

# Eyes-free Text Entry on a Touchscreen Phone

Hussain Tinwala, and I. Scott MacKenzie

Dept. of Computer Science and Engineering

York University

Toronto, Ontario, Canada M3J 1P3

hussain@cse.yorku.ca, mack@cse.yorku.ca

**Abstract** - We present an *eyes-free* text entry technique for touchscreen mobile phones. Our method uses *Graffiti* strokes entered using a finger on a touchscreen. Although visual feedback is present, eyes-free entry is possible using auditory and tactile stimuli. In eyes-free mode, entry is guided by speech and non-speech sounds, and by vibrations. A study with 12 participants was conducted using an Apple *iPhone*. Entry speed, accuracy, and stroke formations were compared between eyes-free and eyes-on modes. Entry speeds reached 7.00 wpm in the eyes-on mode and 7.60 wpm in the eyes-free mode. Text was entered with an overall accuracy of 99.6%. KSPC was 9% higher in eyes-free mode, at 1.36, compared to 1.24 in eyes-on mode.

**Keywords** - *Eyes-free; text entry; touchscreen; finger input; auditory feedback; mobile computing; gestural input; Graffiti.*

## I. INTRODUCTION

### A. The Rise of Touchscreen Phones

Recently, the mobile industry has seen a rise in the use of touchscreen phones. Although touchscreen devices are not new, interest in them has increased since the arrival of the Apple *iPhone*. Following the *iPhone*'s release, a wide array of competing products emerged, such as LG's *Prada*, Samsung's D988, Nokia's *N95* and RIM's *BlackBerry Storm*.

Early touchscreen phones were trimmed down versions of desktop computers. They were operated with a stylus, demanding a high level of accuracy. Such accuracy is not always possible in mobile contexts. As a result, current touch-based phones use the finger for input. The devices allow direct manipulation and gesture recognition using *swiping*, *tapping*, *flicking*, and even *pinching* (for devices with multi-touch). Such novel interactions afford a naturalness that is unparalleled by indirect methods (e.g., using a joystick).

Phones with physical buttons are constrained since all interaction involves pre-configured hardware. Once built, the hardware is fixed and cannot be customized further, which limits the scope of interactions possible. Touchscreen phones use software interfaces making them highly customizable and multipurpose. Use of screen space is more flexible and since there is no physical keypad, the screen size can be bigger.

However, these benefits come at a cost. Without physical keys, a user's ability to engage the tactile, kinesthetic, and proprioceptive sensory channels during interaction is reduced. The demand on the visual channel is increased and this

compromises the “mobile” in “mobile phone”. One goal of our research is to examine ways to reduce the visual demand for interactions with touchscreen phones.

### B. Interacting With Touchscreen Phones

The primary purpose of mobile devices is communication. So, it is important to support alphanumeric entry, even if it is just to enter a phone number. With physical buttons, users develop a sense of the buttons, feel them, and over time remember their locations. This tactile and proprioceptive feedback is priceless. Users build a spatial motor-memory map, which allows them to carry out basic tasks eyes-free, such as making or receiving a call. With experience, many mobile phone users carry out text entry tasks (e.g., texting) eyes-free by feeling and knowing their way around the device.

With touchscreen phones, the feedback afforded by physical buttons is gone. As a result, touchscreen phones are more visually demanding, with users often unable to meet this demand in a mobile context. The overload of the visual channel makes it difficult to use these devices when engaged in a secondary task, such as walking, attending a meeting, or shopping. Furthermore, the inability to use these devices in an eyes-free manner affects people with visual impairments [10].

### C. Mobile Phone Design Space

The natural interaction properties of a touch-sensitive display allow a rich set of applications for touchscreen phones, but the increased visual attention complicates translating the added features into mobile contexts. This brings us to a gap between what was capable with a physical button phone and what is capable with touchscreen phones. Figure 1 serves as a descriptive model to think about the design space and potential elements to consider in bridging this gap. Combining some form of physical feedback, finger tracking, and other feedback



Figure 1. Exploring the design space for eyes-free text entry.

modalities to guide the user through tasks can provide the elements required for eyes-free text entry on a touchscreen phone. Our main purpose is to explore whether eyes-free text entry is even possible on a touchscreen phone. The next sections discuss related work and our proposed solution. This is followed by a description of the evaluation carried out and a discussion of the results.

#### D. Eyes-free Mobile Device Use

##### 1) Button-based strategies

*Twiddler* [5], a one-handed chording keyboard, was used to investigate eyes-free typing. Entry speeds for some participants approached as high as 67 wpm. The difficulty with *Twiddler* is the steep learning curve. This can frustrate users and may result in low acceptability of the technique. Investing substantial time to learn the basic operating modes of a consumer product is generally unacceptable.

Another study compared multitap with the use of an isometric joystick using *EdgeWrite*, a stroke-based text entry technique [15]. The study involved entering *EdgeWrite* strokes using gestures on a joystick. The mobile device used included two isometric joysticks, one on the front and one on the back. Of specific interest is the finding that the front joystick allowed eyes-free entry at 80% of the normal-use speed ( $\approx 7.5$  wpm).

In our previous work, we presented *LetterScroll*, a text entry technique that used a wheel to access the alphabet [14]. As text was entered, each character was spoken using a speech synthesizer. Subjective findings revealed that entering text was easy, but slow. The average text entry speed was 3.6 wpm.

##### 2) Touch-based strategies

Although touchscreen devices have been extensively studied, a literature search revealed very little research on eyes-free text entry solutions for touchscreen devices. Below we present some related work in this area.

*FreePad* [1] investigated pure handwriting recognition on a touchpad. Subjective ratings found that the overall experience of entering text was much better than predictive text entry (aka T9). Text entry speeds were not measured, however.

Wobbrock et al. proposed an enhancement to *EdgeWrite* called *Fisch* [16], which provides in-stroke word completion. After entering a stroke for a letter, users can extend the stroke to one of the four corners of the touchpad to select a word. The authors duly note that the mechanism provides potential benefits for eyes-free use. However, no investigation was carried out to test this.

Yfantidis and Evreinov [18] proposed a gesture-driven text entry technique for touchscreens. Upon contact, a pie menu appears displaying the most frequent letters. Dwelling on the menu updates the pie menu by entering a deeper layer. Users receive auditory feedback to signal their position in the menu hierarchy. Some participants achieved an entry speed of 12 wpm after five trials.

Sánchez and Aguayo [12] proposed a text entry technique that places nine virtual keys on a touchscreen device. Consequently, text entry is similar to multitap. The primary

mode of feedback is synthesized speech. Unfortunately, their work did not include an evaluation.

We were unable to find literature on eyes-free text entry on a touchscreen device using a stroke-based entry mechanism. While occasionally mentioned, no controlled evaluations exist.

##### 3) Alternative modalities

Some studies explored speech as an input mechanism for mobile text entry [2, 11]. General findings reveal that speech is an attractive alternative for text input. However, such modalities are not always appropriate in mobile contexts. By using speech as input for mobile devices, there is a loss of privacy. Furthermore, social circumstances may inhibit the use of such techniques.

We present a method that relies on speech and non-speech modalities as a form of output rather than input. This is acceptable given that most mobile devices include earphones. The next section discusses our solution for eyes-free text input on a mobile touchscreen phone.

## II. EYES-FREE TEXT ENTRY ON A TOUCHSCREEN

### A. The Ingredients

Technology to enable eyes-free text entry on a touchscreen already exists. It is just a matter of knitting it together. The critical requirement is to support text entry without the need to visually monitor or verify input. Obviously, non-visual feedback modalities are important. Desirably, the technique should have a short learning curve so that it is usable seamlessly across various devices and application domains.

Goldberg and Richardson [4] were the first to propose eyes-free text entry using a stroke-based alphabet. Their alphabet, *Unistrokes*, was designed to be fast for experts. A follow-on commercial instantiation, *Graffiti*, was designed to be easy for novices [9]. To maintain a short learning curve, we decided to use the *Graffiti* alphabet (see Figure 2). The similarity of most characters to the Roman alphabet allows users to build upon previous experiences. This encourages quick learning and aids retention.

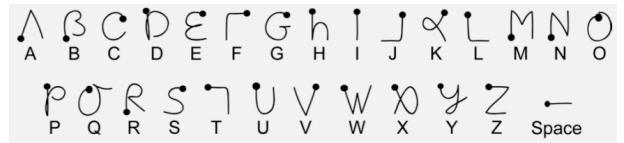


Figure 2. The *Graffiti* alphabet.

Since the device will be occluded from view, the feedback must be non-visual. Given the power of current mobile devices, it is easy to incorporate basic speech synthesis. For certain interactions, such as unrecognized strokes, we used the *iPhone*'s built in actuator to provide a short pulse of vibrotactile feedback.

### B. Design Issues

Previously, many studies investigated the performance of single-stroke text entry using a stylus or pen on touch-sensitive devices (e.g., [1, 15, 17]). Our work differs in that the main input “device” is the finger. Another issue is the drawing surface. Will it be the entire screen or just a region of the touchscreen? Also, what is the ideal approach of incorporating

text entry so as to minimize interference with existing widgets? These are open questions. Our prototype uses the entire screen as the drawing surface eliminating the need to home into a specific area. The drawing surface is overlaid on existing widgets; so, strokes inked do not interfere with other UI elements. Perhaps a “text mode” could be useful, which would lock the underlying interface, but this was not implemented.

It is useful to set the context of our evaluation at this point. We do not expect text entry speeds to rival those attainable on a soft keyboard, but the ability to enter any amount of text eyes-free on a touchscreen phone is noteworthy. Our focus is on tasks that require shorter amounts of text, such as text messaging, calendar entries, or short lists. Earlier work in this area found that novice users are able to reach text entry speeds of 7 wpm while experts reached 21 wpm with *Graffiti* [3].

### III. GRAFFITI INPUT USING FINGERS

#### A. The Interface

Our application allows users to enter text one character at a time. Once the word is completed, it is terminated with a SPACE and appended to the message.

Figure 3 illustrates the interface along with the evolution of the phrase “the deadline is very close”. The *Graffiti* alphabet is overlaid on the screen to promote learning. In previous studies, the strokes were displayed on a wall chart or away from the interface [7, 15]. This demanded visual attention away from the interface and could potentially affect throughput.



Figure 3. Device interface (stroke map enhanced for clarity).

#### B. Text Entry Interaction

To enter text, users draw a stroke on the screen. Digitized ink follows the user’s finger during a stroke. At the end of a stroke, the application attempts to recognize the stroke and identify the intended character. If the stroke is recognized, the *iPhone* speaks the character and appends it to the word entered so far. If the stroke is unrecognized, the *iPhone*’s vibration actuator pulses once. This response communicates the result of the user’s last action with no dependence on audio or visual feedback.

At the end of a word, the user double-taps. This enters a SPACE after the message, and appends the word entered to the message. The system responds with a soft beep to signal that the word has been added to the message. Although a single-tap would suffice, a double-tap was chosen to prevent accidental input that may arise when users hesitate or think twice.

We also leveraged the *iPhone*’s built-in accelerometer to allow users to signal message completion. Shaking the phone ends the message and allows the application to proceed further. The utility of this gesture is noteworthy. A physical motion on the entire device translates to an “end of message” signal to the application. This interaction has no reliance on the visual or auditory channel.

Lastly, we included a mechanism for error correction. Figure 4 depicts a “delete stroke” that deletes the last character entered. The idea is to swipe the finger right-to-left. This gesture was accompanied by a non-speech sound akin to the sound of an eraser rubbing against paper.

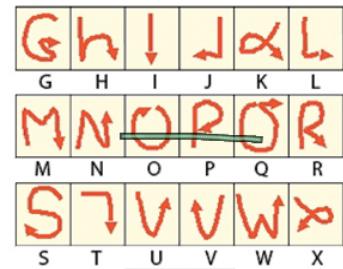


Figure 4. A DELETE stroke is formed by a “left swipe” gesture.

The following summarizes interactions accompanied with non-visual feedback:

Recognized stroke:	character is spoken
Double-tap for SPACE:	soft beep
Unrecognised stroke:	vibration
Delete stroke (←):	erasure sound

### IV. IMPLEMENTATION

#### A. Hardware Infrastructure

The hardware was an Apple *iPhone* and an Apple *MacBook* host system. The host was used for data collection and processing. The two devices communicated via a wireless link over a private, encrypted, ad hoc network. The wireless link allowed participants to move the device freely in their hands.

#### B. Software Architecture

A host application and device interface were developed using Cocoa and Objective C in Apple’s *Xcode*.

The host listened for incoming connections from the *iPhone*. Upon receiving a request and establishing a connection, the program reads a series of 500 phrases ranging from 16 to 43 characters [8]. The host randomly selected a phrase and presented it to the participant for input. It also guided the participant through their tasks by presenting stimuli and responding to user events. *iPhone* generated events (*tracking*, *tapping*, *shaking*, *swiping*, etc.) were logged by the host. As the experiment progressed, the stimulus on the host was updated to alert participants of the text to enter (Figure 5).

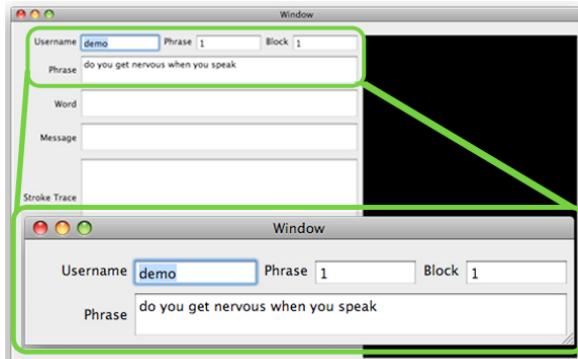


Figure 5. The host interface. Participants saw the zoomed in area. The black region recreates the digitized ink as it is received.

The primary task of the device interface was to detect and transmit all events to the host application, and to respond to any messages received from the host (such as VIBRATE, UPDATE\_WORD, UPDATE\_MESSAGE, etc.).

## V. METHOD

### A. Participants

Twelve paid volunteer participants (8 male, 4 female) were recruited from the local university campus. Participants ranged from 18 to 31 years (*mean* = 23, *SD* = 4.9). All were daily users of computers, reporting 2 to 8 hours usage per day (*mean* = 5.5, *SD* = 2.2). Three participants used a touchscreen phone everyday. Eleven participants were right handed.

### B. Apparatus

The apparatus is described in section IV.

### C. Procedure and Design

Participants completed a pre-test questionnaire soliciting demographic data. The experiment proceeded in three parts: training, eyes-on, and eyes-free. The training phase involved entering the alphabet A to Z three times; entering the phrase “the quick brown fox jumps over the lazy dog” twice; and entering one random phrase [8]. The goal was to bring participants up to speed with the *Graffiti* alphabet. Training was followed by four blocks of eyes-on entry, and then four blocks of eyes-free entry.

Prior to data collection, the experimenter explained the task and demonstrated the software. As a simple control mechanism, error correction was restricted to the previous character only. Participants were instructed to enter text “as quickly and accurately as possible”.

Participants were encouraged to take a short break between phrases if they wished. The software recorded time stamps for each stroke, per-character data, per-phrase data, ink trails of each stroke, and some other statistics for follow up analyses. Timing for each phrase began when the finger was touched to draw the first stroke and ended with the shake of the phone.

Participants sat on a standard study chair with the host machine at eye level (see Figure 6). The interaction was two-handed requiring participants to hold the device in their non-dominant hand and enter strokes with their dominant hand. In the eyes-free condition, participants were required to hold the device under the table, thus occluding it from view.

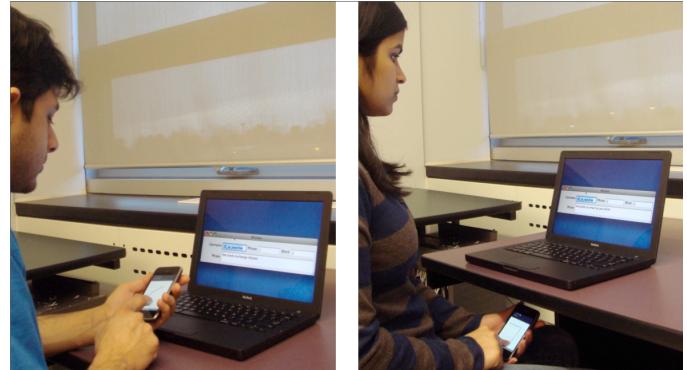


Figure 6. Eyes-on (left); eyes-free (right).

The total amount of entry was 12 participants  $\times$  2 entry modes  $\times$  4 blocks  $\times$  5 phrases/block = 480 phrases.

## VI. RESULTS AND DISCUSSION

### A. Speed

#### 1) Raw entry speed

The results for entry speed are shown in Figure 7. Entry speed increased significantly with practice ( $F_{1,3} = 17.3$ ,  $p < .0001$ ). There was also a difference in speed by entry mode ( $F_{1,11} = 6.8$ ,  $p < .05$ ). The average entry speed for the eyes-on mode was 7.00 wpm, which is the same novice speed reported by Fleetwood et al. [3]. Average entry speed for the eyes-free mode was 8% faster, at 7.60 wpm. It is notable, that there was no degradation in performance in the eyes-free mode. This, alone, is an excellent result and attests to the potential benefits of non-visual feedback modalities for touchscreen phones.

#### 2) Adjusted entry speed

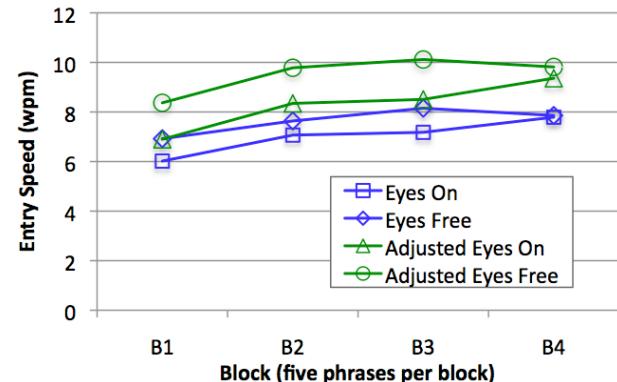


Figure 7. Entry speed (wpm) by entry mode and block. (See text for details of the adjusted results.)

Figure 7 also reports adjusted entry speed. For this metric, we removed the time for entering strokes that were unrecognized. Note that we are not removing errors or corrections, but just the time for unrecognized strokes. Overall, adjusted entry speed differed between modes ( $F_{1,11} = 13.7$ ,  $p < .005$ ). There is a significant effect of block on entry speed between modes as well ( $F_{1,3} = 21.8$ ,  $p < .0001$ ). Participants reached an overall adjusted entry speed of 9.50 wpm in the eyes-free mode; an improvement of 15% over the 8.30 wpm for eyes-on.

### B. Accuracy and KSPC

We analyzed accuracy in two ways. One analysis used the “minimum string distance” (MSD) between the presented and transcribed text [13]. The second analysis used keystrokes per character (*KSPC*) – the number of keystrokes used to generate each character [6]. Of course, “keystrokes” here means “finger strokes”.

#### 1) MSD error rates

Overall, error rates were low at 0.4% (Figure 8). An ANOVA showed no significant effect for error rate, suggesting that the differences were due to variation in participant behavior rather than the modes under test.

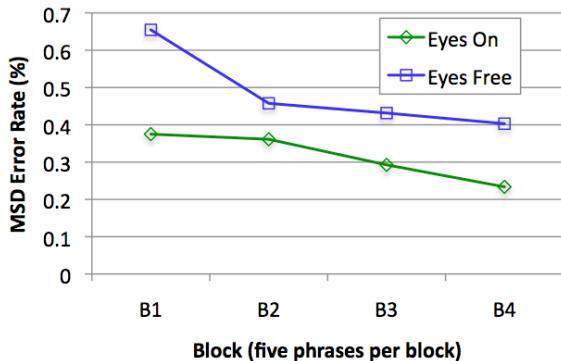


Figure 8. MSD Error rate (%) by block and entry mode.

One interesting observation is that errors in both modes decreased by block. This suggests that users got better with the technique as they progressed. However, it is equally probable that users corrected more errors in later blocks. Since the error rate is so low, we undertook an inspection of the raw data to determine the sort of behaviors present. A closer look revealed that the most common error was forgetting to enter a SPACE at the end of a word (double tap). With practice, users became more alert to this, thus decreasing the error rate across blocks. The SPACE character was forgotten after common words such as “the”, “of”, “is”, etc. Furthermore, if a stroke was unrecognized, it was not logged as an error. Such strokes are accounted for in the *KSPC* measures, discussed next.

#### 2) KSPC analysis

The MSD error analysis only reflects errors remaining in the transcribed text. It is important to also consider strokes to correct errors. For this, we use the *KSPC* metric. Every correction adds two strokes (i.e., delete character, re-enter character). We also add all strokes that were not recognized by the recognizer. If entry was perfect, the number of strokes equals the number of characters; i.e.,  $KSPC = 1.0$ . Results of this analysis are shown in Figure 9. The chart uses a baseline of 1.0. Thus, the entire magnitude of each bar represents the overhead for unrecognized strokes and strokes to correct errors.

Overall, the average *KSPC* in the eyes-free entry mode was 9% higher, at 1.36 compared to 1.24 for eyes-on. The trend was consistent and significant between entry modes ( $F_{1,11} = 38.3, p < .0001$ ), but not within blocks. The difference in *KSPC* between modes is greatest in the 4<sup>th</sup> block. Also, the eyes-on *KSPC* in the 4<sup>th</sup> block is the lowest among blocks, while the eyes-free *KSPC* in the 4<sup>th</sup> block is the highest among blocks. These are opposite extremes, which suggest that

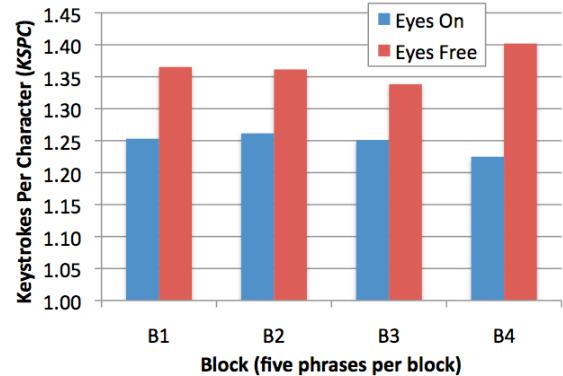


Figure 9. *KSPC* by block and entry mode.

participants were getting better as they progressed in the eyes-on mode. On the flip side, participants invested more effort in the 4<sup>th</sup> block in the eyes-free mode.

#### C. Stroke Analysis

The *KSPC* analysis considered unrecognized strokes at a phrase level. We performed a deeper character-level analysis to find problematic characters. Results are shown in Figure 10. Letters are sorted by their relative frequency in the English language (*x-axis*). Each stroke was captured as an image allowing us to examine unrecognized traces. It was easy to tell what was intended by visually inspecting the images (see Figure 11).

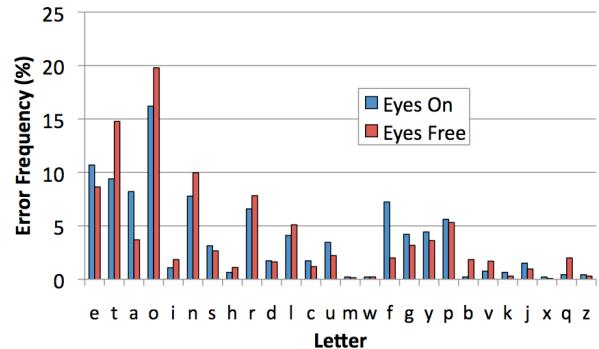


Figure 10. Frequency of unrecognized strokes by character.

Error frequency is defined as the number of times an unrecognized stroke for a given character is found relative to the total number of unrecognized strokes. In Figure 10, the most frequently unrecognized stroke is the character “O”. Nine of twelve participants reported the character as problematic. A closer look at the stroke traces reveals why (see Figure 11).

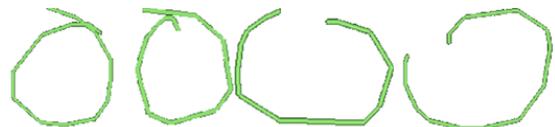


Figure 11. Stroke traces for the letter “O” in eyes-free mode.

Variations in the stroke trace are problematic for the recognizer. Either there is overshoot, undershoot or the begin-end locations are displaced. Participants have the correct mental model, but are unaware of the spatial progress of their finger. Similar issues were reported with “T”, “E”, and “N”.

The number of corrections was not significant between blocks; however, there was a significant main effect found by entry mode ( $F_{1,11} = 8.8, p < .05$ ). Overall, the mean number of corrections was 1.64 per phrase. As expected, the number of corrections in the eyes-free mode was 45% higher at 2.00 per phrase, while the eyes-on mode averaged 1.30 corrections per phrase.

#### D. Other Observations

At the end of testing, participants were asked for feedback on their experience with the entry modes. Many participants felt they could go faster but that doing so may result in unrecognized strokes, which would require additional time in re-entering them. This is a classic example of the speed-accuracy tradeoff. Certainly, this effect would be mitigated with a better recognizer or with continued practice.

The most provocative observation was that some participants exhibited eyes-free behavior in the eyes-on mode. Having quickly learned the strokes, their gaze fixated on the presented text displayed on the host machine, and followed the characters as they progressed. This was accompanied by an occasional glance as needed. When we inquired on this behavior, users responded by saying that “the audio feedback was sufficient to let me know where I am”.

#### VII. CONCLUSIONS AND FUTURE WORK

We have presented a finger-based text entry technique for touchscreen devices combining single-stroke text (*Graffiti*) with auditory and vibrotactile feedback. A key feature is the support for eyes-free text entry. No previous work was found that evaluates such an interaction. From our evaluation, we found an overall entry speed of 7.30 wpm with speeds being 8% higher (7.60 wpm) in the eyes-free mode. This was contrary to our expectations. We expected the occlusion of the device to bring down the overall text entry speed in the eyes-free mode, despite higher overall error rates in the eyes-free mode.

Error rates in the transcribed text were not significant between entry modes. Participants entered text with an accuracy of 99.6%. *KSPC* analyses revealed that eyes-free text entry required an average of 9% more strokes per phrase.

Overall, our results are promising. We plan on extending the work further by adding word prediction using a dictionary in the event of errors. In addition, we plan on evaluating auditory feedback at the word level. We expect that this will increase throughput and provide a resilient mechanism for dealing with unrecognized strokes.

#### VIII. REFERENCES

- [1] Bharath, A. and Madhvanath, S., FreePad: a novel handwriting-based text input for pen and touch interfaces, *Proceedings of the 13th International Conference on Intelligent User interfaces - IUI 2008*, (New York: ACM, 2008), 297-300.
- [2] Cox, A. L., Cairns, P. A., Walton, A., and Lee, S., Tlk or txt? Using voice input for SMS composition, *Personal Ubiquitous Computing*, 12, 2008, 567-588.
- [3] Fleetwood, M. D., Byrne, M. D., Centgraf, P., Dudziak, K. Q., Lin, B. and Mogilev, D., An evaluation of text-entry in Palm OS – Graffiti and the virtual keyboard, *Proceedings of the HFES 46th Annual Meeting*, (Santa Monica, CA: Human Factors and Ergonomics Society, 2002), 617-621.
- [4] Goldberg, D. and Richardson, C., Touch-typing with a stylus, *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems - CHI 1993*, (New York: ACM, 1993), 80-87.
- [5] Lyons, K., Starner, T., and Gane, B., Experimental evaluation of the Twiddler one-handed chording mobile keyboard, *Human-Computer Interaction*, 21, 2006, 343-392.
- [6] MacKenzie, I. S. *KSPC* (Keystrokes per Character) as a characteristic of text entry techniques, *Proceedings of the 4th International Symposium on Mobile Human-Computer Interaction - Mobile HCI 2002*, (London, UK: Springer-Verlag, 2002), 195-210.
- [7] MacKenzie, I. S., Chen, J., and Oniszczak, A., Unipad: Single-stroke text entry with language-based acceleration, *Proceedings of the Fourth Nordic Conference on Human-Computer Interaction - NordiCHI 2006*, (New York: ACM, 2006), 78-85.
- [8] MacKenzie, I. S. and Soukoreff, R. W., Phrase sets for evaluating text entry techniques, *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems - CHI 2003*, (New York: ACM, 2003), 754-755.
- [9] MacKenzie, I. S. and Zhang, S. X., The immediate usability of graffiti, *Proceedings of Graphics Interface '97*, (Toronto, ON, Canada: Canadian Information Processing Society, 1997), 129-137.
- [10] McGookin, D., Brewster, S. and Jiang, W., Investigating touchscreen accessibility for people with visual impairments, *Proceedings of the 5th Nordic Conference on Human-Computer Interaction - NordiCHI 2008*, (New York: ACM, 2008), 298-307.
- [11] Melto, A., Turunen, M., Kainulainen, A., Hakulinen, J., Heimonen, T. and Antila, V., Evaluation of predictive text and speech inputs in a multimodal mobile route guidance application, *Proceedings of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services - MobileHCI 2008*, (New York: ACM, 2008), 355-358.
- [12] Sánchez, J. and Aguayo, F., Mobile messenger for the blind, *Proceedings of Universal Access in Ambient Intelligence Environments - 2006*, (Berlin: Springer), 369-385.
- [13] Soukoreff, R. W. and MacKenzie, I. S., Measuring errors in text entry tasks: an application of the Levenshtein string distance statistic, *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems - CHI 2001*, (New York: ACM, 2001), 319-320.
- [14] Tinwala, H. and MacKenzie, I. S., Letterscroll: Text entry using a wheel for visually impaired users, *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems - CHI 2008*, (New York: ACM, 2008), 3153-3158.
- [15] Wobbrock, J. O., Myers, B. A., Aung, H. H., and LoPresti, E. F., Text entry from power wheelchairs: *Edgewrite for joysticks and touchpads*, *Proceedings of the 6th International ACM SIGACCESS Conference on Computers and accessibility - ASSETS 2004*, (New York, ACM, 2004), 110-117.
- [16] Wobbrock, J. O., Myers, B. A. and Chau, D. H., In-stroke word completion, *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology - UIST 2006*, (New York: ACM, 2006), 333-336.
- [17] Yatani, K., Truong, K. N., An evaluation of stylus-based text entry methods on handheld devices in stationary and mobile settings, *Proceedings of the 9th International Conference on Human Computer Interaction with Mobile Devices and Services - MobileHCI 2007*, (New York: ACM, 2007), 487-494.
- [18] Yfantidis, G. and Evreinov, G. Adaptive blind interaction technique for touchscreens, *Universal Access in the Information Society*, 4, 2006, 328-337.