Evaluation of a New Error Prevention Technique for Mobile Touchscreen Text Entry

Ahmed Sabbir Arif, Wolfgang Stuerzlinger
York University
Toronto, Ontario, Canada
{asarif, wolfgang}@cse.yorku.ca

ABSTRACT

This paper presents a new pressure-based error prevention technique for mobile touchscreen text entry. Two user studies were conducted to compare the new technique with a conventional virtual keyboard, one with novice and another with expert users. Results of the first user study showed that with practice the new technique significantly improves accuracy. Yet, no such indication was observed during the second study.

Author Keywords

Errors; pressure; text entry; touchscreens; virtual keyboard.

ACM Classification Keywords

H5.2 Information Interfaces and Presentation: User Interfaces – Haptic I/O, Interaction styles.

INTRODUCTION

Virtual keyboards are more error prone than physical keyboards and keypads (Clawson et al., 2008). This is likely due to smaller key sizes and the absence of tactile feedback. Prior studies indicate that *substitution errors*, where wrong characters are pressed instead of the correct ones, are the most frequent error committed by mobile users (Clawson et al., 2008; Sad and Poirier, 2009). Here, we present a new pressure-based error prevention technique that attempts to improve touchscreen text entry performance by reducing substitution errors.

THE NEW PRESSURE-BASED TECHNIQUE

The main idea behind the new technique is to use the last entered character to generate in real-time a list of improbable next ones. As those characters are unlikely to appear, their matching keys are then made to be more difficult to activate. That is, users need to apply additional pressure on those keys to input such characters. The assumption here is that input attempts for those characters are unintentional and erroneous. We speculate that the new technique will not compromise text entry in a significant manner, as only normal pressure is required for inputting likely characters. The technique utilizes only two pressure levels, regular and extra, as studies showed that more than two pressure levels do not work well in text entry (McCallum et al., 2009; Wang et al., 2009). Regular pressure represents the force typically applied on touchscreens (approximately 1 N), while extra pressure is relatively more force than that (approximately 3 N). Here, we categorize users as novice if they have never used a touchscreen text entry technique before the study or had a very limited exposure to it. Expert users, on the other hand, are those who use such techniques frequently, i.e. almost every day.

Simulation of Pressure Detection

Currently, three software solutions are used to simulate pressure detection on standard touchscreens. The first is *time-based* and assumes that it takes more time to press a virtual button when extra force is applied (Ramos et al., 2004). This method records the average time it takes to perform a task and uses that as a baseline. When users take longer than the baseline, the system deduces that extra pressure is applied. The second approach is *contact-area-based* and simulates pressure detection by mapping changes in finger contact area to changes in pressure (Forlines and Shen, 2005). The third *touch-point-based* approach detects pressure based on the fact that the touch centre moves more when additional force is applied (Ramos et al., 2004). With this approach the average touch point movement for a task is used as baseline. Then, extra pressure is detected when the touch point moves a greater distance compared to the baseline. The main difference between the contact-area and the touch-point-based approach is that the latter considers *only* the touch centre coordinates. As most current touchscreens do not provide contact area information, many implementations derive contact area from the touch coordinates with various heuristics. In comparison, the touch-point-based approach is simpler and likely more reliable. Hence, we used this approach here.

Bigram Frequency Table

We used a 27×27 bigram frequency table for all letter pairs in English, including the space character (Soukoreff and MacKenzie, 1995), to calculate the probability of a character's appearance based on the preceding one. We use the following equation, also used by Soukoreff and MacKenzie, to calculate the appearance probability P of a character C_n:

$$P(C_n|C_{n-1}) = T(C_{n-1}, C_n)/T_{all} \qquad (1)$$

Here, $T(C_{n-1}, C_n)$ is the total number of occurrences of a specific bigram $(C_n|C_{n-1})$ and T_{all} is the total number of bigrams in the table (here 107,199). Based on several pilots, characters with an occurrence probability less than 0.01% were identified as *unlikely*.

Other Methods for Predicting Unlikely Characters

The use of frequency tables is common in text entry error prevention techniques. For example, the Automatic Whiteout++ technique uses a trigram frequency table along with key proximity information and the time between the previous and the current keystroke to predict unlikely characters (Clawson et al., 2008). Here, we use a bigram frequency table for simplicity. Other choices exist, such as n-grams, a dictionary (Hoffmann, 2009), grammar rules, geometric pattern matching (Kristensson and Zhai, 2005), or language models (Goodman et al, 2002). We assume that if our new technique can show performance improvements with a relatively simplistic prediction method, the use of more advanced methods will only further improve performance.

EXPERIMENT I: NOVICE USERS

We collected raw data from a prior study that compared the pressure-based technique with a timeout-based and the conventional text entry technique with novice users (Arif et al., 2010). Similar to the pressure-based technique, the timeout-based technique used the bigram frequency table to identify and list the unlikely next characters and made those harder to input. In order to input such characters users had to tap-hold the corresponding keys for longer than usual (500 ms).

Apparatus

An Apple iPhone 3G at 320×480 pixel resolution and 163 ppi was used for the study. A custom application, developed with the default iOS SDK, was used. It logged all interactions with timestamps and calculated user performance directly. The application's virtual keyboard was almost identical to iPhone's default virtual keyboard. See Figure 1. It did not feature the default keyboard's key enlargement feedback. Also, the "*.?123*" key to switch to a numerical keyboard and the *Shift* key were removed, as users were not required to input numeric or uppercase characters during the study.

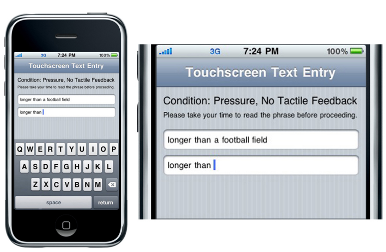

Figure 1. The device (Apple iPhone 3G) and the application used during the exploratory study.

Participants

Twelve participants, aged from 19 to 34 years, average 26, took part in the user study. Five of them were female and all of them were right-handed. They were all novice users. That is, they did not own or use a touchscreen-based device on a regular basis. Note that this study was performed in 2009, before modern smartphones became ubiquitous in Canada.

Procedure and Design

The user study compared three techniques: conventional, timeout-based, and pressure-based. Each technique was examined with and without synthetic tactile feedback in counterbalanced blocks. For the synthetic tactile feedback, the iPhone's vibration motor was activated for 500 ms. During the study participants entered forty short English phrases in two blocks from a set (MacKenzie and Soukoreff, 2003). Phrases were shown to them on the display, all in lowercase. Participants held the device in the portrait position and typed using both of their thumbs. They were asked to take the time to read and understand the phrases in advance, enter those as fast and accurate as possible, and then press the *Return* key when they were done to see the next phrase.

Participants were provided with two practice phrases before each condition to assure that they were reasonably comfortable with the new techniques. Timing started from the entry of the first character and ended with the last. Participants were informed that they could rest between blocks or before typing a phrase. They were also asked to work normally, that is, to correct their errors as they noticed them. Yet they had to exclusively use the *Backspace* key for editing, as we made direct cursor control unavailable to remove a potential confound. Extra pressure was simulated when a users' touch-point moved more than the usual. Based on several pilots we set a threshold of 0.7 mm. The study used a counterbalanced within-subjects design: 12 participants × 3 conditions × 2 blocks × 20 phrases (10 with tactile feedback and 10 with no tactile feedback) = 1440 phrases. The commonly used WPM and Total Error Rate (TER) metrics were used to report text entry speed and accuracy, correspondingly (Soukoreff and MacKenzie, 2003).

Results

D'Agostino Kurtosis tests revealed that the data were normally distributed. A Mauchly's test confirmed that the data's covariance matrix was also circular in form. Thus, repeated-measures ANOVA was used. As prior analysis showed that there was no significant effect of synthetic tactile feedback on performance (Arif et al., 2010), we ignore the tactile feedback factor here.

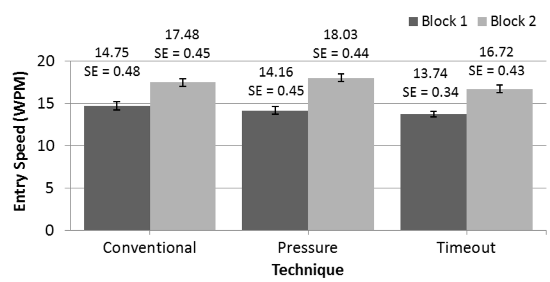

Figure 2. Average entry speed (WPM) with standard error (SE) during the two blocks for all investigated techniques.

Entry Speed

The ANOVA revealed that there was no significant effect of technique on entry speed ($F_{2,11} = 1.35$, ns). There was a significant effect of block ($F_{1,11} = 11.80$, $p < .05$). A Tukey-Kramer test showed that entry speed increased significantly for all techniques during the second block. Yet no significant difference was identified between the techniques. Figure 2 illustrates average entry speed for all techniques during all blocks.

Error Rate

An ANOVA revealed that there was no significant effect of technique on error rate ($F_{2,11} = 0.23$, ns). Also, there was no significant effect of block ($F_{1,11} = 1.52$, ns). Yet a significant effect of block was identified when only the conventional and pressure techniques were considered ($F_{1,11} = 15.38$, $p < .005$). A Tukey-Kramer test showed that the pressure technique significantly reduced error rate during the second block. Further investigation revealed that about 2% of all characters were blocked during the pressure condition. 71% of these were correctly predicted as unlikely, erroneous, or unintended input, while the rest were false positives. Figure 3 illustrates average error rate for all techniques in all blocks.

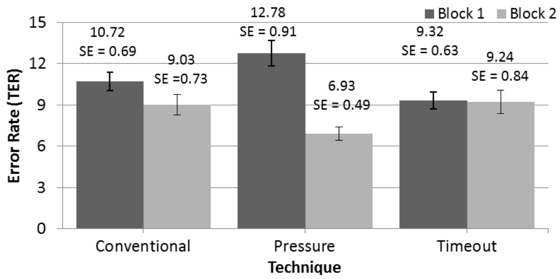

Figure 3. Average error rate (TER) with standard error (SE) during the two blocks for all investigated techniques.

Discussion

No significant effect of technique was identified on entry speed and error rate. However, we found a significance of block, which signifies learning. While text entry speed improved similarly for all techniques, accuracy improved significantly more with the pressure technique compared to the conventional one. On average, errors decreased by 46% with pressure during the final block, compared to 23% for the conventional technique. This indicates the possibility that with practice or training the performance of the pressure technique may increase even more.

Figure 4. The device (Apple iPhone 4) and the application used during the final study.

ISSUES WITH THE FIRST EXPERIMENT

After careful analysis, several issues were identified with the first study setup. First and although insignificant, the synthetic tactile feedback may have influenced the performance of some users. Second, the simulated pressure detection technique detected pressure exclusively based on touch-point movement. This may have failed to detect pressure for some users. Finally, no visual feedback was provided when extra pressure was detected. This may have confused some users, as it was hard for them to identify whether the system detected pressure or not. This also caused delays in text entry, as users had to verify their input more frequently to make sure that the system recognized their input. Therefore, a second study was conducted to further investigate an appropriately modified pressure-based technique.

EXPERIMENT II: EXPERT USERS

We conducted a second user study to further evaluate the new technique with expert users.

Apparatus

We used a custom application, developed with the iOS SDK, on an Apple iPhone 4 at 640×960 pixel resolution and 326 ppi during this user study. Its virtual Qwerty keyboard was identical to the iPhone's default keyboard. See Figure 4. The custom keyboard featured the iPhone default key enlargement feedback. However, no auditory feedback was provided. It logged all interactions with timestamps and calculated user performance directly.

Participants

Twelve participants, aged from 21 to 29 years, average 24, participated in the study. Four of them were female and all of them were right-handed. They were all experienced Apple iPhone or iPad users. That is, they either owned or used those devices on a regular basis.

Procedure

We used the same procedure as the previous study. Yet we made a few changes to address several design issues. We removed the synthetic tactile feedback condition to avoid a potential confound. We also instructed users to hold the device in the portrait position with their dominant hand and then to input text using the thumb of the same hand. This is motivated by a recent survey, which found this to be the most frequently used position with mobile users (Hoober, 2013). Also, to assure better pressure detection accuracy, we used a hybrid of touch-point movement and time-based methods to detect extra pressure. The system simulated detection of extra pressure when users took more than the average time to tap on a key **and/or** when their touch-point moved more than usual (Arif and Stuerzlinger, 2013). Based on several pilots, we used a threshold of 200 ms for the tap time, and a threshold of 0.4 mm for touch centre movement. Finally, users were provided with visual feedback on extra pressure via the default iPhone key enlargement feature. For the blocked (less probable) keys, this feedback was provided only when users applied extra pressure to override the blockade. This permitted users to quickly identify blocked keys without constantly verifying their input. We used a counterbalanced within-subjects design: 12 participants ×2 conditions ×3 blocks × 15 phrases = 1080 phrases in total.

Results

We filtered outliers beyond 3σ from the mean, which was < 1% of the data. D'Agostino Kurtosis tests revealed that the data were normally distributed. Also, a Mauchly's test confirmed that the data's covariance matrix was circular in form. Thus, we used repeated-measures ANOVA.

Entry Speed

An ANOVA revealed that there was no significant effect of technique on entry speed ($F_{1,11} = 0.48$, ns). There was also no significant effect of block ($F_{2,22} = 0.21$, ns). On average, entry speeds for regular and pressure were 25.12 (SE = 0.36) and 24.61 WPM (SE = 0.35), respectively. Figure 5 illustrates average entry speed for both techniques during the three blocks.

Error Rate

An ANOVA revealed that there was no significant effect of technique on error rate ($F_{1,11} = 1.45$, ns). There was also no significant effect of block ($F_{2,22} = 1.91$, ns). On average, TER for regular and pressure were 8 (SE = 0.33) and 7.9%

(SE = 0.33), respectively. Figure 6 illustrates average error rate for both techniques during the three blocks. Further analysis revealed that about 2% of all characters were blocked during the pressure condition, 83% of which were correctly predicted as unlikely, erroneous, or unintended input, while the rest were false positives.

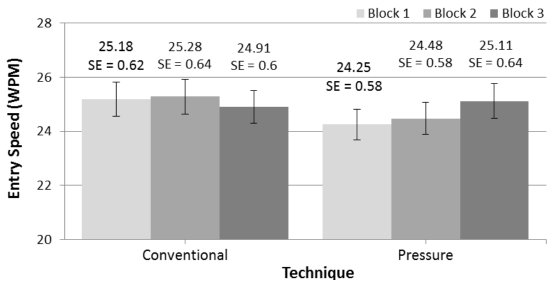

Figure 5. Average entry speed (WPM) with standard error (SE) for both techniques during the three blocks. Note the scale on vertical axis.

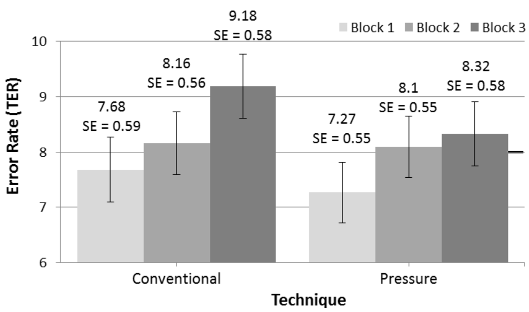

Figure 6. Average error rate (TER) with standard error (SE) for both techniques during the three blocks. Note the scale on vertical axis.

Discussion

As one would expect, average entry speed and accuracy were substantially higher with the expert users than with novices. However, unlike the first study no indication of learning was observed. The overall speed and accuracy remained almost the same for both techniques throughout. A few factors may have caused this. First, different devices were used during the two studies and the virtual keyboards were also slightly different. Moreover, we used an improved pressure detection simulation method for the second study. However and most importantly, in the post-study interviews for the second study almost all users commented that it was hard for them to adapt to the new technique as they were already used to the default iOS error prevention technique, called key-target resizing. In this approach the *invisible* underlying target areas are dynamically resized based on the probabilities associated with each character, instead of the visual representation of the keys. While our technique blocks the most improbable characters, the iOS technique makes the most probable ones easier to enter, which is almost an inverse approach. This seems to have confused several users as they expected the keyboard to act in a certain way.

CONCLUSION AND FUTURE WORK

We presented results of two studies that compared a virtual keyboard augmented with a new pressure-based error prevention technique with the conventional one. The first study used novice and the second expert users. Results of the first study showed that with practice the new technique improves accuracy, while no such indication was observed during the second study. Based on post-study user feedback, we speculate that this is primarily due to users' previous adaptation to an existing error prevention technique. However, as currently we do not have sufficient data to verify this assumption, we intend to investigate this matter further. We would also like to improve the pressure-based technique by providing users with visual feedback on which characters are predicted as *unlikely* and how much pressure one needs to apply to input those characters.

REFERENCES

Arif, A. S. and Stuerzlinger, W. Pseudo-pressure detection and its use in predictive text entry on touchscreens. OzCHI '13, ACM (2013), 10 pages.

Arif, A. S., Lopez, M. H., and Stuerzlinger W. Two new mobile touchscreen text entry techniques. Posters GI '10, CEUR-WS.org/Vol-588 (2010), 22-23.

Clawson, J., Lyons, K., Rudnick, A., Iannucci, R. A., Jr., and Starner, T. Automatic whiteout++: correcting mini-QWERTY typing errors using keypress timing. CHI '08, ACM (2008), 573-582.

Forlines, C. and Shen, C. DTLens: multi-user tabletop spatial data exploration. UIST '05, ACM (2005), 119-122.

Goodman, J., Venolia, G., Steury, K., and Parker, C. Language modeling for soft keyboards. IUI '02, ACM (2002), 194-195.

Hoffmann, A., Spelmezan, D., and Borchers, J. TypeRight: a keyboard with tactile error prevention. CHI '09, ACM (2009), 2265-2268.

Hoober, S. How do users really hold mobile devices? UXmatters, Feb 18, 2013. http://shar.es/jNIP8

Kristensson, P. O. and Zhai, S. Relaxing stylus typing precision by geometric pattern matching. IUI '05, ACM (2005), 151-158.

Lewis, J. R., Potosnak, K. M., and Magyar, R. L. Keys and keyboards. In M. Helander, T. K. Landauer, and P. V. Prabhu, eds. Handbook of Human-Computer Interaction, Elsevier, Amsterdam, 1997, 1285-1311.

MacKenzie, I. S. and Soukoreff, R. W. Phrase sets for evaluating text entry techniques. Ext. Abstracts CHI '03, ACM (2003), 754-755.

McCallum, D. C., Mak, E. Irani, P., and Subramanian, S. PressureText: pressure input for mobile phone text entry. Ext. Abstracts CHI '09, 4519-4524.

Ramos, G., Boulos, M., and Balakrishnan, R. Pressure widgets. CHI '04, ACM (2004), 487-494.

Sad, H. H. and Poirier, F. Using pictographic representation, syntactic information and gestures in text entry. HCII '09, Springer-Verlag (2009), 735-744.

Soukoreff, R. W. and MacKenzie, I. S. Metrics for text entry research: an evaluation of MSD and KSPC, and a new unified error metric. CHI '03, ACM (2003), 113-120.

Soukoreff, R. W. and MacKenzie, I. S. Theoretical upper and lower bounds on typing speed using a stylus and soft keyboard. BIT 14, 6 (1995), 370-379.

Wang, F., Cao, X., Ren, X., and Irani, P. Detecting and leveraging finger orientation for interaction with direct-touch surfaces. UIST '09, ACM (2009), 23-32.