Design of Unimanual Multi-Finger Pie Menu Interaction

Nikola Banovic[1]*, Frank Chun Yat Li*[1]*, David Dearman*[1, 2]*, Koji Yatani*[1]*, and Khai N. Truong*[1]

[1]Department of Computer Science
University of Toronto
Toronto, ON M5S 3G4, Canada
{nikola, frankli, dearman, koji, khai}@dgp.toronto.edu

[2]Nokia Research Center
955 Page Mill Rd.
Palo Alto, CA 94304, USA
david.dearman@nokia.com

ABSTRACT

Context menus, most commonly the right click menu, are a traditional method of interaction when using a keyboard and mouse. Context menus make a subset of commands in the application quickly available to the user. However, on tabletop touchscreen computers, context menus have all but disappeared. In this paper, we investigate how to design context menus for efficient unimanual multi-touch use. We investigate the limitations of the arm, wrist, and fingers and how it relates to human performance of multi-targets selection tasks on multi-touch surface. We show that selecting targets with multiple fingers simultaneously improves the performance of target selection compared to traditional single finger selection, but also increases errors. Informed by these results, we present our own context menu design for horizontal tabletop surfaces.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

General terms: Design, Human Factors

Keywords: Multi-touch, menu selection, unimanual interaction.

INTRODUCTION

On traditional mouse-based interfaces, context menus provide a convenient way of accessing different commands without needing to move the cursor away from the area of focus. Context menus (*e.g.,* right click menus) allow users to access commands that are otherwise located in a faraway menu structure that is usually located on the edges of the interface. However, on tabletop touch-based interfaces, context menus have all but disappeared.

One of the main challenges with implementing a context menu on tabletop touch-based interfaces lies in how to distinguish input and menu invocation [22]. One possible solution is using gestures for invocation [15]; however, even gestures could interfere with regular input and require specific registration phase [36]. Recent advances in hand tracking [1, 29] and finger recognition [2, 21] enable differentiation between the input and menu modalities. For example, context menus for tabletop systems can be supported by tracking the user's hand as if it was a cursor and then requiring the user to select a nearby button (which always trails the user's hand) to activate the menu. Or the user could invoke a context menu by performing a cording gesture [21]. Unfortunately, hand tracking is not yet supported on most hardware, and chording can be difficult for the user to perform because the number of fingers used also increases the cognitive preparation time [21].

Traditionally, to select an item from a context menu is much simpler than that and requires two simple actions – menu invocation and menu item selection. These steps can be treated as a multi-target selection task (*i.e.,* the selection of the menu activation button and the target menu item). Translated directly onto touch surfaces, the user can perform this multi-target selection task using multiple taps (a familiar interaction technique carried over from the traditional mouse input and pen input paradigms), which could require, for example, the user to select the primary and secondary targets serially using her index finger.

Alternatively, selecting multiple targets in parallel can be designed to increase input bandwidth in menu selection [2]. For example, the user can select the menu invocation button with her index finger and immediately use another finger from the same hand to select the desired menu item sequentially. Alternatively, the user could use the two fingers to select the targets simultaneously (*i.e.,* at the same time). However, there is a lack of knowledge to guide the design of such single-handed multi-target selection techniques.

In this paper, we investigate the feasibility of a context menu which is invoked using multiple fingers on a single hand. We study the human performance of the three one-handed techniques for multi-target selection tasks: multiple taps with a single finger, and sequential and simultaneous selection with multiple fingers. More specifically, we first study the anthropometric limitations of the human arm, hand and fingers to scope the experimental design. We then measure user performance of these one-handed techniques for selecting two targets in order to understand applicability of using multiple fingers for such multi-target selection. The results from our study show that the index finger and thumb can both be used to invoke context menus using multiple fingers, but that each finger has different limitations that impact how the context menus should be designed. At the expense of accuracy, users can perform multi-touch target

ITS 2011, November 13-16, Kobe, Japan.

selection with multiple fingers simultaneously faster than performing multiple taps with a single finger. Informed by our results, we present a multi-touch context menu as a partial pie menu that takes advantage of multiple finger selection.

RELATED WORK

We designed a tabletop context menu based on the results of our unimanual multi-finger target selection study. In this section, we review previous work that examines unimanual multi-finger interactions and menu designs on touch screens.

Unimanual Multi-Finger Interactions

Because single-finger input is the most basic form of interactions in tabletop surfaces, its user performance on target selection has been studied [12, 31]. However, multi-touch input can increase the input bandwidth in two ways: unimanual (contacts by multiple fingers with one hand) and bimanual (contacts by one or multiple fingers with two hands). However, unimanual and bimanual multi-touch input mappings are not necessarily interchangeable [25] meaning that interfaces which assume bimanual multi-touch input may not be appropriate for unimanual input, and vice versa. One of our main contributions is an understanding of user performance on unimanual multi-touch menu selection instead of bimanual selection which has been investigated extensively [6, 18, 26].

Research on unimanual multi-touch interactions have investigated how to bring interactions on a mouse to touchscreens. In Fluid DTMouse [10], when two fingers are placed on the screen (*e.g.*, the thumb and middle finger), the system enters a mouse mode. The user can move the cursor by moving the two fingers and perform a right click by tapping the screen with another finger (*e.g.*, the index finger). Matejka *et al.* [23] further explored the design of unimanual multi-touch mouse emulation techniques. Their study found that intuitive and fast is the mapping of the right, middle, and left click to the tapping with the thumb, middle finger, and ring finger, respectively, with the index finger placed on the screen. Bartindale *et al*. [5] designed a special widget to support mouse operation in touchscreens which liberates the user from chording input used by Matejka *et al.* [23]. These projects highlight that tapping with placing one or two fingers on the screen is intuitive and often fast. This motivated us to extend the idea to unimanual menu selection.

Use of various finger combinations has been explored in unimanual multi-touch selection. Past research showed that finger recognition can be implemented on existing hardware based on the position of the palm [2, 3], and that it can be used to invoke different menu items depending on the finger touching the surface. The user can invoke a command associated with each menu by simply tapping it with a finger. Lepinski *et al*. [21] developed marking menus by the combination of the finger chording and gesture direction. Their user study showed that their technique is faster than a traditional single-input marking menu. Bailly *et al*. [4] also used the number of the fingers placed on the screen to specify an item in a menu. These interfaces use the number or combinations (chording) of the fingers on the screen to specify the desired menu item. Unfortunately, chording with multiple fingers becomes difficult as the number of fingers used increases [21]. Thus, we explore menu designs which take advantage of chording input, but reduce the number of required fingers.

Menu Designs on Touch Screens

Menu designs have been deeply explored in the past few decades. To reduce the selection time, researchers have explored the idea of moving the menu closer to the user's area of interest. Toolglasses and magic lenses [7], and popup context menus are example of such menu systems. Another way to improve user experience in menu selection is to design a more efficient menu structure than a linear list. Pie menus [17] and marking menus [19] are known to support fast menu selection. Research has shown that additional improvements can further decrease the selection performance time without sacrificing the accuracy (for example, [20, 37] in marking menus). However, these improvements assume single-point input, and existing menu selection techniques may not exploit the capability of multi-touch input well.

Our literature survey shows that unimanual multi-touch menu selection is still under-explored, particularly, the understanding of the human capability for unimanual multi-touch interaction. Our contributions include an examination of user performance of unimanual multi-touch target selection as well as a novel menu design for multi-touch surfaces.

SELECTION TECHNIQUES

In this work, we explore one-handed multi-target acquisition using one or two fingers. We focus on two-target selection in order to test the feasibility of using this interaction to implement a context menu (*i.e.*, touching a menu button to invoke the menu and selecting a menu item). We refer to a menu button and menu item more generally as the primary and secondary target, respectively. We investigated the following one-handed multi-target selection techniques using one or two fingers:

- *Multi-tap*: Use the index finger to select the primary and secondary targets in serial. We include this traditional interaction technique as a baseline to compare the other techniques against.
- *Sequential*: Use the thumb or index finger to select the primary target followed by another finger to select secondary target (7 finger combinations in total, thumb with index, middle, ring, or pinky; and index with middle, ring, or pinky).
- *Simultaneous*: Use the thumb or index finger to select the primary target and another finger to select secondary target at the same time (7 finger combinations in total).

In the later sections, we refer to the finger used to select the primary target as the anchor finger. Furthermore, *thumb-*

anchored represents the selection of the primary target with the thumb finger (likewise with *index-anchored*).

EXPERIMENT

We conducted a study to evaluate the user performance of the three multi-target selection techniques. We focused on how different locations of the secondary targets affect selection performance in the two multi-touch techniques.

Apparatus

The experiment was conducted using a Microsoft Surface. The Surface was positioned flat and was raised to 101 cm high; this fits within the range of what previous research has reported as a reasonable height for people to interact comfortably with a workspace while standing [26]. A secondary monitor was placed next to the Surface to display trial conditions and progress information. The participants were instructed to stand in a marked 50 × 50cm area in front of the Surface to prevent the participants from moving around the Surface or greatly changing their body stance between conditions. Participants also had to cover the fingers which were not used in the trials with dark green rubber finger cots. These finger cots prevented the Surface from recognizing accidental touches by fingers different from the one we were measuring in a particular block. We did this to ensure that we were able to analyze specific combinations of fingers; these cots are not necessary in actual use settings.

Selectable Area

Although *sequential* and *simultaneous* multi-target selection may offer increased input bandwidth in menu selection [2], they come at the expense of limited area in which two fingers can touch different spots at the same time. This area is restricted due to the short span between fingers and due to the limited rotation of the wrist. To inform the design of our partial pie menu, we conducted an informal pilot study to identify and eliminate secondary target locations that users would have strong discomfort selecting due to these limitations.

Pilot Procedure. Participants in this pilot had to select targets from a full circle pie menu. We examined what radius would enable participants to at least expand and touch the end of the circle with their index and pinky at the same time. We asked participants to touch the surface with combinations of thumb and index fingers and other fingers in the most extreme areas that were allowed with their finger span and wrist rotation.

Pilot Results. Figure 2 shows the areas that all participants were able to reach without strong discomfort. Our observations showed that most participants were able to touch areas with different combinations of fingers within the 100mm radius, with *thumb-anchored* selection allowing for selection even outside of this radius. Also, most participants were able to select targets with both anchor fingers between 0° and 90° of wrist rotation. Based on these results, we developed a partial pie menu (Figure 3) that has nine selectable arcs (3 rings and 3 wedges) and used it in our study. This allowed for a large enough primary target area and the secondary targets in the smallest ring (Ring 0) that can be accurately selected with a thumb and index finger [16].

Approach Directions

Unlike mouse-based interactions, approach directions to the target may impact selection performance in unimanual finger techniques. For instance, selection with approaching from the bottom side might be faster than one with approaching from the top side because it is less likely that the primary and secondary targets are occluded by the hand during the selection. We, thus, ran a second pilot study to understand the effects of approach directions. In this second pilot, we examined eight approach directions (we used the compass notation: N, NE, E, SE, S, SW, W, NW) with the three selection techniques (see Figure 4). For example, E meant that the participant would approach the primary target from the east side (from right to left).

Pilot Procedure. Each trial required the participant to select the start button with the palm of their hand first, and then select the primary and secondary targets using one of the three selection techniques (Figure 4). The secondary targets in this pilot were the three arcs on Wedge 1. The participants were instructed which fingers and which technique to use for acquiring the two targets before starting a trial. However, they were always asked to touch the start button with their palm. In this manner, we prevented the participants from assuming a particular hand posture which could bias performance. The system asked the participants to repeat the same trial when they failed to acquire any of the targets correctly. At the start of the study, participants completed a practice block of 50 trials for each technique. This pilot took around 90 minutes to complete.

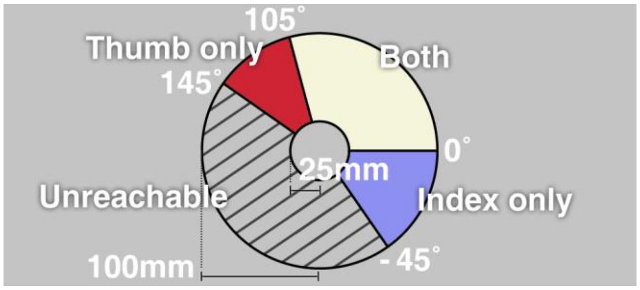

Figure 2. Selectable areas based on hand span and wrist rotation by anchor finger.

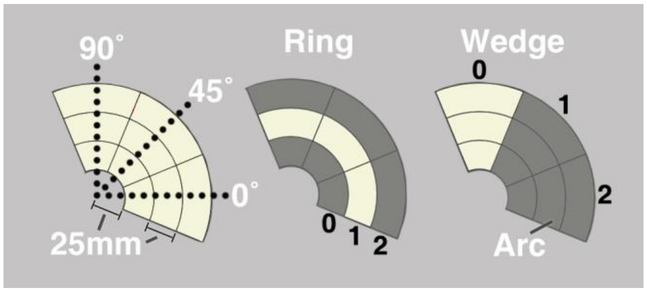

Figure 3. Orientation and components of the pie-menu used in the study.

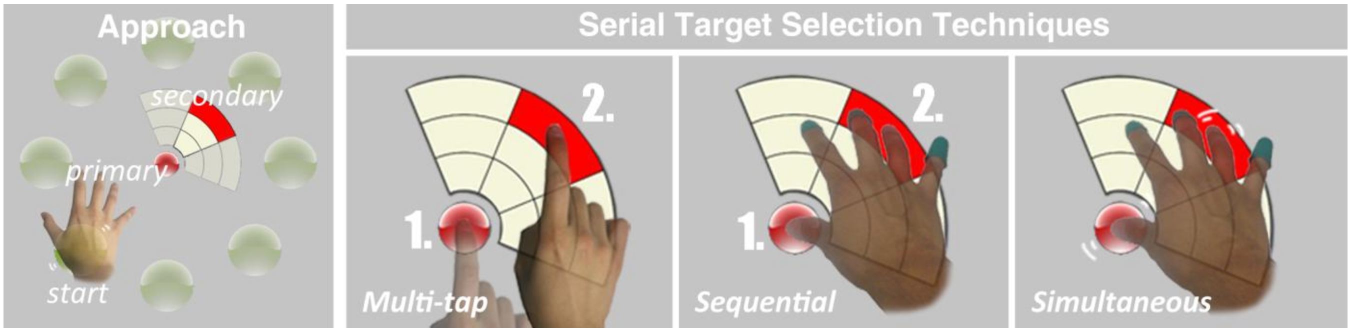

Figure 4. Study interface: 8 approach directions (N, NE, E, SE, S, SW, W, NW) from the start button to the primary target, and three different serial target selection techniques: *Multi-tap*, *Sequential*, and *Simultaneous*.

Pilot Results. Our analysis indicates that the task completion time was affected by the approach direction: the *E*, *SE*, and *S* group was the fastest and *NW* the slowest for each of the three techniques. We did not see significant differences in the number of errors across the eight directions due to the large variances. Based on these results, we decided to use a fastest and the slowest direction, S and NW, in our user study.

Setup

The study interface (Figure 4) includes a start button (a circular button 78mm in diameter) that appears at S and NW directions centered on the primary target (a circular button 36mm in diameter). The primary target was placed in the center of the display. The distance between the start button and the primary target was fixed to 180mm. The secondary target was one of nine arcs in the partial pie menu. For the rest of the paper, we refer to the rings and wedges as Ring 0, 1, and 2 (innermost to outermost) and Wedge 0, 1, and 2 (clockwise). The thickness of each arc was 25mm. The start button and the targets were color-coded; red, green, and grey meant disabled, enabled, and already selected, respectively. Audio feedback was given after every selection.

Participants

Twelve participants (7 female and 5 male) with a median age of 29 (min=21, max=48) took part in the study. Participants were recruited through an online classified ad. All participants were right-handed and reported no motor impairments. Participants' mean measures include: height of 170.33cm (SD=9.55), hand span of 19.83cm (SD=1.65), hand breath of 8.75cm (SD=0.78), hand length of 18.08cm (SD=0.97), upper limb length of 73.54cm (SD=3.49), and elbow length of 43.21cm (SD=2.90). Ten participants had previous experience with touch screen devices. We compensated each participant $45.

Study Procedure

In this study, we looked at the fastest and slowest direction found from the pilot, S and NW, and expanded to all nine arcs. Participants had to complete four repetitions for each task condition and two blocks of all conditions; thus, 2 *Directions* × 9 *Arcs* × 15 *Finger Combinations* (1 for *Multi-tap*, 7 for *Sequential*, 7 for *Simultaneous*) × 4 (repetition) × 2 (blocks) = 2160 trials for each participant. The order of presentation for techniques within a block was counter-balanced, and the order of *Direction*, *Arc* and *Finger Combination* were randomized across participants. Aside from these differences, the procedure was the same as the approach direction pilot. We recruited the same participants from the approach direction pilot study for this study. This study took around 60 minutes to complete.

We measured the following:

- *ApproachTime*: The time between when the participant released her palm from the start button and when she touched the primary target.
- *2ndTargetTime*: The time between when the primary target was touched and when the secondary target was touched.
- *TotalTime*: The sum of *ApproachTime* and *2ndTargetTime*.
- *ApproachError*: The number of cases in which the participants failed to correctly acquire the primary target.
- *2ndTargetError*: The number of cases in which the participants selected the primary target, but failed to correctly acquire the secondary target.
- *SelectionError:* The sum of *ApproachError* and *2ndTargetError.*

RESULTS AND DISCUSSION

Trials with selection time greater than 2SD from the mean were removed as outliers; ~1.6% of trials were removed. To account for the variability in human performance, we used the mean selection time for each participant when performing the analysis. Unless explicitly described, the main effect for selection time was analyzed with repeated-measure ANOVAs using the Greenhouse-Geisser correction when sphericity was violated, and post-hoc pairwise comparisons were conducted using paired t-tests. Event-count measures such as errors were analyzed with nonparametric Friedman tests and post-hoc pairwise comparisons were conducted with the Wilcoxon test. We report the effect size (*r*) for pairwise comparison conducted with the Wilcoxon tests. All post-hoc pairwise comparisons used the Holm's sequential Bonferroni correction.

Selection Speed

Our results showed a significant effect of the techniques on *TotalTime* ($F_{(2,22)}$=19.23, p<.001, η_p^2=0.64). *Simultaneous* on average performed 127ms faster than *Multi-tap* (p<.001) and 97ms faster than *Sequential* (p<.01) (Figure 5). Figure 6 shows the average *ApproachTime* and *2ndTargetTime* for each technique. *Sequential* and *Simultaneous* were significantly slower than *Multi-tap* in *ApproachTime* ($F_{(2,22)}$=48.51, p<.001, η_p^2=0.82). *Multi-tap* was on average 49ms and 77ms faster in *ApproachTime* than *Sequential* (p<.001) and *Simultaneous* (p<.001), respectively, while *Simultaneous* was on average 28ms faster than *Sequential* (p<.001). A paired t-test on *2ndTargetTime* revealed that *Sequential* was on average 79ms faster than *Multi-tap* ($t_{(11)}$=6.44, p<.001, Cohen's d=1.86). We did not analyze *Simultaneous* for *2ndTargetTime* because the secondary target selection happened concurrently with the primary target selection.

Discussion. The results imply that in *Simultaneous*, participants tended to adjust the posture of their hand during the approach to the primary target (*ApproachTime* for *Simultaneous* is significantly slower than for *Multi-tap*). Once the hand was postured during the approach, it was used to select both the primary and secondary target. Participants also postured their hands during the approach in *Sequential*. However, additional posturing and movement was required after the primary selection for the secondary digits to select the secondary target. Thus, the time to select the primary target was closer to that of *Multi-tap*, while the time for selecting the secondary target was faster than that of *Multi-tap*.

Selection Error

As shown in Figure 7, the multi-touch selection techniques resulted in more errors than *Multi-tap* ($\chi^2_{(2)}$=12.17, p<.01). *Multi-tap* was significantly more accurate than *Simultaneous* by 8.05% (p<.05, r=0.82) but not significantly more accurate than *Sequential*. We again analyzed the breakdowns of errors (Figure 8). *ApproachError* was significantly affected by *Technique* ($\chi^2_{(2)}$=24, p<.001), but our test did not find a significant effect of *Technique* on *2ndTargetError*. *Multi-tap* resulted in no *ApproachError* because of the simplicity and serial nature of the technique, and had a lower error rate than *Sequential* at about 2% and *Simultaneous* at 8%, which was more error-prone than the other two (all p<.01, r=0.88).

Discussion. Sequential and *Simultaneous* was significantly more error-prone than *Multi-tap*. We speculate the reason is that multi-touch techniques required specific hand postures which can be hard to make and control. Thus, some of the errors could have been motor errors (where the user performed the right intention inaccurately). Additionally, the rubber cots prevented the system from recognizing touches by the wrong fingers. When using the system without cots, it is likely that slips (where the user performed the right intention with the wrong finger) will occur. Allowing any finger to select the secondary target could minimize the impact of such slips. Furthermore, our

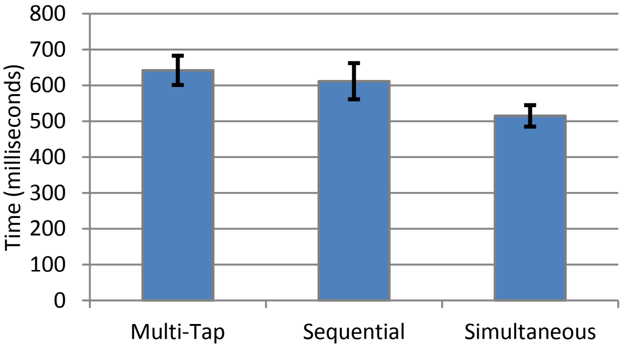

Figure 5. Average *TotalTime* by *Technique*. In this and all later charts, error bars represent 95% confidence intervals.

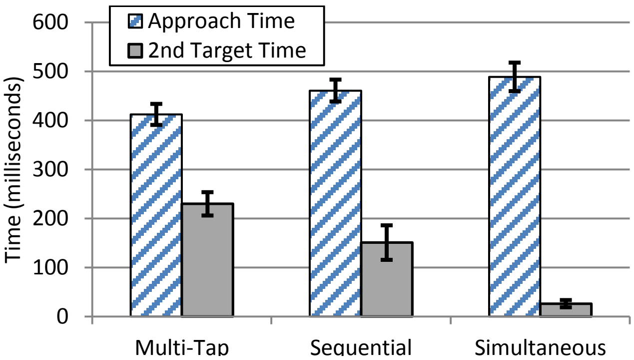

Figure 6. Average *ApproachTime* and *2ndTargetTime* by Technique.

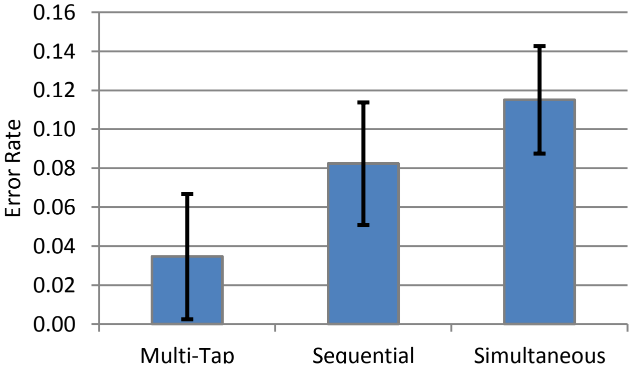

Figure 7. Median *SelectionError* by *Technique*. Error bars denote 95% confidence interval.

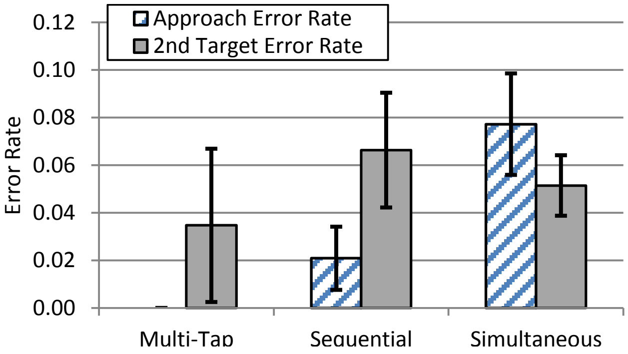

Figure 8. Median *ApproachError* and *2ndTargetError* by *Technique*.

results indicate that there is a trade-off between speed and accuracy in the three selection techniques. It is not possible to distinguish the specific reasons why *Sequential* and *Simultaneous* resulted in more errors than *Multi-tap*. Thus, we decided to analyze the effects of anchor fingers on multi-target selection.

Effects of Anchor Finger

Errors in *Sequential* and *Simultaneous* may be attributed to some specific combinations of the fingers. We, therefore, examined how different combinations of the fingers affected selection performance. We begin with examining the effects of the anchor finger. A paired t-test did not show a significant difference of the anchor finger in *TotalTime*, but a Wilcoxon test showed a difference in *SelectionError* (W=11, Z=2.20, p<.05, r=0.63, for *Sequential*, and W=9, Z=2.35, p<.05, r=0.68, for *Simultaneous*) with *thumb-anchored* resulting in more errors. We observed that most of the selection errors with *thumb-anchored* were caused by the underlying system's inability to successfully detect the participant's thumb due to the rotation of the finger. We explore this further in section on *thumb-anchored* selection.

Index-anchored Target Selection. The wedges influenced the speed and accuracy of acquiring the targets in the *index-anchored* condition in both *Sequential* ($F_{(2,22)}$=36.13, p<.001, η_p^2=0.77, and $\chi^2_{(2)}$=20.67, p<.001) and *Simultaneous* ($F_{(1.25,13.71)}$=77.57, p<.001, η_p^2=0.88, and $\chi^2_{(2)}$=18.17, p<.001). Wedge 0 was the slowest (all p<.001) and most error-prone region (p<.05, r=0.66 for Wedge 1 and p<.01, r=0.88 for Wedge 2) in both *Sequential* and *Simultaneous* (Figures 9 and 10).

Our results also showed a significant effect of the rings on *TotalTime* in both *Sequential* ($F_{(1.15,12.70)}$=33.00, p<.01, η_p^2=0.75) and *Simultaneous* ($F_{(1.19,13.12)}$=8.24, p<.05, η_p^2=0.43). The results indicate that closer rings were faster to acquire than the furthest ring (both p<.001; Figure 10). Similarly, the results showed a significant effect (Figure 11) on *SelectionError* for *Sequential* ($\chi^2_{(2)}$=14.09, p<.001) and *Simultaneous* ($\chi^2(2)$=10.50, p<.01): the closer targets were generally less error-prone to acquire than further targets in Ring 2 in both techniques (all p<0.05).

Thumb-anchored Target Selection. We found a significant effect of the wedges in *TotalTime* only for *Simultaneous* ($F_{(1.35,14.86)}$=9.54, p<.01, η_p^2=0.47), where Wedge 1 was analyzed to be faster than Wedge 0 (p<.001) and Wedge 2 (p<.05). Our tests did not find a significant effect of the wedges on *SelectionError* in each technique. Figure 8 and Figure 9 show the performance and accuracy of the different wedges.

We also revealed that the rings had a significant effect on *TotalTime* in both *Sequential* ($F_{(1.08,11.88)}$=14.60, p<.01, η_p^2=0.57), and *Simultaneous* ($F_{(1.38,15.16)}$=22.33, p<.001, η_p^2=0.67). But their effect was different from the one reported for the *index-anchored* condition. Post-hoc analysis of *TotalTime* revealed that Ring 0 was slower than Ring 1 and Ring 2 in both techniques (p<.01 for both). The results also showed a significant effect of the rings on

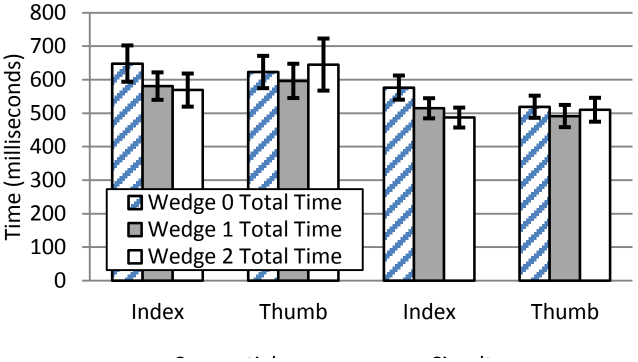

Figure 9. Average *TotalTime* by *Wedge* and *Technique*.

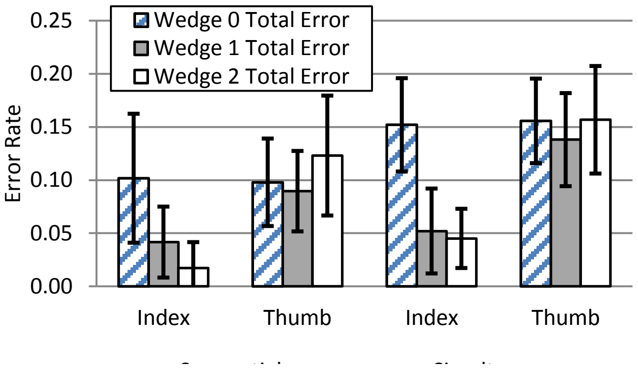

Figure 10. Median *SelectionError* by *Wedge* and *Technique*.

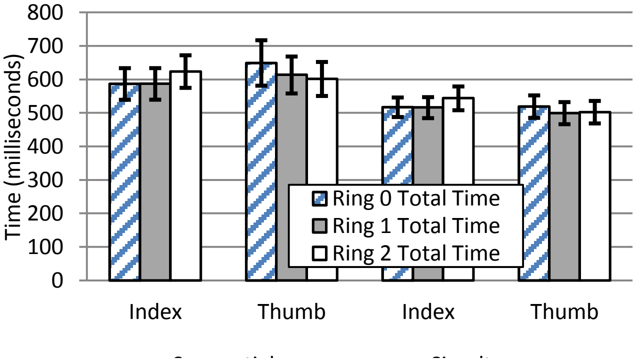

Figure 11. Average *TotalTime* by *Ring* and *Technique*.

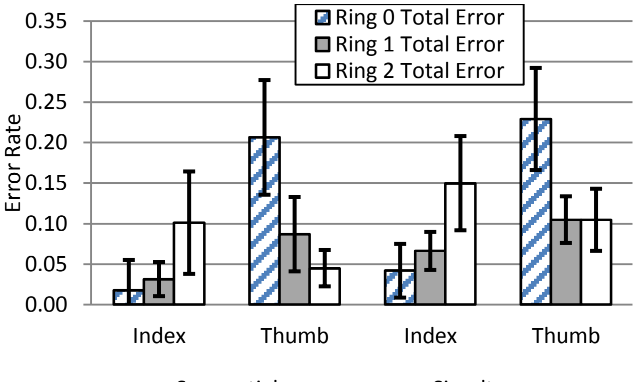

Figure 12. Median *SelectionError* by *Ring* and *Technique*.

SelectionError ($\chi^2_{(2)}$=19.45, *p*<.001 for *Sequential*; and $\chi^2_{(2)}$=11.17, *p*<.01 for *Simultaneous*). The closest ring was more error-prone to acquire than the other two. Figure 10 and Figure 11 show the performance and accuracy of the different rings.

Discussion. The results highlight different trends in the speed and accuracy in *index-anchored* and *thumb-anchored* selection. This can be explained by the dexterity of the arm and wrist. In *index-anchored* selection, participants had to rotate their arm and wrist to a fairly uncomfortable position to acquire targets in Wedge 0. However, they did not need to do so for targets in Wedge 0 in the *thumb-anchored* selection. Furthermore, the minimal wrist rotation seemed to contribute to fast target acquisition for Wedge 1 in *thumb-anchored* selection.

The distance between the primary and secondary targets affected selection performance and the performance differed depending on the anchor finger. Particularly, secondary targets close to the primary target were error-prone in *thumb-anchored* selection. One explanation for this is the large amount of finger rotation that was caused by the contraction of the hand (*e.g.*, when selecting a target with the pinky finger while the thumb is anchoring to the primary target). However, an approach which models finger orientation could be applied to address this issue (*e.g.*, the approach described by Holz and Baudisch [16]), or alternatively designs should avoid using this region for *thumb-anchored* interaction.

Effects of Finger Combination

Finally, we analyzed the effects of fingers used for acquiring the secondary target. Overall, we did not find a significant effect of *Finger Combination* on *TotalTime*. However, there was its effect in *Sequential* ($F_{(2.04,22.44)}$=4.44, *p*<.05, η_p^2=0.29). The combination of *Thumb-Pinky* was slower than *Thumb-Index* (*p*<.01), *Thumb-Middle* (*p*<.01), and *Thumb-Ring* (*p*<.05).

We also examined the effect of *Finger Combination* on *SelectionError*. However, due to the large variances of the data, post-hoc tests did not reveal any significant difference. A larger sample size is necessary to draw a conclusion on the effect of *Finger Combination* on *SelectionError*.

Discussion. Thumb-Pinky was the slowest finger combination in *thumb-anchored* selection in the *Sequential* technique. Again, this finger combination requires much contraction of the hand to acquire secondary targets closely located to the primary target. This may be one reason why *Thumb-Pinky* was slow. Our results also show that all finger combinations except *Thumb-Pinky* performed comparably in terms of selection speed. *Thumb-Index* has been shown to be the most flexible finger combination [28]. However, our results suggest that different finger combinations except *Thumb-Pinky* can be used interchangeably for multi-target selection with either the thumb or index finger anchored to the primary target.

DESIGN GUIDELINES FOR MULTI-TARGET SELECTION

Our results show that the scope of possible interactions with two fingers on a single hand is impacted by the limitation of the user's arm, wrist, and fingers. Informed by our study results, we propose the following interface design guidelines for single hand multi-target selection using multi-touch interactions:

DG1. Interfaces should encourage users to approach from S, SE, or E to the primary target.

DG2. Interfaces should take the trade-off between the speed and accuracy into account: *Simultaneous* can be faster but more error-prone while *Sequential* is slower but less error-prone.

DG3. Interfaces can use either the index finger (*index-anchored*) or the thumb (*thumb-anchored*) to select the primary target; there are no performance differences.

DG4. Interfaces should avoid placing the secondary target directly above the primary target for *index-anchored* selections, and below the primary target for *thumb-anchored* selections.

DG5. Interfaces should avoid placing the secondary target outside of a 100mm radius from the primary target, unless anchoring with thumb.

DG6. Interfaces should avoid placing the secondary target within 50mm of the center of the primary target for *thumb-anchored* selections.

We note that these guidelines are based only on data from right-handed users.

DESIGN OF A TABLETOP CONTEXT MENU

We now present a tabletop context menu design that follows the above guidelines. We will use a simple drawing application to illustrate our menu design.

Invoking Context Menus

In our design, there are two types of context menus, *thumb-anchored* and *index-anchored*. We use the *thumb-anchored* menu as a high level application wide context menu (*i.e.*, independent on the current cursor). In our drawing application, the *thumb-anchored* menu always shows the same menu items regardless of where the user is touching. We use the *index-anchored* menu as a cursor specific context menu (*i.e.*, menu items are based on the current cursor).

Detecting the Anchor Finger. We use the detection of multi-touch finger combinations to support triggering different context menus. We choose this approach because, unlike hand tracking, it is more readily supported on existing hardware [2, 21]. Lepinski *et al.* [21] have previously demonstrated techniques for identifying different combinations of fingers on multi-touch surfaces. More specifically, their technique can accurately identify if the thumb is touching the surface or not, and if it is touching in combination with other fingers or not. Therefore, the system can differentiate touches with and without the thumb. Anchoring the thumb on the surface (on its own or with another finger) could then be used to differentiate from the normal input modality (usually done with the index finger [34]) and thus invokes a *thumb-anchored* menu. Removing the thumb from the surface without pressing with another finger cancels the menu. *Index-anchored* menu, on the other hand, requires time delayed invocation (*i.e.*, the user must place her finger on the table, without moving for 100ms, upon which the context menu is invoked) or explicit invocation button because the technique cannot accurately differentiate the index finger.

Thumb-Anchored Menu. In our drawing application, tools can be selected through a *thumb-anchored* menu (see Figure 13). The user anchors the thumb in any empty space to activate the tools context menu. Then she can use any other digit to select the brush tool. From this point, any single finger input except for thumb inputs acts as brush input. The user can simply lift her thumb and continue to use the index finger to draw in one smooth motion (without needing to lift her index finger first). Alternatively, she can lift all fingers from the surface and the next time she touches the screen with a non-thumb finger, it will act as a brush.

Index-Anchored Menu. When interacting with touchscreens, the users mostly gesture with the index finger for one-point touches or paths [34]. When the system recognizes an anchor touch point that is not with the thumb, the system assumes it was the index finger and displays a transparent *index* menu (after a 100ms delay) to allow the user to quickly adjust parameters for the current tool. The menu is changed based on the currently selected tool. Figure 14 illustrates how the user can adjust the stroke size for the brush tool. When pressing her index finger on the surface, the *index* menu associated with the brush tool is shown. The user uses another finger to switch to the desired stroke size. This does not interfere with the inputs of current tool (*e.g.,* brush inputs). The index-anchored menu fades as soon as the user begins to move the index finger (*e.g.,* to use the brush).

Objects on the screen can also include a tool-tip which the user can press with her index finger to invoke a menu specific to that item. For example, in Figure 13, above the smiley face's right eye is a small tool-tip which the user can press with his index finger to activate a menu specific to the eye. Note that there is no need to uniquely identify index finger in this case, because any finger can invoke the tool-tip menu. Once the tool-tip menu is active the user uses any of his remaining fingers to select one of 9 items from the menu (see Figure 15). The commands in this object-based context menu can include object-specific commands like copy and move. Our pilot study showed that approaching the primary target from the S, SE, and E direction is the fastest (DG1). In our design, tool-tips are placed in the top, top-right, or right of the objects so the user can approach them from the fastest directions (see Figure 15).

Layout of the Context Menus

Our study results show that when the thumb or index finger are used as the anchor, the user can more easily posture these fingers to select a second within a circular area, 10cm in diameter, around the anchor finger (DG5). As a result, pie menus naturally emerge as a possible implementation

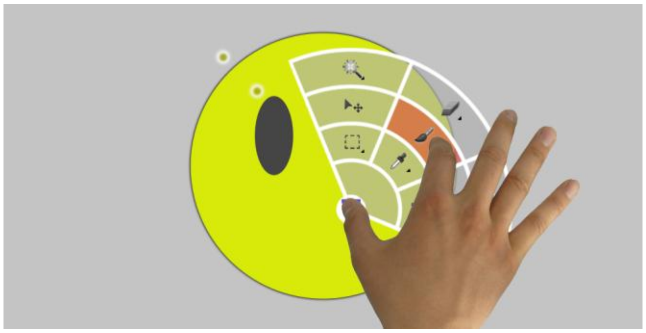

Figure 13. The user selects the paint brush by anchoring her thumb anywhere on the screen.

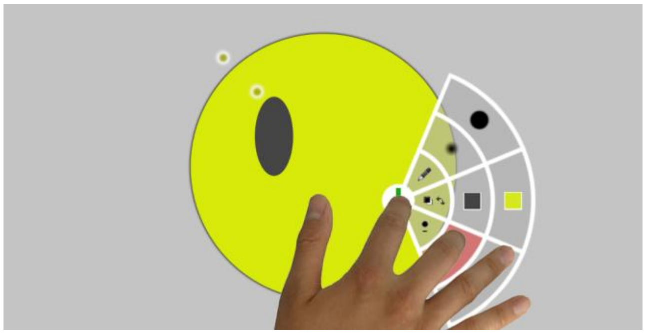

Figure 14. The user changes the stroke size of the brush by anchoring the index finger where she wants to start a line representing the mouth on the screen. Then she can begin to draw the mouth.

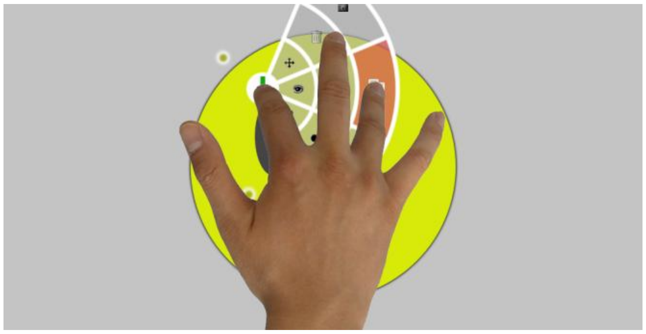

Figure 15. The user triggers the copy command by anchoring the index finger on the eye's tooltip.

Figure 16. The completed smiley face with an object tooltip for each object.

that maximizes the area the user can select from [9] (*e.g.*, compared to say a linear menu). However, due to limitations of the wrist (DG4) and past research on potential hand occlusion with pie menus on touch surfaces [8, 35], we designed our context menu based on the menu used in the study, which was implemented as a partial pie consisting of 3 rings × 3 wedges allowing for 9 possible items to be displayed at a time.

Our study results also show that the rotation of the wrist and the distance from the secondary target to the primary target impact the total selection time. Moreover, the impact is dependent on whether the anchor finger is the index or thumb. In our design, a *thumb-anchored* context menu has the inner most ring disabled because it is the slowest (DG6, Figure 13). However, the increased span between the thumb and the other fingers could allow for another ring to be added on the outside to compensate for the loss of the inner ring (4 rings × 25mm is still well within the average hand span). For *index-anchored* context menus, the partial pie menu is rotated clockwise 45˚ because Wedge 0 was the slowest (DG4, Figure 14).

More commonly accessed commands should reside in faster and/or less error-prone arcs. Based on our results, for *thumb-anchored* context menus, more frequent commands are placed in Wedge 1. For *index-anchored* context menus, more frequent commands are placed in Ring 0.

Supporting Transition from Novice to Expert Use

The system supports both *Sequential* as well as *Simultaneous* multi-touch input. Our study result shows that although *Simultaneous* was fastest, *Sequential* was more accurate (DG2). We use this insight to support the transition from novice to expert use. While still a novice, the user can sequentially invoke the context menus with her thumb or index finger, and then use a second finger to invoke the desired menu item. As a user becomes more proficient with using the menu system (*i.e.*, an expert) she will become familiar with the orientations and positions of the commands in each of the menus. Thus, she can interact with the menus and invoke menu items by simultaneously placing the correct finger combinations in the relevant orientations and positions without needing visual feedback.

Distinguishing from other Multi-touch Gestures

The system must track the finger combinations and finger movements on the surface to disambiguate multi-touch menu input from other multi-touch gestures. For example, if the user wants to zoom-in or -out using the pinching gesture [35], the user first presses down with her thumb and then index finger. This action also invokes the *thumb-anchored* context menu. By tracking the finger movements, it is possible to detect when the user starts to slide her thumb and index figure towards or away from each other; the system can then deactivate the menu and pinch-to-zoom is invoked.

CONCLUSION AND FUTURE WORK

In this paper, we presented a design for a multi-touch context menu that is invoked using one of two multi-touch selection techniques – *sequential* and *simultaneous*. In order to inform the design of the menu, we conducted a study in which we explored the limitations of the wrist rotation, fingers span, and the range of motion of arm for approach direction. Then, we measured the user performance of two multi-touch techniques compared to a single-touch baseline technique. We identified key design guidelines for unimanual multi-touch menus on horizontal tabletop surfaces. We found that the technique of selecting the two targets with two different fingers concurrently (*Simultaneous*) achieved faster performance than the technique of selecting the two targets with two different fingers sequentially (*Sequential*) and the technique of selecting them with one finger (*Multi-tap*).

Based on the findings from our studies, we suggested design guidelines for context menus invoked using the single hand multi-touch techniques on touchscreen tabletop systems. We did not cover all aspects of user performance of multi-target acquisition techniques. For example, we have not examined the full effects of various target sizes and distances. Additionally, an experiment with left-handed users is also necessary to generalize our findings reported in this paper. However, we believe that this work covers many of the interesting aspects of unimanual multiple target acquisition. Previous work has already demonstrated how to detect the finger combinations being pressed on the Surface [21]. In this work, we used this capability in a solution to distinguish input and menu invocation. We then applied this solution and the design guidelines to illustrate how a tabletop context menu system can be developed.

REFERENCES

1. Annett, M., Grossman, T., and Fitzmaurice, G. Medusa: A Proximity-Aware Multi-touch Tabletop. To appear in *Proc. UIST'11*. ACM, NY, 2011.
2. Au, O. K. C., and Tai, C. L. Multitouch finger registration and its applications. In *Proc. OZCHI '10*, ACM, NY, 2010, pp. 41-48.
3. Bailly, G., Demeure, A., Lecolinet, E., and Nigay, L. MultiTouch menu (MTM). In *Proc. IHM '08*, ACM, New York, 2008, pp. 165-168.
4. Bailly, G., Lecolinet, E., and Guiard, Y. Finger-count & radial-stroke shortcuts: 2 techniques for augmenting linear menus on multi-touch surfaces. In *Proc. CHI '10*, ACM/SIGCHI, NY, 2010, pp. 591-594.
5. Bartindale, T., Harrison, C., Olivier, P., and Hudson S. E. SurfaceMouse: supplementing multi-touch interaction with a virtual mouse. In *Proc. TEI '11*, ACM, NY, 2010, pp. 293-296.
6. Benko, H., Wilson, A. D., and Baudisch, P. Precise selection techniques for multi-touch screens. In *Proc. CHI '06*, ACM/SIGCHI, NY, 2006, pp. 1263-1272.
7. Bier, E., Stone, M., Pier, K., Buxton, W., DeRose, T., Toolglass and magic lenses: the see-through interface. In *Proc. SIGGRAPH '93*, ACM , NY, 1993, pp. 73-80
8. Brandl, P., Leitner, J., Seifried, T., Haller, M., Doray, B., and To, P. Occlusion-aware menu design for digital tabletops. In *Proc. CHI '09*, ACM/SIGCHI, NY, 2009, pp. 3223-3228.

9. Callahan, J., Hopkins, D., Weiser, M., and Shneiderman, B. 1988. An empirical comparison of pie vs. linear menus. In *Proc. CHI '88*. ACM/SIGCHI, New York, 1988, pp. 95-100.
10. Esenther A. and Ryall, K. 2006. Fluid DTMouse: better mouse support for touch-based interactions. In *Proc. AVI '06*, ACM, NY, pp. 112-115.
11. Forlines, C., Vogel, D., and Balakrishnan, R. Hybrid-Pointing: Fluid switching between absolute and relative pointing with a direct input device. In *Proc. UIST '06*, ACM/SIGCHI, NY, 2006, pp. 211-220.
12. Forlines, C., Wigdor, D., Shen, C., and Balakrishnan, R. Direct-touch vs. mouse input for tabletop displays. In *Proc. CHI '07*. ACM/SIGCHI, NY, 2007, pp. 647-656.
13. Häger-Ross, C. and Schieber, M. H. Quantifying the independence of human finger movements: comparisons of digits, hands, and movement frequencies. *The Journal of Neuroscience* (2000), pp. 8542-8550.
14. Hesselmann, T., Flöring, S., and Schmitt, M. Stacked Half-Pie menus: navigating nested menus on interactive tabletops. In *Proc. ITS '09*, ACM, NY, 2009, pp. 173-180.
15. Hinckley, K., Baudisch, P., Ramos, G., and Guimbretiere, F. Design and analysis of delimiters for selection-action pen gesture phrases in scriboli. In *Proc. CHI '05*. ACM, New York, (2005), pp. 451-460.
16. Holz, C. and Baudisch, P. 2010. The generalized perceived input point model and how to double touch accuracy by extracting fingerprints. In *Proc. CHI '10*, ACM/SIGCHI, NY, 2010, pp. 581-590.
17. Hopkins, D. The design and implementation of pie menus. *Dr. Dobb's Journal*, 16, 12 (1991), 16-26.
18. Kin, K., Agrawala, M., and DeRose, T. Determining the benefits of direct-touch, bimanual, and multifinger input on a multitouch workstation. In *Proc. GI 2009*, Canadian Information Processing Society, Toronto, 2009, pp. 119-124.
19. Kurtenbach, G., and Buxton, W. User learning and performance with marking menus. In *Proc. CHI'94*, ACM/SIGCHI, NY, 1994, pp. 258-264
20. Kurtenbach, G., Fitzmaurice, G. W., Owen, R. N., and Baudel, T. The Hotbox: efficient access to a large number of menu-items. . In *Proc. CHI'99*, ACM/SIGCHI, NY, 1999, pp. 231-237
21. Lepinski, G. J., Grossman, T., and Fitzmaurice, G. The design and evaluation of multitouch marking menus. In *Proc. CHI '10*, ACM/SIGCHI, NY, 2010, pp. 2233-2242.
22. Li, Y., Hinckley, K., Guan, Z., and Landay, J. A. Experimental analysis of mode switching techniques in pen-based user interfaces. In *Proc. CHI '05*. ACM, New York, 2005, pp. 461-470.
23. Matejka, J., Grossman, T., Lo, J., and Fitzmaurice, G. The design and evaluation of multi-finger mouse emulation techniques. In *Proc. CHI '09*, ACM/SIGCHI, NY, 2009, pp. 1073-1082.
24. Micire, M., Desai, M., Drury, J. L., McCann, E., Norton, A., Tsui, K. M. and Yanco, H. A. Design and validation of two-handed multi-touch tabletop controllers for robot teleoperation. In *Proc. IUI '11*, ACM, NY, 2011, pp. 145-154.
25. Moscovich, T. and Hughes, J. F. Indirect mappings of multi-touch input using one and two hands. In *Proc. CHI '08*, ACM/SIGCHI, NY, 2008, pp. 1275-1284.
26. North, C., Dwyer, T., Lee, B., Fisher, D., Isenberg, P., Robertson, G., and Quinn, K. I. Understanding multi-touch manipulation for surface computing. In *Proc. HCII'09*, Springer-Verlag, Berlin, 2009, pp. 236–249.
27. Pheasant, S. and Hastlegrave, C. Bodyspace: anthropometry, ergonomics and the design of the work. CRC, 2006.
28. Reilly, K. and Hammond, G. Human Handedness: Is there a Difference in the Independence of the Digits on the Preferred and Non-preferred Hands. *Experimental Brain Research*, 156, 2 (2004), 255-262.
29. Rekimoto, J. SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. In *Proc. CHI '02*, ACM/SIGCHI, NY, 2002, pp. 113-120.
30. Santello, M., Soechting, J. Matching object size by controlling finger span and hand shape. *Somatosensory & Motor Research*, 14, 3(1997), 203-212.
31. Sasangohar, F., MacKenzie, I. S., & Scott, S. D. Evaluation of mouse and touch input for a tabletop display using Fitts' reciprocal tapping task. In *Proc. HFES'09*, Human Factors and Ergonomics Society , Santa Monica, CA, 2009, pp. 839-843.
32. van Doren, C. Cross-modality Matches of finger span and line length. *Perception & Psychophysics*, 57, 4 (1995), 555-568.
33. Vogel, D. and Balakrishnan, R. Occlusion-aware interfaces. In *Proc. CHI '10*, ACM/SIGCHI, NY, 2010, pp. 263-272.
34. Wobbrock, J. O., Morris, M. R., and Wilson, A. D. User-defined gestures for surface computing. In *Proc. CHI '09*, ACM/SIGCHI, NY, 2009, pp. 1083-1092.
35. Wu, M. and Balakrishnan, R. Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. In *Proc. UIST '03*, ACM/SIGCHI, NY, 2003, pp. 193-202.
36. Wu, M., Shen, C., Ryall, K., Forlines, C., and Balakrishnan, R. Gesture Registration, Relaxation, and Reuse for Multi-Point Direct-Touch Surfaces. In *Proc. IEEE TABLETOP 2006*. pp. 183-190.
37. Zhao, S., Agrawala, M., Hinckley, K. Zone and polygon menus: using relative position to increase the breadth of multi-stroke marking menus. In *Proc. CHI '06*, ACM/SIGCHI, NY, 2006, pp. 1077-1086.