One-Handed Touchscreen Input for Legacy Applications

Amy K. Karlson and Benjamin B. Bederson
Human-Computer Interaction Lab, Department of Computer Science
University of Maryland, College Park, 20742
karlson@microsoft.com, bederson@cs.umd.edu

ABSTRACT
Supporting one-handed thumb operation of touchscreen-based mobile devices presents a challenging tradeoff between visual expressivity and ease of interaction. ThumbSpace and Shift—two new application-independent, software-based interaction techniques—address this tradeoff in significantly different ways. ThumbSpace addresses distant objects while Shift addresses small object occlusion. We present two extensive, comparative user studies. The first compares ThumbSpace and Shift to peripheral hardware (directional pad and scrollwheel) and direct touchscreen input for selecting objects while standing and walking. The data favored the Shift design overall, but suggested ThumbSpace is promising for distant objects. Our second study examines the benefits and learnability of combining Shift and ThumbSpace on a device with a larger screen (3.5"). We found their combined use offered users better overall speed and accuracy in hitting small targets (3.6 mm^2) than using either method alone.

Author Keywords
Mobile devices, touchscreens, one-handed interaction.

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

INTRODUCTION

Mobile phones today exist in a wide spectrum of designs. The most widespread styles in circulation feature the classic combination of numeric keypad and non-touchscreen display, but the attention garnered by new devices such as Apple's iPhone indicate that larger touchscreen devices are gaining ground. Yet as devices evolve, certain constraints on users' visual, physical, and mental resources will remain [15]. One example is the mobile user who operates a device with one hand because the other is needed to carry items and interact with the environment [15]. Not only can users benefit from interfaces that free one hand for the physical and intellectual demands of mobile tasks [17], but a survey [14] suggests that users generally prefer to use mobile devices with one hand. Unfortunately, devices with touchscreens have traditionally been designed for two-handed use, often requiring a stylus for small targets.

Designing touchscreen interfaces for one-handed operation presents a challenging trade-off between visual expressivity and ease of interaction. On one end of the spectrum, existing (legacy) mobile UI toolkits tend to use small, stylus-oriented widgets for interaction, resulting in information-rich interfaces composed of targets that are too small [18] or too far to be hit reliably with the thumb [14]. On the other end of the spectrum, the "lowest common denominator" approach of ensuring that all targets are large and within reach [12] wastes the precious screen space of small devices, and can slow down users when two hands *are* available. These factors have led us to consider an alternative design strategy.

We developed ThumbSpace (Figure 1) to address both the reach and accuracy problems that users experience when operating arbitrary touchscreen interfaces with thumbs [13]. Its design is inspired by the substantial body of research that has addressed the challenges that distance plays in large display interaction. ThumbSpace applies existing techniques for accessing objects out of *arm's* reach on large displays, adapting them to the problem of accessing objects out of *thumb's* reach on handheld devices.

Conceptually, ThumbSpace serves as an absolute touchpad superimposed on the display, to which all screen objects are mapped (Figure 1b). Reach limitations are addressed by allowing users to personalize ThumbSpace's size and placement (Figure 1a), thereby accommodating user

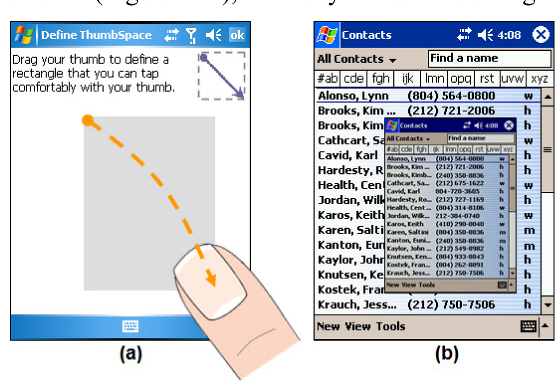


Figure 1. (a) Defining a ThumbSpace. (b) ThumbSpace in use.

differences in hand preference, geometry, motion range, and grip. ThumbSpace supports access to all display objects within an accessible portion of the screen, and applies dynamic visual feedback and selection tuning to improve accuracy. Finally, ThumbSpace is user-activated and so is unobtrusive when not needed.

ThumbSpace is based on users' natural inclination to touch the interface with their fingers when a stylus is not available. But problems of high visual demand, targeting accuracy, and finger occlusion suggest peripheral hardware solutions might offer advantages. Blackberry devices, for example, have shown how effective thumb wheels and trackballs can be for controlling a non-touchscreen device. So after first introducing the ThumbSpace technique, we present an initial user study that compares ThumbSpace to two touchscreen techniques—direct touch and Shift [22]—and two peripheral hardware input methods—directional pad (DPad) and scrollwheel—for selecting small targets in a dense 2D interface. The results show that Shift offers the most desirable combination of speed and accuracy for small (occluded) targets within thumb reach. They also suggest that ThumbSpace may provide added benefit for distant objects. Since occlusion and reach are orthogonal problems, the first study raises the question: *can Shift and ThumbSpace be usefully combined*?

We present a follow-up study that addresses this question by investigating the relative benefits of using ThumbSpace *together* with Shift. To more fully quantify ThumbSpace's solution to reach issues, this second study uses a larger screen than does the first (3.5" vs. 2.8"). Here our results show that ThumbSpace and Shift compose effectively—their combined use significantly outperforms their individual use—while simultaneously addressing both occlusion and reach. Further, users find the combination intuitive and preferable. With some practice, users naturally choose to use the method that is quantitatively more effective for the given region: ThumbSpace for distant objects, and Shift for near objects.

RELATED WORK

One-handed interaction

Research in one-handed mobile interaction has largely focused on specific user tasks, such as media play [19], text entry [23], and application navigation [12]. Karlson et al. [13] looked more generally at human factors requirements for one-handed use of mobile devices, including situational and task preferences for hand use as well as biomechanical limitations of thumbs. They found widespread interest in single-handed operation of mobile devices, but discovered that current designs, especially for touchscreen devices, do not accommodate one-handed scenarios well. Their results also show that the areas of a device users are comfortable interacting with vary by user and device size.

Many current touchscreen interfaces consist of widgets similar in size and function to those featured on a desktop PC. While acceptable for interaction with a 1 mm stylus tip, research suggests touchscreen targets smaller than about 1 cm [18] can result in unacceptably high error rates when accessed with the thumb due to finger occlusion. Recently, Vogel and Baudisch [22] developed the Shift technique as an improvement over Sears and Shneiderman's offset cursor [21]. The offset cursor couples a selection cursor positioned off the tip of a user's finger with a stabilization algorithm to achieve character-level selection accuracy. The downside to the offset cursor is that users have to aim *below* the intended target. Shift instead allows users to aim directly at the target. After a variable delay, it displays a callout of the screen area covered by the user's finger. The callout includes a crosshair cursor to show the position of finger contact, which users can adjust before lifting their finger to perform selection. The callout is displayed after a short delay which is longer for larger targets. The result is that Shift is only shown when users are likely to need it.

While Shift holds great potential for one-handed selection of targets within reach of the thumb, further investigation is necessary to understand whether pixel-level selection is appropriate under mobile conditions, and whether Shift works equally well for objects along the perimeter of the screen, which occur frequently in today's designs. ThumbSpace, on the other hand, is designed to support targets at edges just as well as those away from the edges. More importantly, Shift was designed for two-handed index finger interaction, and so does not address the limits of thumb reach that ThumbSpace aims to support.

Reaching Distant Objects

ThumbSpace was inspired by wall-sized displays, which pose problems with out-of-reach and widely-separated interface objects. Unfortunately, Fitts' Law dictates that increasing travel distance without a commensurate increase in target size will increase access time. Solutions have thus typically focused on: 1) decreasing movement distance to targets; and/or 2) increasing target sizes.

Improving target acquisition for mouse-based interaction has often involved clever manipulation of the control-display (CD) ratio. Some approaches include slowing mouse movement over interaction targets (Semantic Pointing [6]), jumping the cursor to the nearest target (Object Pointing [9]), and predicting the user's intended target (Delphian desktop [2]). But these approaches become less effective as the number of nearby objects increase. Other approaches in smart cursor control make targets easier to hit by increasing the cursor size (e.g., Area Cursor [11] and Bubble Cursor [8]). Unfortunately, these techniques are not directly applicable to touchscreen interaction; touching the screen directly implies a 1:1 correspondence between motor and display movement, so there is no CD ratio or cursor to manipulate.

Direct screen interaction with fingers or pens is common in tablet, mobile, and wall computing. Techniques to improve object access speed in these arenas have focused on minimizing the movement distance to targets. However, most research that focuses on icon placement or selection [3, 5, 10] is inappropriate for PDA interfaces because drag-and-drop and object placement are used much less frequently than interactions such as tapping buttons, links, and check boxes. Another approach has been to provide a nearby miniaturized version of the display, or Radar View [16], that can be manipulated directly. However, both the Radar View and the pen-based extension to the Bubble Cursor, the Bubble Radar [1], again focus on object placement tasks, rather than general application interaction.

THUMBSPACE DESIGN

ThumbSpace is an interaction technique that allows arbitrary touchscreen interfaces to be controlled with a thumb, thereby preserving the richness of data presentation and the efficiency of navigation when two hands are available. In this section we describe the current design and highlight changes from the original design that follow directly from the formal evaluation reported in [13].

ThumbSpace supports individual differences by allowing users to operate a device using their most comfortable and stable one-handed grip. Each user defines a *ThumbSpace*—a region of the touchscreen surface that she considers easy to reach and interact within—by dragging her thumb along a diagonal that defines the upper left and lower right corners of a rectangular region (Figure 1a). During its definition, we constrain the ThumbSpace to match the aspect ratio of the display area.

When needed, a thumbnail version of the display is presented within the user's ThumbSpace for use as a Radar View; selecting an object in the thumbnail selects the object in the original display, thereby allowing all display elements to be accessed within a personalized interaction region. However, since Radar View objects can be too small to hit reliably with the thumb, visual feedback and input tuning are offered to help users refine their selections.

ThumbSpace Interaction

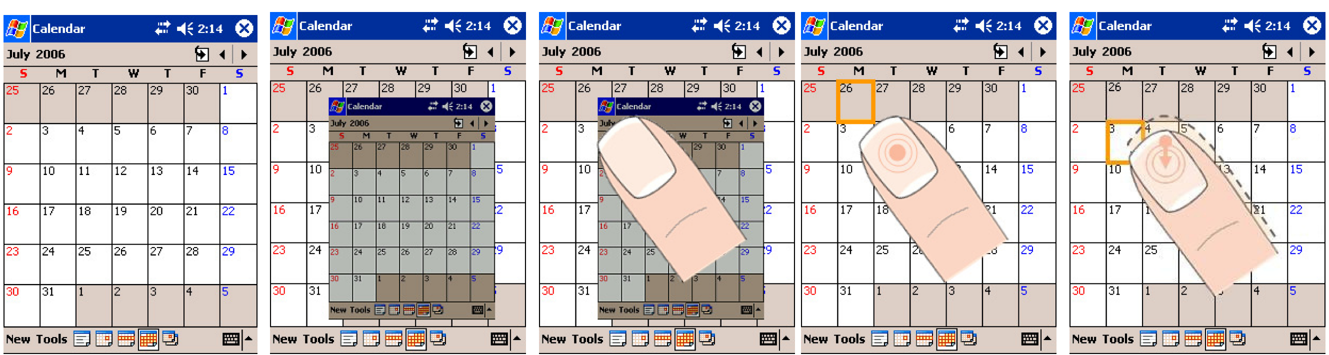
Object selection with ThumbSpace is performed in four phases: *trigger*, *aim, adjust,* and *lift*. In the *trigger* phase, the user presses the center of the DPad ("enter") to launch the radar thumbnail within her ThumbSpace (Figure 2b). The user then *aims* her thumb at the object in the thumbnail that represents the one she wants to select in the main display. Here, the ThumbSpace acts like an absolute touchpad. The object in the main display associated with the selected object in the radar display is shown highlighted by a thick orange object selection cursor, and ThumbSpace disappears (Figure 2d).

Before lifting her thumb, the user can *adjust* her selection by using the ThumbSpace as a relative touchpad for controlling the object cursor. If the user rolls or drags her thumb more than a fixed number of pixels, the object cursor animates to the closest object in the direction of movement (Figure 2e). Tuning a selection in ThumbSpace is similar to Object Pointing [9], which jumps a desktop cursor to the nearest object in the direction of mouse movement once the cursor leaves an object's bounds. However, our *adjust* strategy differs somewhat from Object Pointing in that the *adjust* threshold is independent of the display object size. Finally, the user confirms the selection by *lifting* her thumb. This selection design is inspired by Potter's lift-off strategy for touchscreen object selection [20], which allows users to verify and adjust a selection before committing to an action.

The current ThumbSpace interface design differs from the one reported in [13] in three ways. First, the initial design did not show a Radar View, only a whitewashed region. Second, the use of ThumbSpace was mandatory in the original design and was displayed for use at all times. Last, the aspect ratio of the original ThumbSpace was not constrained to that of the screen area as we now enforce.

STUDY 1: TOUCHSCREEN VS. PERIPHERAL INPUT

Despite the flexibility that new touchscreen interaction methods offer, peripheral input hardware has the potential to be more accurate and demand less attention. We therefore conducted an initial study to quantify the relative speed and accuracy of peripheral vs. touchscreen input for one-handed object selection on touchscreen devices.



(a) (b) (c) (d) (e)

Figure 2. ThumbSpace interaction. (a) A calendar. (b) Pressing "enter" on the directional pad launches a Radar View within the user's predefined ThumbSpace. (c) To select July 3rd, the user aims for the 3rd in the Radar View. (d) Touching the screen highlights the object associated with the Radar View proxy actually hit (June 26th), and the ThumbSpace disappears. (e) To adjust the cursor the user rolls her thumb downward to move it from June 26th to July 3rd, and lifts her thumb to perform the selection.

Independent Variables

Input Methods. We wanted to compare ThumbSpace to the most promising peripheral hardware alternatives on devices today. We chose the scrollwheel (ScrollWheel) for the acclaim it brought to the non-touchscreen Blackberry devices, and the DPad for its ubiquity. As a baseline for touchscreen interaction we chose direct finger touch (DirectTouch). Under the assumption that ThumbSpace would offer users more accurate targeting than DirectTouch at the expense of speed, we also chose a technique that specifically addresses targeting accuracy for fingers on touchscreens. Recently, Vogel and Baudisch [22] showed that users made fewer errors using Shift (Figure 3c) than direct touch for hitting targets ≤ 2.9 mm^2, but not ≥ 4.3 mm^2. Given the likelihood that Shift would help users in hitting the small targets used in our study (3.6 mm^2), we included it as a more competitive variant of DirectTouch. Although Shift was designed for index-finger operation, our study explored its viability for thumb use (Figures 3a,c).

Mobility. Many previous studies have established the negative impact that mobility has on mental demand and task performance (e.g., [17]). Assuming that peripheral input methods offer more stability and less mental, visual, and physical demand than touchscreen methods, we would expect higher activity to degrade performance more when using touchscreen vs. hardware input methods. To understand this relationship, we studied users performing tasks while both standing and walking. During the walking condition, users chose a comfortable walking pace along a 19’x7.5’ figure-8 path.

Tasks

The study tasks were modeled after touchscreen selection activities that would typically be performed with a stylus. Given our focus on arbitrarily rich, legacy touchscreen designs, we opted for a 2D input space of small targets. Since an earlier study found ThumbSpace performance to be independent of target size [13], we studied a single target size of 20x20 pixels (3.6 mm^2), which is representative of standard Windows Mobile widgets (e.g., checkboxes: 15 px; buttons: 21 px; text boxes: 19 px).

Targets were placed within a 6x8 grid of 40x40 px cells. Because the targets were smaller than their assigned cells (20 vs. 40 px), the target for each trial was placed in the center of the designated cell. Distractors were randomly assigned one of two sizes (20 px and 13 px, with probabilities 0.2 and 0.8 respectively), and positioned at random locations within their cells. By randomizing the locations of non-target objects, we hoped to create the illusion of a non-uniform layout space, and by shrinking some of the non-targets, we increased the percentage of the map background displayed, which we expected to be helpful context during the use of Shift (Figure 3c).

Trials were introduced by a message dialog that pointed to the goal target, distinguished by its yellow center, and described the user action required to start the trial timer: to begin a ScrollWheel trial, users pressed the scrollwheel; for DPad and ThumbSpace, users pressed the center of the DPad to begin; for DirectTouch and Shift, users tapped the dialog to begin. The dialog was centered on the user’s ThumbSpace as long as it did not overlap the target, and was otherwise placed within the bounds of the ThumbSpace, either above or below the target.

(a) (b) (c)

Figure 3. Study 1. (a) PDA use in the standing condition. (b) ThumbSpace, and (c) Shift representations.

A rectangular orange cursor was used as the input focus. For ScrollWheel trials, the cursor started at the top left target and moved between objects in a left-to-right, top-to-bottom pattern, or the reverse, for downward and upward scrolls respectively (excluding top to bottom wrapping). DPad trials also began in the upper left corner, but the cursor could be moved up, down, left or right (excluding all wrapping). For DirectTouch, ThumbSpace, and Shift, the cursor was shown around a target whenever it had the input focus (e.g., Figure 3c). Upon target selection, the cursor faded out, and an audio tone indicated success or failure.

Implementation and Apparatus

The ThumbSpace prototype was developed as an input handler to custom applications written in C# using the PocketPiccolo.NET graphics toolkit [4]. The software ran on a Cingular 8525 PocketPC phone (11.2 x 5.8 x 2.2 cm) with a 2.8” display. We chose this device because it was the only one available that had all the hardware components we wanted to study, thus avoiding a potential confound.

Resistive touchscreens recognize only a single average point from a finger touch, which is not only hard to predict, but also unstable, due to the varying deformation the finger tip on the screen surface. This problem is well known to make pixel-level targeting difficult, so various approaches have been used to stabilize finger input [21, 22]. We used the recursive dynamic filter of Shift [22], with cutoff frequencies of 5 and 20 Hz interpolated between 54 and 144 mm/s. Given the small size of our targets, Shift was configured to display the callout as soon as the finger touched the screen, and we did not correct for users’ perceived contact points as in [22]. For DirectTouch we used a land-on strategy with input stabilization.

Method

The study was a 5 (*Input*: DPad, ScrollWheel, DirectTouch, Shift, ThumbSpace) x 2 (*Mobility*: standing, walking) x 48

(*Position*) within-subjects design. Presentation of *Input* and *Mobility* were counterbalanced across participants, and the 48 *Position* trials were randomized within blocks. Dependent variables collected included task time, error rate, satisfaction ratings, and input preference rankings.

Participants and Procedure

Twelve right handed volunteers (8 male, 4 female) ranging in age from 21 to 31 (μ = 26) were recruited via fliers posted in the Department of Computer Science. Participants received $15 for 1.5 hours of their time.

Before each block of trials, the study administrator explained and demonstrated the *Input* method that would be used. Participants were instructed to select each target as quickly as possible without sacrificing accuracy. Participants then took a standing position or began walking the figure-8. After 24 practice trials, users performed the 48 timed tasks. After each block, participants filled out a short subjective questionnaire about the *Input* x *Mobility* condition completed, and proceeded to the next block.

The study wrapped up with a "usability" phase. First users performed a block of "Choice" trials, for which they could use either DirectTouch or ThumbSpace. The study software then presented 10 selection tasks for each of six *Inputs* (DPad, ScrollWheel, DirectTouch, Shift, ThumbSpace, and Choice) in that order (roughly familiar to unfamiliar) for a Windows Mobile Calendar interface (Figure 2). Following the usability phase, users ranked the input methods from 1 ("favorite") to 6 ("least favorite") for general device use, including a Choice option for using ThumbSpace when desired, and Shift otherwise. While we expected users to draw from their experience with Choice in the usability phase, we assumed that Shift would complement ThumbSpace better in real usage than DirectTouch would.

Study 1 Results

Accessing targets with one thumb on a touchscreen can be hindered by several factors that vary with target location, such as thumb reach limits, bezel interference, and differences in the size and shape of the thumb contact area. One problem with studying a small set of targets is that they may not fully capture the locations where these factors impact users in everyday use, so by studying screen locations at high density, we captured a broad range of conditions that affect selection performance. Yet since statistical methods are unlikely to uncover trends across such a large number of locations, for analysis purposes, we partitioned the 48 targets into 12 equivalence classes, arranged as a 3x4 grid of regions (Figure 5). Region data was derived by averaging the time and error data of the 4 targets belonging to the region. Trials with selection errors and outliers more than three standard deviations from the mean within each *Input* type were excluded from the aggregation.

Task Time

Task time was measured from the onset of the trial (when the scrollwheel or center of the DPad was released, or the user's finger was lifted from the task dialog) to the completion of the trial (when the scrollwheel or center of the DPad was pressed, or the user's finger was removed from the screen). Huynh-Feldt corrections are reported where sphericity did not hold and post hoc analyses of main effects were conducted using Bonferroni correction.

A 5 (*Input*) x 2 (*Mobility*) x 12 (*Region*) repeated measures Analysis of Variance (RM-ANOVA) revealed main effects of *Input* $F(4,24)=67.7$, $p<.001$, and *Region* $F(11,66)=68.5.4$, $p<.001$. Interestingly, there was no significant main effect for *Mobility*, perhaps indicating that the device was easy to control in one hand, the task was not mentally challenging, or the self-paced walking did not require much visual attention. However, there was an interaction of *Input* x *Region* $F(44,264)=50.0$, $p<.001$, as seen in Figure 5. Post hoc analyses showed that DirectTouch was significantly faster (865 ms), and ScrollWheel was significantly slower (3,311 ms) than the other input methods. While post hoc tests comparing Shift, ThumbSpace, and DPad did not reach significance, Figure 4 shows a clear trend that at 1,581 ms, Shift supports consistently faster interaction than either ThumbSpace (2,027 ms) or DPad (2,191 ms).

Figure 5 explains both the main effect of *Region* and the *Region* x *Input* interaction: task time increased with *Region* because the DPad and ScrollWheel times generally increased with region while task times for the remaining three input methods were relatively constant across regions. The increase in time by region for DPad and ScrollWheel

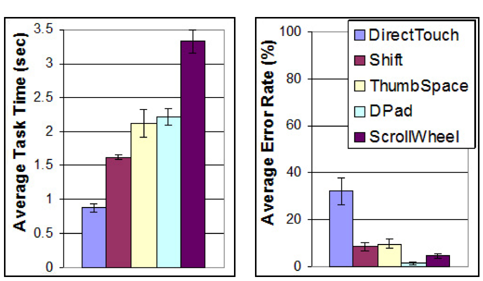


Figure 4. Study 1: average task times and error rates.

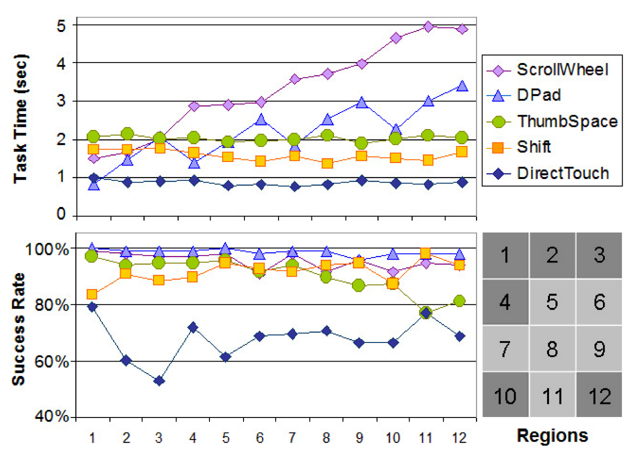


Figure 5. Study 1: task times and success rates by region (dark gray="hard to reach", light gray="easy to reach").

reflect their serial nature and starting position in region 1. Figure 5 again illustrates that Shift generally offered a speed advantage over ThumbSpace, but the two methods were more closely matched in topmost regions (1-3).

Error Rate

We performed a 5 (*Input*) x 2 (*Mobility*) x 12 (*Region*) RM-ANOVA on the average percent error data. Main effects of *Input* $F(1.4,14.9)=21.2$, $p<.001$, *Mobility* $F(1,11)=5.2$, $p=0.04$, and an interaction of *Input* x *Region* $F(44,484)=2.6$, $p<0.001$, were found. Post hoc tests revealed that with a 32% error rate, DirectTouch was significantly less accurate than any of the other input methods (Figure 4). DPad (1% error), on the other hand, was significantly more accurate than all other input methods. While no significant differences were found between the error rates of ThumbSpace (10%), Shift (8%) and ScrollWheel (5%), user comments suggested that the 8525 scrollwheel was susceptible to unwanted movement during presses, so 5% error may not be representative of scrollwheels in general.

As shown at the bottom of Figure 5, the error rates of Shift, ThumbSpace, and ScrollWheel were generally stable and comparable across regions. The few exceptions were the 8-20% higher rates for ThumbSpace in regions (9,11,12) and Shift's 14-16% higher rate in region 1. ThumbSpace averaged 3-14% lower error than Shift in regions (1-4).

Satisfaction

We performed a 5 (*Input*) x 2 (*Mobility*) x 13 (*Question*) RM-ANOVA on participant satisfaction ratings (1=low, 7=high satisfaction). A main effect of *Question* $F(12,120)=7.9$, $p<0.001$ and an interaction of *Input* x *Question* $F(48,480)=4.3$, $p<0.001$ were found. The absence of a main effect of *Input* together with presence of the *Input* x *Question* interaction suggests that the input methods varied in their perceived strengths. While overall scores for DPad, Shift, and ThumbSpace were fairly high (majority $\geq$ 5), their differences are corroborated by the data: Thumbspace was rated 1 point lower than the others for accessing "easy to reach" targets, while DPad was rated 1 point higher than the others on selection accuracy. Otherwise, ratings for DirectTouch and ScrollWheel were more mixed in terms of satisfaction than the other three.

We also asked participants to provide an absolute ranking of the five *Input* methods, plus a sixth option of using a Shift+ThumbSpace combination, from 1=Best to 6=Worst for the majority of device interaction. A chi-square revealed no reliable effect of preferred (rank of 1) input method $\chi^2(5,n=12)=7.0$, $p=0.22$. However, to account for the fact that the Shift+ThumbSpace input was not tested officially, we also performed a chi-square test on the number of participants who ranked each input technique either 1st or 2nd , which was significant $\chi^2(5,n=24)=11.0$, $p=0.05$. If methods were generally preferred equally, we would expect each method to be ranked 1st or 2nd by roughly one-third of participants. Instead 75% of participants ranked Shift+ThumbSpace either 1st or 2nd, while 0% of participants chose ScrollWheel 1st or 2nd. The remaining inputs four inputs were comparably preferred, each capturing the interest of 25-42% of participants.

Discussion

With respect to our broad question about the relative value of peripheral hardware vs. touchscreen interaction for mobile one-handed computing, we found little performance or subjective evidence that one approach should be recommended over the others. While DPad was relatively fast, and supported the most accurate selection, it was not preferred over any of the touchscreen inputs, and was the only method that generated unsolicited comments about hand fatigue. The fact that ScrollWheel was the least preferred input method is almost certainly related to the large number of targets presented in the study, which made average selection time slow. We should only conclude, therefore, that scrollwheel input is less suitable than other methods for dense 2D interfaces.

Of the touchscreen methods studied, it is clear that using the thumb directly (DirectTouch) to hit small targets is unacceptably error-prone, at least when considering use with standard resistive touchscreen technology. Yet the study data suggest that Shift and ThumbSpace are two software solutions that can greatly improve user accuracy in hitting small (3.6 mm^2) targets with the thumb. Although we did not find a statistical difference between Shift and ThumbSpace in terms of speed or error, trends in the data suggest that Shift has a speed advantage over ThumbSpace, especially in "easy to reach" regions—precisely those for which we assumed ThumbSpace would not be needed.

Given that users felt they would prefer using Shift *with* ThumbSpace over using either method exclusively suggests that users perceived each technique to have different advantages based on a target's location. Indeed, performance trends showed Shift and ThumbSpace had the most similar target access times in the top regions (1-3) of the screen, which are the same regions for which ThumbSpace averaged higher accuracy than Shift. ThumbSpace may have offered users increased relative advantage to using Shift in those regions because users found them relatively hard to reach, as shown by the region diagram in Figure 5. One reason we did not see a stronger effect may be that accessing those regions was less *comfortable* than accessing the others, but that they did not pose true *physical* challenge for the participants.

STUDY 2: THUMBSPACE VS. SHIFT

Because the results of our first study leave open several questions regarding the relative benefits of ThumbSpace and Shift, we conducted a second study with the goals of: 1) evaluating Shift and ThumbSpace on a "large" touchscreen and 2) understanding usage patterns when participants are given a real-time choice between the two.

Despite satisfying a range of hardware requirements for Study 1, the Cingular 8525's 2.8" (7.1 cm) screen is 20% shorter than many common devices that have 3.5" (8.9 cm) displays, including several models of the HP iPAQ (whose screens are 1.0 cm wider and 1.5 cm taller) and the iPhone (whose screen is 0.8 cm wider and 1.8 cm taller). To explore the question of whether ThumbSpace would provide an advantage over Shift in regions that are even farther than the top regions in Study 1, we wanted to reevaluate ThumbSpace and Shift over a larger screen.

If ThumbSpace outperforms Shift in far regions, maximizing overall selection efficiency would require users to make real-time choices between Shift and ThumbSpace. Unfortunately, the act of making such a choice adds a mental calculation to every action, which would tend to increase the attention demands of the task, as well as the overall selection time [7]. Thus to understand the practical viability of using Shift with ThumbSpace, we sought to answer the following question: Are users *willing* to trigger ThumbSpace given the choice? More precisely, do the comfort and stability offered by a personalized ThumbSpace sufficiently compensate for a longer selection time, or does the cost of making a decision result in users making no decision at all, instead using the technique that is less demanding on average for all targets (e.g., Shift)?

Independent Variables

Four input methods (*Input*) were studied: Shift, ThumbSpace, Shift with ThumbSpace (referred to as Combined), and a baseline of using the thumb to hit targets directly (DirectTouch). For consistency with the previous study, we evaluated only one target size (3.6 mm^2), with targets again arranged in a 6x8 grid of 48 positions (*Position*). Finally, to account for learning effects within each *Input*, tasks were repeated across two blocks (*Block*).

Implementation and Apparatus

The study was performed on an HP iPAQ 4155 (11.4 x 7.1 x 1.3 cm). Although the implementation of the Shift and ThumbSpace techniques were unchanged from Study 1, a mm/px constant was updated from 0.18 mm/px to 0.24 mm/px so that the absolute measurements of the Shift camera offset, the Shift camera diameter, and the target were the same between the Cingular 8525 and the iPAQ.

Tasks

Selection tasks for Study 2 were modeled after those of Study 1 with the following differences. Because of concerns in Study 1 that the yellow target was too similar to the yellow roads of the map, the trial target was shown in red. In addition, users started the trial timer by pressing the center of the DPad for all input methods. For Shift, users then aimed for the desired target. For ThumbSpace, users pressed the center button a second time to launch ThumbSpace. For the Combined *Input* condition, users decided for each trail whether to aim directly at the target using Shift or whether to press the center of the DPad a second time to launch ThumbSpace.

We chose to use the same start procedure for all input methods to: 1) standardize users' grips across all trials; and to 2) abstract away the cost of launching ThumbSpace (which will vary for different trigger mechanisms) by equalizing the movement burden across the input techniques. Results of the study should therefore be interpreted as having used the most optimistic trigger penalty of "a button press" for ThumbSpace.

Method

The study was a 4 (*Input:* DirectTouch, Shift, ThumbSpace, Combined) x 48 (*Position*) x 2 (*Block*) repeated measures within-subjects factorial design. Presentation of three *Inputs* (DirectTouch, Shift, ThumbSpace) were counterbalanced across participants, but the Combined input was always presented last to ensure participants had encountered both Shift and ThumbSpace prior to their combined use. For each *Input*, users performed 24 evenly distributed practice trials, followed by two blocks of randomized *Position* test trials. Prior to the Combined trials, users performed a "usability" phase in which they were given two minutes of self-directed exploration using Combined input for both a Calendar (Figure 2) and Windows Mobile Start Menu.

Dependent variables collected included task time, error rate, satisfaction ratings, and interface preference rankings. For the Combined input condition, the user input choice for each trial was also recorded.

Participants and Procedure

Twelve new right handed participants (5 male, 7 female), ages 18-29 ($\mu = 21$) were recruited from the general campus population. Only one participant studied a technical field. Because of the potential for hand size to bias users toward one input technique or another (e.g., users with small hands might benefit most from ThumbSpace), we aimed for an even distribution of hand size in our participant population. Participants' hands were classified into three broad categories S (n=3), M (n=5), and L (n=4), based on relative comparison (±1 cm) to the administrator's own hand (M), which fit a M/L women's glove.

The procedure for Study 2 was modeled directly after that of Study 1. For the Combined input condition, participants were urged to use whichever technique (Shift or ThumbSpace) they felt offered them the highest speed and accuracy for each trial. Total session time lasted between 45 and 75 minutes for which participants were paid $15.

Study 2 Results

Task Time

Task time was measured from the onset of the trial (when the center of the DPad was released) to the completion of the trial (when a user removed her finger from the screen). For each block, task time data were aggregated by *Position* into 12 *Regions* as in Study 1.

A 4 (*Input*) x 12 (*Region*) x 2 (*Block*) RM-ANOVA was carried out on the average task time data. Since there was no main effect of *Block*, it was removed from further analysis. Main effects of *Input* $F(3,33)=38.0$, $p<0.001$, and *Region* $F(6,65.9)=4.7$, $p<0.001$, were observed. In addition, an interaction between *Input* x *Region* $F(33,363)=1.8$, $p=0.006$, was also found.

Post-hoc comparison of task times by *Input* found DirectTouch was significantly faster (1,017 ms) on average than the other input methods. Although ThumbSpace was the slowest input method on average (1,993 ms), it was not found to differ significantly from Shift (1,753 ms) or Combined (1,724 ms). By *Region*, task times were highest in top regions (1-4) and lowest in central regions (5,7-9,11).

To make better sense of the trends in the *Input* x *Region* data, we felt it would be useful to aggregate the regions into "Far" and "Near" classes based on user opinion data. The 3x4 grid of Figure 6a depicts the 12 screen regions we analyzed. The color of each region conveys the majority user opinion about whether the region was "hard to reach" (dark gray) or "easy to reach" (light gray); region 10 did not have majority support for either classification. It is reasonable to conclude that the "hard to reach" distinction of the top regions (1-3) was because those regions were "Far" from the trial start point. Similarly, "easy to reach" regions were considered such because they were relatively "Near". Based on distance from the start point, we conclude that regions 10 and 12 are also both "Near", giving the categories Far (1-3) and Near (4-12).

We ran a 4 (*Input*) x 2 (*Distance:* Near, Far) RM-ANOVA on the aggregated region data. In addition to the main effect of *Input* reported above, the analysis revealed a main effect of *Distance* $F(1,11)=12.5$, $p=0.005$ and an interaction of *Input* x *Distance* $F(3,33)=3.4$, $p=0.03$. Post-hoc comparison of task times by *Distance* confirmed that users took significantly longer performing tasks in Far regions (1,774 ms) than they did in Near regions (1,570 ms). Considering the *Input* x *Distance* interaction data, Figure 6b suggests that Shift was more sensitive to differences in task distance than the other three input methods. Planned comparisons of the *Input* x *Distance* data using Shift as the reference input revealed that Shift/Far was significantly slower than Combined/Far (2,037 ms vs. 1,816 ms, $p=0.018$) but that Shift/Near was significantly faster than ThumbSpace/Near (1,659 ms vs. 1,958 ms, $p=0.013$).

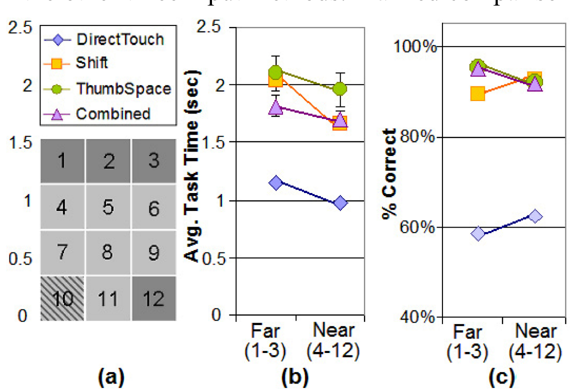


Figure 6. Study 2: (a) device regions, where dark gray = "hard to reach" and light gray = "easy to reach"; (b) average task times for distances 'Far' (1-3) and "Near" (4-12); (c) average percent correct by distance type.

Shift was likely faster than ThumbSpace in Near regions due to the occlusion problems that can occur when using ThumbSpace for targets that overlap the user-defined ThumbSpace. However, because Shift/Near did not differ significantly from Combined/Near, users apparently made appropriate decisions when choosing between Shift and ThumbSpace for Near targets. It is somewhat surprising that Combined/Far was significantly faster than Shift/Far, which was faster than ThumbSpace/Far, since participants were restricted to using either Shift or ThumbSpace in the Combined condition. One explanation is that users became more efficient with the techniques over the course of the study, and that this learning effect benefited Combined disproportionately since it was always presented last. This could be true even though our block design did not reveal learning effects for time or error rate *within* input types.

Error Rate

A 4 (*Input*) x 12 (*Region*) RM-ANOVA was performed for average input error data. A main effect of *Input* $F(1.4,15.2)=40.5$, $p<0.001$ and an interaction between *Input* x *Region* $F(33,363)=2.3$, $p<0.001$ were observed. Despite the speed advantage offered by DirectTouch, users were significantly less accurate using DirectTouch (39% error) than the other three input methods, which had between 7.4-8.4% error and did not differ significantly from each other.

We again ran a 4 (*Input*) x 2 (*Distance:* Near, Far) RM-ANOVA on the aggregated region data to better understand the *Input* x *Region* error trends for "Near" and "Far" regions. As reported above, a main effect of *Input* was found, as well as an interaction of *Input* x *Distance* $F(3,33)=4.5$, $p=0.009$.

As seen in Figure 6c, participants tended to be more accurate using Shift and DirectTouch in Near regions than in Far regions, but the opposite trend was found for ThumbSpace and Combined. Planned comparisons of the *Input* x *Distance* data using Shift as the reference input revealed that Shift/Far was significantly less accurate than ThumbSpace/Far ($p=0.013$) and Combined/Far ($p=0.027$). For Near targets, Shift, ThumbSpace and Combined did not differ significantly from one another. Thus for both distances, Combined supported equivalent or better performance results as Shift and ThumbSpace did individually.

Input Choice

To understand user strategies for using Shift and ThumbSpace in the Combined condition, we looked at the frequencies with which participants chose Shift vs. ThumbSpace by *Region* for Combined input. Figure 7 shows that ThumbSpace was chosen more often than Shift

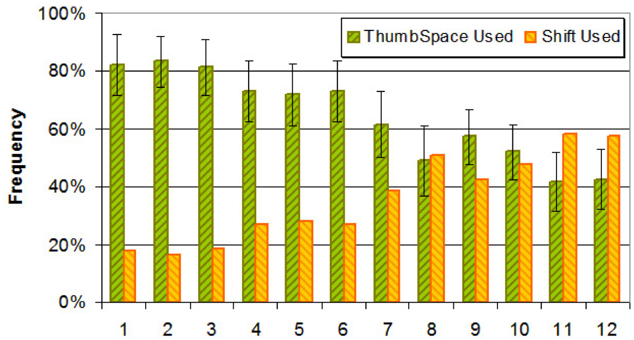


Figure 7. ThumbSpace vs. Shift usage for Combined input.

for regions in the top half of the device (1-6), and that Shift was used increasingly toward the bottom of the device.

The fact that participants nearly always used ThumbSpace for Combined input in Far regions (1-3) explains why the error rates of ThumbSpace and Combined matched one another closely in Far regions (Figure 6c). Again, given that Combined enjoyed faster access times over ThumbSpace in Far regions, even though users chose ThumbSpace the majority of the time, suggests a learning effect.

The usage patterns of Shift and ThumbSpace for the Combined condition are intriguing. Users modified their input strategies based on the device region (Figure 7), and the choices users made were "good" with respect to both time and errors (Figure 6). Moreover, the benefit of allowing users to make a real-time choice for *Input* type according to the *Region* of access outweighed any time cost associated with the decision.

Satisfaction

A 4 (*Input*) x 13 (*Question*) RM-ANOVA was performed on participant satisfaction ratings, selected from a 7 point scale, (1=low, 7=high satisfaction). Main effects of *Input* $F(3,33)=11.9$, $p<0.001$ and *Question* $F(12,132)=3.9$, $p<0.0001$ as well as an interaction between *Input* x *Question* $F(36,396)=3.9$, $p<0.001$ were found.

Post hoc tests found DirectTouch received significantly lower satisfaction scores on average than ThumbSpace and Combined (4.6 vs. 5.9 and 6.2), while Shift (5.5) was not found statistically more or less satisfying than the others. It is unsurprising that average scores would differ across *Question*, but in fact most were rated relatively high on average (≥ 5.1), especially learnability of the techniques (6.5), which rated significantly higher on average than nearly all the other measures. Overall, Shift, ThumbSpace and Combined were rated similarly to one another across satisfaction measures, except for Comfort and Stability, for which Shift rated about 1 point lower than the other two.

We asked participants to provide an absolute ranking of the four input methods from 1=Best to 4=Worst. A chi-square test on preferred input method (rank of 1) was significant $\chi^2(3,n=12)=18.0$, $p<0.001$. Seventy-five percent of the participants chose Combined as their preferred method, while the remaining 25% chose ThumbSpace.

Discussion

In switching from a smaller to a larger device between Study 1 and Study 2, we observed that device size affects both the utility and perception of ThumbSpace. With smaller screens, the main problem users face is not that of thumb reach, but hitting small targets accurately with a finger, for which the Shift technique is well suited. By moving to a larger device with a 3.5" screen in Study 2 and explicitly testing the use of ThumbSpace and Shift both individually and together, we confirmed that: 1) Shift and ThumbSpace traded advantages according the position of the targets; 2) users developed an intuition for the relative benefits of each, choosing Shift for its speed in near regions and ThumbSpace for its accuracy in far regions; 3) the act of choosing between Shift and ThumbSpace had no noticeable impact on selection speed, and 4) users preferred having a choice to using either method alone.

To understand the practical implications of Study 2, we must examine the parameters under which the results were obtained. First, we studied thumb access to small (3.6 mm^2) touchscreen targets only, since they are common to traditional interfaces and present significant challenges for finger interaction. For interfaces with larger targets, Shift delays the onset of its callout in proportion to target size so that users do not experience a visual disruption if objects are large enough to hit easily with a finger (e.g., ≤ 10 mm^2). Yet because the cost of launching ThumbSpace is constant, users may become less willing to use ThumbSpace if their success rate increases when "stabbing for" large but distant targets—even if the action requires a grip adjustment or momentary device instability. Furthermore, the effective physical cost of triggering ThumbSpace in Study 2 was a button press, and so the results reflect an idealized design that may be difficult to replicate in practice.

Even with these considerations, the Study 2 results make a compelling case for touchscreen devices to include Shift as a default interaction technique to support finger use in general, and ThumbSpace for aiding in one-handed operation. The ability to personalize ThumbSpace's size, position, and use occasion means it can flexibly and unobtrusively accommodate variations in hand size, interface design, and device dimension. That both Shift and ThumbSpace were considered fast, accurate, easy to learn and easy to use make them a very promising combination for generalized one-handed touchscreen operation.

CONCLUSION

We performed two user studies to evaluate the relative benefits of peripheral hardware and software-based interaction techniques for supporting one-handed use of touchscreen-based mobile devices. Although we found no single approach that dominated in terms of both preference and performance, it may not be reasonable to expect a single strategy to fit all legacy applications; our studies focused on a dense 2D layout, where touchscreen-based methods had an edge. But for sparse or 1D layouts, peripheral devices may be better suited. Indeed Blackberry devices are a great example of how thoughtful, complementary software designs can render scrollwheels effective and enjoyable. But for legacy applications, one generally does not have the luxury of redesigning the underlying software, thus interface-independent solutions such as ThumbSpace or Shift may prove advantageous.

An interesting finding from both of our studies was that participants preferred the combination of Shift and ThumbSpace to using either method exclusively. Users quickly and effectively modified their input strategy between these two methods to match the input task: using Shift for near targets and ThumbSpace for far targets. These findings motivate and lend credence to a general approach of composing multiple input methods, each of which is tuned to a different type of task. Perhaps, for instance, peripheral hardware and software-based methods could be used for sparse and dense layouts, respectively. Hopefully our work will inspire further research in composing input methods, where answers to questions such as “how do the attention demands of realistic mobile scenarios affect user ability to combine interaction methods?” will be crucial for assessing the pragmatic viability of input composition.

REFERENCES

1. Aliakseyeu, D., Subramanian, S., Gutwin, C. and Nacenta, M. Bubble radar: efficient pen-based interaction. *Proc. AVI 2006*, ACM Press (2006), 19-26.
2. Asano, T., Sharlin, E., Kitamura, Y., et al. Predictive interaction using the delphian desktop. In *Proc. UIST 2005*, ACM Press (2005), 133-141.
3. Baudisch, P., Cutrell, E., Robbins, D. C., et al. Drag-and-Pop and Drag-and-Pick: Techniques for Accessing Remote Screen Content on Touch- and Pen-operated Systems. In *Proc. INTERACT 2003* (2003), 57-64.
4. Bederson, B. B., Meyer, J. and Good, L. Jazz: An Extensible Zoomable User Interface Graphics Toolkit in Java. In *Proc. UIST 2000)*, ACM Press (2000), 171-180.
5. Bezerianos, A. and Balakrishnan, R. The vacuum: facilitating the manipulation of distant objects. In *Proc. CHI 2005*, ACM Press (2005), 361-370.
6. Blanch, R., Guiard, Y. and Beaudouin-Lafon, M. Semantic pointing: improving target acquisition with control-display ratio adaptation. In *Proc.CHI 2004*, ACM Press (2004), 519-526.
7. Card, S. K., Moran, T. P. and Newell, A. The keystroke-level model for user performance time with interactive systems. *Comm. of the ACM 23*(1980), 396-410.
8. Grossman, T. and Balakrishnan, R. The bubble cursor: enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proc. CHI 2005*, ACM Press (2005), 281-290.
9. Guiard, Y., Blanch, R. and Beaudouin-Lafon, M. Object pointing: a complement to bitmap pointing in GUIs. In *Proc. GI 2004*, Canadian Human-Computer Communications Society (2004), 9-16.
10. Hascoët, M. Throwing models for large displays. In *Proc. HCI 2003*, British HCI Group (2003), 73-77.
11. Kabbash, P. and Buxton, W. The "Prince" technique: Fitts' Law and selection using area cursors. In *Proc. CHI 1995)*, ACM Press (1995), 273-279.
12. Karlson, A. K., Bederson, B. B. and SanGiovanni, J. AppLens and LaunchTile: two designs for one-handed thumb use on small devices. In *Proc. CHI 2005*, ACM Press (2005), 201-210.
13. Karlson, A. K. and Bederson, B. B. ThumbSpace: generalized one-handed input for touchscreen-based mobile devices. In *Proc. INTERACT 2007*, Springer (2007), 324-338.
14. Karlson, A. K., Bederson, B. B. and Contreras-Vidal, J. L. Understanding One Handed Use of Mobile Devices. In Lumsden, J. ed. *Handbook of Research on User Interface Design and Evaluation for Mobile Technology*, Information Science Reference, 2008.
15. Kristoffersen, S. and Ljungberg, F. “Making place” to make IT work. In *Proc. SIGGROUP 1999*, ACM Press (1999), 276-285.
16. Nacenta, M. A., Aliakseyeu, D., Subramanian, S. and Gutwin, C. A comparison of techniques for multi-display reaching. In *Proc. CHI 2005*, ACM Press (2005), 371-380.
17. Oulasvirta, A., Tamminen, S., Roto, V. and Kuorelahti, J. Interaction in 4-sec. bursts: the fragmented nature of attentional resources in mobile HCI. In *Proc. CHI 2005*, ACM Press (2005), 919-928.
18. Parhi, P., Karlson, A. K. and Bederson, B. B. Target Size Study for One-Handed Thumb Use on Small Touchscreen Devices. In *Proc. MobileHCI 2006*, ACM Press (2006), 203-210.
19. Pirhonen, P., Brewster, S. A. and Holguin, C. Gestural and audio metaphors as a means of control in mobile devices. *Proc. CHI 2002*, ACM Press (2002), 291-298.
20. Potter, R. L., Weldon, L. J. and Shneiderman, B. Improving the accuracy of touch screens: an experimental evaluation of three strategies. In *Proc. CHI 1988*, ACM Press (1988), 27-32.
21. Sears, A. and Shneiderman, B. High-precision touchscreens: design strategies and comparisons with a mouse. *Int. J. Man-Mach. Stud. 34*, 4 (1991), 593-613.
22. Vogel, D. and Baudisch, P. Shift: a technique for operating pen-based interfaces using touch. In *Proc. CHI 2007*, ACM Press (2007), 657-666.
23. Wigdor, D. and Balakrishnan, R. TiltText: using tilt for text input to mobile phones. In *Proc. UIST 2003*, ACM Press (2003), 81-90.