

User Learning and Performance with Bezel Menus

Mohit Jain

Department of Computer Science
University of Toronto
40 St. George St, Toronto, ON
mjain@cs.toronto.edu

ABSTRACT

Touchscreen phones tend to require constant visual attention, thus not allowing eyes-free interaction. For users with visual impairment, or when occupied with another task that requires a user's visual attention, these phones can be difficult to use. Recently, marks initiating from the *bezel*, the physical touch-insensitive frame surrounding a touchscreen display, have been proposed as a method for eyes-free interaction. Due to the physical form factor of the mobile device, it is possible to access different parts of the bezel eyes-free. In this paper, we first studied the performance of different bezel menu layouts. Based on the results, we designed a bezel-based text entry application to gain insights into how bezel menus perform in a real-world application. From a longitudinal study, we found that the participants achieved 9.2 words per minute in situations requiring minimal visual attention to the screen. After only one hour of practice, the participants transitioned from novice to expert users. This shows that bezel menus can be adopted for realistic applications.

Author Keywords

Bezel Menu; Eyes-Free Interaction; Touchscreen Phones; Text Entry.

ACM Classification Keywords

H.5.2 [Information Interfaces and Presentation]: User Interfaces - Graphical User Interfaces;

General terms

Design; Human Factors; Performance; Experimentation.

INTRODUCTION

Since the launch of the iPhone, touchscreen phones have dominated the smartphone market. Touchscreen phones allow for direct manipulation, but require constant visual attention, thus making eyes-free interaction difficult. For users with visual impairment, or when occupied with other tasks that require a user's visual attention causing situational impairments (such as attending a meeting, or walking through a shopping center) [29], touchscreen phones can be difficult to use. Furthermore, because touch input is overloaded in functionalities, it is hard to use without being able to see the interface. For example, when a user touches a

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI'12, May 5–10, 2012, Austin, Texas, USA.

Copyright 2012 ACM 978-1-4503-1015-4/12/05...\$10.00.

Ravin Balakrishnan

Department of Computer Science
University of Toronto
40 St. George St, Toronto, ON
ravin@dgp.toronto.edu

menu item, it can be either to select that item or to browse the list of menu items, which can cause mode switching problem. Finally, the small screen area limits the size devoted to user interface widgets, resulting in hard-to-hit, small targets. The fat-finger problem exacerbates the issue [13, 15].

Marking gestures provide a solution for eyes-free interaction, as they can be learned into the muscle memory [16]. However, gesture recognition requires a mode switch to distinguish pointing input from simple inking action, where a user input is by default treated as 'ink'. Recently, researchers [6, 27] have explored using the bezel of the phone for eyes-free interaction. *Bezel* refers to the physical touch-insensitive frame surrounding a touchscreen display (black-colored area in Figure 1a, 1b). Bezel menus are built on the periphery of the touch screen (*activation area* in Figure 1a) without occluding the screen space [27]. The initiation of a bezel gesture starts from outside of the screen (Figure 1b), solving the mode-switching problem. Bragdon *et al.* [6] showed that for bezel gestures users performed better in the eyes-free mode than when looking at the phone. They also found that mark-based bezel gestures as in marking menus [16] are faster and more accurate than free-form bezel gestures (including pigtail, spiral, Z, X, etc.) under different mobile and distracting environments.

In this research, we build upon this previous work [6] by exploring the range of possible mark-based bezel gestures, along with evaluating crucial aspects of bezel menus like threshold distance, feasibility of off-axis items and performance of different bezel menu layouts, which has not been studied before. Based on the obtained results, we developed a text-entry application to study the performance of bezel gestures in a real-world task. From the application, we

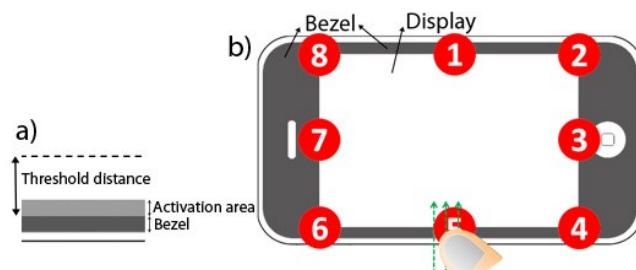


Figure 1. a) *Activation area* (grey) and *threshold distance*, b) Eight level-1 menu items, numbered 1 to 8 at bezel (black), and the start position of the finger to select item-5.

stroved to understand the applicability of bezel-based interactions for a specific type of scenario where the user can only devote varying levels of limited visual attention to the screen. The text entry user study demonstrates that the bezel menus are easy to learn and use in situations not permitting much visual attention to the screen. Text entry requires more than 26 targets on the screen which is hard to achieve. Also, users tend to prefer text entry with minimal visual attention [5, 28, 33, 35]. Using bezel menus, we made the 26 letters accessible by at most needing to access two levels of menu operations in an eyes-free manner. Further, typing requires the user to perform more input than other interactions (*e.g.*, typing a short response to an email requires more input than launching an application). Thus, being able to access various combinations of letters through the bezel menu system without needing to devote full visual attention towards the interface was used to evaluate the learning and performance of bezel menus. Through a longitudinal study, we show that after an hour of practice, the user's transition to expert user. They achieved 9.2 wpm (words per minute) in situations requiring minimal visual attention to the screen. We conclude with some design implications for bezel menus, such as accessibility considerations and optimal threshold distance for bezel invocation.

RELATED WORK

Bezel Menus

Bezel-initiated marks are a combination of crossing-based interfaces [1] and marking menus [16]. Crossing-based interfaces require crossing a boundary for menu selection. Marking menus [16], a variant of radial or pie menus [7], allow menu selection by either selecting an item from a popup radial menu appearing directly under the cursor or by drawing a mark in the direction of the desired menu item. Marking menus enable an efficient transition from novice to expert, as every time a novice user makes a mark to select an item from the popup menu; he/she is basically rehearsing the gesture that an expert user would have made without popping up the menu. However, on a small touchscreen phone, the finger can cause an occlusion, making it difficult to view the popup radial menu.

Bezel-initiated marks [6, 27] are similar to marking menus but instead of using the display screen for menu invocation, the phone's bezel is used. To select a menu item, a threshold distance starting from the bezel needs to be swiped, similar to a crossing-based interface. Both marking menus and crossing-based interfaces use the display area for menu invocation which might result in mode-switching problems. Invoking from the bezel helps in solving this issue. Simple marking menus [38], in which series of separate strokes (*e.g.*, two strokes for up-left) are used were shown to be more accurate and faster than hierarchical marking menu. However, similar to Bragdon *et al.* [6], we use hierarchical marks, as single marks require a 'transient mode' between the two strokes; a waiting time might be confusing for a user in the eyes-free mode.

The bezel space has been previously explored by other researchers [6, 11, 13, 27]. Barrier Pointing [11] relies on elevated bezels, and is designed for people having motor impairments. It uses the bezel as an end point with the inside of the display area as a starting point, contrary to the approach adopted by others [6, 13, 27]. This can result in a mode-switching problem. Manual Deskterity [13] uses the bezel menu to create a new object, implying making something out of nothing. Bezel Swipe [27] restricts the bezel usage to support multiple selections, cut, copy, paste and other similar operations. An item is selected by swiping the finger from the bezel to over the item and lifting the finger off the display, but the finger might occlude the item. Following Bragdon *et al.* [6], we use a crossing-based interface for selecting an item; hence finger occlusion has no influence. The bezel space has also been used for tasks not requiring visual attention [6], and it was found that users performed better in the eyes-free condition than looking at the phone. The recently unveiled BlackBerry PlayBook [4] has a touch-sensitive bezel which can detect many of the bezel interactions described above; however, our approach does not require any hardware changes.

Much of the previous work has explored bezel menus for specific applications contexts, such as creating a new object [13], multiple target selections [27]. Bragdon *et al.* [6] restricted their mark-based bezel gesture set to 12 axis-aligned marks. The general space of bezel-initiated marks has not been studied before.

Gesture-based Text Entry Systems

Although many gesture-based text entry systems have been proposed [23, 26, 33, 34, 35, 37], not many has been evaluated for eyes-free mobile phone usage scenario. Graffiti [33] uses free-form unistroke gestures resembling English letters. Because it leverages a skill that people are often familiar with, imaginably it can be operated eyes-free. Tinwala *et al.* [33] show that it supports a text entry speed of 7.6 wpm in the eyes-free mode. Bragdon *et al.* [6] confirm that mark-based gestures perform better than free-form gestures both in terms of speed and accuracy. T-Cube [34] uses simple mark-based gestures and supported ~20 wpm, but no eyes-free study has been reported for T-Cube. EdgeWrite [35] provides an easier gesture-set for text entry. It was built for people with motor impairments. It uses a physical square guide imposed over the usual text input area, and text is entered by traversing the square's edges and diagonals. EdgeWrite requires specialized hardware. It can potentially be useful in eyes-free scenarios; however, no evaluation of its potential in eyes-free usage scenarios has been reported. Yfantidis and Evreinov's [37] technique uses very simple directional gestures on a pie-menu of characters. Three out of 12 users were able to achieve 12 wpm after five trials on a touchscreen monitor; however, users' ability to achieve the same speed on a small phone screen has not been explored.

Eyes-Free Mobile-based Text Entry Systems

Prior to touchscreen phones, standard and QWERTY physical keypads were most common with the physical buttons providing tactile feedback, thus enabling users to develop

muscle memory over time and allowing for text entry with low visual attention. Most touchscreen phones support soft-keypads. Due to small screen size, the fat-finger problem, high visual demand, and lack of tactile feedback from physical keys, accessing virtual keypads is difficult [24].

The iPhone offers VoiceOver [3], which provides audio feedback to guide visually-impaired users to access the phone, including text entry using a soft keypad. It has been shown [5] that entering text is easy with VoiceOver, but the text entry speed is very low at 0.66 wpm. Recently, researchers have proposed techniques [5, 28, 33] enabling faster eyes-free text entry on touchscreen phones. No-Look Notes [5] and Mobile Messenger [28] divides the phone screen into 8 or more easy-to-reference parts, and allocates 3-4 letters to each part, with Text-to-Speech providing feedback. Performance of Mobile Messenger has not been reported, while No-Look Notes reports a low text entry speed of 1.32 wpm with minimal practice.

Most of the above text entry methods [3, 5, 28, 37] were developed for visually-impaired users and, thus rely on speech, auditory, or vibration feedback to input text. These systems were tested with either visually-impaired or blindfolded participants. In contrast, our research focuses on situation-impaired users. Everyone experiences situational vision impairments such as attending a meeting or walking through a busy shopping center, when paying high visual attention to the phone is difficult. This design paradigm differs in that users can look at the phone once in a while to get visual feedback, unlike visually impaired users. This design space has been previously explored [6, 18], but have not examined text entry specifically. Other text entry researches [12, 15] have mentioned this design space but none have conducted a longitudinal study exploiting situational impairment.

BEZEL MENU DESIGN

We divided the bezel into eight regions – 4 corners and 4 sides (Figure 1b), and associated different menu items with each region, numbered clockwise from 1 to 8, starting from the top. *On-axis items* are at the primary compass directions (N, E, S, W, numbered 1, 3, 5, 7), while *off-axis items* are those at 45° angles (NE, SE, SW, NW, numbered 2, 4, 6, 8) (Figure 3a). The 8 regions are distinguishable from each other, and allows for both portrait and landscape mode usage. It is also in accordance with prior work on marking menus [16], which states that optimum performance is achieved with 8 items. For generality, instead of directions we use numbers which do not have any existing mental mapping with the regions. This enabled us to study novice behavior.

To enter the selection mode, the user has to start swiping the finger from the bezel inwards to the touchscreen (Figure 1b). Although the finger is detected only when it touches the screen, swiping from the bezel makes sure that the first point of contact is detected on a very narrow edge of the screen, called the *activation area* (Figure 1a, 2a). The activation area is kept as a thin strip of 30 pixels around the periphery. Thus,

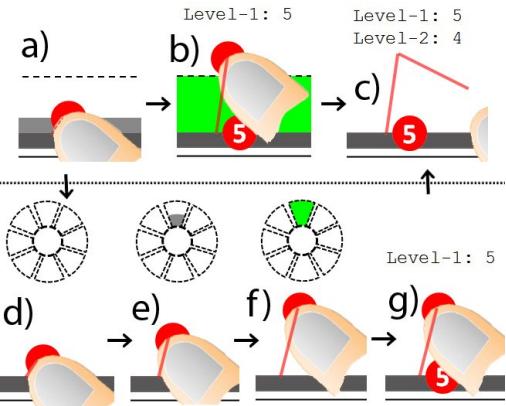


Figure 2. Expert usage: a) invocation of bezel menu selection from *activation area* (grey), b) crossed the threshold distance to select item-5 of level-1, c) swipe towards item-4 and lift the finger up, to select item-4 of level-2.

Novice usage: a) invocation of bezel menu selection from *activation area*, d) long pause and appearance of the pie menu, e) selecting item-5 from the pie menu, f) item-5 selected, g) screen showing item-5 is selected, c) swipe towards item-4 and lift the finger up, to select item-4 of level-2.

the bezel menu does not consume any accessible screen space, as it tries to exploit the unoccupied display areas which are not commonly used during regular interaction. Eight or fewer menu items can be accommodated in level-1 of the bezel menu design (Figure 1b, numbered 1 to 8). To select a *level-1* item, the finger entering from the bezel needs to cross a threshold distance (Figure 1a, 2b). For more than 8 menu items, a *level-2* selection is defined, such that after crossing the threshold distance in level-1, the user needs to swipe towards the correct intended direction, to select a menu item out of the 64 (8x8) possible items (Figure 2c). On lifting the finger from the screen, the item gets selected. For example, to select item-5 in level-1 and item-4 in level-2, the required mark is shown in Figure 2c. Level-2 items can be submenu for level-1 items; e.g., “*save*” is an example of a level-2 menu item under the “*file*” in level-1 menu.

Starting from the bezel leads to a temporal mode change, but the gesture ends as soon as the finger is lifted—causing no mode-switching related errors. To select a menu item again, the same process of swiping from the bezel needs to be repeated. Using a threshold distance to transition to level-2 removes ambiguity of sequential strokes in the same direction (e.g., selecting item-5 of level-1 followed by item 1 of level-2) which has been pointed as one of the limitations of hierarchical marking menus [38]. To reduce accidental bezel menu invocation, no item is selected until the threshold distance is not crossed.

To identify the optimal threshold distance such that the value is neither too small leading to accidental selection, nor too large resulting in tiring long marks, we conducted a study (*Pilot Study 1*) with 4 right-handed male graduate

students (average age 23.3 years). Only the eight level-1 items were used for this study. A cardboard arrangement was used to hide the phone screen. A random number between 1 and 8 was spoken by the phone, as the question. For a correct response, the participant needed to swipe his/her finger starting from the bezel corresponding to the spoken number and make a mark in the required direction. No threshold was used; hence any mark of any length initiated from the correct bezel was taken as a correct mark. A beep sound was played for mark initiation from the wrong bezel. Participants were instructed to keep trying until a correct response was received. The 8 items were randomized, forming a *set* of 8 tasks. Each participant completed 24 sets of trials (plus 2 practice sets). Thus, the total number of trials = 4 participants × 24 sets × 8 items = 768. Participants took 20 minutes to complete the study. Overall, the participants achieved a high accuracy with 97.33% of the marks drawn correctly on the first attempt. On average, the mark length was 70.02 pixels ($sd = 20.8$). For all further studies, 70 pixels have been used as the threshold distance.

Similar to [6], we did not provide any audio/vibration feedback, as we were not targeting the visually-impaired users. Also the user was allowed to look at the phone once in a while to get visual feedback, thus not necessitating the use of vibrotactile feedback. An ink trail, consistent with the metaphor of marking with a pen, is provided. The trails remain till the user no longer touches the screen. Andersen and Zhai [2] showed that an ink trail actually slows down the average speed in stroke production. However, we used the trails as it reinforces learning which helps in correcting errors for subsequent use. For smooth transitioning from novice to expert mode, marking menu based methodology is adopted. On swiping from the bezel, a long pause (of 1/3 sec) results in a pie menu appearing (Figure 2d-g) to enable selection. A similar pie menu also appears for level-2. Over time, the user can get used to making such bezel marks, and smoothly transitions to the expert mode of selecting items eyes-free. To solve the occlusion problem, the pie menu is displaced visually, such that it is always shown at the center of the screen, which also helps in avoiding the pie menu from getting clipped-off.

Kurtenbach [16] showed that the source of poor performance at higher breadths and depths is due to the selection of off-axis items. We conducted a study (*Pilot Study 2*) with the same 4 participants to check the feasibility of off-axis items of bezel-menu. The other objective of this study was to find, for level-1 off-axis items, which of the level-2 items – on-axis or off-axis

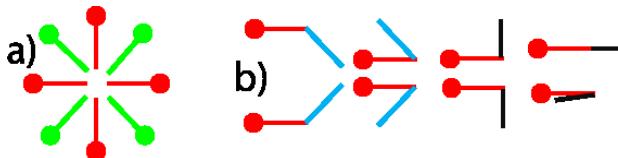


Figure 3. a) Level-1 marks: On-axis (red) and Off-axis (green), b) Level-2 marks for one of the level-1 on-axis mark; level-2 off-axis (blue), level-2 on-axis (black).

– performs better in terms of accuracy. Only the off-axis level-1 items (Figure 3a green) were used for this study. Level-2 resulted in two conditions – a) on-axis items (Figure 3b black), b) off-axis items (Figure 3b blue). The four level-1 items with four level-2 items form a *set* of 16 tasks. A randomly selected (without repetition) task (e.g., “5 → 4”, correct response Figure 2c) appears in the center of the screen and the participants were asked to keep on trying until a correct response was received. A beep sound was played for incorrect responses. Each participant took around 40 minutes to complete 16 trial sets (plus 2 practice sets) for each of the two conditions. So total number of trials = 4 participants × 2 conditions × 16 tasks × 16 sets = 2048. The participants were able to achieve high accuracy of 96.6%, showing that level-1 off-axis marks are feasible. Among the two conditions, participants made less errors per set in the level-2 on-axis condition ($m=2.06$, $sd=2.11$) compared to level-2 off-axis condition ($m=4.32$, $sd=4.75$). This result is in accordance with [38] that markings with no inflections can be ambiguous. Also, any kind of inflection after crossing the bezel can help in reducing unintentional menu invocation.

As the touch points are not uniformly distributed, the resample method from the \$1 recognizer [36] is used on the obtained touch-points. Identification of the level-1 item is dependent on the entry point where the finger first touches the screen, and the direction of the mark until the threshold distance is reached. Assuming the deflection point breaks a single stroke into two lines, we used a ‘two best fitting straight lines’ algorithm [17] to obtain the angle at the point of deflection for calculating the level-2 item. Similar to marking menus [16], all marks are treated as scale invariant.

BEZEL MENUS STUDY

We conducted a study to understand how different menu layouts, in terms of the number of items at each level, impact performance. We studied accuracy, types of errors, and length of the marks, to gain an insight into the feasible directions for eyes-free interaction. The study was conducted on an HTC Desire Android phone with a 533×320 pixels (3.7" diagonal) capacitive touchscreen.

Participants. Twelve right-handed graduate students (1 female) with an average age of 26.5 years ($sd=3.08$) participated. All but two had been using a touchscreen phone for more than a year. Participants self-reported an average 62.8 minutes per day ($sd=8.32$) of touchscreen phone usage. None reported interacting eyes-free with their touchscreen phone.

Study Design. Two different menu layouts were used: $L_{4 \times 8}$ and $L_{8 \times 4}$. In $L_{4 \times 8}$, level-1 consisted of only the four on-axis items (Figure 1a, item number 1, 3, 5 and 7; Figure 3a red-colored) and level-2 comprised of all the 8 items. $L_{8 \times 4}$ consists of all the 8 items for level-1, but only the *opposite axis* items for level-2 (e.g., Figure 3b blue), i.e., if the level-1 item is on-axis then the level-2 item will be off-axis and vice-versa. This decision of taking opposite axis is in accordance with the results obtained from Pilot Study 2. So both the menu layouts – $L_{4 \times 8}$ and $L_{8 \times 4}$ – have 32 items each, which we refer to as a *set*. Only $L_{4 \times 8}$ and $L_{8 \times 4}$ were used for

the study, because we believe that 32 options is a reasonable upper limit for most mobile applications, motivating these layout choices. For the same reason, we did not study the L_{8x8} (providing 64 items) layout.

Participants were given verbal instructions with demonstrations on how to perform the menu selection. To make the experiment ecologically realistic, the participants were free to hold the phone in any manner, though they were asked to use it in the landscape mode, using 2-hands. We believe that complex realistic operations on phones are preferable in landscape mode, using both the hands as it allows easy access to all the parts of the screen. The participants were asked to perform the task as fast and as accurately as possible. At the completion of the task, participants were briefly interviewed about their preference and the problems faced.

A random menu choice (same as Pilot Study 2) appears in the center of the screen. It was ensured that no two same menu choices appeared consecutively. A beep sound was played for each wrong gesture, and the participants were asked to keep trying until the correct gesture was received. A long pause resulted in a pie menu appearing. The task started with performing a trial session with 2 sets each of L_{4x8} and L_{8x4}. Next the participant performed 4 sets each of L_{4x8} and L_{8x4} with the phone screen visible (*eyes-on mode*), and 4 sets each of L_{4x8} and L_{8x4} in the *eyes-free mode*. In the eyes-free mode, a cardboard arrangement was used to hide the phone screen, and audio-based question and feedback were provided. The phone's inbuilt-speaker was used for the audio cues. For complete within-subjects counter-balancing, an ABBA design was used such that each participant completed 2 sets each of L_{4x8} and L_{8x4} in eyes-on mode, followed by 4 sets in eyes-free mode, and then again 2 sets in eyes-on mode. The order of presentation of the menu layouts were L_{4x8} followed by L_{8x4}, as starting with the four on-axis bezels, provides a better learning curve. Each participant took around 1.5 hours to complete the study, with breaks in between sets. Total number of trials in eyes-free mode = Total number of trials in eyes-on mode = 12 participants × 8 sets × 32 items = 3072.

Results. We define accuracy as the number of times the correct mark was performed on the first attempt. In terms of accuracy per set, participants performed significantly better in the eyes-free mode ($m=29.3$, $sd=2.37$) compared to eyes-on mode ($m=26.67$, $sd=2.2$) with $t_{11}=4.2$, $p<0.005$. This is not surprising, as similar results were obtained by [6]. This confirms that bezel-initiated marks are better suited for eyes-free interaction. The post-test interview revealed the same as all participants preferred the eyes-free mode, with comments such as “*It is easy to memorize*”, “*Eyes-on mode is surprisingly tougher than expected, whereas eyes-free mode is surprisingly easier than expected*.”

In the eyes-free mode, L_{8x4} performed better with 95.8% accuracy ($m=30.67$, $sd=1.5$), compared to L_{4x8} 87.5% ($m=28$, $sd=2.37$). Multiple errors were recorded for a single trial, as the participants needed to try until the correct mark

was received. A paired t-test analysis of the number of errors per set in the two layouts shows L_{8x4} ($m=5.27$, $sd=2.23$) performed significantly better than L_{4x8} ($m=8.32$, $sd=3.67$) with $t_{11}=4.44$, $p<0.001$. Three of the participants complained that “*there was no direct mental model associated with the numbers and the bezel menu items*”; hence it required “*thinking before doing, leading to lag and more errors*.”

For the L_{8x4} eyes-free mode, the errors were further analyzed and it was found that marks originating from the upper left corner (number 8 in Figure 1b) contributed to the majority of the total errors (26.6%). This might be because all participants were right-handed. Out of the 32 items, 1→2, 1→4, 2→7, 3→8, 8→1, 8→5, and 8→7 (here x→y refers to item x in level-1 and item y in level-2) were found to be most inaccurate, and together contributed to 44.7% of the total L_{8x4} eyes-free errors (Figure 4).

The error results were also dependent on the way the phone was held. Nine participants used both thumbs to perform the task, while the remaining held the phone with their left hand and used the right-hand index finger for interaction. These three participants were among the four most inaccurate, and contributed to 41.2% of the total errors, much more erroneous than participants using both thumbs.

In the eyes-free mode, the average length of the marks for L_{4x8} was 374.9 pixels ($sd=68.2$), and for L_{8x4} was 406.87 pixels ($sd=80.78$). Marks, in which the correct response was received on the first attempt, were only considered. This shows that participants were making long marks. We believe that similar to marking menus, the mark length will reduce as user's transition from novice to expert.

TEXT ENTRY SYSTEM DESIGN

Based on the results obtained from the bezel menu study, we designed a text entry application to evaluate the learning and performance of bezel-initiated marks in a realistic application scenario (as opposed to a contrived menu-selection task). Using text entry to study the performance of bezel menu for realistic operations was a conscious decision. Text entry requires more than 26 targets on the screen which is hard to achieve. Also users tend to prefer text entry with minimal visual attention [5, 28, 33, 35]. Using bezel menus, we made the 26 letters accessible by at most needing to access two levels of menu operations in an eyes-free manner.

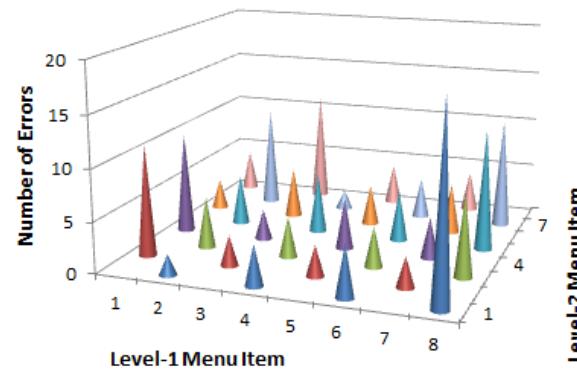


Figure 4. L_{8x4} eyes-free mode error analysis.

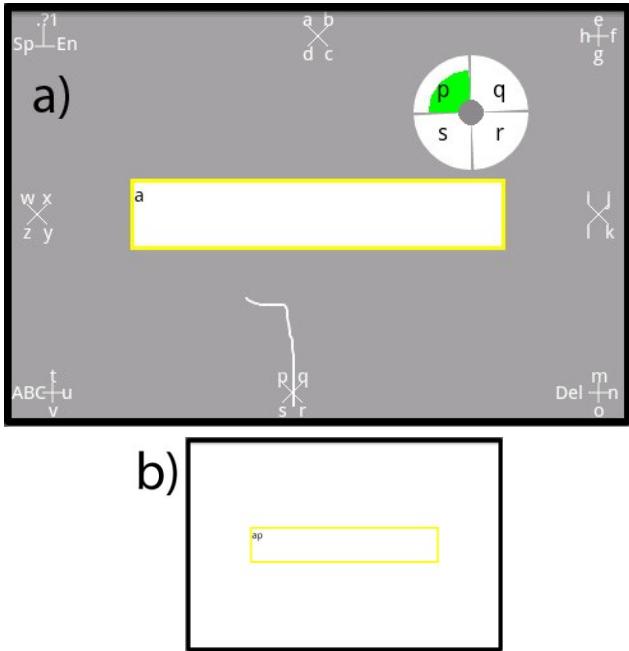


Figure 5. a) Text entry application showing the letter layout and pie menu appearing on long pause, **b)** The default state of the screen. The alphabetical layout is visible only when a user taps on the textbox, situated at the center of the screen. After entering a letter, the layout disappears.

We used the L_{8x4} layout to design the system as it was found to be more accurate. A letter is assigned to each item of the L_{8x4} menu layout (Figure 5a). An alphabetical layout is used to minimize the learning curve. Initially, we thought of using the T9-design [32], but due to the limitation of 8 bezels, we designed a modified version of T9.

On every bezel, the layout of the block of letters is alphabetical with letters in the clockwise direction starting from top/top-left (Figure 5a). For letters ‘a’ to ‘l’, we assigned four letters to each bezel such that the first letter of the bezel is a vowel, which would help the user to remember the layout. The combined frequency of vowels is 37.98% [10]. For ‘m’ to ‘z’, we used the same layout as T9. This design allowed us to effectively distribute the functional characters such as ‘Del’ (Delete), ‘Sp’ (Space), ‘Ent’ (Enter), ‘ABC’ (CapsLk), and ‘.?1’ (Punctuations and Numbers). We assigned ‘Sp’ and ‘Del’ to different thumbs, as confusing the two might result in many mistakes. As marks originating from bezel number 8 are highly inaccurate (results from Pilot Study 2), least-frequently used items were assigned to it. Note that our goal is not to design the best layout for an eyes-free text entry system, but to gain insight in the performance of a bezel-based system for a realistic task requiring minimal visual attention.

Based on the results of Pilot Study 2, a mark requires an inflection to select a letter. As the system is designed for right-handed users, most of the high frequency letters are assigned to the right thumb. For left-handed users, a mirror image of

this layout can be used. The *textbox* in the center of the screen contains the text being typed so far. To minimize screen usage, the visual hint showing the letters layout and the ink trail are not always visible (Figure 5b). A user can tap on the *textbox* to view the layout (Figure 5a), and after entering a letter with visible ink trail, the layout disappears (Figure 5b). The default state of the screen is shown in Figure 5b. This provides an added benefit that users would not need to switch between the content screen and keypad screen. While the layout is visible, a long pause results in the appearance of a pie menu (Figure 5a). Unlike Bonner *et al.* [5], no audio feedback is provided, as we are not designing for visually impaired people, but for situational impairments.

TEXT ENTRY STUDY

Participants

Eight right-handed participants (1 female, mean age 27, $sd=3.38$) were recruited from the local university campus. Only two were native English speakers. Participants self-reported an average 9.8 hours ($sd=3.52$) of daily computer usage, though only five of them perform touch-typing. Their average computer typing speed is 50.75 wpm ($sd=22.7$) with a high accuracy of 98.25%. Except one, all had previously used a touchscreen phone on a regular basis for 6 months or longer. None of them reported ability to interact eyes-free with their touchscreen phone. Each participant was compensated \$10 per session.

Study Design

All the participants participated in eleven 40-minutes sessions spread over a period of 15 days, with each session separated by at least twelve hours and no more than two days. Replicating the real-world scenario, where a typical user looks at his/her cellphone in burst of 4 sec [25], we decided to use a visual distracter task to assess situational vision impairment usage. Similar to [18], as a visual distracter, participants were required to repeat verbally a periodically-changing random number (between 1 and 9). This tries to replicate typical low attention mobile usage. To avoid confusion, we ensured that no two same numbers appeared consecutively. The number of times participants missed/misspoke a number was recorded.

The sessions were divided into three distraction-level: (a) *no distraction*: no visual distracter was used (session 1-4 and session 11 which was the last session and intended to see if the other intervening conditions resulted in changed behavior for the base condition), (b) *low distraction*: random numbers changing periodically at 5 sec interval (session 5-7), and (c) *high distraction*: numbers changing at 2.5 sec interval (session 8-10). The distraction levels were based on previous research [25], and were not randomized for the study so that all the participants have similar learning curve. For the study, the same phone HTC Desire was used, with all the hard buttons being disabled. The experiment was conducted only with the lower case letters and with no punctuations and numbers, so ‘ABC’ and ‘.?1’ were disabled, and hence not discussed further.

Each participant was given verbal and visual instructions describing the task and goal of the experiment. The researcher demonstrated how to hold the phone and enter text – holding the cellphone with both hands in the landscape mode, and using both the thumbs for text entry. Unlike Bragdon *et al.* [6], we opted for 2-hand usage, as it allows easy access to all the 8 bezels, as confirmed in the previous study. Also we believe that realistic operations such as text entry benefit from two-hand usage for optimum performance, as two-thumb typing using mini-QWERTY has been studied and found to support 60 wpm, much faster than any one-handed technique [9]. Participants were told that the layout is alphabetical and for each block of letters assigned to a bezel, the letters are in clockwise direction starting from top/top-left. A mark initiated from an on-axis bezel must end towards an off-axis bezel, and vice-versa, to input a letter (Figure 5a). The positions of the vowels were stressed for a quick start; also position of ‘Del’ and ‘Space’ was emphasized. Participants were free to tap on the textbox to view the letter’s layout.

A within-subject design was used. For every session, participants were asked to enter as many phrases as possible in 30-minutes duration, similar to [9]. The instruction was to “enter the text as quickly and accurately as possible, as if typing an email to a colleague.” No error correction was enforced. At the start of every session, the participants were required to type the practice sentence (“the quick brown fox jumps over the lazy dog”) twice. Participants were encouraged to take a short break between phrases, anytime they wished. They were facing two monitors, one showing the presented phrase and other showing the periodically changing random number. For sessions with visual distraction, the participants were explicitly told that the primary task was to speak out the appearing number (1-9), and the secondary task was to enter text. Participants were seated on a reclining chair and were asked to adjust the height of the chair to their comfort before starting the task.

The phrases were randomly chosen from a published phrase set for mobile phone text entry by Wobbrock [31], with no phrase longer than 27 characters in length. All the participants received the phrases in the same order. Participant’s face including eye movements was video-recorded for all the sessions, totaling 58 hours 12 minutes of video, which was analyzed offline.

Results

All eight participants completed 11 sessions. In total, participants entered 4385 phrases (1825 phrases in 5 sessions with no distraction, 1252 phrases in 3 sessions with low distraction, and 1308 phrases in 3 sessions with high distraction).

Speed. We measured text entry speed in terms of words per minute (*wpm*), calculated using (characters per second)*60/5, with the definition that a word consists of 5 characters [22]. The average text entry rates over all the sessions across all the participants were 8.47 wpm (*sd*=1.99). The mean speed for session 1 were 4.86 wpm (*sd*=0.98), while for session 11 were 12.08 wpm (*sd*=0.83), clearly showing that performance in-

creased with practice. Approximately 8 wpm was achieved just after 1 hour of usage, showing a short learning curve and quick transition from novice to expert user. The effect of introducing distractors is visible in Figure 6a, as there was a significant decrease in the text entry speed during session 5 and session 8, as low distraction and high distraction were introduced in these respective sessions. Also, if the curve (Figure 6a) from session 1-4 is extended to join the average-wpm of session 11, the curve obtained is very similar to the characteristic learnability curve for text entry system [19]. This shows that learning was not affected much by distracter.

We conducted a Distraction-level(3) × Day(3) RM-ANOVA (Repeated Measures Analysis of Variance) on the text entry speed, with the no distraction condition as baseline. Session 1 and session 11 data were dropped, thus allowing us to analyze three days of data for each distraction-level. The result shows a significant effect for distraction-level ($F_{2,14}=8.58$, $p<0.005$) and for day ($F_{2,14}=42.2$, $p<0.0001$). To analyze this further, we conducted a post-hoc analysis with Bonferroni correction. On pairwise comparisons, significant differences ($p<0.05$) were observed between no distraction ($m=7.75$, $sd=1.22$) and low distraction ($m=8.57$, $sd=1.06$) (Figure 6a). Among the three distraction conditions, the participants performed fastest at 9.07 wpm ($sd=1.35$) with high distraction. This can be attributed to the slow development of the muscle memory over time. Also the reason could be that higher distraction requires higher attention, resulting in high speed.

The recorded videos were analyzed at the frame level by an independent coder, to reduce the possibility of bias. Every time a participant looked at the phone and the associated duration were noted. During session 10, the average time that the participants looked at their phone was 3.1 minutes ($sd=2.3$), proving that the experiment successfully captured the minimal visual attention scenario. A high speed of 9.2 wpm was maintained during session 10. It shows that users can enter text comfortably while paying minimal attention to the screen. To generalize, using bezel-initiated marks a user can interact with realistic touchscreen applications, even during minimal visual attention scenario.

Intra-time is the time taken to perform a mark, measured as the time difference between finger-down and finger-up; while *Inter-time* is the time in between two consecutive marks. Intra-time averaged over all the sessions and participants was 469.6 ms (*sd*=52.03), and Inter-time was 821.48 ms (*sd*=219.1). For session 11, the average Intra-time was 381.8 ms and Inter-time was 521 ms. This shows that both, time taken to perform a mark and time taken to think before making a mark, reduce with practice over time. These values are comparable with [6], which reports that the mean completion time of bezel marks was 1092.87 ms. Also, marking menu’s response time was 1.69 sec [16], while for crossing-based menu, the average response time was 0.9 sec [1].

Interestingly, a strong correlation was found between the bezel-based text entry speed and the participants’ computer typing speed with Pearson’s $r(6)=0.79$, $p<0.05$. The average length of the marks was 235.3 pixels (*sd*=42.01).

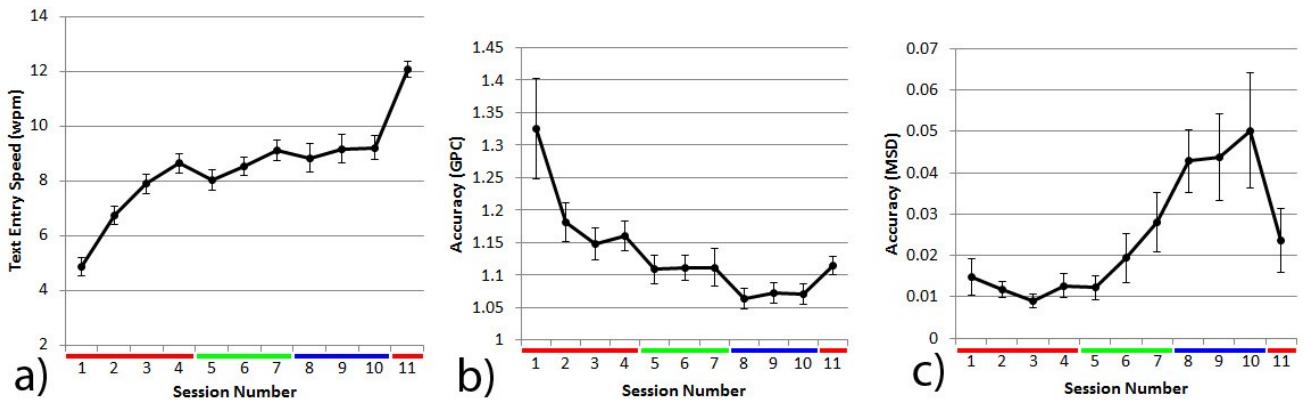


Figure 6. a) Speed: Words per minute (wpm), b) Accuracy: Gesture per Character (GPC), c) Accuracy: Minimum String Distance (MSD). (Red: No Distraction, Green: Low Distraction, Blue: High Distraction)

Our technique performed comparably in terms of speed (session 5: 8.02 wpm, session 10: 9.2 wpm, session 11: 12.08 wpm), compared to other existing techniques such as VoiceOver (0.66 wpm) [3], No-Look Notes (1.32 wpm) [5], and Graffiti (7.6 wpm in eyes-free [33], 11.4 wpm in eyes-on [8]). Compared to Graffiti and Unistroke (15 wpm in eyes-on [8]) where users are required to make free-form gestures, our technique uses basic marking gestures which might have contributed to the speed. The video analysis confirms that not much visual attention is required to enter text with our bezel-based technique. However, we recognize that a direct comparison is not fair, as previous work [3, 5, 33] were tested for complete eyes-free usage.

Accuracy. Two metrics were used to measure text entry accuracy: (a) Gestures per Character (GPC) [20] – the number of gestures (marks, for our case) required to input a character, (b) Minimum String Distance (MSD) [30] between the presented and transcribed phrase. MSD accounts for the uncorrected errors in the final transcribed text; while GPC measures the corrected errors as every correction adds multiple gestures, *i.e.*, delete character, re-enter character. The calculation of GPC also includes non-character producing marks, which constitutes only 1.72% of the total marks. Please note that a low value of GPC, close to 1, is considered as the ideal value.

Out of the total 1.72% of non-character producing marks, 0.15% consists of tap used to view the alphabet-layout, showing that on-screen visible menu items is not necessary as users can easily remember after practice; 0.07% marks missed the bezel activation area, highlighting that users can comfortably access all the 8 bezels; and the remaining 1.5% marks consists of such marks where the threshold distance was not crossed in spite of correct bezel initiation, hinting that the chosen threshold distance works. This could be because as user speed increases, he/she tends to make smaller marks, and thus missed the threshold distance. An adaptive threshold distance might work better.

A Distraction-level(3) \times Day(3) RM-ANOVA of GPC shows a main effect for distraction level ($F_{2,14}=13.6$,

$p<0.001$), but no significant effect for days. A post-hoc analysis with Bonferroni correction showed that for GPC there was significant difference between no distraction ($m=1.16$, $sd=0.02$) and high distraction ($m=1.06$, $sd=0.01$), $p<0.05$ (Figure 6b). Even MSD of no distraction ($m=0.011$, $sd=0.001$) and high distraction ($m=0.046$, $sd=0.01$) have a main effect of $p<0.01$, showing that participants were correcting significantly more mistakes in the no distraction mode compared to the high distraction mode. Interestingly, on a pairwise comparison, MSD for different distraction-level were significantly different from each other ($p<0.01$) (Figure 6c).

Speed, GPC and MSD were co-related, as for session 8-10, the speed and MSD increased considerably while GPC dropped. This shows that as participants gain speed, error correction was minimized, highlighting the speed-accuracy trade-off. For session 10, three participants have high MSD (above 0.05) which could be because of high speed, high distraction level, novice user, and/or non-native speaker [14].

The average number of misses in speaking out the number aloud was 1.79 ($sd=2.1$) in low distraction mode, and 6.6 ($sd=6.3$) with high distraction. In low distraction, the number changes 360 times in a session, and 720 times in high distraction. Hence the number of misses is not significant.

Confusion Matrix. A confusion matrix represents the frequency of character-level errors [21]. Figure 7 shows the total number of times when a presented letter (horizontal axis) was transcribed with an incorrect letter (depth axis). The most prevalent mistakes were: transcribing ‘h’ instead of ‘e’, and ‘j’ instead of ‘i’. Both of these confusing-letters are adjacent to each other in their respective bezel. These errors can be either due to incorrect gesture by the participants, or incorrect gesture-recognition by the application, it is hard to differentiate them. The number of wrong gestures for ‘e’ only constitute 3.3% of the total gestures for ‘e’ (for ‘i’ it constitutes 4.5%). However a better recognition system might improve the performance. Also, as most of the errors in the confusion matrix are neighbors with the same

level-1 item, in future we can build error correction into the system to boost the text entry performance.

Qualitative Feedback. At the end of session 11, participants were asked to fill a questionnaire with questions rated on a Likert scale (1 = strongly disagree, 5 = strongly agree) and free-form questions about the limitations of the current system and suggestions for improvement. High ratings were obtained for ease of learning ($m=4.25$, $sd=0.7$), ease of usage ($m=3.37$, $sd=0.8$), fun to use ($m=3.5$, $sd=0.9$), and high speed of text entry ($m=3.5$, $sd=0.8$). Easy to undo mistakes, received the lowest rating ($m=2.75$, $sd=1.16$). All, except one, were affirmative of using this technique for text entry, if it were available as an app. Participants like the technique: “*no need to worry about the small [QWERTY] keys*”, “*eyes-free is possible*”, “*faster*”, “*quick to learn*”, and “*mechanical, patterns such as ‘the’ are very easy to perform eyes-free, once learnt*”.

Most participants were of the opinion that the alphabetical keypad layout was easy to learn and easy to remember, though three participants did complain about the positioning of the Delete key. Only one participant was of the opinion that a single hand usage would be better. We believe that the system can be easily modified for a single hand use scenario. Three participants complained that the task was “*taxing on the mind* (in the distracter condition)”. On the contrary, on average, participants took only 0-3 breaks during a session, hinting that the task was not too tiring.

Our current system lacks a cursor, so a correction requires deleting even all the correct letters that were typed after a mistake. Apart from the cursor, participants asked for features such as auto-correction, audio/vibration feedback, and a better gesture recognizer.

DESIGN IMPLICATIONS

Accessibility. The facilitator observed that all the participants, except one, used their right-thumb to access the bezels containing letters ‘abcd’ and ‘pqrs’, in spite of these bezels being equidistant from both the thumbs. Participants stated that they were more comfortable accessing a bezel with their right-thumb. In the future, such keypad layouts which place the higher frequency letters to the bezels which

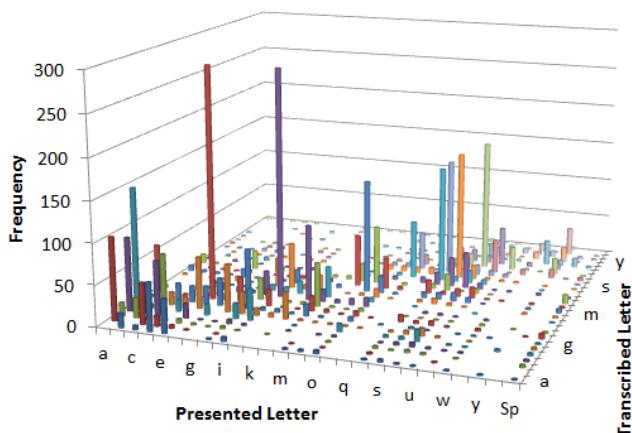


Figure 7. Confusion Matrix.

are easily accessible by the right thumb (*i.e.*, bezel number 1 to 5 in Figure 1b) can be explored, but that might result in a steeper learning curve.

To generalize, for building any novel bezel-based application, bezel numbered 1 to 5 are preferable; these bezels would work well in both landscape and portrait mode. Left-hand users should also be taken into consideration. One way to achieve this is to provide a way to switch the position of the menu items to its mirror image.

Preferable Layout. Based on the results, if the number of menu items is 8 or fewer, a level-1 bezel with 8 items could be used, as participants made very few level-1 errors. If menu items are 16 or fewer, an L_{4x4} design could be used with level-1 on-axis items and level-2 off-axis items. For 32 items or fewer, an L_{8x4} design (discussed earlier) could be used. If possible, items at bezel 8 should be avoided, and preference should be given to bezel number 1 to 5 (Figure 1b). For more than 32 options, an L_{8x8} design with all the items in level-1 and level-2 can be used, providing 64 items. As noted earlier, on-screen menus are not necessary and can be shown only when required, providing more screen space for the actual content.

Ergonomics. Accessing a phone’s bezel is dependent on the ergonomics of the phone. For example, a few Android phones provide touch-sensitive buttons at bezel number 3. In such phones, that specific bezel cannot be used for initiating a bezel menu. Further investigation is needed to use the bezel technique for other touchscreen-based handheld devices having different screen size.

Threshold distance. The threshold distance is dependent on the screen size. While designing a bezel-based application, a pilot study to find the optimal threshold distance should be considered. We used ~1/7th of the screen size as the threshold distance, and achieved low error rate. An adaptive threshold distance might work better.

CONCLUSION AND FUTURE DIRECTIONS

Bezel menus enable interaction with a touchscreen phone with minimal visual attention, along with solving the occlusion and mode-switching problem. They ameliorate the fat-finger problem. Marks do not have to be very precise. Bezel menus can work under direct sunlight, when it is difficult to access the on-screen controls. They can make the display icon-free, resulting in more screen space for the actual content. Complex realistic applications such as video editor, word processor, text entry, which requires numerous controls along with large content viewing area can take advantage of bezel menus. One of the demerits is that the number of menu items is limited to 64, and only 32 for best performance, but we believe that 32 menu items is a reasonable upper limit for most mobile applications. Also users would need to learn different command sets for different applications, but with regular practice, accessing frequently-used items eyes-free would be achievable.

The study shows that highly accurate eyes-free interaction is achievable with L_{8x4} layout. To gain insight into the per-

formance of a bezel-based system we developed a bezel-based text entry technique. We found it to be competitive with existing techniques in terms of speed, accuracy, and ease of learning and usage. This shows that bezel-initiated marks can be used to interact with realistic touchscreen applications, while paying minimal visual attention to the screen. While encouraging, these results must be interpreted with caution. The small sample size, non-native speakers as participants, limited our analyses. More participants are required to make a stronger claim.

As the accuracy of originating the mark from the correct bezel is very high, different variations of bezel menu such as (a) both level-1 and level-2 marks starting from the bezel similar to simple marking menus [38], and (b) marks starting and ending at the bezels, are worth exploring. Bezel menu can provide a 2-layer interaction on a touchscreen phone, as the first layer can be on-screen controls, and the second layer of menus can be pulled out from the bezel. The obtained results are not limited to text entry, and can be readily applied to other applications. We hope that our work will inform future designers to design better bezel-based interaction techniques.

REFERENCES

1. Accot, J. and Zhai, S. More Than Dotting the i's - Foundations for Crossing-based Interfaces. *CHI 2002*, ACM, 73-80.
2. Andersen, T.H. and Zhai, S. "Writing with Music": Exploring the Use of Auditory Feedback in Gesture Interfaces. *ACM Transactions on Applied Perception* 7, 3 (2010), 1-24.
3. Apple - Accessibility - iPhone - VoiceOver. <http://www.apple.com/accessibility/iphone/vision.html>
4. PlayBook. <http://us.blackberry.com/playbook-tablet/>
5. Bonner, M.N. *et al.* No-Look Notes: Accessible Eyes-Free Multi-touch Text Entry. *Pervasive 2010*, 409-426.
6. Bragdon, A. *et al.* Experimental Analysis of Touch-Screen Gesture Designs in Mobile Environments. *CHI 2011*, ACM, 403-412.
7. Callahan, J. *et al.* An Empirical Comparison of Pie vs. Linear Menus. *CHI 1988*, ACM, 95-100.
8. Castellucci, S.J. and MacKenzie, I.S. Graffiti vs. Unistrokes: An Empirical Comparison. *CHI 2008*, ACM, 305-308.
9. Clarkson, E., Clawson, J., Lyons, K. and Starner, T. An Empirical Study of Typing Rates on mini-QWERTY Keyboards. *Ext. Abstracts CHI 2005*, ACM, 1288-91.
10. Concise Oxford Dictionary Letter Frequency: <http://www.oxforddictionaries.com/page/133>
11. Froehlich, J., Wobbrock, J. and Kane, S. K. Barrier Pointing: Using Physical Edges to Assist Target Acquisition on Mobile Device Touch Screens. *ASSETS 2007*, ACM, 19-26.
12. Goldberg, D. and Richardson, C. Touch-Typing with a Stylus. *CHI 1993*, ACM, 80-87.
13. Hinckley, K. *et al.* Pen + Touch = New Tools. *UIST 2010*, ACM, 27-36.
14. Isokoski, P. and Linden, T. Effect of Foreign Language on Text Transcription Performance: Finns Writing English. *NordiCHI 2004*, ACM, 109-112.
15. Kristensson, P. and Zhai, S. SHARK²: A Large Vocabulary Shorthand Writing System for Pen-based Computers. *UIST 2004*, ACM, 43-52.
16. Kurtenbach, G. and Buxton, W. The Limits of Expert Performance Using Hierarchic Marking Menus. *INTERCHI 1993*, ACM, 482-487.
17. Least Squares Fitting. <http://mathworld.wolfram.com/LeastSquaresFitting.html>
18. Li, F.C.Y. *et al.* Virtual Shelves: Interactions with Orientation Aware Devices. *UIST 2009*, ACM, 125-128.
19. Lyons, K., *et al.* Twiddler Typing: One-Handed Chording Text Entry for Mobile Phones. *CHI 2004*, ACM, 671-678.
20. MacKenzie, I.S. KSPC (Keystrokes per Character) as a Characteristic of Text Entry Techniques. *HCI with Mobile Devices 2002*, Springer-Verlag, 195-210.
21. MacKenzie, I.S. and Soukoreff, R.W. A Character-level Error Analysis Technique for Evaluating Text Entry Methods. *NordiCHI 2002*, ACM, 243-246.
22. MacKenzie, I.S. and Soukoreff, R.W. Text Entry for Mobile Computing: Models and Methods, Theory and Practice. *HCI 17*, (2002), 147-198.
23. Mankoff, J. and Abowd, G.D. Cirrin: A Word-level Unistroke Keyboard for Pen Input. *UIST 1998*, ACM, 213-214.
24. McGookin, D., Stephen, B. and WeiWei, J. Investigating Touchscreen Accessibility for People with Visual Impairments. *NordiCHI 2008*, ACM, 298-307.
25. Oulasvirta, A., Tamminen, S., Roto, V. and Kuorelahti, J. Interaction in 4-second Bursts: the Fragmented Nature of Attentional Resources in Mobile HCI. *CHI 2005*, ACM, 919-928.
26. Perlin, K. Quikwriting: Continuous Stylus-based Text Entry. *UIST 1998*, ACM, 215-216.
27. Roth, V. and Turner, T. Bezel Swipe: Conflict-Free Scrolling and Multiple Selection on Mobile Touch Screen Devices. *CHI 2009*, ACM, 1523-1526.
28. Sanchez, J. and Aguayo, F. Mobile Messenger for the Blind. *ERCOM 2007*, Springer-Verlag, 369-385.
29. Sears, A. *et al.* When Computers Fade: Pervasive Computing and Situationally-induced Impairments and Disabilities. *HCI: Theory and Practice 2003*, 1298-1302.
30. Soukoreff, R.W. and MacKenzie, I.S. Measuring Errors in Text Entry Tasks: An Application of the Levenshtein String Distance Statistic. *Ext. Abstracts of CHI 2001*, ACM, 319-320.
31. StreamAnalyzer. <http://depts.washington.edu/ewrite/downloads/texteval.zip>
32. T9 The Global Standard for Mobile Text Input. <http://www.nuance.com/for-business/by-product/t9/index.htm>
33. Tinwala, H. and Mackenzie, I. S. Eyes-free Text Entry on a Touchscreen Phone. *TIC-STH 2009*, IEEE, 83-89.
34. Venolia, D. and Neiberg, F. T-Cube: A Fast, Self-Disclosing Pen-Based Alphabet. *CHI 1994*, ACM, 265-270.
35. Wobbrock, J.O., Myers, B.A. and Kembel, J.A. EdgeWrite: A Stylus-based Text Entry Method Designed for High Accuracy and Stability of Motion. *UIST 2003*, ACM, 61-70.
36. Wobbrock, J.O., Wilson, A.D. and Li, Y. Gestures Without Libraries, Toolkits or Training: a \$1 Recognizer for User Interface Prototypes. *UIST 2007*, ACM, 159-168.
37. Yfantidis, G. and Evreinov, G. Adaptive Blind Interaction Technique for Touchscreens. *Universal Access in the Information Society* 4, 4 (2006), 328-337.
38. Zhao, S. and Balakrishnan, R. Simple vs. Compound Mark Hierarchical Marking Menus. *UIST 2004*, ACM, 33-42.