975$, $p<0.001$) but none for *Technique*×*Length* ($F_{2,20} = 1.359$, $p=.28$) or for *Width*×*Length* ($F_{2,20} = 1.377$, $p=.275$).

Post-hoc pair-wise comparisons (unequal variance assumed) show significant differences across each of the three pairs of the three techniques ($p<0.001$). The fewest attempts were made with Back alone (1.02 attempts, s.e. 0.03), then with Dual-Surface (1.14 attempts, s.e. 0.03), and the most with Front alone (2.22 attempts, s.e. 0.03).

Post-hoc pair-wise comparisons also show significant differences across each of the three pairs of tunnel widths ($p<0.001$). The fewest attempts were made with width 18 (1.02 attempts, s.e. 0.03), then with width 15 (1.14 attempts, s.e. 0.03), and most with width 12 (2.22 attempts, s.e. 0.03).

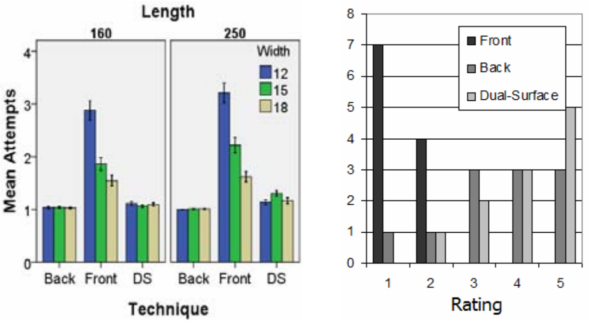


**Figure 8 (left) Average number of attempts for each technique, by tunnel length and width (bars represent +/-1 standard error). (right) Frequency of ratings from 1 (least preferred) to 5 (most preferred) for each of the three techniques.**

### 6.7.4 *Subjective Preference*

Participants rated each of the three techniques, from 1 (least preferred) to 5 (most preferred). Half of the participants rated Dual-Surface a 5, and one third rated it a 4. None of the participants gave Dual-Surface a rating of 1. On the other hand 70% of the participants rated Front a 1 and none rated it above a 2. Back was marginal and rated average across all participants. The rankings for each technique are provided in Figure 8 (right).

# 7. DISCUSSION

We first discuss the results of the second experiment and then based on the results of both experiments we present some recommendations to designers. We also propose the use of Dual-Surface input in a number of applications and finally present some of the limitations of this form of interaction.

## 7.1 Discussion of Experiment 2

The results of the second experiment showed the benefits of using a virtual enhancement with the behind-the-surface cursor. This is only possible with a relative cursor, which we made available in Dual-Surface. Even with visual feedback, participants still performed the tunneling task significantly slower with the thumb than with cursor. One interesting finding is that the average time to complete the last steering attempt for the Back was shorter than with the Dual-Surface input. For several reasons, we expected these two techniques to perform similarly as they both use the Back for the tunneling task. We noticed that, participants used different grips in favor of different techniques. For the Front, they held the device at about a 45° angle to the index finger. By holding the device in this orientation, they maximized the region reachable by the thumb. For the Back, the device was held in the same orientation as the extension of the index finger. This position helped to better perform vertical tunneling using the index finger. However, to perform this task comfortably with the Dual-Surface input, the participants needed to switch frequently from one grip to another. This tired their hands, and may have led to the unexpected performance.

Even though Back led to better performance in the last-attempt time, overall Dual-Surface was the most efficient technique. This resulted primarily from taking longer to clutch with the relative input on the Back, whereas, with the Dual-Surface technique users could simply move the cursor to the top of the tunnel with their thumb and then attempt to scroll again. As mentioned before, it took participants approximately 1.5 times longer to complete all attempts with Dual-Surface, approximately 4 times longer with Back alone, compared to the last attempt time. This shows that, with Back, participants spent a large proportion of their time clutching the cursor towards the goal. This finding supports our hypothesis that one-handed interaction could benefit from the effective use of absolute thumb input and relative cursor input.

Analysis on the number of attempts showed that participants made significantly more mistakes when using the thumb, which lead to the highest number of attempts with this technique. One interesting finding is that participants did not make more mistakes on the 250px compared to the 160px long tunnel. As shown in Figure 6, the tunnel of length 250px is almost the full length of the screen. Given that it is more difficult to steer through a vertical tunnel than a horizontal tunnel [5, 26], we suspect the highest average attempts shown in Figure 8 (left) are close to the upper bound of the number of failures in scrolling a scrollbar on the tested PDA. This suggests that designers may want to reconsider the design of scrollbars on smaller devices, either in software or with a solution such as the one proposed in this paper.

Tunneling tasks on mobile devices usually require both hands, one to hold the device and the other to perform the steering (sometimes with a stylus). We demonstrated that attaching a virtual enhancement to a cursor can assist in tunneling with one hand. This provides support for the use of Dual-Surface interaction, but also supports the need for easily accessible relative input on mobile device. While we only showed the advantage of relative input with the steering task, many other tasks including pointing and selection can benefit from the use of easily accessible relative input on mobile devices.

## 7.2 Applications

Numerous applications can benefit from one-handed interaction. We have demonstrated that small and typically unreachable targets can benefit from Dual-Surface interaction. Our results also reveal that complex tasks, such as steering, can benefit if the Dual-Surface input is augmented with virtual enhancements.

*Map Navigation.* Panning is a common operation in map browsing applications. Often, information that would normally be available to a user resides off-screen. In this case, panning is used for bring an off-screen target into view, and extra cursor movements towards the target are often required as the panning operation lands the cursor a distance away from the desired target. For instance, to select the off-screen target in Figure 9, users need to make a ← (right-to-left) gesture to bring the target into the display, and then make a → (left-to-right) gesture to move the cursor onto the target for selection. With Dual-Surface interaction, off-screen target selection can be performed with less effort. The user can simply leave the cursor in its original position. Instead of manipulating the cursor, the user pans the map to the left to move the target towards the cursor. Once the target is underneath the cursor, the user simply taps on the back to make a selection.

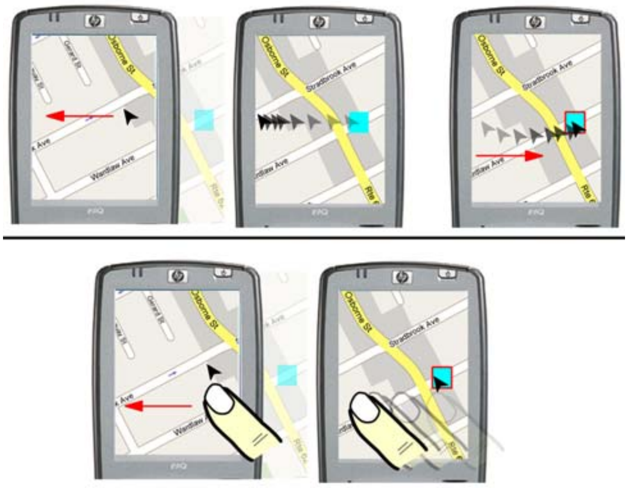


**Figure 9 Off-screen target selection. Top-left: pan the map using cursor requires a left gesture. Top-middle: after the panning, cursor is on the left edge. Top-right: cursor moves back to the right to make selection. In contrast, bottom-left: pan the map to left using thumb. Bottom-right: the target is moved underneath the cursor.**

*3D Object Rotation.* 3D manipulations such as rotations can take advantage of Dual-Surface interaction. Users can select a rotational axis anywhere on the screen with the back and then can pan their thumb to rotate the object along the selected axis. Intuitively, it would require more effort to perform the task with either the back or the front alone. Note that the coordination suggested in this application does not necessarily follow that proposed earlier, i.e. the back can also initiate the movement and the front can follow it.

*Simultaneous Input.* Although Dual-Surface is performed sequentially, one can also take advantage of naturally occurring simultaneous actions. One example of such an interaction is zooming. Zooming can be triggered by having the thumb and index fingers make simultaneous opposite gestures. For example, a user can trigger a zoom-in action by having the thumb make a ← gesture and the index finger make a → gesture.

*Dual-Surface Widgets.* We can design a new class of widgets that support Dual-Surface input, which we refer to as DS-widgets. For instance, we described the use of a virtual enhancement with the scrollbar. This could eventually become a DS-scrollbar which would behave just like a normal scrollbar with the front input, but would also use any possible enhancement when coordinating the front with the back input. Other similar widgets, such as toolbar buttons could be made to work with Dual-Surface. Items on a DS-toolbar could expand and shrink under the influence of the position of the relative cursor as in [16].

*Hardware Alternatives.* Many hardware design options are available. In our study, the manipulation of cursor movement was through a touchpad. Alternately, this could be replaced with a mini joystick. The joystick could be similar to a trackpoint. Note that, previous research has reported that, due to the kinematic limitations of the index finger, complex gestures involving vertical motions are difficult with the index finger [26]. A joystick, however, requires minimal finger motion to control the cursor movement and also allows for rate-based control of the cursor. Furthermore, it is possible with a joystick to provide real force-feedback to support richer interactions. Another option to replace the touchpad is to use an optical sensor. Cursor movement can be controlled by the movement of the finger tip against the optical sensor.

## 7.3 Recommendations

Based on the findings of our experiments, we make the follow recommendations for enhancing one-handed use:

- Complement mobile devices with relative cursor input that is accessible such as a touchpad behind the display;
- Relative cursor input could be associated with virtual enhancements for making complex tasks easier;
- Large targets that are accessible with the thumb should be placed in accessible areas for the thumb;
- Smaller targets which would be difficult to access with the thumb but relatively easy to access with the cursor, could be placed in areas that are best suited for cursors, such as in corners and edges (the width factor in the Fitts equation increases significantly when targets are on the edges, reducing the index of difficulty);
- To facilitate steering tasks widgets requiring tunneling should be made larger for easy access in the front.

## 7.4 Limitations of Dual-Surface Interactions

While our results show the advantages of Dual-Surface input, this interaction needs to be further developed to overcome some limitations. For instance, the results of our studies showed that Dual-Surface input was faster than Back alone. This may not remain true in situations where the goal is too far to be reached by the thumb. In this case, most of the input will take place on the back. Moreover, in different tasks, users may have to use different grips to facilitate input on the Back or on the Front. Frequently changing grips may discourage users from using Dual-Surface input. Furthermore, our experiments evaluated Dual-Surface input in only two, albeit common, tasks. It is left for future work to determine how Dual-Surface input would operate on other routine tasks, and whether users would employ it in real-world mobile settings, such as when walking or driving a car.

## 8. CONCLUSION AND FUTURE WORK

In this paper, we presented two experimental studies in measuring the performance of Dual-Surface input in facilitating one-handed interaction on mobile devices. We found Dual-Surface input outperformed Front input and Back input in both tunneling and target selection tasks. This clearly shows the promise of Dual-Surface interaction. The benefit of input via both sides of a mobile device is that it takes advantage of the best of both relative and absolute input. Based on the findings, we recommend Dual-Surface interaction to be used in handheld devices

While we conducted our experiments in a lab environment, it would be interesting to repeat the experiment using different virtual enhancements in situations where participants are mobile, e.g. while walking or sitting in a bus. We plan this for future work. Furthermore, it would be interesting to have the Back support both relative and absolute input to support smoother transactions between relative and absolute input. Augmenting the back with multitouch input will also provide support for richer interactions with one-handed input.

## 9. REFERENCES

[1] Ahlström, D. (2005). Modeling and improving selection in cascading pull-down menus using Fitts' law, the steering law and force fields. *CHI*, 61–70.

[2] Ahlström, D., Hitz, M., and Leitner, G. (2006). An evaluation of sticky and force enhanced targets in multi target situations. *NordicHCI*, 58–67.

[3] Benko, H., Wilson, A., and Baudisch, P. (2006). Precise selection techniques for multi-touch screens. *CHI*, 1263-1272.

[4] Baudisch, P. and Chu, G. (2009). Back-of-device Interaction allows creating very small touch devices. *CHI* (in press).

[5] Dennerlein, J., Martin, D., Hasser, C. (2000). Force-feedback improves performance for steering and combined steering-targeting tasks. *CHI*, 423–429.

[6] Engelbart, D. C. (1968), SRI-ARC. A technical session presentation at the Fall Joint Computer Conference (NLS demo: The computer mouse debut), Engelbart Collection, Stanford University Library, Menlo Park (CA).

[7] Esenther, A. and Ryall, K. (2006). Fluid DTMouse: better mouse support for touch-based interactions. *AVI*, 112-115.

[8] Forlines, C., Wigdor, D., Shen, C., and Balakrishnan, R. (2007). Direct-touch vs. mouse input for tabletop displays. *CHI*, 647-656.

[9] Forlines, C. and Balakrishnan, R. (2008). Evaluating tactile feedback and direct vs. indirect stylus input in pointing and crossing selection tasks. *CHI*, 1563-1572.

[10] Grossman, T. and Balakrishnan, R. (2005). The Bubble Cursor: Enhancing target acquisition by dynamic resizing of the cursor's activation area. *CHI*, 435–459.

[11] Guiard, Y. (1987). Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model. *J. Motor Behavior*, 19 (4), 486-517.

[12] Karlson, A., Bederson, B., and SanGiovanni, J. (2005). ApLens and launchTile: two designs for one-handed thumb use on small devices. *CHI*, 201-210.

[13] Karlson, A., Bederson, B., Contreras-Vidal, J. (2006) Understanding Single-Handed Mobile Device Interaction. *Tech Report HCIL*.

[14] Karlson, A. and Bederson, B. (2007). Thumbspace: Generalized one-handed input for touchscreen-based mobile devices. *Interact*, 201–210.

[15] Karlson, A. and Bederson, B. (2008). One-handed touchscreen input for legacy applications. *CHI*, 1399-1408.

[16] McGuffin, M. and Balakrishnan, R. (2002). Acquisition of expanding targets. *CHI*, 57-64

[17] Meyer, S., Cohen, O., and Nilsen, E. (1994). Device comparisons for goal-directed drawing tasks. *Extended Abstracts of the CHI*, 251-252.

[18] Olwal, A., Feiner, S., and Heyman, S. (2008). Rubbing and tapping for precise and rapid selection on touch-screen displays. *CHI*, 295-304.

[19] Parhi, P., Karlson, A., and Bederson, B. (2006). Target size study for one-handed thumb use on small touch screen devices. *MobileHCI*, 203-210.

[20] Potter, R., Weldon, L., Shneiderman, B. (1988). Improving the accuracy of touch screens: an experimental evaluation of three strategies. *CHI*, 27-32.

[21] Rodgers, M., Mandryk, R. and Inkpen, K. (2006). Smart sticky widgets: Pseudo-haptic enhancements for multi-monitor displays. *Smart Graphics*, 194-205.

[22] Sears, A. and Shneiderman, B. (1991). High precision touchscreens: design strategies and comparisons with a mouse. *IJMMS*, 34(4), 93–613.

[23] Sugimoto, M. and Hiroki, K. (2006). Hybridtouch: an intuitive manipulation technique for pdas using their front and rear surfaces. E*xtended Abstracts MobileHCI*, 137–140.

[24] Vogel, D. and Baudisch, P. (2007). Shift: a technique for operating pen-based interfaces using touch. *CHI*, 657-666.

[25] Wigdor, D., Forlines, C., Baudisch, P., Barnwell, J., and Shen, C. (2007). LucidTouch: A See-through Mobile Device. *UIST*, 269–278

[26] Wobbrock, J. O., Myers, B. A., and Aung H. (2008). The performance of hand postures in front- and back-of-device interaction for mobile computing. *IJHCS*, 66(12), 857-875.

[27] Yatani, K., Partridge, K., Bern, M., and Newman, M. (2008). Escape: A target selection technique using visually-cued gestures. *CHI*, 285-294.