Zone and Polygon Menus: Using Relative Position to Increase the Breadth of Multi-Stroke Marking Menus

Shengdong Zhao[1]   Maneesh Agrawala[2, 3]   Ken Hinckley[3]

[1]University of Toronto
Toronto, Ontario, Canada M5S 3G4
sszhao@dgp.toronto.edu

[2]University of California, Berkeley
Berkeley, CA 94720-1776
maneesh@cs.berkeley.edu

[3]Microsoft Research
Redmond, WA 98052
kenh@microsoft.com

ABSTRACT

We present Zone and Polygon menus, two new variants of multi-stroke marking menus that consider both the relative position and orientation of strokes. Our menus are designed to increase menu breadth over the 8 item limit of status quo orientation-based marking menus. An experiment shows that Zone and Polygon menus can successfully increase breadth by a factor of 2 or more over orientation-based marking menus, while maintaining high selection speed and accuracy. We also discuss hybrid techniques that may further increase menu breadth and performance. Our techniques offer UI designers new options for balancing menu breadth and depth against selection speed and accuracy.

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

Author Keywords: Marking menus, pie menus, position-based menus, pen-based interfaces.

INTRODUCTION

Marking menus are a gesture-based menu selection technique in which menu items are arranged radially and a user draws a stroke towards a desired item in order to select it. Comprehensive user studies [8, 9] demonstrate that marking menus offer selection gestures that are fast and easy to draw, as well as an efficient transition path for novice users to become experts. However, a drawback of marking menus is that selection accuracy depends on the number of items that appear in the menu (*breadth*). Accuracy decreases substantially when breadth exceeds 8 items.

Hierarchical marking menus increase the total number of menu items available by allowing users to select from multiple submenus using a compound zigzag stroke. Kurtenbach and Buxton [8] have studied such *compound-stroke marking menus* and show that for users to maintain a reasonable accuracy rate of greater than 90%, a breadth-8 menu can have a *depth* of at most 2 levels, thereby providing access to 64 unique items.

Recently, Zhao and Balakrishnan [18] have proposed breaking compound selection strokes into a sequence of inflection-free strokes with pen lifts between each straight line stroke (Figure 1a). Such *multi-stroke marking menus* allow users to work with breadth-8 menus up to depth-3 (512 items), at an accuracy rate of 93%. Multi-stroke marking menus also use space more efficiently than compound-stroke marking menus.

Although increasing menu depth increases the total number of menu items available, deep hierarchies suffer from several drawbacks. Deeper items take longer to access because the user must draw more complex strokes to select them. Increasing menu breadth yields shallower hierarchies, but to maintain acceptable accuracy rates, the current compound-stroke and multi-stroke marking menu techniques cannot exceed breadth-8.

This limitation on menu breadth may force UI designers to employ awkward groupings of menu items. Consider a painting application that allows users to select brush color from a palette of 16 choices. Since marking menus are limited to breadth-8, the application designer must use a two level marking menu for color selection. The menu thus divides the 16 colors into arbitrary groups (Figure 1a). Studies [5, 6, 11] of the breadth-depth tradeoff for general menu hierarchies have concluded that in most cases breadth is preferable to depth.

In this paper, we introduce new variants of multi-stroke marking menus that are designed to increase menu breadth. Almost all previous marking menu designs (compound-stroke or multi-stroke) consider only one attribute of the selection stroke – its *orientation* – to determine which menu item is selected. The key idea of our approach is to consider another geometric attribute - the *position* of each stroke - in addition to its orientation.

We present Zone and Polygon menus (Figure 1b,c), two position-based variants of multi-stroke marking menus. Our experimental studies show that these menus can increase breadth by a factor of 2 or more, while maintaining high selection speed and accuracy. As in previous work [8, 18], we maintain a seamless transition path to expert usage. A menu visualization appears if a novice user pauses for a short interval, while in expert mode only the strokes are drawn. We also contribute new hybrid menu designs that may further increase performance and breadth. Our tech-

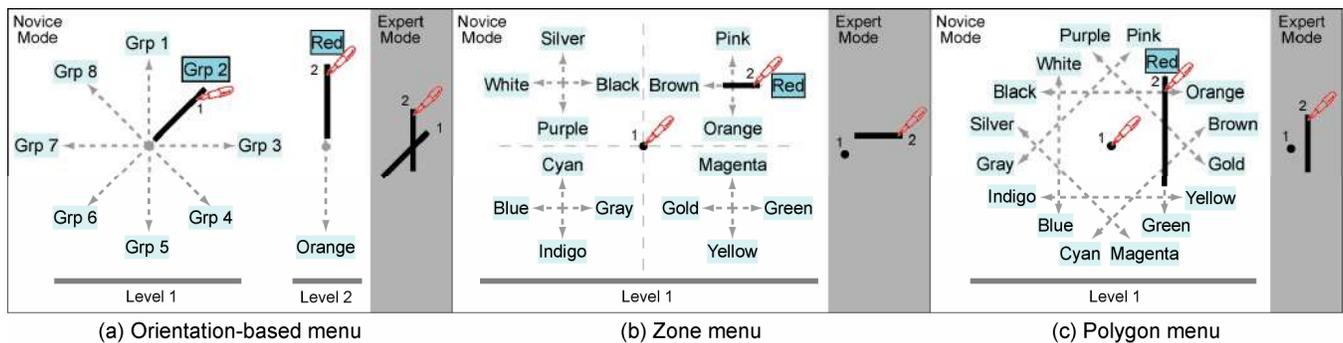(a) Orientation-based menu   (b) Zone menu   (c) Polygon menu

Figure 1. Multi-stroke marking menus require pen lifts between straight line strokes to traverse the menu hierarchy. (a) Orientation-based multi-stroke menus [18] use only the orientation of each stroke to determine which menu item is selected. We introduce (b) Zone and (c) Polygon multi-stroke marking menus. These menus require an initial tap to set the menu origin. For each subsequent stroke, both its position relative to the tap and its orientation are used to determine which menu item is selected.

niques offer UI designers new options for balancing menu breadth and depth against selection speed and accuracy.

PROPERTIES OF EFFECTIVE MARKING MENUS
Previous studies [7, 8, 9, 16, 18] have considered three aspects of usability in the design of effective marking menus:

- **Speed:** Users should be able to quickly draw the selection strokes necessary to access any menu item.
- **Accuracy**: Users should be able to easily and reliably draw the strokes necessary to access any menu item.
- **Learnability**: Users should be able to rapidly acquire the physical skills necessary to operate the menu.

Based on these earlier studies as well as our own experience with marking menus we identify specific properties that affect speed, accuracy and learnability. These properties impose constraints on the design space of marking menus and will help us to reject ineffective designs.

Stroke Complexity & Precision: Selection strokes should be geometrically simple and require just enough precision to easily select each menu item. Complex strokes are slower to draw and difficult to execute reliably. Similarly, menus that require strokes to be drawn more precisely tend to reduce selection accuracy.

Menu Breadth & Depth: The menu should balance breadth and depth with accuracy and speed. As we have seen, there is a tradeoff between menu breadth and depth that affects both speed and accuracy. In many designs, as breadth increases, the strokes must be drawn more precisely and therefore selection accuracy and speed decrease. Similarly, as depth increases more complex strokes may be necessary. As a result speed and accuracy decrease.

Scale-Independence: Users should be able to draw the selection strokes at almost any size. Drawing strokes at a fixed scale can be difficult for users. Therefore, a scale-independent technique can improve accuracy. The ability to draw shorter strokes with ballistic motions [12] may also improve selection speed.

In-Place: Users should be able to draw the selection strokes right where they are working. Menu selection techniques

such as taskbars and tool palettes that force users to move to a specific invocation location may reduce selection speed. Performing a round-trip to acquire the menu pulls the user away from their work. In-place selection is particularly useful for large-screen applications.

Attention-Free: Users should be able to draw the selection strokes without diverting attention from their primary task. The selection strokes should be simple enough that they can be drawn using motor skills alone, without any kinesthetic or visual feedback from the system.

Novice to Expert Transition: Novice users should draw exactly the same selection strokes as expert users. In this way novices rehearse the movements necessary for expert performance and can learn the transition effortlessly.

Conceptual Hierarchy: To increase learnability, the location of each item in the menu hierarchy should reflect some conceptual hierarchy of the items. For example, in many applications the main menu bar offers a file menu that contains sub-items related to file operations. As we noted with the paintbrush color example in the previous section, limitations on menu breadth may force menu designers to group the items in an unnatural manner.

THE GEOMETRIC DESIGN SPACE OF MARKING MENUS
If we consider a stroke as a straight line segment connecting the starting point (pen down event) to the ending point (pen up event), it is fully defined by three basic geometric attributes; *orientation*, *length* and the *position* of some point on the segment - we typically use the starting point of the stroke. These three attributes are independent of one another. More formally, the set of line segments embedded in the 2D plane is 4D (orientation and length are 1D each, while position requires 2D). Thus, we can extract at most 4 unique numbers from any line segment.

The advantage of treating strokes as line segments is that the exact internal shape of the stroke does not matter, and are therefore very easy to draw. To handle curved strokes we must also consider *curvature*, an attribute that may differ at each point on the stroke. For example compound-stroke marking menus consider stroke curvature to find the

inflections. In general, curvature-based techniques may be slower and less accurate than techniques that treat strokes as simple line-segments [18]. For these reasons we focus on multi-stroke marking menus and we limit our analysis to the three basic geometric attributes of straight line strokes.

Orientation: As shown by Zhao and Balakrishnan [18] multi-stroke orientation-based marking menus are very effective in terms of speed, accuracy, and learnability. The main drawback of orientation-based selection is that as breadth increases beyond 8 items, the angular precision required for each stroke also increases. Beyond breadth-8, off-axis strokes are particularly difficult to draw. As a result, both speed and accuracy are reduced. In addition, the breadth-8 limitation can force unnatural groupings of menu items (Figure 1a), or result in a situation where adding one more command requires redesigning the hierarchy.

Length: Reliably drawing strokes of a given length without any visual reference to provide a sense of scale can be difficult. Bull's Eye menus [7], for example, place menu items in concentric rings and force users to draw a stroke of appropriate length to choose the desired item. Kurtenbach [7] has shown that such menus are neither scale-independent nor attention-free. While Freidlander et al. [3] have developed an eyes-free version of Bull's Eye menus, their system replaces visual feedback with aural or tactile feedback. These approaches prevent ballistic selection because users must continuously monitor the feedback.

Although stroke length is not well suited for discrete menu selection, it can be used to provide interactive control over a continuous parameter. Both Control menus [14] and FaST sliders [12] take this approach, combining standard orientation-based marking menus with length-based dragging to control parameter values.

Position: One way to exploit stroke position for menu selection is to divide the interaction surface into zones and assign one item to each zone. The user taps in the appropriate zone to select the corresponding item. For in-place selection users can specify the origin of the menu with their first tap and then specify the zone relative to this origin with the second tap as in Figure 2a. However, rectangular zones are not scale-independent. We can provide scale-independence by creating a single open-ended zone in each radial direction from the origin of the coordinate system as shown in Figure 2b. Because the zones are arranged radially, such position-based menus are similar to orientation-based marking menus and we believe they are likely to share the same advantages and limitations.

While FlowMenus [4] and Quickwriting [13] also use radial zones, the radial zones are arranged about a finite-sized central zone. To select an item, users must draw a single continuous stroke from the central zone through the outer zones and back to the center. Thus, these techniques are not scale-independent, and they force users to draw complicated curving strokes.
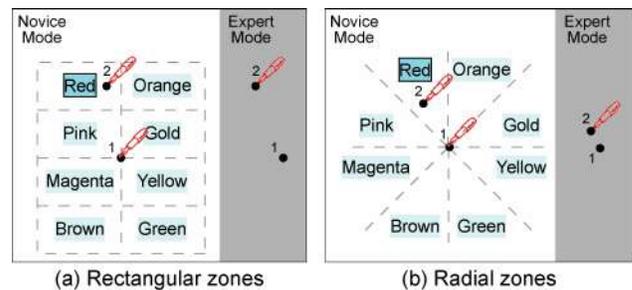

Figure 2. Position-based multi-stroke marking menus use the position of the second tap relative to the first to determine which item is selected. (a) Rectangular zones are not scale-independent because the zone depends on the distance between taps. (b) Open-ended radial zones are scale-independent.

MENU SELECTION USING POSITION & ORIENTATION

Orientation, length and position are independent attributes of a straight line stroke. Therefore we can design menu selection techniques that consider two or more of these attributes simultaneously. However, as we have seen, length is difficult for people to precisely control because it is neither scale-independent nor attention-free. Thus, we propose two new forms of multi-stroke marking menus that simultaneously consider position and orientation.

Zone Menus

Our first approach extends the pure position-based design shown in Figure 2b. To select an item the user first taps to specify the menu origin and then draws one or more straight line strokes (Figure 1b). The item selected depends on the zone in which the starting point of each stroke falls, and the orientations of the strokes. For example, in Figure 1b to select the menu item Red, the starting point of the second stroke is located in the upper left zone, and it is drawn in the horizontal orientation. The breadth of a Zone menu depends on the number of zones and the number of stroke orientations allowed within each zone. Figure 1b shows a 4 zone menu where each zone allows 4 stroke orientations. The total menu breadth is 16 items.

Our Zone menu approach was inspired by and shares some design properties with Kurtenbach et al.'s [10] Hotbox. The Hotbox also splits the input area into zones and allows users to access a different marking menu in each one. However the Hotbox is not hierarchical as it only provides single level marking menus within each zone. In contrast, our Zone menu is hierarchical: the user draws a separate stroke for each level of the hierarchy and can choose a different zone and orientation for each stroke. Note that the initial tap used to set the menu origin only needs to be drawn once.

One drawback of both Zone menus (Figure 1b) and the purely position-based menus introduced earlier (Figure 2) is that the user must carefully position the starting point of a stroke within a particular zone. As the number of zones increases, the size of each zone decreases and this positioning can become difficult. We have developed Polygon menus to mitigate this issue.
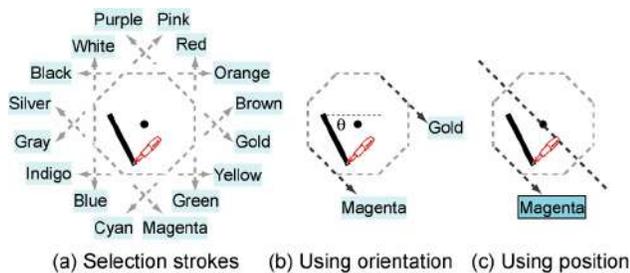
(a) Selection strokes　　(b) Using orientation　　(c) Using position

Figure 3. (a) Selection strokes for an 8-sided Polygon menu. (b) Matching the orientation θ of the stroke to the polygon edges eliminates all but two possibilities. The interaction surface is split into two half-plane sized zones in the direction of the remaining edges. (c) The stroke and the edge it represents lie in the same half plane. The stroke could lie anywhere in this half-plane and would select the same menu item.

Polygon Menus

Polygon menus allow users to select menu items by drawing strokes corresponding to edges of an N-sided polygon. Each edge can be used to select one of two items depending on the direction in which the stroke is drawn. Thus, the breadth of an N-sided Polygon menu is 2N.

As with Zone menus, an initial tap sets the origin of the menu. The orientation and position of subsequent strokes are analyzed as shown in Figure 3. In an even-sided polygon, opposite edges have the same orientation. Therefore the orientation of the stroke eliminates all but two edges of the polygon. Note that these two edges lie on opposite sides of the menu origin. To disambiguate between them, we split the interaction surface into two half-plane zones oriented parallel to the edge, and running through the menu origin. The stroke and the polygon edge it represents will then lie in the same half-plane.

Polygon menus dynamically adjust the orientation of the half-plane sized zones based on the orientation of the stroke. In contrast, Zone menus fix the shape of each zone *a priori* and the size of each zone may be much smaller than a half-plane depending on the total number of zones. Thus, Polygon menus are likely to require less precision than Zone menus when positioning the starting point of a stroke.

Setting the Menu Origin

Both of our position-based menu designs use an initial tap as the menu origin and consider the position of subsequent strokes relative to this origin. The initial tap not only allows for in-place menu selection but also allows users to place their hand so that it is easy to draw the subsequent strokes about the origin. However, the drawback of explicitly specifying the menu origin is that tapping takes time and therefore reduces selection speed.

An alternative approach is to fix the menu origin to a specific point on the input area and then force the user to always position their strokes relative to this fixed origin. Given a small, tightly constrained input area, like the touchpad of a laptop, or the screen of a small PDA, the user's hand will remain roughly centered over the input area. Thus, the center of the input area can serve as the menu origin and the user no longer has to draw the initial tap, thereby increasing selection speed.

EXPERIMENT 1

The primary design goal of both Zone and Polygon menus is to increase menu breadth while maintaining the speed, accuracy and learnability of orientation-based multi-stroke marking menus. We expect to maintain learnability because both of our new designs encourage novices to use exactly the same selection strokes they would use as experts. Therefore, our first experiment focuses on comparing the speed and accuracy of our two new position-based multi-stroke menu designs with the orientation-based multi-stroke marking menus of Zhao and Balakrishnan [18]. We do not compare our techniques against compound-stroke orientation-based marking menus [8] because Zhao and Balakrishnan have already shown the multi-stroke design to be superior.

We have seen that selection speed and accuracy depend on menu breadth and depth, as well as the complexity of the strokes and the precision required to draw them. If our position-based menus increase menu breadth, they will permit shallower menu hierarchies than purely orientation-based menus and may thereby increase selection speed. However, because the position-based designs consider position and orientation, they may require strokes to be drawn more precisely and thereby reduce speed or accuracy. In addition, with large input areas these designs require an initial tap to set the menu origin, which may decrease selection speed.

Previous studies [7, 18] of purely orientation-based menus have shown that at depth-1 users can draw 12 orientations with acceptable accuracy, but at depth-2 users can only draw the 8 basic compass directions accurately. Since users have trouble drawing strokes at more than 8 different orientations, we limit our investigation to Zone and Polygon menu designs with at most 8 stroke orientations. This limit puts an upper bound of breadth-64 on Zone menus (8 radial zones with 8 possible stroke orientations in each zone) and breadth-16 (8-sided polygon) on Polygon menus. The accuracy of Zone and Polygon menus also may depend on the shape and size of their zones. The smaller fixed shape zones of Zone menus demand increased drawing precision and may be less accurate than Polygon menus.

Participants

Nine right-handed volunteers (5 women, 4 men) ranging from 26 to 60 years old participated in Experiment 1. All participants had little or no previous experience with the TabletPC. None had previously used marking menus.

Apparatus

The experiment was conducted using a Toshiba Portege 3500 TabletPC running MS WindowsXP TabletPC edition. The tablet's stylus was used for input. Our software was implemented in C# using Microsoft Visual Studio .Net.
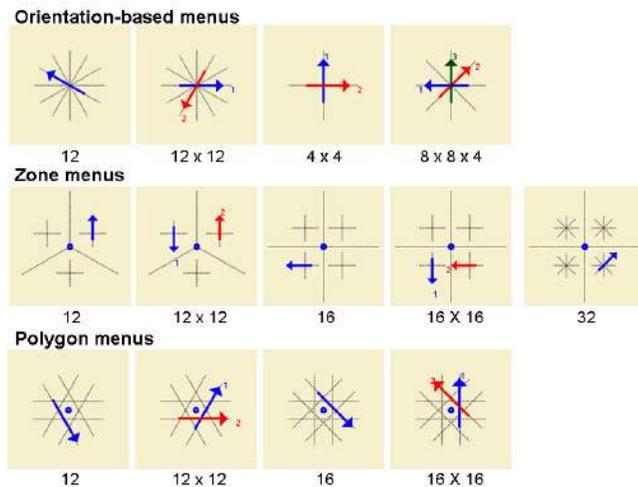
Figure 4. Examples of experimental stimuli. Strokes are numbered in the order to be drawn and color coded - blue first, red second, green third. The initial tap is also drawn in blue.

Task and Stimulus

Marking menu users achieve expert level performance when they are familiar with the menu layout and can make selections without waiting for the menu visualization to be displayed. Experts may chunk together the marks required to make a selection and thus execute them very quickly [2]. Earlier studies of orientation-based marking menus have assisted users in achieving expert level performance by using compass labels for breadth-8 menus and clock labeling for breadth-12 [8, 18]. Users are given a sequence of labels and asked to produce the strokes necessary to generate those labels. However, no such obvious labelings exist for our position-based designs. Thus, we adopted the approach of Balakrishan and Patel [1] and directly display the strokes we wish the users to reproduce (Figure 4). Users did not have to learn the layout of menu items, because this is not what we were interested in evaluating.

In our experimental setup (Figure 5), a trial began as soon as users tapped the "next trial" button. The stimulus for the trial appeared and the user had to respond by emulating the corresponding strokes. To encourage users to chunk the stimuli and achieve expert level performance the stimulus disappeared as soon as users started drawing the selection command. Thus, users had to remember the sequence of strokes long enough to execute them. To reinforce learning and aid users in correcting errors in subsequent trials, the strokes left visible ink trails as they were drawn. After users finished drawing the strokes, the stimulus reappeared along with a message explaining whether or not the trial was successful. The computer beeped softly on each unsuccessful trial. Users tapped the "next trial" button to proceed.

Experimental Design

Experiment 1 investigated the effects of menu technique (*orientation, zone, polygon*) and menu layout (four combinations of breadth and depth) on selection speed and accuracy. Each subject performed the menu selection task using
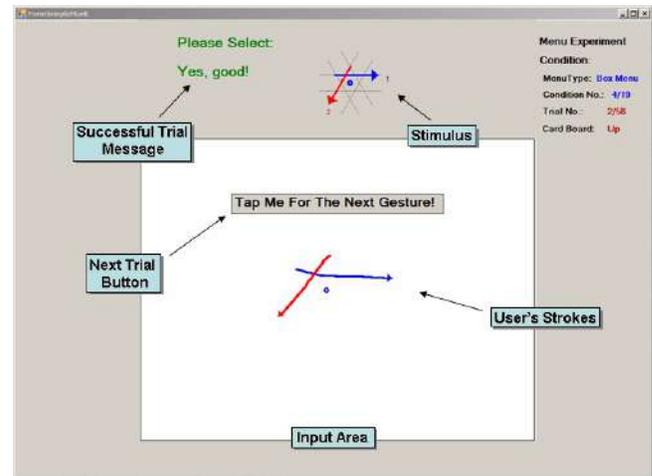

Figure 5. Experimental setup as it appeared immediately after successful completion of a trial for a 12x12 Polygon menu. The input area covered a space of 6" x 4.5" and users were required to initially tap the screen to specify the menu origin.

all three menu techniques. The ordering of the three techniques was counterbalanced across subjects using a Latin square design.

To analyze the effects of menu layout on performance we initially planned on testing all combinations of breadth-12 and breadth-16 at depth-1 and depth-2, yielding 4 layouts (*12, 12x12, 16,* and *16x16*). However, pilot tests revealed that at breadth-16 both the speed and accuracy of orientation-based menus decrease to the point that these techniques are not worth testing in a formal experiment. Therefore, for the orientation-based menus, we replaced the *16* and *16x16* layouts with more practical *4x4* and *8x8x4* layouts. These replacements trade breadth for depth while maintaining the same total number of menu items. Because we replaced these conditions only for the *orientation* technique, our design was not fully factorial in menu breadth and depth. However, the resulting conditions are more representative of practical alternatives a designer might consider to support 16 and 256 item menus, and thus make the experimental results more relevant to practical design questions.

For each menu technique we used a fixed ordering of the 4 layouts from easy to hard (*12, 12x12, 16* or *4x4, 16x16* or *8x8x4*) so that subjects could ease into the more complex layouts[1]. For each menu layout subjects performed 3 blocks of trials. As the number of menu items increases more trials are necessary to obtain good coverage over all possible selection gestures. Our layouts allowed access to 12, 144, 16 or 256 items and we used 12, 18, 16 or 24 trials respectively within each block. On each trial we randomly chose a menu item from amongst all the items in the layout.

---

[1] Kurtenbach [7] studied compound-stroke orientation-based marking menus up to breadth-12. Zhao and Balakrishnan [18] compared multistroke against compound stroke orientation-based marking menus, but did not go beyond breadth-8. Ours is the first study of orientation-based multistroke marking menus at breadth-12.

	Technique					
	Orientation		**Zone**		**Polygon**	
Layout	12	12x12	12	12x12	12	12x12
	4x4	8x8x4	16	16x16	16	16x16

Table 1. The 12 main conditions in our experiments. Due to asymmetries in the conditions we partition the data into three subsets for analysis. Subset A consists of all breadth-12 conditions (blue cells), Subset B of all 16 item conditions (yellow cells) and Subset C of all 256 item conditions (green cells).

At the start of each new menu technique we gave subjects a practice block of 21 trials to help them become familiar with the task. In addition, at the beginning of each menu layout condition we added four practice trials. Participants completed the experiment in approximately 1.5 hours, including required breaks between each menu technique.

Thus our design (excluding practice trials) has a total of:

> 9 subjects x
> 3 menu techniques (orientation, zone, polygon) x
> 3 blocks x
> (12+18+16+24) trials per block for 4 layouts
> = 5670 menu selections

Dependent variables were *accuracy*, *reaction time*, *execution time* and *total time*. Accuracy was computed as the fraction of successful trials to total trials. Reaction time was measured as the interval between the appearance of the stimulus and the pen down event for the first stroke or tap. This interval represents the time participants took to understand the stimuli before making a selection. Execution time was measured as the interval between the first pen down event and the pen up event of the last stroke. It represents the time required to physically draw the strokes. The total time is reaction time + execution time.

Results

Figure 6 shows means and standard errors for the dependent variables. Because our experiment was not fully factorial in menu breadth and depth, it was not possible to analyze all conditions with a single ANOVA. Instead we partitioned the data into three fully-crossed subsets (Table 1) and ran a separate ANOVA on each.

Subset A - All Breadth-12 Conditions (blue cells)

We began by conducting a repeated measures ANOVA on the factors of menu technique and menu depth for the breadth-12 conditions only.

Accuracy: We found a significant main effect for technique ($F_{2,16}$=15.71, $p<.01$). Pairwise t-Tests (with Bonferroni correction) showed that the accuracy of *orientation* (87.1%) was significantly worse than for *zone* (97.7%) (T_{53}=6.89, $p<.001$) and *polygon* (98.2%) (T_{53}=6.84, $p<.001$). There was also a significant main effect for menu depth ($F_{1,8}$=66.44, $p<.01$), indicating that selections requiring two strokes are less accurate than selections requiring one stroke. Finally, there was a significant technique x depth interaction ($F_{2,16}$=20.72, $p<.01$), suggesting that the *orientation* technique dropped in accuracy faster than *zone* and *polygon* between depth-1 and depth-2.

Reaction Time: Menu technique did not have a significant effect on the reaction time indicating that there was little difference in mental preparation time between the stimuli we used for the three techniques. There was a significant effect for menu depth ($F_{1,8}$=31.68, $p<.01$), implying that more time was needed to react to stimuli of deeper menus.

Execution Time: We found a significant main effect for technique ($F_{2,16}$=4.72, $p<.05$). Pairwise t-Tests showed that the average execution times for *orientation* (675 ms) differed significantly from both *zone* (869 ms) (T_{53}=5.60, $p<.001$) and *polygon* (887 ms) (T_{53}=6.25, $p<.001$). This result confirms that drawing a tap and stroke as required by position-based menus is slower than drawing a stroke alone as required by orientation-based menus. We also found a significant effect for menu depth ($F_{1,8}$=310.44, $p<.001$), since depth-2 selections took longer to complete. In addition we found a significant technique x depth interaction ($F_{2,16}$=8.93, $p<.001$).

Total Time: While technique did not have a significant effect, there was a significant effect for depth ($F_{1,8}$ =121.28, $p<.001$), as deeper selections took more time.

Subset B – All 16 item conditions (yellow cells)

Due to the asymmetry in breadth-16 menu layouts, we conducted a second ANOVA on just the three menu configurations *orientation-4x4*, *zone-16* and *polygon-16* containing
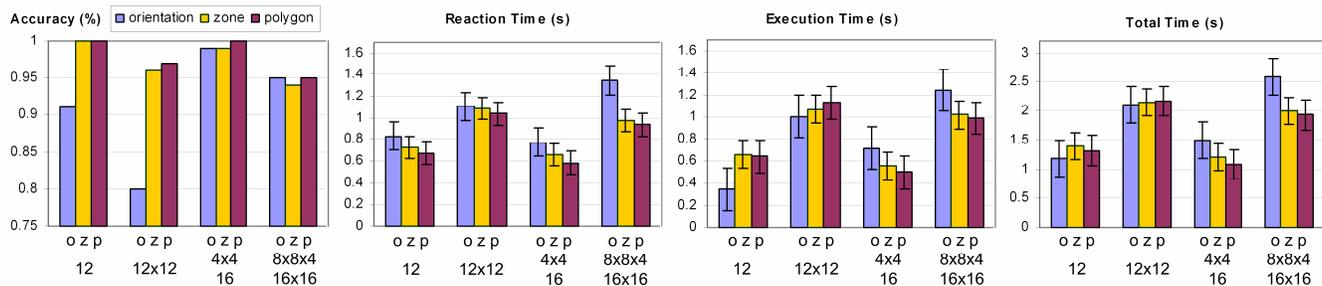

Figure 6. Performance data for Experiment 1 in which a large sized input area was provided and users had to initially tap to specify the menu origin. In the breadth-12 conditions (first 2 sets of bars in each chart) Zone and Polygon menus are significantly more accurate than the orientation-based menu, but their speed is slightly slower. For 16 and 256 menu items, (last 2 sets of bars), accuracy is similar across all techniques, but the shallower Zone and Polygon menus are faster than the orientation-based menu.

sixteen items. This comparison allowed us to investigate the tradeoff between increasing depth for an orientation-based menu versus increasing breadth for position-based menus.

Accuracy: As shown in Figure 6, all three configurations provided very high accuracy (between 97.4% and 99.8%). Menu configuration did not significantly affect accuracy.

Reaction Time: There was a significant effect for menu configuration ($F_{2,16}$=8.04, p<.01). Pairwise t-Tests revealed that reaction time for *orientation-4x4* (935 ms) was significantly slower than for *zone-16* (775 ms) (T_{26}=3.81, p<.001) and *polygon-16* (773 ms) (T_{26}=5.91, p<.001). Reacting to the 2-level *orientation* stimulus was slower than reacting to the 1-level *zone* or *polygon* stimulus.

Execution Time: We found a significant main effect for menu configuration ($F_{2,16}$=12.57, p<.01). Pairwise t-Tests showed that *orientation-4x4* (714 ms) was significantly slower than *zone-16* (554 ms) (T_{26}=4.89, p<.001) and *polygon-16* (497 ms) (T_{26}=8.78, p<.001). The difference was about 200 ms and it suggests that drawing an extra *orientation* stroke was more costly than drawing the initial tap required by the position-based menus.

Total Time: We found a significant main effect for menu configuration ($F_{2,16}$=12.72, p<.01). Pairwise t-Tests revealed that *orientation-4x4* (1489 ms) was significantly slower than *zone-16* (1208 ms) (T_{26}=5.31, p<.001) and *polygon-16* (1081 ms) (T_{26}=8.15, p<.001). For 16 items the shallower breadth-8, position-based menus are faster to use than the deeper breadth-4, orientation-based menu.

Subset C – All 256 item conditions (green cells)

We conducted a third ANOVA on the three menu configurations *orientation-8x8x4*, *zone-16x16* and *polygon-16x16* containing 256 items. This comparison allowed us to investigate the same breadth-depth tradeoff as Subset B, but with menus containing more items. Our analysis yielded exactly the same significant effects as Subset B. Again we found that the increased breadth position-based designs are faster to use than the lower breadth orientation-based menus. For brevity we omit the details of the analysis for Subset C.

Learning Effects

To check whether or not users had learned to select menu items at expert performance levels we checked for differences between the blocks in each condition. We found that block had no significant effect on accuracy. However, as is typical of repeated measures studies, there was a significant effect for block on reaction time ($F_{2,32}$=18.44, p<.01), execution time ($F_{2,32}$=15.68, p<.01) and total time ($F_{2,32}$=24.42, p<.01). Pairwise t-Tests showed that in all three cases block 1 was significantly slower than block 2 (all p<.0167) and block 3 (all p<.0167), but there was no significant difference between blocks 2 and 3. Excluding data from block 1 did not change the results of our analysis.
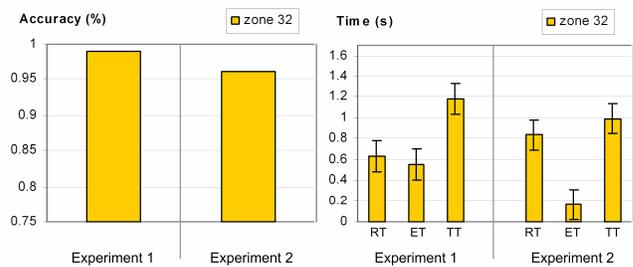

Figure 7. Performance data for the *zone-32* condition show good accuracy and speed in both experiments.

Zone-32

We also wanted to check whether Zone menus are viable menu designs breadth-32 and breadth-64. Pilot tests eliminated *zone-64* as users produced unacceptably low accuracy rates. The *zone-32* menus fared well in the pilots and so we added this layout to our experiment. This layout was always the final condition tested to avoid unequal learning with the Zone menu in the 12 main conditions. We used 24 trials per block in this condition for a total of:

9 subjects x 3 blocks x 24 trials = 648 menu selections

Since *zone-32* was not part of the factorial design of our main experiment we simply report the descriptive statistics. As shown in Figure 7, accuracy was 96%, average reaction time was 626 ms, average execution time was 553 ms, and average total time was 1179 ms, suggesting that *zone-32* is a viable design option.

EXPERIMENT 2

As we noted earlier, when the input area is relatively small, Zone and Polygon menus can work with a fixed menu origin at the center of input area and thereby eliminate the need for the initial tap. Our second experiment focuses on comparing these tapless versions of our position-based techniques with purely orientation-based menus.

We recruited a new set of 9 right-handed volunteers (5 women, 4 men) with little or no previous TabletPC experience, ranging in age from 28 to 54. We made two small changes to the set-up for Experiment 1. First, to simulate a small input area device, we placed a cardboard cutout with a 1 inch square hole in it over the input area of the Tablet PC. Users were instructed to draw their selection strokes within the hole. Second, we modified the software so that the system assumed the center of the hole was the menu origin. Thus, users did not have to draw the initial tap. The experimental conditions were exactly the same as in the previous experiment.

Results

Figure 8 presents an overview of the performance data for Experiment 2 using the small input area. We analyzed the data from this second experiment exactly the same way we analyzed data from the first. Almost all significance effects were replicated suggesting that the position-based variants
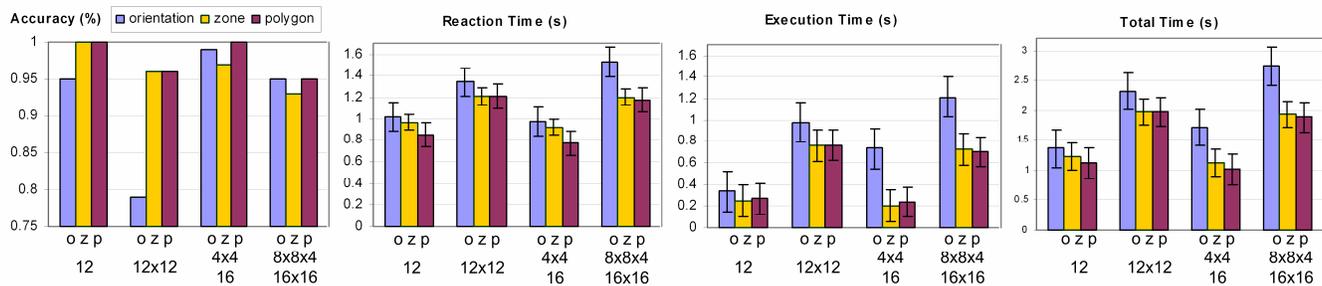
Figure 8. Performance data for Experiment 2 in which a small 1 inch square input area was used. The menu origin was fixed to the center of this area, thereby eliminating the initial tap for Zone and Polygon menus. The results are very similar to those of Experiment 1. The main difference is that these tap-less menus are faster than orientation-based menus in all conditions.

are effective even when the input area is small and the menu origin is fixed. For brevity we only report on the differences between the two experiments.

Subset A - All Breadth-12 Conditions (blue cells)
Execution Time: Although menu technique was not quite significant ($F_{2,16}$=3.62, p=.051) we found that *zone* (505 ms) and *polygon* (517 ms) required less execution time than *orientation* (673 ms). This is the reverse of the trend we saw in Experiment 1, where the position-based variants required an initial tap and were therefore significantly slower to execute than the orientation-based technique.

Subset B – All 16 item conditions (yellow cells)
Reaction Time: There was a significant main effect for menu technique ($F_{1,8}$=8.239, p<.05). The *polygon-16* condition (773 ms) was significantly faster than both *orientation-4x4* (978 ms) (T_{26}=4.88, p<.001) and *zone-16* (935 ms) (T_{26}=4.07, p<.001). This result was somewhat surprising because in Experiment 1, both *zone-16* and *polygon-16* were significantly faster than *orientation-4x4*. With a small input area, however many of the strokes required by *zone-16* are limited to a very small quadrant of the input area. Thus, *zone-16* may require more mental preparation than *polygon-16* which has larger half-plane zones that dynamically adapt their orientation for each stroke.

DISCUSSION AND DESIGN IMPLICATIONS
The goal of our experiments was to compare the performance of orientation-based multi-stroke marking menus with our position-based multi-stroke menus. Our results can be summarized as follows:

Exceeding Breadth-8: As menu breadth grows beyond 8 items per level, Zone and Polygon menus provide much better accuracy than orientation-based menus. Moreover, even though tapping to set the menu origin decreases execution speed for the position-based designs, the total time does not differ significantly between the three techniques. When tapping is not necessary Zone and Polygon are significantly faster than orientation-based menus. These results strongly suggest that interface designers should use position-based menus when breadth is greater than 8.

Breadth-Depth Tradeoff: If we limit orientation-based menus to breadth-8 we must increase depth to maintain the same number of items in a menu. However, increasing the depth of orientation-based menus significantly slows down their total selection speed compared to shallower position-based menus. It is faster to tap and draw one stroke with a 1-level position-based menu than to draw two strokes with a 2-level orientation-based menu. Similarly, drawing a tap and 2 strokes is faster than drawing 3 strokes. These results indicate that for a fixed number of menu items it is better to use shallower position-based menus than deeper orientation-based menus.

We believe that tapping may require less mental effort than drawing a longer stroke. In all three multi-stroke menu designs drawing a stroke requires users to choose a path through the menu hierarchy while tapping simply allows users to set the menu origin. Users do not have to mentally navigate the menu hierarchy when drawing the initial tap. We leave an in-depth study of the effort required for tapping versus stroking as future work.

Zone vs. Polygon: Although there are few significant differences in performance between Zone and Polygon menus, the general trend in our data shows that Zone menus are less accurate and slower than Polygon menus. In addition we have performed a small pilot test on four subjects in which we simulated eyes-free selection by eliminating all visual feedback, including ink trails, from our system. Although users were able to successfully use Zone and Polygon menus in this eyes-free condition, we found an even stronger performance gain for Polygon menus. We believe these benefits likely occur because Polygon menus require less stroke precision than Zone menus.

Yet, Polygon menus are unlikely to scale beyond breadth-16, while our results show that Zone menus of breadth-32 are viable. Therefore, interface designers should use Polygon menus up to breadth-16 and then consider Zone menus if they need to go up to breadth-32.

DESIGN ISSUES AND ALTERNATIVES
Several design issues and alternatives arise when implementing multi-stroke menus in real-world settings.
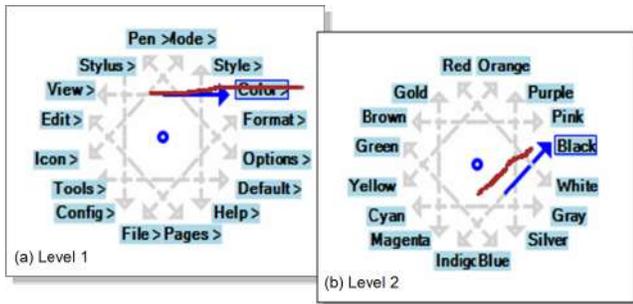
Figure 9. Novice mode menu visualization showing items found in a real drawing application. After selecting the menu item Color at the first level (a) , the menu fades out and a new submenu fades in at the same place. We select Black from the second level menu (b).

Novice Mode Menu Visualization

In real-world applications a novice mode that pops up a menu visualization is essential for learning how to use the menu. We have implemented novice mode versions of orientation-based menus, as well as Zone and Polygon menus, that fade-in the visualization for the current menu level if the user pauses at any point in the interaction for more than 300 ms. We populated the menus with items commonly found in drawing applications and in an informal pilot study asked six novice users to select specific items in the hierarchy as shown in Figure 9. We found that while these novice users were able to use the menu visualizations to find and select the appropriate menu items, they sometimes had to try several paths through the menu hierarchy.

One recurring comment from the novice users was that they preferred menu visualizations which showed the most menu choices at once. They pointed out that the breadth-8 orientation-based menus required traversing more of the hierarchy to find an item than the breadth-16 Zone and Polygon menus. This feedback suggests that increasing menu breadth may improve learnability; as the menu visualizations display more information, novices can more easily find the items they are searching for.

Abandoning an Ongoing Menu Selection

Zhao and Balakrishan [18] suggest a timeout-based approach for aborting a selection. However, working with novice users we found that it was impossible to set a reasonable value for this abort timeout. Initially, when users require more time to read the menu items, a longer abort timeout is necessary to avoid accidentally canceling the interaction while users are still reading. As users become familiar with the menus, a shorter timeout is warranted so that users can quickly abort as necessary. Thus, we use a spring-loaded mode (quasimode) [15, 16] to both activate and abort the menu. Users press and hold a button (ctrl key in our implementation) for the entire interaction. Users can abort by releasing the button at any time before completing the selection. Note that this abort mechanism was not included in the implementation we used in our experiments.
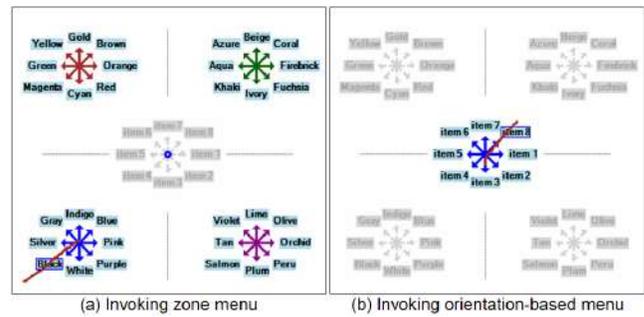

Figure 10. Hybrid Zone-Orientation menu of breadth-40. (a) If the first stroke is a short tap, it invokes the Zone menu. (b) A longer first stroke invokes the orientation-based menu.
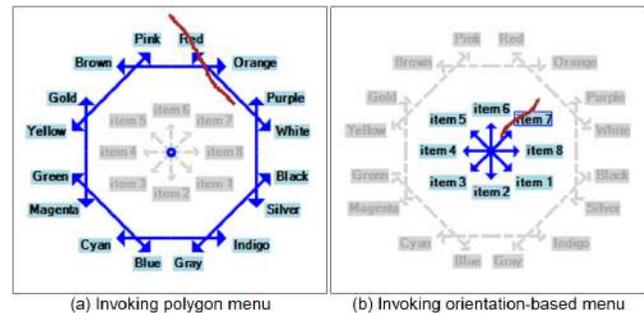

Figure 11. Hybrid Polygon-Orientation menu of breadth-24. (a) A short tap, invokes the Polygon menu. (b) A longer first stroke invokes the orientation-based menu.

Hybrid Menu Designs

We implemented two menus that hybridize our position-based designs with pure orientation-based menus. The goal of these hybrids is to further increase menu breadth while providing faster access to a small number of frequently used items. The key idea is to consider the length of the first stroke to decide which menu is invoked. A tap indicates that the user wishes to access an item in the position-based menu, while a (longer) stroke is used to directly access an item in the orientation-based menu (Figures 10 and 11)[2]. Since the items in the orientation-based menu do not require the initial tap they are faster to access.

These hybrid designs increase the breadth of Zone and Polygon menus by including up to 8 items in the orientation-based menu. However the increase in breadth does not extend beyond the top menu level. All strokes after the top level are interpreted as a position-based stroke. When the first stroke is treated as an orientation-based stroke we use the center of this stroke as the menu origin for subsequent Zone or Polygon strokes.

CONCLUSIONS AND FUTURE WORK

We have presented improved multi-stroke marking menu designs that consider both the relative position and orienta-

---

[2] The hybrids are not completely scale-independent because they make use of the length of the first stroke. In practice, however, we have found that users can easily separate drawing a short tap versus a longer stroke.

tion of selection strokes. Zone and Polygon menus extend breadth to 16 items, while providing good speed and accuracy. Our results also suggest that zone-32 is a viable design, but zone-64 is not. Our hybrid menu designs may further increase breadth and performance. These new variants of multi-stroke marking menus offer interface designers a wider set of options when choosing how to balance menu breadth and depth against speed and accuracy.

There are several directions for future work. A longitudinal user study within the context of a real-world application may provide further insight on the limits of expert level performance. A drawback of the experimental stimuli we used in the experiments reported here is that they explicitly depict the strokes users must draw to select an item. Thus, the stimuli may lead users to draw the strokes in a particular way. A longitudinal study in which user have time to become familiar with the menus could employ a sequence of menu item labels instead of depicting the strokes.

Inspired by the work on EdgeWrite [17], we would like to explore techniques for extending Polygon menus to support motion impaired users. A four-sided Polygon menu essentially forms a square box about the menu origin and provides breadth-8 menus. All the selection strokes are along edges of the box and therefore this technique may be amenable to use with the same physical template used in the EdgeWrite system.

While we have focused on using the relative position of strokes, other geometric attributes such as curvature might provide additional means for increasing menu breadth. Similarly time-based information such as stroke velocity and acceleration, or hardware specific properties such as pen tilt could provide additional channels of information for improving gesture-based menu selection.

REFERENCES
1. Balakrishnan, R., & Patel, P. (1998). The PadMouse: Facilitating selection and spatial positioning for the non-dominant hand. *ACM CHI Conference on Human Factors in Computing Systems.* pp. 9-16.
2. Buxton, W. (1986) Chunking and phrasing and the design of human-computer dialogues. *Readings in human-computer interaction: Towards the year 2000*, R. Baecker, J. Grudin, W. Buxton, S. Greenberg., Editors. 1986, Morgan Kaufmann. pp. 494-499.
3. Friedlander, N., Schlueter, K. & Mantei, M. (1998). Bullseye! when Fitts' law doesn't fit. *ACM CHI Conference on Human Factors in Computing Systems.* pp. 257-264.
4. Guimbretière, F., Stone, M. & Winograd, T. (2001). Fluid interaction with high-resolution wall-size displays. *ACM UIST symposium on User interface software and technology.* pp. 21-30.
5. Jacko, J. A. & Salvendy, G. (1996). Hierarchical menu design: Breadth, depth, and task complexity. Perceptual and Motor Skills, 82, pp. 1187-1201.
6. Kiger, J. I. (1984). The depth/breadth tradeoff in the design of menu-driven interfaces. International Journal of Man-Machine Studies, 20, pp. 201-213.
7. Kurtenbach, G. (1993). The design and evaluation of marking menus. Ph.D. Thesis. University of Toronto.
8. Kurtenbach, G., & Buxton, W. (1993). The limits of expert performance using hierarchical marking menus. *ACM CHI Conference on Human Factors in Computing Systems.* pp. 35-42.
9. Kurtenbach, G., & Buxton, W. (1994). User learning and performance with marking menus. *ACM CHI Conference on Human Factors in Computing Systems.* pp. 258-264.
10. Kurtenbach, G., Fitzmaurice, G.W., Owen, R.N., & Baudel, T. (1999). The Hotbox: efficient access to a large number of menu-items. *ACM CHI Conference on Human Factors in Computing Systems.* pp. 231-237.
11. Larson, K., & Czerwinski, M. (1998). Web page design: Implications of memory, structure and scent for information retrieval. *ACM CHI Conference on Human Factors in Computing Systems.* pp. 25-32.
12. McGuffin, M.J., Burtnyk, N., & Kurtenbach, G. (2002). FaST Sliders: Integrating Marking Menus and the Adjustment of Continuous Values. *The Graphics Interface Conference.* pp. 35-41.
13. Perlin, K. (1998). Quikwriting: Continuous stylus-based text entry, *ACM UIST Symposium on User Interface Software and Technology,* pp. 215–216.
14. Pook, S., Lecolinet, E., Vaysseix, G. & Barillot, E. (2000). Control menus: Execution and control in a single interactor. *ACM CHI Conference on Human Factors in Computing Systems.* pp. 263-264.
15. Raskin, J. (2000). *The Humane Interface: New Directions for Designing Interactive Systems.* ACM Press.
16. Sellen, A., Kurtenbach, G., & Buxton, W. (1992). The Prevention of Mode Errors through Sensory Feedback. *Journal of Human Computer Interaction.* pp. 141-164.
17. Wobbrock, J.O., Myers, B.A., & Kembel, J.A. (2003). EdgeWrite: a stylus-based text entry method designed for high accuracy and stability of motion. *ACM UIST symposium on User interface software and technology.* pp.61-70.
18. Zhao, S. & Balakrishnan, R. (2004). Simple vs. compound mark hierarchical marking menus. *ACM UIST Symposium on User Interface Software and Technology,* pp. 33-44.