Escape-Keyboard: A Sight-Free One-Handed Text Entry Method for Mobile Touch-Screen Devices

Nikola Banovic, Department of Computer Science, University of Toronto, Toronto, Canada

Koji Yatani, HCI Group, Microsoft Research Asia, Beijing, China

Khai N. Truong, Department of Computer Science, University of Toronto, Toronto, Canada

ABSTRACT

Mobile text entry methods traditionally have been designed with the assumption that users can devote full visual and mental attention on the device, though this is not always possible. The authors present their iterative design and evaluation of Escape-Keyboard, a sight-free text entry method for mobile touch-screen devices. Escape-Keyboard allows the user to type letters with one hand by pressing the thumb on different areas of the screen and performing a flick gesture. The authors then examine the performance of Escape-Keyboard in a study that included 16 sessions in which participants typed in sighted and sight-free conditions. Qualitative results from this study highlight the importance of reducing the mental load with using Escape-Keyboard to improve user performance over time. The authors thus also explore features to mitigate this learnability issue. Finally, the authors investigate the upper bound on the sight-free performance with Escape-Keyboard by performing theoretical analysis of the expert peak performance.

Keywords: *Devices, Escape-Keyboard, Mobile Text Entry, Sight-Free, Touch-Screen*

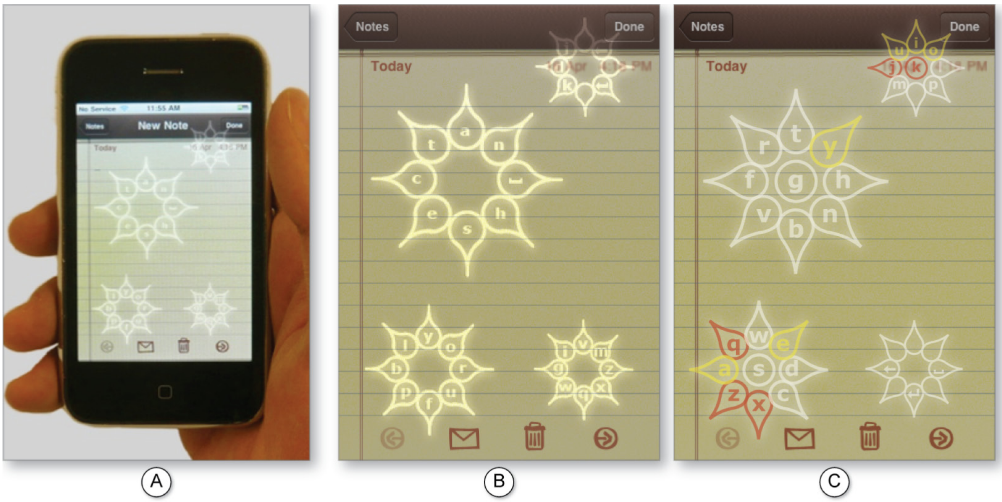
INTRODUCTION

There are many situations in which the user cannot devote full visual attention towards her actions on a mobile device (Brewster *et al.*, 2003). For example, social protocols often discourage multi-tasking during certain situations (*e.g.*, business meeting); as a result, the user often must reply to emails in a discrete manner, if done at all. The high visual demand of some tasks or environments also might cause a *situationally-induced impairment* (Sears, Lin, & Jacko, 2003) which, for example, could prevent the user from quickly sending a message while walking. Thus, the ability to interact with mobile devices sight-free could allow the user to complete important tasks without needing to pull much visual attention away from the main task towards the text entry method.

We present our iterative design of Escape-Keyboard, a soft keyboard which supports sight-free text entry on mobile touch-screen devices without requiring any hardware modification to the device (Figure 1a). We designed Escape-

DOI: 10.4018/jmhci.2013070103

Figure 1. (a) The Escape-Keyboard. (b) Initial and (c) final layouts determined through our iterative design process.



Keyboard for one-handed use, where the user enters text using the thumb of the hand holding the device. Based on the results from two small formative studies, we determined a layout with four target regions on the screen (Figure 1b). The user can place her thumb easily in these regions without looking at the device or needing any physical guides. To type a letter, the user touches one of the four regions, and then performs a flick gesture in one of eight directions as indicated by the Escape icon (Yatani *et al.*, 2008).

We examined the performance of Escape-Keyboard through two experiments and a theoretical analysis. The first evaluation included 16 sessions in which participants typed in sighted and sight-free conditions; its results highlighted the importance of reducing the mental load with using Escape-Keyboard to improve user performance over time. We then implemented and evaluated features specifically designed to mitigate the issue caused by the user's unfamiliarity with Escape-Keyboard. Finally, to examine the upper bound of the performance, we conducted a theoretical analysis of the entry speed of Escape-Keyboard. Our work demonstrates that marking gestures can be used from four specific target areas on the screen to create a sight-free text-entry method with low physical demand. Additionally, the layout can adopt a QWERTY-like design (Figure 1c) to create a familiar interface that is easy to learn, without compromising much in terms of user performance.

RELATED WORK

Sight-Free Interaction on Mobile Devices

Existing research has explored different input and output modalities to enable mobile sight-free interactions, such as speech-based input. However, the user might prefer not to vocalize her intended interaction (*e.g.*, speaking a text message she wants to send) in public space. On-body interactions (*e.g.*, Imaginary Phone (Gustafson, Holz, & Baudisch, 2011)) and physical body movements offer another way to interact with mobile devices sight-free (Cockburn, *et al.*, 2011; Gustafson, Bierwirth, & Baudisch, 2010; Li, Dearman, & Truong, 2009; Li, Dearman, & Truong, 2010; Oakley & Park, 2009; Scott, *et* *al.,* 2010). But, they might be inappropriate to perform in some public settings, and may lack enough resolution to support rapid text entry.

Previous research has also used audio (Kane, Bigham, & Wobbrock, 2008; Li, Baudisch, & Hinckley, 2008; Su, *et al.,* 2010) and tactile (Yatani & Truong, 2009) feedback to support on-device sight-free interaction. Earpod (Zhao, *et al.,* 2007) enables accurate sight-free menu selection by reading out the menu item that the user is selecting. Li *et al*.'s BlindSight (2008) uses audio feedback and physical keys located on the backside of the mobile phone to provide sight-free access to information, such as a calendar. These projects showed that audio cues can convey semantic information about an object and support sight-free interactions. Yatani and Truong (2009) explored using spatial tactile patterns to provide feedback about the object that the user is touching on a device.

Unlike menu selection and simple target selection tasks, text entry involves a series of interactions with the device (*i.e.*, typing multiple letters for a word). Waiting for long audio or tactile feedback on each interaction could degrade text entry speed. Prior studies (Pirhonen, Brewster, & Holguin, 2002; Bragdon, *et al.,* 2011) have showed that users can perform simple on-screen gestures reliably in a sight-free setting. This motivated us to further investigate the use of gestures specifically for sight-free text entry.

Sight-Free Mobile Text Entry

According to Clawson *et al.* (2005), there are two important feedback attributes in text entry: *keyboard visibility* (the level at which the input space information is visible to the user; present, limited, or absent) and *on-screen feedback* (the level typing feedback besides what the input space already implicitly offers is provided to the user; present, limited or absent). In sight-free text entry, both *keyboard visibility* and *on-screen feedback* are absent. However, non-visual feedback about the user's typing may be available. We thus define *output availability* as the level of the available information about what the user is typing (full, limited, and no) to represent different settings of sight-free text entry. *Full output availability* means that the user is able to know what she has typed. With *limited output availability*, the system only acknowledges that it has received some input from the user without offering any feedback about what she has typed. In *no output availability,* no feedback is present.

Several projects have investigated the user performance with text entry methods having different levels of *output availability.* Tinwala and MacKenzie (2009; 2010) showed that user performance with a Graffiti-based method with *limited output availability* is 7.6 words per minute (WPM) with 95% accuracy and 10 WPM with 96% accuracy when word-level error correction is added. Wobbrock, Chau, and Myers (2007) studied the performance of EdgeWrite with a joystick on a mobile phone with *full output availability*. They found that EdgeWrite supports about 8.1 WPM and 97% accuracy. In contrast, PocketTouch (Saponas, Harrison, & Benko, 2011) is a technique that offers sight-free through-fabric handwriting text entry with *no output availability*; it has not yet been formally evaluated.

Sight-free text entry method is also beneficial to people with visual impairments. Prior work has investigated the performance of different methods for this user population. Bonner *et al.* (2010) developed and evaluated No-Look Notes, which uses multi-touch gestures to type a letter with *limited output availability* (letter-based audio feedback was available). Their study showed that participants with visual impairments achieved faster typing speed and higher accuracy with their method than VoiceOver (Apple Inc., 2012) (No-Look Notes: 1.3 WPM with 89% accuracy; VoiceOver 0.66 WPM with 40% accuracy).

Braille-based techniques, such as BrailleType (Oliveira, *et al.*, 2011), Perkinput (Azenkot, *et al.*, 2012), and BrailleTouch (Southern, *et al.*, 2012), support text-entry specifically for people with visual impairments. These keyboards contain three soft-keys or six soft-keys (three keys on each side), and offer *limited* *output availability* via audio feedback for the letter the user is typing. A combination of the pressed keys is associated with a specific letter (*e.g.*, touching the two keys atop both sides types "c"). With BrailleTouch, proficient or expert Braille typists can achieve the typing speed of 7-24 WPM. Our keyboard is designed mainly for the general user population. Our work also focuses on one-handed interaction as it is shown to be often more desirable than two-handed interaction for small mobile devices (Karlson & Bederson, 2007).

A key design insight we explore in our work is that marking gestures from key spots on the screen can enable sight-free text entry. Previously, Jain and Balakrishnan (2012) extended Bragdon *et al*.'s (2011) findings to develop a bezel-based text entry method. In their technique, the user selects characters by performing directional marking gestures initiating from distinct locations along the screen bezel. Participants were able to achieve an average of 9.2 WPM using their technique sight-free with *limited output availability*. The technique uses all borders and corners of the screen to cover the alphabetic input with the minimal number of marking gesture levels and directions. The interface therefore requires the user to hold and interact with the mobile device in both hands.

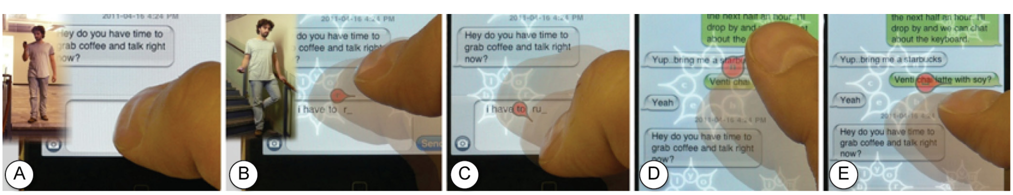
Yfantidis and Evreinov (2006) developed a gesture-based text entry method for desktop touch-screen devices similar to Escape-Keyboard. In their method, the user selects the group which contains the desired letter by dwelling on the screen. Then, she presses on a square button on a touch screen and flicks in one of eight different directions. The user then performs another marking gesture in one of eight directions to select the desired symbol from that group. Although their method could be applied for mobile devices, dwelling can slow down typing.

ESCAPE-KEYBOARD: DESIGN AND IMPLEMENTATION

Escape-Keyboard is a gesture-based sight-free text entry method for touch-screen mobile devices. It is based on Escape (Yatani, *et al.,* 2008), a fast target selection technique for touch screens. With Escape, the user establishes an approximate position of interest (*i.e.*, a region containing a group of letters), and then makes a flick gesture to select an intended target quickly (*i.e.*, an individual letter from the group). Additionally, Escape-Keyboard allows the user to type with one hand, which is often a preferred interaction with mobile devices (Karlson & Bederson, 2007). Figure 2 illustrates the concept of typing with Escape-Keyboard.

Although the basic interactions are similar to MessagEase (Nesbat, 2003), the key difference is that Escape-Keyboard is specifically designed for sight-free one-handed typing. Because it is intended as a sight-free text entry method, Escape-Keyboard exploits the entire screen for input (Figure 1a).

Figure 2. An interaction scenario with Escape-Keyboard: The user receives a message as he approaches a staircase on his way to a meeting. (a) The user reads the message and presses on the text area to respond. (b) The Escape-Keyboard slides over the message, and the user starts to enter text without looking at the keyboard while rushing down the stairs. (b-e) The user enters word "run" followed by the space character one letter at a time.



Layout Design Iterations

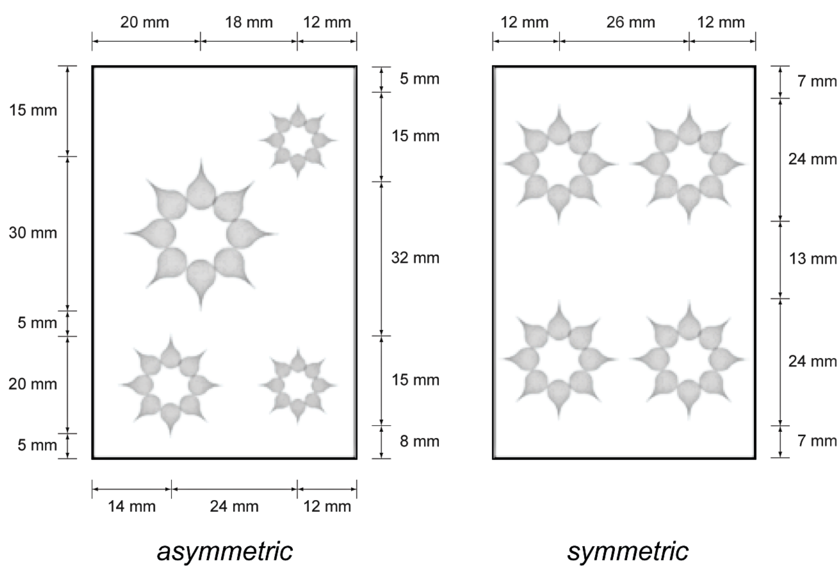
We employed an iterative design approach to define the key layout for Escape-Keyboard. We first conducted a small formative study with five right-handed participants to examine where the thumb can be placed on a touch-screen and perform flick gestures in eight different directions with the most ease and accuracy. Participants pressed and flicked from twelve rectangular regions on the screen. From this study, we learned:

1. The top-left corner is hard to contact. Participants often accidently pressed the bottom-right part of the touch screen with the thenar of the hand holding the device when trying to select regions close to the top-left corner of the screen. Karlson and Bederson (2007) also showed that these regions are hard to acquire with the thumb; thus, this area should be removed from our keyboard design;
2. The regions in the middle of the screen were difficult for participants to select accurately without looking at the device. Participants often confused adjacent regions in the middle of the screen as the intended target area. Thus, the middle area of the screen should not be segmented into small regions.

Hence, we designed two layouts (asymmetric and symmetric) with four target regions (Figure 3). The size of each region is large enough to be acquired by the thumb comfortably. The asymmetric layout is optimized for the thumb's dexterity. The thumb is anchored and pivots from the bottom-right corner of the device. The asymmetric layout does not have a separate region in the top-left corner to avoid the less comfortable expansion movement. On larger devices, the layout can be anchored in the bottom-right corner without scaling. On smaller devices, it can be scaled down.

We conducted the second formative study with another ten right-handed participants to examine the effect that the two layouts had on user performance and NASA-TLX subjective workload (Hart, 1988). Participants pressed and flicked different directions from the four regions in each layout. Overall, the participants were more accurate (error rate=21.1% compared to 23.1%; W=2, Z=2.26, $p < 0.05$, Cohen's d=0.71) and reported lower physical demand (W=3,

Figure 3. Possible layouts for Escape-Keyboard for iPhone-sized screens



Z=1.95, $p < 0.05$, Cohen's d=0.62) with the asymmetric layout than the symmetric layout. We did not find any significant difference in the gesturing speed. Therefore, we chose the asymmetric layout for our keyboard.

Letter Allocation

Using data gathered from the second formative study, we mapped the alphabet to specific regions and gesture directions in the asymmetric layout using a simulated annealing algorithm. Although there are more robust optimization techniques (*e.g.*, multidimensional pareto optimization (Dunlop & Levine, 2012)), we used simulated annealing for the sake of simplicity. The algorithm takes the frequency of each letter in the Corpus of Contemporary American English (COCA) (Davis, 2012), and optimizes its placement based on the speed and accuracy at which participants were able to perform specific gestures in the asymmetric layout. The energy function that we minimized in the algorithm was determined as the sum of the products of average time and average error for each region and gesture direction combination and the frequency of the letter mapped to that region. The algorithm calculated the additional weight of the letters that form letter bigrams in proportion to the occurrence frequency in the corpus so that these letters are more likely to be mapped to the same region.

We ran the algorithm with 100 random starting layouts for 100 million iterations each. We then performed another 100 million iterations on the 10 layouts with the smallest energy value to assess whether restarting from another configuration would give better results. All layouts converged to the same layout as shown in Figure 1b.

Flick Gesture Recognition

Escape-Keyboard only accepts touch movements longer than 2mm on the screen as a flick gesture. In a sight-free setting, flick gestures may not be in a straight line, and the user may fail to select the desired letter accurately. Thus, our prototype uses two classifiers (region and gesture classifiers) to predict the user's desired letter based on the flick gesture.

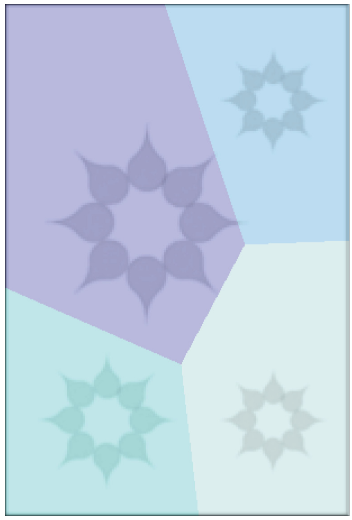
When the user makes a contact with the screen, the region classifier calculates the likelihood for each region. After the user performs a gesture, the gesture classifier calculates the likelihood for each gesture direction based on the angle between the initial and the last contact point. The region and direction with the highest likelihood are used, but likelihood values for the other regions and gestures are also used for error correction (explained later). This approach, allows the user to change the gesture direction without any penalty unless she releases the thumb from the screen.

We built the classifiers with the logistic regression method, and trained both classifiers with 6,400 input samples collected from the second formative study (Figure 4). A test with 10-fold cross validation showed accuracy improvement of 5.0% for region detection and 1.7% for gesture detection over the fixed region boundaries and gesture direction thresholds. Although the accuracy could be improved further by training the classifier for each individual user, we use these generic classifiers in our current implementation.

Letter Classification

Escape-Keyboard includes an error correction algorithm for letters. When the user finishes a gesture, the likelihood for each letter is calculated as a product of the likelihood for the region and likelihood for the gesture direction. If the likelihood for one letter is over a threshold (experimentally set to 0.75 in the current prototype), the system assumes that it is the desired letter. Otherwise, the keyboard keeps all the letters whose likelihood is over another threshold (0.25) in the buffer as the candidate letters. As the user types, the system calculates the likelihood for each possible string (referred as the string likelihood) from the product of letter likelihoods. When the user finishes entering the word (*i.e.*, enters the space or carriage return),

Figure 4. Four regions classified using the logistic regression region classifier



the system determines the likelihood of words constructed from the possible strings based on their frequency in COCA (referred as the corpus likelihood). It then enters the word with the maximum product of the two likelihoods. In our early informal evaluations of the system, we noticed that this approach more often resulted in correct input when entering text sight-free than existing letter disambiguation algorithms that are based simply on letter bi-gram frequencies (*e.g.*, work by MacKenzie *et al.* (2001)).

Sound Feedback

Our prototype provides non-speech audio feedback about which region of the keyboard the user is pressing. The prototype uses dual-tone multi-frequency signals, where each audio cue is a sinusoidal tone: 697 and 1209 Hz for the middle region; 697 and 1477 Hz for the top-right region; 770 and 1336 Hz for the bottom-left region; and 852 and 1209 Hz for the bottom-right region. The device generates a short beep sound with these tones when the user makes the initial contact on a region. When the user enters a letter, it plays another audio click. Space, backspace, and carriage return have different unique sounds to indicate a special symbol.

Mode Switching

To change the set of the symbols in the layout (*e.g.*, from the alphabets to the numbers and symbols), the user swipes from the left bezel to the right bezel or vice versa. We use bezel gestures because they can be robustly differentiated from other gestures around the bezel with a simple algorithm (Roth & Turner, 2009) and easy to perform sight-free (Bragdon, *et al.*, 2011).

EXPERIMENT 1: TEXT ENTRY PERFORMANCE STUDY

We conducted an experiment to examine the user performance of Escape-Keyboard in both sighted and sight-free settings. We used EdgeWrite (Wobbrock, Myers, & Kembel, 2003) as a reference technique. EdgeWrite is a unistroke-based text entry method, and is highly accurate even in a sight-free setting (Wobbrock, Chau, & Myers, 2007). The user enters text by moving her finger between corner regions on the phone in a specific pattern to write a character (Figure 5a). A physical guide can be used with EdgeWrite to support finger stroke

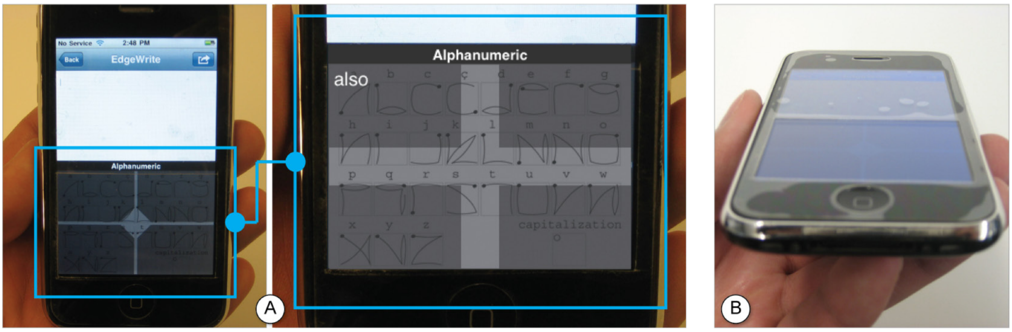


Figure 5. (a) EdgeWrite for iPhone and (b) added physical guides to support sight-free text entry

input in visually-demanding settings (*e.g.*, typing while driving (González, *et al.*, 2007)). We thus believe that EdgeWrite is an appropriate reference technique.

Participants

We recruited ten right-handed participants (PA1–PA10; 5 male and 5 female; the median age of 26 years, *SD*=3.7) for our first user study. All participants were in occupations which require at least a high-school level of English literacy (undergraduate students, graduate students, a research assistant, and a project manager). All participants reported no visual or motor disability. The participants had average hand span of 20.1 cm (*SD*=1.8), hand length of 18.6 cm (*SD*=1.2), and handbreadth of 8.4 cm (*SD*=0.9). All participants reported some experience with touch-screen interfaces. Participants received $200 for successfully completing all 16 sessions. Additionally, one participant won a $150 gift certificate in a draw.

Apparatus

We used an iPhone 3G for the study. For EdgeWrite, we used the iPhone version provided by Wobbrock and Kane. We attached to the screen a physical guide made from a screen protector to provide passive haptic feedback indicating the boundaries of the input space (Figure 5b). A computer was used to run the server program presenting phrases to participants (Figure 6). It also collected the phrases entered by participants, and calculated the performance metrics. The mobile phone and server communicated through a wireless network.

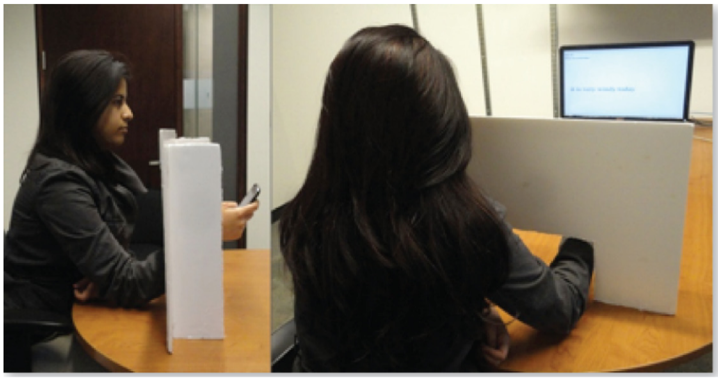
Procedure

We asked participants to enter short phrases presented on a desktop display as fast and accurately as possible using the given text entry method. We prepared phrases based on MacKenzie and Soukoreff's (2003) phrase set. We randomized the order of the phrases and grouped them into blocks of ten phrases, with no repeated phrases within a block.

Each participant completed sixteen sessions in total in this study. Any two consecutive sessions were scheduled at a 2–72 hour interval depending on the participants' availability, but participants were not allowed to complete more than three sessions in the same day. The first session also included the explanation of the two text entry methods.

In the first five sessions and last (16^{th}) session, participants were allowed to look at the screen of the mobile device (the *Sighted* condition). In addition to examining the performance of the keyboards in a sighted condition, these first five sessions were intended to help the participants become familiar with the keyboards. For the remaining sessions, participants typed phrases without looking at the screen (the *Sight-free* condition). We placed a barrier between the participant and the mobile device in the *Sight-free* condition (Figure 6). The input

Figure 6. The experimental setup for the Sight-free condition. The barrier hides the mobile device from view, while text phrases are presented on a nearby screen.



phrases were visible on the desktop display, but the keyboard was invisible. This study design offered *limited output availability* through the keyboard audio cue feedback. Participants sat at a desk, and had to hold the mobile device in their right hand and place the arm on or above the table. The participants were allowed to change their arm and hand posture if they wished during the experiment.

Each session consisted of two 20-minute half-sessions (one for each text entry method). Each half-session started with two practice phrases, followed by multiple blocks of 10 phrases. Participants had to complete typing as many phrases as they could. We counter-balanced the presentation order of the text-entry methods (*Technique*) across the participants for the first session, and alternated their presentation order across the sessions for the same participant. After completing each half-session, the participants filled out a NASA-TLX questionnaire (Hart, 1988) to assess the subjective workload associated with a given text-entry method.

Performance Metrics

We measured five performance metrics. In the following formulae, C means the number of letters typed correctly; INF means the number of letters typed incorrectly and not fixed; IF means the number of letters typed incorrectly but fixed; and AF means the number of letters the system automatically corrected:

- **TextEntrySpeed:** The average text entry speed in WPM;
- **NotCorrectedErrorRate:** The rate of errors remaining in the transcribed text, including insertion, substitution, and deletion errors (Soukoreff & MacKenzie, 2003). This was calculated as follows:

$$NotCorrectedErrorRate = \frac{INF}{C + INF + IF + AF} \times 100\%$$

- **CorrectedErrorRate:** The rate of errors that the user corrected and that did not remain in the transcribed text (Soukoreff & MacKenzie, 2003). This was calculated as follows:

$$CorrectedErrorRate = \frac{IF}{C + INF + IF + AF} \times 100\%$$

- **AutoCorrectedErrorRate:** The rate of errors that the system auto-corrected and that did not remain in the transcribed text. This was calculated as follows:

$$AutoCorrectedErrorRate = \frac{AF}{C + INF + IF + AF} \times 100\%$$

- **Workload:** The subjective workload of the two *Techniques* participants reported using NASA-TLX.

EXPERIMENT 1 RESULTS

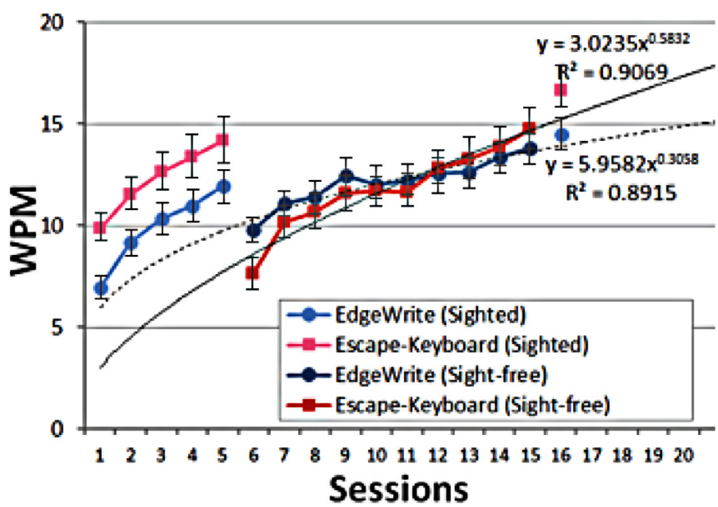
We used StreamAnalyzer (Wobbrock & Myers, 2006) to analyze data we gathered. We modified StreamAnalyzer to account for the substitution error correction functionality in Escape-Keyboard and auto-completion, prediction, and word-deletion functionalities in EdgeWrite. Modifications include changes to the original *NotCorrectedErrorRate* and *CorrectedErrorRate* metrics (Soukoreff & MacKenzie, 2003) and the addition of *AutoCorrectedErrorRate*.

To analyze the effects of *Technique* and *Session* on the dependent variables, we performed analysis on data from *Sighted* and *Sight-free* sessions separately. We used two-way repeated measures ANOVA to test the effects on *TextEntrySpeed*, and used Greenhouse-Geisser correction when data failed the test for sphericity. We performed post-hoc tests using a paired t-test with Bonnferroni correction. For event-count metrics, such as error rates and workload, we used the nonparametric Aligned Rank Transform (ART), which enables running parametric F-tests on nonparametric data while preserving the correctness of interaction effects (Wobbrock, *et al.*, 2011). We used the same tests on transformed data as described above for *TextEntrySpeed*.

Speed

Figure 7 shows the typing speed of each text entry method across the 16 sessions. In *Sighted* condition, our tests found that Escape-Keyboard was significantly faster than EdgeWrite ($F_{(1,9)}$=13.08, p=.006, η_p^2=.59), and also found a significant main effect of *Session* ($F_{(5,45)}$=77.75, p<.001, η_p^2=.59) on *TextEntrySpeed*, but did not find any significant interaction effect between

Figure 7. Mean TextEntrySpeed for the two methods over the 16 sessions and regression curves fitted to the data in the Sight-free conditions. Error bars are 95% confidence interval.



Technique and *Session* ($F_{(5,45)}$=0.65, *p*=.66, η_p^2=.07). Also, Escape-Keyboard (*M*=16.8 WPM, *SD*=4.1) was significantly faster than EdgeWrite (*M*=14.6 WPM, *SD*=2.44) in the last sighted session in the experiment (*p*=.023).

In the *Sight-free* condition, we found a significant effect of *Session* ($F_{(2.65,23.85)}$=33.3, *p*<.001, η_p^2=.79) on *TextEntrySpeed*. However, our tests did not find a significant main effect of *Technique* ($F_{(1,9)}$=.18, *p*=.68, η_p^2=.02) and *Technique*-by-*Session* interaction ($F_{(2.67,24)}$=3.02, *p*=.054, η_p^2=.25). Our post-hoc tests did not find significant differences in any sight-free session other than the first (*p*=.045). In the last sight-free session (session 15), the mean *TextEntrySpeed* for Escape-Keyboard was 14.7 WPM (SD=3.6) and for EdgeWrite was 13.8 WPM (SD=2.6).

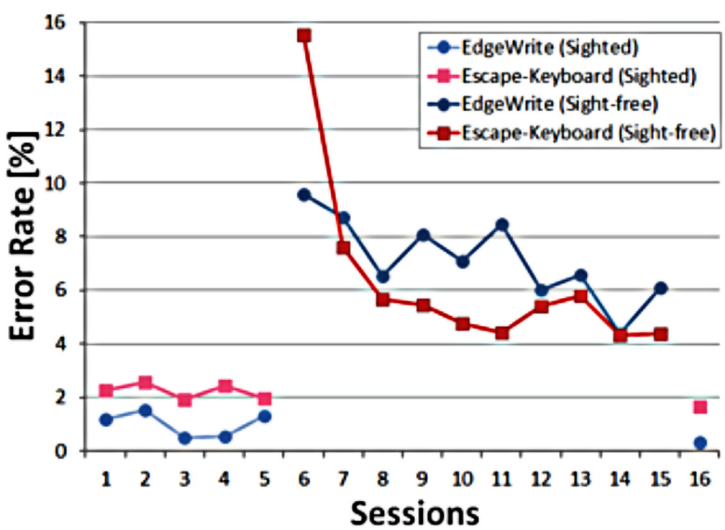
We performed regression analysis with a power function on the typing speed across the sessions. The typing speeds of both techniques were fitted well (R^2=.90 for Escape-Keyboard, and R^2=.89 for EdgeWrite). The models indicate that Escape-Keyboard provides a steeper improvement on the typing speed than EdgeWrite, and starts to outperform EdgeWrite after the 12th sessions.

Accuracy

Figure 8 shows median *NotCorrectedErrorRate* over all sessions. In *Sighted* condition, our tests found a significant main effect of *Technique* ($F_{(1,9)}$=11.08, *p*=.008, η_p^2=.55) with lower error rates in EdgeWrite. We did not find a significant effect of *Session* ($F_{(5,45)}$=2.11, *p*=.08, η_p^2=.19) or significant interaction ($F_{(5,45)}$=0.48, *p*=.78, η_p^2=.05). In the first session, median *NotCorrectedErrorRate* was 2.28% for Escape-Keyboard and 1.19% for EdgeWrite, and dropped to 1.66% for Escape-Keyboard and 0.33% for EdgeWrite in the last sighted session.

In *Sight-free* condition, there was a significant effect of *Session* ($F_{(9,81)}$=4.48, *p*<.001, η_p^2=.33) on *NotCorrectedErrorRate*, but our tests did not find any significant effect of *Technique* ($F_{(1,9)}$=.06, *p*=.80, η_p^2=.01) or significant interaction ($F_{(2.37,21.32)}$=2.34, *p*=.11, η_p^2=.21). In the first *Sight-free* session, the median *NotCorrectedErrorRate* was 15.34% for Escape-Keyboard and 9.60% for EdgeWrite. The error rate dropped down to 4.36% for Escape-Keyboard and 6.12% for EdgeWrite in the last *Sight-free* session.

Figure 8. Median NotCorrectedErrorRate for the two methods over the 16 sessions



Our tests also found significant main effects of *Technique* ($F_{(1,9)}$=44.76, p<.001, η_p^2=.83) and *Session* ($F_{(5,45)}$=5.02, p<.001, η_p^2=.36) on *CorrectedErrorRate*, and significant interaction ($F_{(1.77,15.95)}$=4.35, p=.035, η_p^2=.33) in sighted sessions. The post-hoc tests found significant differences in the first and last sessions. The participants fixed 2.77% of errors with Escape-Keyboard and 7.39% with EdgeWrite (p=.040) in the first session. In the last session, median *CorrectedErrorRate* was 3.25% with Escape-Keyboard and 4.59% with EdgeWrite (p=.003).

In the *Sight-free* sessions, our tests did not find a significant effect of *Technique* ($F_{(1,9)}$=.68, p=.43, η_p^2=.07), but did find a significant effect of *Session* ($F_{(9,81)}$=2.76, p=.007, η_p^2=.23) and *Technique*-by-*Session* interaction ($F_{(4.13,37.14)}$=2.76, p=.005, η_p^2=.32). Participants fixed significantly fewer errors with Escape-Keyboard than with EdgeWrite in the first (median EK=2.18%, EW=3.80%, p=.042) and second (median EK=1.57%, EW=2.51%, p=.048) sight-free sessions. But the test did not find a significant difference in other sight-free sessions. In the last sight-free session, the median *CorrectedErrorRate* was 2.25% with Escape-Keyboard and 1.21% with EdgeWrite.

One reason for low *NotCorrectedErrorRate* of Escape-Keyboard was its letter disambiguation. Escape-Keyboard's median *AutoCorrectedErrorRate* over the whole experiment was 17.41%. Participants reported that this feature helped them in both sighted and sight-free sessions as it allowed them to type faster with less concern about being very accurate with their flick gestures.

Workload

In *Sighted* sessions, our tests did not find a significant effect of *Technique* ($F_{(1,9)}$=5.00, p=.052, η_p^2=.36) on *Workload*, although we observed that Escape-Keyboard tended to require more workload than EdgeWrite (Figure 9). Our tests showed a significant effect of *Session* ($F_{(1.95,17.53)}$=3.76, p=.045, η_p^2=.29), but did not find any significant interaction ($F_{(1.81,16.28)}$=1.97, p=.17, η_p^2=.18). In *Sight-free* sessions, Escape-Keyboard required significantly more workload than EdgeWrite ($F_{(1,9)}$=6.14, p=.035, η_p^2=.41). Our tests also found a significant effect of *Session* ($F_{(1.99,17.89)}$=7.19, p=.005, η_p^2=.44), but again did not show a significant interaction ($F_{(3.06,27.56)}$=2.64, p=.068, η_p^2=.23).

We analyzed mental and physical demand components because they seemed to contribute the most to the overall subjective workload (Figures 10 and 11). In *Sighted* sessions, Escape-Keyboard was significantly more mentally demanding than EdgeWrite ($F_{(1,9)}$=16.06, p=.003,

Figure 9. Total Workload Index in NASA-TLX

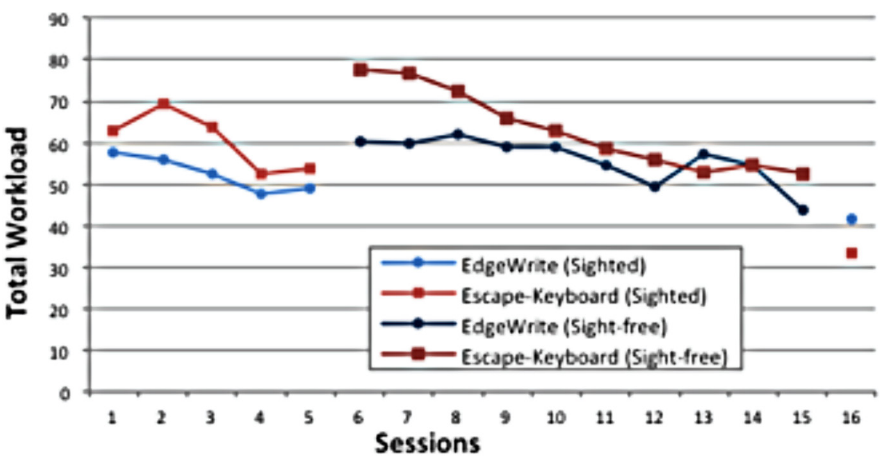


Figure 10. Mental Demand Index in NASA-TLX

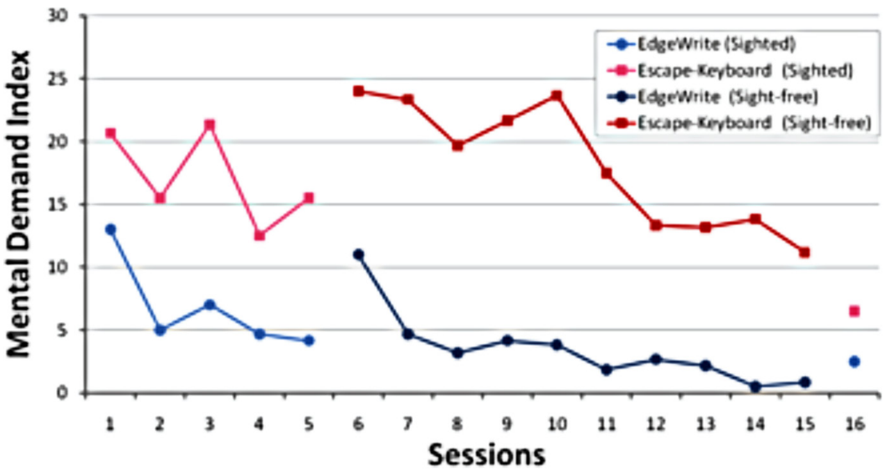
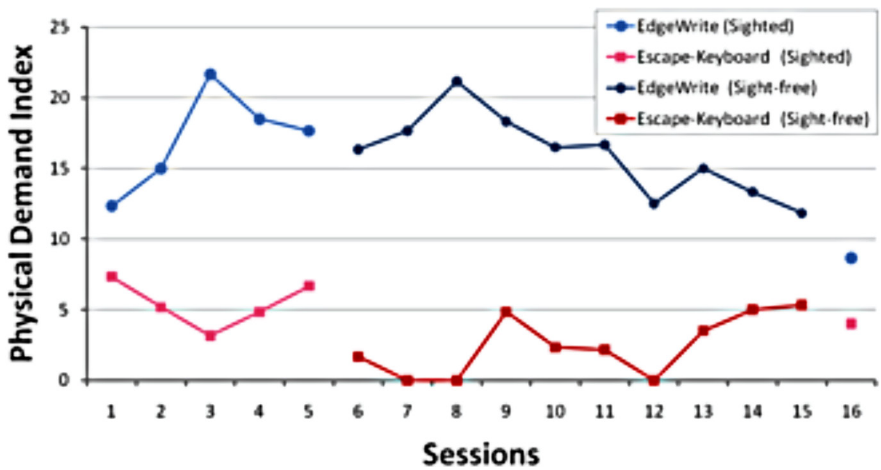


Figure 11. Physical Demand Index in NASA-TLX



η_p^2=.64), and there were also a significant effect of *Session* ($F_{(5,45)}$=4.10, *p*=.003, η_p^2=.31) and significant interaction ($F_{(5,45)}$=2.52, *p*=.043, η_p^2=.22). Our post-hoc tests found significant difference only in the third sighted session (*p*=.049). On the other hand, Escape-Keyboard overall required significantly less physical demand ($F_{(1,9)}$=5.13, *p*=.049, η_p^2=.36), but our tests did not find any significant effect of *Session* ($F_{(5,45)}$=1.29, *p*=.28, η_p^2=.13) or significant interaction ($F_{(5,45)}$=1.33, *p*=.27, η_p^2=.13).

In *Sight-free* sessions, Escape-Keyboard had significantly higher mental demand than EdgeWrite ($F_{(1,9)}$=31.09, *p*<.001, η_p^2=.78), and our tests found significant effect of Session ($F_{(3.3,29.72)}$=6.01, *p*=.002, η_p^2=.40), but no significant interaction ($F_{(9,81)}$=.82, p=.60, η_p^2=.08). Again, Escape-Keyboard had significantly lower physical demand ($F_{(1,9)}$=12.91, p=.006, η_p^2=.78), but our tests did not find significant effect of *Session* ($F_{(2.64,23.71)}$=.83, p=.59, η_p^2=.08), or significant interaction ($F_{(3.35,30.13)}$=2.09, p=.12, η_p^2=.19).

Qualitative Findings

Escape-Keyboard and EdgeWrite perform comparably well in terms of typing speed and accuracy in a sight-free setting. While Escape-Keyboard's low physical demand can potentially enable rapid text entry, reducing its mental demand is key to improving its sight-free use.

Escape-Keyboard was faster in the first five sighted sessions. Participants attributed their speed (13.97 WPM) to low physical demand required to enter individual letters:

[Escape-Keyboard] requires less movement per letter so you can actually be faster on it. – PA2

However, in the first sight-free session, participants dropped in speed and accuracy with both techniques. Participants also reported increased mental demand with both techniques in this session. However, mental demand for EdgeWrite dropped faster than for Escape-Keyboard.

One reason why participants felt Escape-Keyboard required high mental demand was its unfamiliar key layout. This is in contrast with more familiar gestures in EdgeWrite which participants related to hand writing:

[EdgeWrite] is more intuitive so I don't have to like keep searching for the letters like with [Escape-Keyboard]. It is kind of like writing and you know what the movements are. – PA1

Participants also reported issues with memorizing the layout of Escape-Keyboard which was another contributor to its high metal demand. Most participants were able to memorize only the positions of most-frequent letters by the start of the sight-free sessions. In contrast, they felt memorizing EdgeWrite gestures was easier:

Writing the whole letters [with EdgeWrite] was easier at first ... It takes less thinking and less memorizing. – PA8

However, participants became more familiar with the Escape-Keyboard layout over the sight-free sessions. We also observed a decrease in the mental demand (Figure 10) near the crossover point in the typing speed we derived from the learning curves that we fitted to the empirical data (Figure 7):

The mental demand with Escape-Keyboard was beginning to decrease and I was getting more proficient at using it. Whereas EdgeWrite, the physical demand, there was no way out because of how many gestures I needed to make. – PA10

IMPROVEMENTS FOR LEARNABILITY

As our study has shown, a high mental workload contributed to lower user performance with Escape-Keyboard in the initial stage of sight-free use. User feedback suggested that we must focus on improving the learnability of the technique. Thus, we implemented several changes to Escape-Keyboard to increase the familiarity of the technique and improve the memorizability of the layout.

Letter Layout Changes

To address the issue of familiarity, we adapted the layout for Escape-Keyboard to resemble the QWERTY layout (Figure 12). The QWERTY layout is familiar to many users. Thus, it could help the user develop the mental model of the keyboard and infer letter positions. We divided the QWERTY layout into three parts. We added tap gestures to our keyboard to maintain the QWERTY layout. The user enters letters "s", "g", and "k" by tapping the corresponding region (Figure 12b).

Speech Feedback

Our previous Escape-Keyboard design did not allow the user to explore the layout and to learn letter positions sight-free. Thus, participants had to guess letter positions in the early sight-free sessions. To solve this issue, we added speech-based feedback to the keyboard. When the user presses the screen or performs a flick gesture but continues to dwell with the thumb on the screen for 100 milliseconds, the system reads out the corresponding letter. The user can change the position of her thumb before lifting it off the screen to receive speech feedback identifying other letters in the initial contact region.

Visual Cues

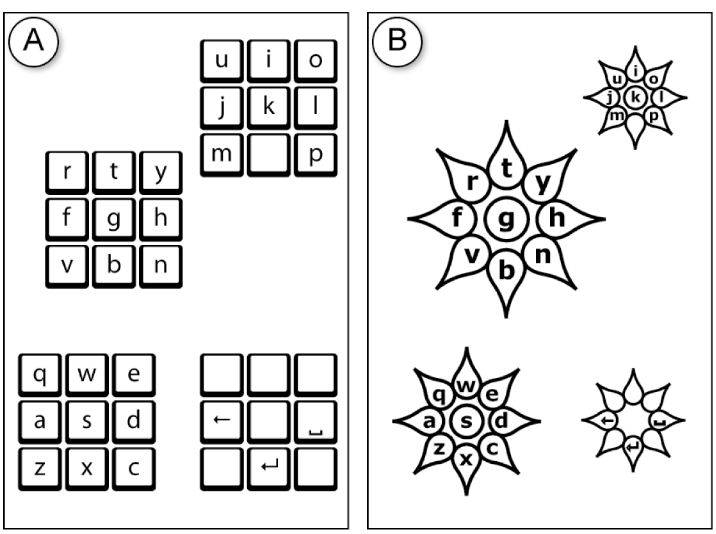
We introduced additional visual cues to the keyboard layout (Figure 1c) to facilitate memorization of commonly-used and infrequently-used letters. We added a set of static color cues: yellow to draw attention to the vowels, and red to differentiate the five least frequently-used consonants from the more frequent characters. We also added dynamic visual cues to highlight the eight most likely consonants based on the previous two letters typed. These cues cannot be used sight-free, but can help the user remember the letter positions more quickly in the early stages of sighted use.

EXPERIMENT 2: LEARNABILITY STUDY AND RESULTS

We conducted another experiment to examine whether the new Escape-Keyboard design addresses the learnability issues of the first design. In this study, 5 different participants (PB1–PB5) interacted with the revised Escape-Keyboard with sight to become familiar with the system before being asked to type with Escape-Keyboard sight-free for one 20-minute session. We did not find any statistically significant result in the sight-free text-entry performance between this and the previous study. However, participants' qualitative feedback after this session about the learnability of the new design is promising.

Participants' subjective ratings about their confidence and learning experience with this keyboard layout indicate that they were able to

Figure 12. (a) QWERTY-based layout (b) applied to Escape-Keyboard



build a relatively accurate mental model of the layout; four participants stated that they knew the position of most letters. Furthermore, most participants rated the challenge associated with memorizing the layout and the challenge associated with finding unknown letters as neutral for both; this indicates that although neither was effortless, they also were not necessarily difficult.

Participants' familiarity with QWERTY helped them develop the mental model and learn where to locate letters:

I could do a lot of those frequent letters without looking at them. I scaffold on the knowledge of QWERTY I already have. Having that special knowledge and mapping it to this new keyboard helps. And I was able to figure out when I made a mistake, for example, I would like know: "right region, wrong direction for 'm'. – PB4

Furthermore, speech feedback enabled the participants to find letters for which the positions could not be quickly recalled:

The QWERTY layout helped me know what group the letter was in. I can listen in that region if I don't know where it was. What helped me the most without looking was hearing it with the hand motion – it helped me learn. – PB5

The combination of the improvements resulted in less mental effort with using Escape-Keyboard overall:

If I just autopilot, my fingers knew where to go. – PB5

PREDICTING PEAK EXPERT PERFORMANCE

To examine the peak expert sight-free performance, we developed a theoretical model similar to Keystroke Level Model (KLM). Instead of using the time for keystrokes, we created the model with the time for performing the individual gestures sight-free to enter text. By only examining the linguistics and motor component of the interaction, we aim to estimate the theoretical upper bound of the entry speed with Escape-Keyboard.

Model Overview

To construct a KLM-like model, we break down typing into a series of movements. For example, when the user attempts to type the word "to" with the revised Escape-Keyboard design, she first presses the top-left region, and gestures upwards. Then she moves the thumb to the top-right region and makes an up-and-rightward gesture to type "o." Thus, the time of typing two letters "to" can be calculated as the sum of 1) moving the thumb to the top-left region; 2) making the gesture; 3) moving the thumb from the top-left region to the top-right region; and 4) making another gesture. By combining the time for moving the thumb to one region and pressing it, and the time for performing gestures, we can calculate the overall time for typing any word. In typical typing situations, every word is followed by a space. Our model thus assumes that every word starts with a space and ends with another space.

Empirical Data Collection

We measured abstract physical movement time for moving the thumb from one area to another, and performing flick gestures in different area sight-free. Participants who completed the text entry performance study were asked to take part in this study.

We first asked participants to tap one region, and then tap the same or another region, and tap again in the starting region. The participants repeated this task five times for each combination of regions. We then asked participants to complete another task which involved performing flick gestures from four different regions. Participants repeated this task five times for each combination of region and direction.

Model Predictions

We used a similar approach to the method previously described by MacKenzie and Soukoreff (2002) to calculate the peak expert performance. We first calculated the typing time for each word in a corpus. The corpus we used contained the 500,000 most frequently-used words in American English. We removed all entries in the corpus containing non-alphabetic characters, leaving 406,919 words (2.23E9 letters). To type these words, it would take 6.58E8 and 6.85E8 seconds with the original and QWERTY-based Escape-Keyboard, respectively. Therefore, the predicted peak expert performance of sight-free text entry is 2.23E9 / 6.58E8 × 60 / 5 = 40.6 WPM for the original layout, and 2.23E9 / 6.85E8 × 60 / 5 = 39 WPM for the QWERTY-based layout.

The difference between the measured performance and the predicted peak performance is large. This suggests that reducing mental demand is necessary in improving user performance with Escape-Keyboard. This analysis also shows that the changes we introduced to reduce the mental demand, such as using an adapted QWERTY layout, will not have a large impact on peak expert performance.

CONCLUSION AND FUTURE WORK

We presented our iterative design of Escape-Keyboard, a sight-free text entry method. We conducted a series of experiments and analysis to examine the performance of Escape-Keyboard. Our evaluation revealed that Escape-Keyboard and EdgeWrite are comparably fast and accurate in a sight-free setting. However, the findings from that study outlined the importance of improving the learnability of Escape-Keyboard. We then explored another design based on the QWERTY layout, and validated its potential to mitigate some initial learnability issues. Our KLM-like model predicted peak expert performance of the original Escape-Keyboard layout to be 40.6 WPM while the easier to learn QWERTY-based layout is only somewhat slower, at 39 WPM.

We have thus far designed and tested Escape-Keyboard layouts for only right-handed users, but we plan to adapt Escape-Keyboard to left-handed users. One solution is to mirror the asymmetric layout along the y-axis. However, future work should study the impact of that layout on Escape-Keyboard's performance.

The design features we proposed improved initial learnability of the keyboard. However, our present study does not uncover how the user would adopt our keyboard in a realistic setting. In order to examine this, we will release Escape-Keyboard publicly.

REFERENCES

Apple Inc. (2012). *Apple - accessibility - iPhone - vision*. Retrieved May 15, 2012 from http://www.apple.com/accessibility/iphone/vision.html

Azenkot, S., Wobbrock, J. O., Prasain, S., & Ladner, R. E. (2012). Input finger detection for nonvisual touch screen text entry in Perkinput. In *Proceedings of the Graphics Interface (GI '12)* (pp. 121-129). Toronto, Canada: Canadian Information Processing Society.

Bonner, M. N., Brudvik, J. T., Abowd, G. D., & Edwards, W. K. (2010). No-look notes: Accessible eyes-free multi-touch text entry. In *Proceedings of the Pervasive Computing (Pervasive'10)* (pp. 409-426). Berlin, Germany: Springer-Verlag.

Bragdon, A., Nelson, E., Li, Y., & Hinckley, K. (2011). Experimental analysis of touch-screen gesture designs in mobile environments. In *Proceedings of the Human Factors in Computing Systems (CHI '11)* (pp. 403-412). New York, NY: ACM.

Brewster, S., Lumsden, J., Bell, M., Hall, M., & Tasker, S. (2003). Multimodal 'eyes-free' interaction techniques for wearable devices. In *Proceedings of the Human Factors in Computing Systems (CHI '03)* (pp. 473-480). New York, NY: ACM.

Clawson, J., Lyons, K., Starner, T., & Clarkson, E. (2005). The impacts of limited visual feedback on mobile text entry for the twiddler and mini-QWERTY keyboards. In *Proceedings of the International Symposium on Wearable Computers (ISWC '05)* (pp. 170-177). Washington, DC: IEEE Computer Society.

Cockburn, A., Quinn, P., Gutwin, C., Ramos, G., & Looser, J. (2011). Air pointing: Design and evaluation of spatial target acquisition with and without visual feedback. *International Journal of Human-Computer Studies*, *69*(6), 401–414. doi:10.1016/j.ijhcs.2011.02.005.

Davis, M. (2012). *Word frequency data from the Corpus of Contemporary American English (COCA)*. Retrieved May 15, 2012 from http://www.wordfrequency.info

Dunlop, M., & Levine, J. (2012). Multidimensional pareto optimization of touchscreen keyboards for speed, familiarity and improved spell checking. In *Proceedings of the Human Factors in Computing Systems (CHI '12)* (pp. 2669-2678). New York, NY: ACM.

González, I. E., Wobbrock, J. O., Chau, D. H., Faulring, A., & Myers, B. A. (2007). *Eyes on the road, hands on the wheel: Thumb-based interaction techniques for input on steering wheels. Graphics Interface (GI '07)* (pp. 95–102). New York, NY: ACM.

Gustafson, S., Bierwirth, D., & Baudisch, P. (2010). Imaginary interfaces: Spatial interaction with empty hands and without visual feedback. In *Proceedings of the User Interface Software and Technology (UIST '10)* (pp. 3-12). New York, NY: ACM.

Gustafson, S., Holz, C., & Baudisch, P. (2011). Imaginary phone: Learning imaginary interfaces by transferring spatial memory from a familiar device. In *Proceedings of the User Interface Software and Technology (UIST '11)* (pp. 283-292). New York, NY: ACM.

Hart, S. G. (1988). Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Proceedings of the Human Mental Workload*.

Jain, M., & Balakrishnan, R. (2012). User learning and performance with bezel menus. In *Proceedings of the Human Factors in Computing Systems (CHI '12)* (pp. 2221-2230). New York, NY: ACM.

Kane, S. K., Bigham, J. P., & Wobbrock, J. O. (2008). Slide rule: Making mobile touch screens accessible to blind people using multi-touch interaction techniques. In *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (Assets '08)* (pp. 73-80). New York, NY: ACM.

Karlson, A. K., & Bederson, B. B. (2007). ThumbSpace: Generalized one-handed input for touchscreen-based mobile devices. In *Proceedings of the Human-Computer Interaction (INTERACT'07)* (pp. 324-338). Berlin, Germany: Springer-Verlag.

Li, F. C., Dearman, D., & Truong, K. N. (2009). Virtual shelves: Interactions with orientation aware devices. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '09)* (pp. 125-128). New York, NY: ACM.

Li, F. C., Dearman, D., & Truong, K. N. (2010). Leveraging proprioception to make mobile phones more accessible to users with visual impairments. In *Proceedings of the ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '10)* (pp. 187-194). New York, NY: ACM.

Li, K. A., Baudisch, P., & Hinckley, K. (2008). Blindsight: Eyes-free access to mobile phones. In *Proceedings of the Human Factors in Computing Systems (CHI '08)* (pp. 1389-1398). New York, NY: ACM.

MacKenzie, I., Kober, H., Smith, D., & Jones, T. (2001). LetterWise: Prefix-based disambiguation for mobile text input. In *Proceedings of the User Interface Software and Technology (UIST '01)* (pp. 111-120). New York, NY: ACM.

MacKenzie, I. S., & Soukoreff, R. W. (2002). A model of two-thumb text entry. [Toronto, Canada: Canadian Information Processing Society.]. *Proceedings of the Graphics Interface*, *2002*, 117–124.

MacKenzie, I. S., & Soukoreff, R. W. (2003). Phrase sets for evaluating text entry techniques. In *Proceedings of the CHI '03 Extended Abstracts on Human Factors in Computing Systems (CHI EA '03)* (pp. 754-755). New York, NY: ACM.

Nesbat, S. B. (2003). A system for fast, full-text entry for small electronic devices. In *Proceedings of the Multimodal Interfaces (ICMI '03)* (pp. 4-11). New York, NY: ACM.

Oakley, I., & Park, J. (2009). Motion marking menus: An eyes-free approach to motion input for handheld devices. *International Journal of Human-Computer Studies*, *67*(6), 515–532. doi:10.1016/j.ijhcs.2009.02.002.

Oliveira, J., Guerreiro, T., Nicolau, H., Jorge, J., & Gonçalves, D. (2011). BrailleType: Unleashing braille over touch screen mobile phones. In *Proceedings of the Human-Computer Interaction (INTERACT 2011)* (pp. 100-107). Berlin/Heidelberg, Germany: Springer.

Pirhonen, A., Brewster, S., & Holguin, C. (2002). Gestural and audio metaphors as a means of control for mobile devices. In *Proceedings of the Human Factors in Computing Systems (CHI '02)* (pp. 291-298). New York, NY: ACM.

Roth, V., & Turner, T. (2009). Bezel swipe: conflict-free scrolling and multiple selection on mobile touch screen devices. In *Proceedings of the Human Factors in Computing Systems (CHI '09)* (pp. 1523-1526). New York, NY: ACM.

Saponas, T. S., Harrison, C., & Benko, H. (2011). PocketTouch: Through-fabric capacitive touch input. In *Proceedings of the User Interface Software and Technology (UIST '11)* (pp. 303-307). New York, NY: ACM.

Scott, J., Dearman, D., Yatani, K., & Truong, K. N. (2010). Sensing foot gestures from the pocket. In *Proceedings of the User Interface Software and Technology (UIST '10)* (pp. 199-208). New York, NY: ACM.

Sears, A., Lin, M., & Jacko, J. (2003). When computers fade: Pervasive computing and situationally-induced impairments and disabilities. *HCI International*, *2*, 1298–1302.

Soukoreff, R. W., & MacKenzie, I. S. (2003). Metrics for text entry research: An evaluation of MSD and KSPC, and a new unified error metric. In *Proceedings of the Human Factors in Computing Systems (CHI '03)* (pp. 113-120). New York, NY: ACM.

Southern, C., Clawson, J., Frey, B., Abowd, G., & Romero, M. (2012). BrailleTouch: An evaluation of mobile eyes-free touchscreen text entry for the visually impaired. In *Proceedings of the Human Computer Interaction with Mobile Devices and Services (MobileHCI '11)*. New York, NY: ACM.

Su, J., Rosenzweig, A., Goel, A., Lara, E. D., & Truong, K. N. (2010). *Timbremap: Enabling the visually-impaired to use maps on touch-enabled devices*. New York, NY: ACM. doi:10.1145/1851600.1851606.

Tinwala, H., & MacKenzie, I. S. (2009). Eyes-free text entry on a touchscreen phone. In *Proceedings of the Toronto International Conference – Science and Technology for Humanity (TIC-STH '09)* (pp. 83-88). New York, NY: IEEE.

Tinwala, H., & MacKenzie, I. S. (2010). Eyes-free text entry with error correction on touchscreen mobile devices. In *Proceedings of the Nordic Conference on Human-Computer Interaction (NordiCHI '10)* (pp. 511-520). New York, NY: ACM.

Wobbrock, J. O., Chau, D. H., & Myers, B. A. (2007). An alternative to push, press, and tap-tap-tap: Gesturing on an isometric joystick for mobile phone text entry. In *Proceedings of the Human Factors in Computing Systems (CHI '07)* (pp. 667-676). New York, NY: ACM.

Wobbrock, J. O., Findlater, L., Gergle, D., & Higgins, J. J. (2011). The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *Proceedings of the Human Factors in Computing Systems (CHI '11)* (pp. 143-146). New York, NY: ACM.

Wobbrock, J. O., & Myers, B. A. (2006). Analyzing the input stream for character-level errors in unconstrained text entry evaluations. [TOCHI]. *Transactions on Computer-Human Interaction*, *13*(4), 458–489. doi:10.1145/1188816.1188819.

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. In *Proceedings of the User Interface Software and Technology (UIST '03)* (pp. 61-70). New York, NY: ACM.

Yatani, K., Partridge, K., Bern, M., & Newman, M. W. (2008). Escape: A target selection technique using visually-cued gestures. In *Proceedings of the Human Factors in Computing Systems (CHI '08)* (pp. 285-294). New York, NY: ACM.

Yatani, K., & Truong, K. N. (2009). SemFeel: A user interface with semantic tactile feedback for mobile touch-screen devices. In *Proceedings of the User Interface Software and Technology (UIST '09)* (pp. 111-120). New York, NY: ACM.

Yfantidis, G. (2006). Adaptive blind interaction technique for touchscreens. *Universal Access in the Information Society*, *4*(4), 328–337. doi:10.1007/s10209-004-0109-7.

Zhao, S., Dragicevic, P., Chignell, M., Balakrishnan, R., & Baudisch, P. (2007). Earpod: Eyes-free menu selection using touch input and reactive audio feedback. In *Proceedings of the Human Factors in Computing Systems (CHI '07)* (pp. 1395-1404). New York, NY: ACM.

Nikola Banovic is a Masters Student in the Department of Computer Science at the University of Toronto. His research interests are in Human-Computer Interaction and Ubiquitous Computing, primarily focusing on assistive technologies and mobile interaction techniques.

Koji Yatani is an Associate Researcher in the Human-Computer Interaction group at Microsoft Research Asia. His research interests lie in Human-Computer Interaction and ubiquitous computing. He is interested in developing new sensing technologies to support user interactions in mobile/ubiquitous computing environments, and interactive systems with computational linguistics methods.

Khai Truong is an Associate Professor in the Department of Computer Science at the University of Toronto; his research interests are in Human-Computer Interaction and Ubiquitous Computing, specifically in the areas of assistive technologies, mobile interaction techniques, and sustainability.