Pseudo-Pressure Detection and Its Use in Predictive Text Entry on Touchscreens

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

ABSTRACT
In this article we first present a new hybrid technique that combines existing time- and touch-point-based approaches to simulate pressure detection on standard touchscreens. Results of two user studies show that the new hybrid technique can distinguish (at least) two pressure levels, where the first requires on average 1.04 N and the second 3.24 N force on the surface. Then, we present a novel pressure-based predictive text entry technique that utilizes our hybrid pressure detection to enable users to bypass incorrect predictions by applying extra pressure on the next key. For inputting short English phrases with 10% non-dictionary words a comparison with conventional text entry in a study showed that the new technique increases entry speed by 9% and decreases error rates by 25%. Also, most users (83%) favour the new technique.

Author Keywords
Mobile phone; touchscreen; mobile text entry; predictive text; pressure; virtual or soft keyboard.

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

General Terms
Performance; Design; Human Factors.

INTRODUCTION
Although much work has been targeted at pressure-based user interfaces and widgets for tabletops and large displays, few attempts focus on mobile devices. The main reason for this is technological. No current mobile device provides hardware support for measuring pressure. However, recent work (Graham-Rowe, 2010; Nurmi, 2009) indicates that future mobile phones may include pressure-sensitive touchscreens as an alternative interaction modality. A recent opaque touchpad already provided support for detecting pressure levels (Synaptics, 2013).

Several software solutions are available to detect pressure on touchscreens. Yet, none of these are broadly applicable as they either increase the time to perform tasks that involve additional pressure, or are user specific, e.g., due to different finger sizes. Thus, we present a new hybrid pseudo-pressure detection technique that combines the existing time-based and a new touch-point-based approach to detect pressure. We evaluate the new technique in a study for two different pressure levels. We investigate whether users interpret general terms such are *regular* and *extra* pressure in a reasonably consistent manner and explore how much force is really applied for each level in a separate user study.

Almost all recent virtual keyboards augment text entry with prefix-based word prediction and auto-correction. These methods suggest the most probable word(s) based on what users are typing and automatically correct "probably" misspelled words. Almost all these methods require users to tap on an area outside the virtual keyboard to *reject* or *bypass* a suggestion. This requires additional mental preparation, visual scan time, as well as a finger movement to the target. Due to the small target sizes used, users may need several attempts to reject a prediction. This increases the possibility of accidentally selecting the wrong word as well. Here, we present a new pressure-based technique for prediction rejection that does not require tapping outside the keyboard. Instead, it requires users to apply more pressure for the tap on the next key, which may be any key. We compare the performance of this technique with the conventional technique in a user study. Finally, we also study the user experience of the new hybrid pressure detection simulation and the new pressure-based predictive text entry technique.

RELATED WORK
Herot and Weinzapfel (1978) were the first to investigate the ability of humans to apply pressure and torque on a computer screen. Buxton et al. (1985) also explored touch-sensitive tablet input. They concluded that although pressure control can be difficult in the absence of button clicks or similar tactile feedback, it is a promising research area. Srinivasan and Chen (1993) asked users to control the force applied to a sensor under several different conditions as well as different forms of feedback. Their results suggest that pressure interfaces need to have a force resolution of at least 0.01 N to make full use of human capabilities. Mizobuchi et al. (2005) examined the properties of force-based input on a mobile device by asking participants to apply force in ten predetermined target levels, ranging from 0 N to 4.0 N, with and without visual feedback. They suggested that pressure levels from 0 N to 3.0 N are comfortable and controllable for users. Ramos et al. (2004) investigated users' ability to perform discrete target selection tasks by varying a stylus's pressure, with full or partial visual feedback. Based on their results they proposed a number of pressure widgets for tasks such as zooming and selection. Similar to Mizobuchi et al. (2005), they concluded that users could control 6 ± 1 pressure levels without major difficulties. Subsequent studies (McCallum et al., 2009; Wang et al.,

2009) showed that this does not work well for text entry, as techniques with more than two pressure levels suffer from relatively higher error rates. In a different avenue of work, Zeleznik et al. (2001) proposed an alternative to binary button switches on mice. With their technique one had to press a button lightly to activate its first state and harder to activate its second state. Likewise, Cechanowicz et al. (2007) permitted users to apply different pressure levels on a mouse to simultaneously control cursor position and multiple levels of discrete selection modes in desktop tasks. They, however, did not evaluate their techniques.

Pressure in Text Entry

McCallum et al. (2009) introduced a pressure-based text entry technique for the standard 12-key mobile keypad that utilized three pressure levels. Their technique yielded a higher *expert* text entry rate than Multi-tap, but at the expense of an 8.7% error rate, compared to a baseline of 2.8%. Tang et al. (2001) developed a three-key chorded keyboard with three pressure levels, which again suffered from high error rates, ~18% after three trials. Hoffmann et al. (2009) created a full-size physical Qwerty keyboard that uses key resistance to prevent errors. The keyboard used dictionary, grammar, and context tests to identify probably erroneous characters, and then made those keys harder to press by increasing the resistance. This reduced erroneous keystrokes by 87% and correction attempts by 46%, on average. Similarly, Dietz at al. (2009) developed a pressure sensitive physical keyboard that used different pressure levels to enable users to delete one character or a word using the *Backspace* key. However, they did not evaluate their work. Jong et al. (2010) presented a tactile input method for pressure sensitive keyboards based on the detection and classification of pressing movements on already pressed-down keys. Yet, they too did not compare the new techniques with conventional ones. Brewster et al. (2009) presented several pressure-based techniques to switch between uppercase and lowercase letters on a virtual Qwerty keyboard. Some of their techniques were more accurate and faster than the standard *Shift* key. However, to our knowledge, no work has been done on using pressure in predictive text entry.

Existing Pressure Detection Simulation Techniques

Currently, two software-level solutions are widely used on touchscreens to simulate pressure detection: time-based and contact-area-based. The time-based approach simulates pressure detection based on the assumption that it takes more time to perform a task when extra pressure is applied (Cechanowicz et al., 2007; Ramos et al., 2004). It records the average time it takes to perform a task and uses that as a baseline. When users take more time than the baseline, the system deduces that extra pressure is applied. Several mobile applications, such as Doodle Buddy[1] and TypeDrawing[2], use this to simulate pressure detection. The limitation of this approach is that it forces users to take additional time to perform all tasks that require extra pressure. Yet Raisamo (1999) showed that many tasks take almost the same time regardless of the level of pressure applied. Thus, a time threshold for pressure may unnecessarily slow users down.

The contact-area-based approach relies on the fact that human fingertips spread wider over the point of contact when additional pressure is applied (Buxton, 2013). It simulates pressure detection by mapping the changes in a finger's contact area to changes in pressure. More specifically, this approach maps different finger areas to different pressure levels, and simulates pressure detection based on that. This approach was first implemented by Forlines and Shen (2005), although they did not elaborate on their implementation. A detailed explanation was later provided by Benko et al. (2006). They also demonstrated how this approach could be used in touchscreen user interfaces. Boring et al. (2012), also investigated pressure detection simulation using the thumb's contact area. The fundamental limitation of this approach is that finger contact areas depend not only on the amount of pressure applied, but also on finger sizes and different touch types, i.e., vertical and oblique (Wang at al., 2009). Thus, this approach cannot be used with all users or with interactive pens/styli. Besides, most current touchscreens provide touch coordinates, not contact area information.

Touch-Point Movements

Several studies identified that touch-points or coordinates move with extra pressure (Wang et al., 2009; Ramos et al., 2004; Boring et al., 2012). Wang et al. (2009) argued that most touch interactions are oblique due to the common practices in handing physical objects and to accommodate long fingernails of some users. They observed that touch-points shift more when users apply extra pressure. Boring et al. (2012) also reported this. Interestingly, a similar phenomenon was observed for stylus-based interactions as well (Ramos et al., 2004).

Additional Methods

Hwang and Wohn (2012) proposed an alternative pressure detection simulation technique. They monopolize a mobile device's built-in microphone to detect five different pressure levels by mapping different sound amplitude to different pressure levels. In a pilot study their technique was found to be 94% accurate. Heo and Lee (2011) used acceleration data along the z-axis to differentiate between two pressure levels on touchscreens. In an investigation, their technique was about 90% accurate. It is unclear whether this method will work in mobile settings or not. Watanabe et al. (2012) used the light transmitted by touchscreens onto fingernails to estimate the level of force applied, which changes the intensity of the transmitted light. This method is impractical in many situations, as it requires a light sensor attached to be attached to one's fingernail(s).

NEW HYBRID METHOD

We propose a new hybrid method that combines the time- and touch-point-based approaches, which avoids the limitations of existing pressure detection simulation techniques. The new method uses the average time it takes to perform a task and the average touch-point movement for that specific task as baselines. Then, it simulates extra pressure when users take more time

[1] http://blog.pinger.com/tag/doodle-buddy

[2] http://www.storyabout.net/typedrawing

and/or their touch-point moves a larger distance than the baselines while performing that task. We hypothesize that the new hybrid technique will simulate pressure detection faster and more reliably. If the touch-point threshold is crossed before the time threshold, users will not have to wait to trigger extra pressure detection. In contrast, tasks that require additional time to perform with extra pressure will be detected through the time-based approach. Thus, this approach does not only save time but also increases the probability of detecting extra pressure.

As discussed earlier, the touch-point moves further when additional pressure is applied. This movement is somewhat proportional to the force applied on the screen. Our approach simulates pressure detection based on this, which makes it somewhat similar to the contact-area-based approach. The main difference is that our approach does not use contact area but considers *only* the touch centre coordinates (x- and y-axis). This makes it simple, straightforward, and theoretically even applicable to styli-based interactions (Ramos et al., 2004). As most current mobile touchscreens do not provide contact area information, many implementations derive contact areas from the touch coordinates with heuristics (Boring et al., 2012). As our approach works directly on the touch point movement, it is more reliable.

EXPERIMENT I: PRESSURE DETECTION CRITERIA

We conducted a study to validate our assumption that our hybrid technique can detect pressure more efficiently. We examined two pressure levels: *regular* and *extra*, where regular represents the level of pressure typically applied on touchscreens, and extra represents relatively stronger pressure. We investigated only two levels, as our main target is pseudo-pressure in text entry and more than two pressure levels do not work well in text entry (McCallum et al., 2009; Wang at al., 2009).

Apparatus

We used a custom application, developed with the iPhone SDK, on an Apple iPhone 4, 115.2×58.6×9.3 mm, 137 grams, at 640×960 resolution for the user study. The application's virtual Qwerty keyboard was visually identical to the iPhone's default keyboard. See Figure 1. However, we disabled the *Shift* and the "*?.123*" keys, as these were not required during the study. The custom keyboard featured the key enlargement feedback of the iPhone's default keyboard. No auditory feedback was provided. The application calculated all metrics directly and logged all action events with timestamps.

Participants

Twelve participants, aged from 21 to 29 years, average 24, participated in the study. We recruited experienced touchscreen users, which ensured familiarity with touchscreens. They were recruited through online social communities, local university e-mailing lists, and by word of mouth. Five of them were female and all were right-handed. They were all familiar with the Qwerty layout and had prior experience with touchscreen devices. They all received a small compensation for participating.

Procedure

During the study participants were asked to input random characters by tapping on the keys of the virtual keyboard with regular and extra pressure. There were four blocks. In each block participants inputted 27 uppercase and 27 lowercase characters, presented one at a time in random order and case. Participants had to input the lowercase character by applying regular pressure and the uppercase ones with extra pressure. The lower- and uppercase space characters were displayed as "sp" and "SP". Characters were presented in random order and case to avoid ordering effects. Each tap time was recorded from the moment users touched the keyboard until they lifted their fingers. Likewise, touch point movement distance was calculated from the touch-down to the touch-up point.

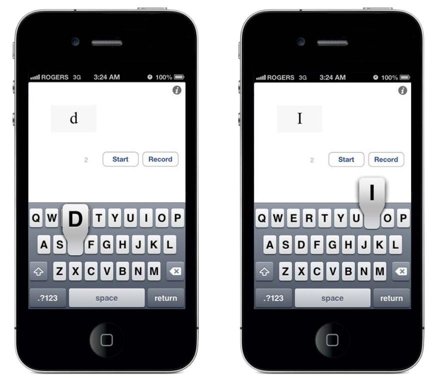

Figure 1. The custom application used during Experiment I. In the first screenshot users have to tap on the *D* key with regular pressure. In the second screenshot they have do the same on the *I* key with extra pressure.

Upon touch-up the next character was automatically presented. Participants could take short breaks (max. 5 minutes) between blocks. They were instructed to hold the device with their dominant hand in portrait orientation and then to type using the thumb of that hand. We used portrait orientation as mobile users use it most frequently (Hoober, 2013). Participants were instructed to first examine the presented character, understand the level of pressure they need to apply, and then to perform the specified task. They were not provided with practice trials. Participants were not required to fix mistakes, as this study focused on differences between regular and extra pressure, in terms of touch duration and touch point movement, not on input accuracy.

Design

We used a within-subjects design. The two factors were regular and extra pressure. In summary, the design was:

12 participants ×
4 blocks ×
54 characters (27 regular, 27 extra, randomized) ×
= 2,592 characters in total.

Results

Both Anderson-Darling and D'Agostino Kurtosis tests revealed that the study data was not normally distributed. Thus, we used Wilcoxon Signed-Rank test for all analysis.

Tap Time (Milliseconds)

A Wilcoxon Signed-Rank test indicated that there is a significant difference between regular and extra pressure in terms of tap time (z = -30.49, p < .0005). The average tap times for regular and extra pressure were 116.9 ms and 390.07 ms, respectively. Figure 2 illustrates this.

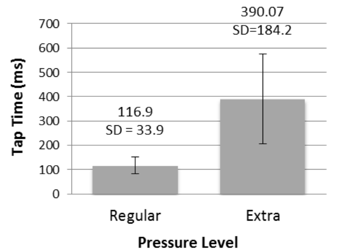

Figure 2. Average tap time (ms) for different pressure levels. Error bars represent ±1 standard deviation.

Touch Point Movement (Millimetre)

A Wilcoxon Signed-Rank test revealed a significant difference between regular and extra pressure in terms of touch point movement ($z = -17.76$, $p < .0005$). The average touch point movements for regular and extra pressure were 0.289 mm and 0.503 mm, correspondingly. Figure 3 illustrates this.

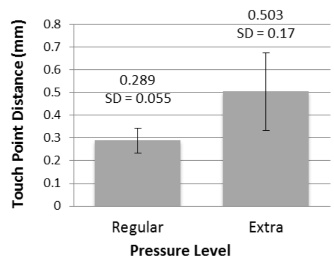

Figure 3. Average touch point movement (mm) for different pressure levels. Error bars represent ±1 standard deviation.

Discussion

The results of the study establish that there is a significant difference between regular and extra pressure both in terms of tap time and touch point movement. On average, taps took more time and the touch point moved more when extra pressure was applied. This verifies our claim that a hybrid criterion can be useful to simulate pressure detection, at least for two pressure levels. Further study identified three distinct user groups. About 67% of users took significantly more time to tap. But their touch point did not move significantly. The touch point of 8% moved significantly more. Yet these did not take significantly more time. The remaining 25% took both significantly more time and their touch point moved more with extra pressure. This indicates that an approach based on time, contact-area, or touch-point *alone* cannot accommodate all users, as user behaviour varies too much. In contrast, our new hybrid approach supports all three groups.

The data was further analysed to investigate the effect of key positions on tap time and touch point movements by segmenting the virtual keyboard into 3×1 and 2×2 grids, similar to Parhi et al. (2006). No statistical significance was identified and we did not pursue this further.

EXPERIMENT II: REGULAR VS. EXTRA PRESSURE

The results of the first study verified that the two different pressure levels are easily distinguishable through a combination of tap time and touch point movement. Also, different users seem to interpret regular and extra pressure in a reasonably consistent way.

While some have investigated the amount of force applied on flat surfaces (Srinivasan and Chen, 1993; Mizobuchi et al., 2005; Ramos et al., 2004), their results do not apply directly to our system as they either explored more than two pressure levels or used a pressure-sensitive stylus. Thus, we conducted a separate study to detect the force users apply when limited to two pressure levels.

Apparatus

A *DYMO* M5 digital postal scale was used for this study. The scale had 5 lb weight capacity. It displayed the weight of an object in 0.1 oz increments with ±0.1 oz accuracy.

Participants

Fourteen participants, aged from 23 to 46 years, average 31.4 years, participated in the study. They were recruited through online social communities, local university e-mailing lists, and by word of mouth. Three of them were female and all of them were right-handed. They all owned and frequently used a touchscreen-based mobile device.

Figure 4. A participant tapping on the digital scale.

Procedure

The user study used a finger posture akin to holding a touchscreen device with one hand and then tapping on it with the thumb of the same hand. For this, the digital scale was placed on the table and participants were asked to sit in front of it. They were then instructed to place the closed fist of their dominant hand on the table, and to tap on the scale with only the thumb of that hand, as if tapping on a virtual keyboard. See Figure 4. This design eliminates the option of using arm strength to apply pressure and limits users to using only their thumb. There were two conditions: *regular* and *extra* pressure. In the regular pressure condition participants were asked to tap on the scale six times with *regular* pressure. For the extra condition they were asked to do the same with *extra* pressure. Conditions were counterbalanced to avoid asymmetric skill transfer. Similar to the first study and during the regular pressure condition, participants were asked to tap on the scale with the amount of pressure they usually use on a virtual keyboard. In the extra pressure condition they were asked to apply relatively more pressure than that. The experimenter recorded readings in ounces (oz) with pen and paper. We later converted that to newton (N). Participants could not see the scale readings as this might influence their performance.

Design

A within-subjects design was used for the two factors: regular and extra pressure. In summary, the design was:

14 participants ×
2 conditions (*regular* and *extra*, counterbalanced) ×
6 taps on the scale
= 168 taps in total.

Results

Both Anderson-Darling and D'Agostino Kurtosis tests on the dependent variables revealed that the data were not normally distributed. Hence, we used a Wilcoxon Signed-Rank test for all analysis.

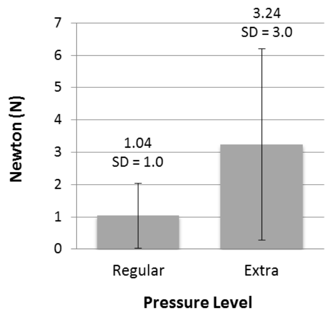

Figure 5. Average force (N) applied for regular different pressure levels. Error bars represent ±1 standard deviation.

Force Applied (Newton)

A Wilcoxon Signed-Rank test indicated that there is a significant difference between regular and extra pressure in terms of applied force (z = -2.86, p < .004). The average force applied for regular and extra pressure was 1.04 N and 3.24 N, respectively. Figure 5 illustrates this.

Discussion

The forces applied during regular and extra pressure were significantly different. Users applied on average 1.04 N for regular and 3.24 N for extra pressure. This matches Mizobuchi et al.'s (2005) work, where they identified force levels between 0 and 3 N to be comfortable and 4 N to be (too) strong. Similarly, we found a force level well below 3 N for regular and about 3 N for extra pressure. We recognize that our results are only an approximation as data was collected on a postal scale instead of a pressure sensitive touchscreen.

PRESSURE-BASED PREDICTIVE TEXT ENTRY

The results of the first two studies established that users comprehend regular and extra pressure in a reasonably consistent manner. Also, the first study confirmed that a hybrid of time- and touch-point-based approaches can detect pressure reliably on touchscreens. Now we apply this to text entry, where we developed and evaluated a new pressure-based, predictive text entry technique.

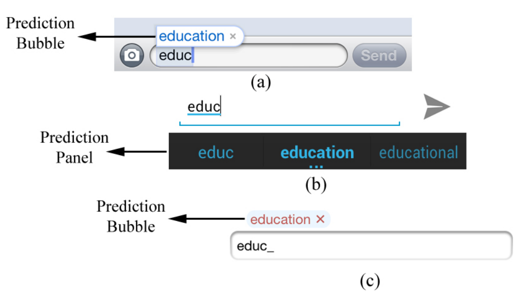

Figure 6. Default word prediction systems on the (a) Apple iPhone, (b) Android OS, and (c) our custom application.

Word Prediction in Virtual Keyboards

Today, almost all virtual keyboards augment text entry with prefix-based word prediction and autocorrection. These suggest the most probable word(s) based on what users are typing and often automatically correct a "likely" misspelled word. Figure 6**Error! Reference source not found.** (a) illustrates word prediction on the iPhone keyboard, where the most probable word completion "education" is suggested based on the input (or prefix) in a prediction bubble. When a word is suggested, users can perform any of the following:

1) *Accept* the prediction by tapping on the *Space* key. This will replace the partially inputted word with the suggested word, followed by a space character.
2) *Reject* or bypass prediction for that word by tapping on the prediction bubble. This will remove the prediction bubble along with the predicted word.
3) *Ignore* the prediction and continue typing. Here, the system will keep updating the suggestion based on the prefix. For instance, if we continue typing and input "educab", the system will update the suggestion to "educable", which is the most probable word starting with that prefix. When the system fails to find a match based on the prefix, it often assumes that a spelling mistake has been made. It then suggests the "closest" most likely word. For instance, if we input "educc", the system will assume that we made a spelling mistake, hence, will continue suggesting "education".

Some virtual keyboards suggest more than one word. The default Android keyboard suggests the two most probable words in a prediction panel, placed above the keyboard. See Figure 6 (b). With this keyboard users can again perform any of the above-mentioned actions. Note that the system highlights the word "education", to signify that this word will be used for auto-completion when hitting *Space*. To reject or bypass this suggestion, one has to tap either on the typed text (in the left of the panel) or the second most probable word (elsewhere in the panel). All of these actions require users to tap in an area away from the virtual keyboard, on both platforms. After bypassing or rejecting a prediction, both keyboards resume suggestions when users input a space character or tap on the *Return* or *Backspace* keys.

A New Pressure-Based Text Entry Technique

Cancelling a prediction requires the user to tap on an area outside of the virtual keyboard, a relatively distant target. The time to do this depends not only on mental preparation and visual scan times, but also on the distance and width of the target (the Fitts' law parameters). Furthermore, the small target size increases the potential for errors. For example, while attempting to tap on a prediction bubble to reject a prediction, one may miss the target. Tapping then on the *Space* key without visual verification will result in input of an entirely wrong word.

To address these issues, we present a new pressure-based predictive technique that does not require tapping outside the keyboard. The new technique resembles and behaves like the default iPhone keyboard. However, one can apply extra pressure on the next key (which may be any key) to bypass prediction. Figure 6 (c) illustrates word prediction in the new technique, where the system predicted the most probable word based on the inputted prefix. Now, one can perform any of the above-mentioned tasks: accept, reject, or ignore the prediction. To *reject* the prediction, one only has to tap on the next key with extra pressure. For example, to input "educo", one taps on the

O key with extra pressure. As the new technique reduces the average finger movement distance, we hypothesize this will not only improve text entry speed but also reduce errors significantly. We used the default iPhone keyboard as a baseline as most users use this or a similar keyboard on their devices (Arif, 2012). Also, the intent of this work is not to evaluate the quality of the predictive system, but to evaluate pressure as a modality in predictive text entry, which is mostly independent.

Word Prediction

A straightforward word prediction system was created for our study. For this, we used a list of the most frequent 5000 English words (Davies, 2011), extracted from the 450 million-word Corpus of Contemporary American English (COCA). Each time users input a character the system attempts to find matches in the list and suggests the most frequent word in a prediction bubble. See Figure 6 (c). Based on several pilots, the following conditions were applied in the prediction system:

1) At least two characters have to be inputted for the system to suggest a word. For example, users have to input at least "ed" to get the prediction "education".

2) If no match was found, the system will assume that the user made a spelling mistake and will suggest the most frequent word with a Levenshtein string distance (Levenshtein, 1966) less than 3 to the inputted prefix. For example, with "edution" as input, the system will suggest "education", with an edit distance of two.

3) Similar to many other predictive systems and after the user rejects a prediction, the system resumes suggestions on a *Space*, *Return*, or *Backspace*.

We informally tested our prediction system *without* pressure detection with three experienced Apple iPhone users. They all inputted random texts for 10 minutes. None of them noticed any notable difference between the tested and the default iPhone prediction system, in terms of prediction accuracy or processing time.

Pressure Detection

Pressure detection was simulated based on the proposed hybrid approach. A threshold of 200 ms was used for the tap time, and a threshold of 0.389 mm for touch point movement. Extra pressure was detected when users took more time **and/or** their fingers slid more than the above-mentioned thresholds. These values were picked based on the results of the first study, by selecting the "deepest" spot between the two alternatives as thresholds.

EXPERIMENT III: PRESSURE IN TEXT ENTRY

This study compared the new pressure-based predictive text entry technique with the conventional technique (the default iPhone method). It also explored user preference for pressure as an alternative modality and (indirectly) evaluated the hybrid pressure detection approach.

Apparatus

The same physical apparatus as in the first user study was used. The custom application was modified to support predictive text entry, as discussed previously. We again disabled the *Shift* and the "*?.123*" keys as users were not required to use these during the study.

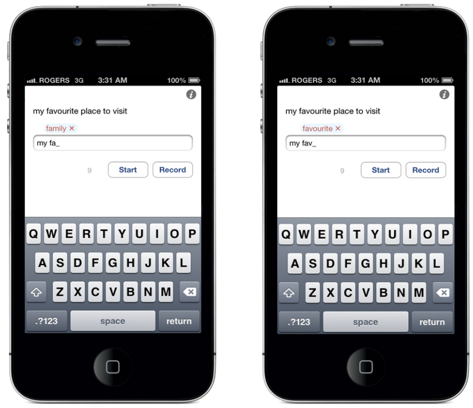

Figure 7. The custom application for Experiment III. Note the change in the prediction in the two screenshots.

Participants

Twelve new participants, aged from 22 to 32 years, average 28 years, participated in the study. They were recruited through online social communities, local university e-mailing lists, and by word of mouth. Two participants were female and one was left-handed. They were all proficient in the English language. That is, they were either native speakers or had spent at least five years in an English-speaking environment. All were frequent mobile phone users and had prior experience with touchscreens (on average 2 years). They all used a virtual Qwerty keyboard on their mobile device to input text. Amongst them, six used both word prediction and auto-correction, one used only word prediction, two used only auto-correction, and the rest none of the features.

Figure 8. The experiment setup for the final user study. Here, a user is inputting short English phrases in a seated position with the custom software.

Procedure

We compared two virtual keyboards, both of which use prefix-based word prediction: the *pressure*-based one with the new pressure-based prediction rejection technique, and the *conventional* one. During the study participants inputted short English phrases with both techniques. We took phrases from a widely used corpus (MacKenzie and Soukoreff, 2003) that correlates very well with the English language character frequency. Sixty random phrases without uppercase, numeric, or special characters were selected for each technique, which users inputted in the same order during the two conditions. For each condition, the same phrases were used to ensure relatively similar prediction rate and accuracy for all users. To reject a prediction with the pressure-based technique participants had to apply extra pressure on the next key, while with the conventional technique they had to tap on the prediction bubble. The conditions were counterbalanced to avoid asymmetric skill transfer.

Users were instructed to hold the device in the portrait orientation with their dominant hand and then to type using the thumb of that hand. See Figure 8. The system displayed one phrase at a time and users had to transcribe that phrase. They were asked to take the time to read and understand the phrases in advance, then to enter them as fast and accurate as possible, and to press the *Return* key when they were finished to see the next one. No practice was given, but both methods were briefly demonstrated before the study. During this and for the pressure-based technique, we emphasized on how extra pressure could be applied on any key to bypass predictions, including *Space* and *Backspace*. We did this, as users showed uncertainty on this issue during a pilot.

Participants were informed that they could take a short break (max. 5 minutes) between conditions. Timing started from the entry of the first character and ended with the last. All key actions were performed on touch-up, similar to the default Apple iPhone keyboard. Hence, when users touched a wrong key, they could drag their finger to the right key before lifting it. They were asked to work normally, that is, to correct their errors as they noticed them. However, they had to exclusively use the *Backspace* key for editing, as we disabled direct cursor control to remove a potential confounding factor.

Both keyboards used the same method for word prediction, as discussed earlier. We verified that the frequency list contained all words used in the selected 120 phrases. Then we deliberately deleted 10% of the words from the list for each condition. This replicates the scenario where an incorrect prediction is provided and the user is forced to bypass it. This is not uncommon in predictive text entry, as users have to input non-dictionary words, such as abbreviations, names, alphanumeric text, and slang in real life. Some users also input text with the wrong prediction dictionary activated on occasion. MacKenzie et al., (2001) also highlighted the necessity for adequate handling of non-dictionary words in evaluations of predictive text entry. The 10% deleted words were selected randomly, subject to the restriction that they consist of at least three characters and do not appear more than once in the phrases. This guaranteed that an incorrect prediction would not be offered more than once, to prevent user adaptation.

We used the Words per Minute (WPM) metric to measure speed and the Total Error Rate (TER) metric for errors, which unifies the effect of accuracy during and after text entry (Soukoreff and MacKenzie, 2003; Arif and Stuerzlinger, 2009). We also recorded the sum of the mental preparation and physical movement time for each task. This was measured from the end of the previous task, i.e. touch-up, to the beginning of the next task, i.e. touch-down. We also recorded user actions on a prediction, including the rate at which predictions were accepted, rejected, and ignored. Finally, upon completion of the study users completed a questionnaire.

Design

A within-subjects design was used for the two factors: conventional and pressure-based techniques. In summary, the design was:

12 participants ×
2 conditions (*conventional*, *pressure*, counterbalanced) ×
60 phrased per condition
= 1440 phrases in total.

Results

After filtering outliers beyond three standard deviations from the mean (1.11% of the data) D'Agostino Kurtosis tests on the dependent variables confirmed that the data were normally distributed. In addition, a Mauchly's test confirmed that the data's covariance matrix was circular in form. Hence, repeated-measures ANOVA was used for all analysis. A Wilcoxon Signed-Rank test was used to analyse the nonparametric questionnaire data.

Entry Speed (WPM)

An ANOVA identified a significant effect of technique on entry speed ($F_{1,11}$=13.30, p<.005, η^2=.02). The average entry speeds for the conventional and the pressure-based techniques were 16.69 and 18.24 WPM, correspondingly. Figure 9**Error! Reference source not found.** illustrates this.

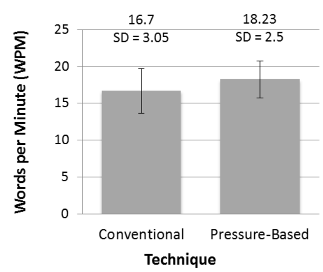

Figure 9. Average entry speed (WPM) for both techniques. Error bars represent ±1 standard deviation.

Error Rate (Total Error Rate)

An ANOVA identified a significant effect of technique on error rate ($F_{1,11}$= 11.99, p < .01, η^2 =.02). The average TER for the conventional and the new techniques was 9.31 and 7.02%, respectively. Figure 10 illustrates this**Error! Reference source not found.**.

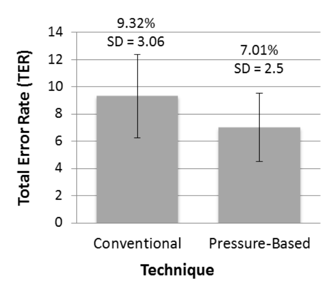

Figure 10. Average error rate (TER) for both techniques. Error bars represent ±1 standard deviation.

Corrective Operations (Backspace Use)

An ANOVA identified a significant effect of technique on corrective operations ($F_{1,11}$ = 6.81, p < .05, η^2 =.09). Average corrective operations for the conventional and new techniques were 8.31 and 6.49%, respectively. Figure 11 shows this. We considered only *Backspace*, as direct cursor control was disabled during the study.

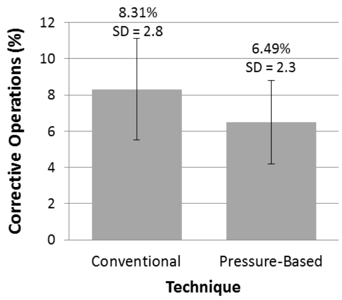

Figure 11. Average corrective operations (%) for both techniques. Error bars represent ±1 standard deviation.

Mental Preparation and Movement Time (Milliseconds)

An ANOVA on the data did not identify a significant effect of technique on the sum of mental preparation and physical movement time ($F_{1,11} = 3.65$, $p = .08$, $\eta^2 = .10$). The averages for the conventional and new techniques were 848.72 ms and 721.99 ms, respectively. Figure 12 illustrates this.

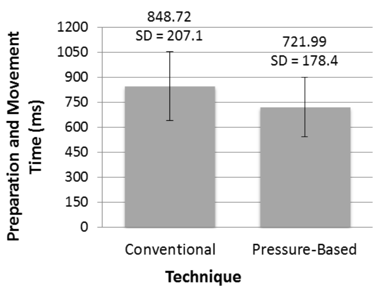

Figure 12. Average sum of mental preparation and physical movement time for both techniques. Error bars represent ±1 standard deviation

User Actions on Predictions (Accepted, Rejected, Ignored)

There was no significant effect of technique on accepted prediction rate ($F_{1,11} = 0.32$, ns). However, there was a significant effect on rejected prediction rate ($F_{1,11} = 6.48$, $p < .05$, $\eta^2 = .09$), and also on ignored prediction rate ($F_{1,11} = 5.93$, $p < .05$, $\eta^2 = .05$). Figure 13 illustrates the average user actions on predictions for both techniques.

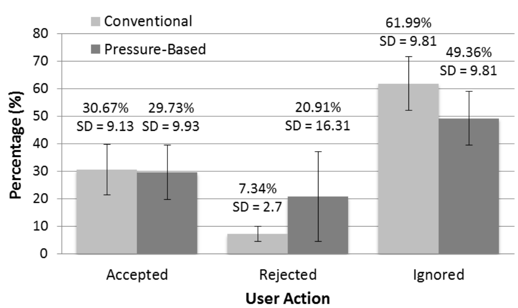

Figure 13. The average user actions on predictions (accepted, rejected, or ignored) for both techniques. Error bars represent ±1 standard deviation.

Hybrid Pressure Detection

The data from the pressure-based condition was further analysed to identify the rate at which the individual pressure detection simulation criteria were used by the hybrid method. Results showed that 58.59% of the time the hybrid technique detected extra pressure with the time-based approach, 30.80% with the touch-point-based approach, and the remaining 10.61% with both criteria simultaneously. Figure 14 illustrates this. A Friedman test found these three to be significantly different from one another ($\chi^2 = 17.92$, $p < .0005$, $df = 2$).

User Evaluation

Upon completion of the study participants responded to several questions on a seven-point Likert scale. We used a Wilcoxon Signed-Rank test to analyse the questionnaire data. The seven-point scales were later converted to three-point scales using linear transformation to calculate ratios (%). All ratings below four on the seven-point scale were mapped to one, all fours to twos, and all ratings above four to three. Some responses were converted to binomial data. Everything above four was rated as *accept* and below four as *reject* or vice versa, depending on the phrasing of the question. Ratings of four were disregarded. Such a mapping is common practice in statistics (Dawes, 2008).

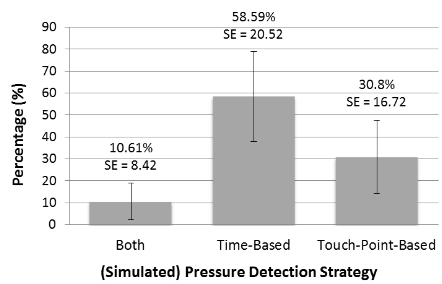

Figure 14. The average use of the extra pressure detection simulation criteria by the hybrid method. Error bars represent ±1 standard deviation.

Ease of Use

A Wilcoxon Signed-Rank test revealed that the two techniques differ significantly in their perceived ease of use ($z = -2.72$, $p < .05$). Average user ratings for the conventional and the new techniques were 3.08 and 5.75, respectively. See Figure 15. On average, 83% found the new technique easier to use than the conventional one. Also, most users (83%) responded that they did not feel any fatigue or discomfort while using the new technique.

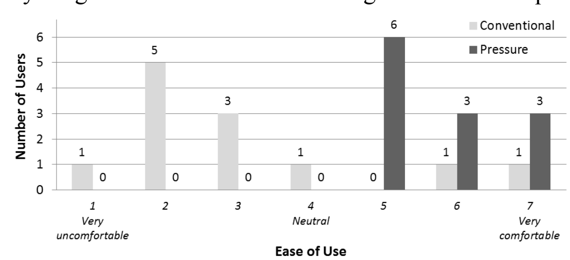

Figure 15. User feedback on how easy users found inputting text with the techniques, on a seven-point Likert scale.

Speed and Accuracy

A Wilcoxon Signed-Rank test identified significance with respect to user perceived entry speed ($z = -2.85$, $p < .005$) and accuracy ($z = -2.05$, $p < .05$). Average user ratings for the conventional and the new techniques were 3.75 and 5.25 for entry speed, and 4.17 and 5.25 for accuracy. Figure 16 illustrates this. 83% users found inputting text with the new technique faster and 58% found it more accurate compared to the conventional technique.

Pressure Detection Simulation

Participants were also asked to rate the accuracy of the pressure detection simulation technique. Results showed that 75% found our pressure detection approach accurate. See Figure 17. A Chi-squared test on the three-point scale derived from the original seven-point Likert scale found this to be significant ($X^2_{(2)}$=9.5, p<.01).

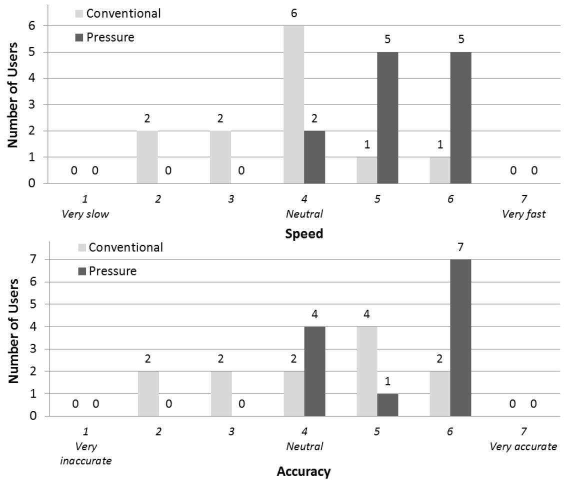

Figure 16. User feedback on how fast (above) and how accurate (below) they thought their text entry was with the two techniques on seven-point Likert scales.

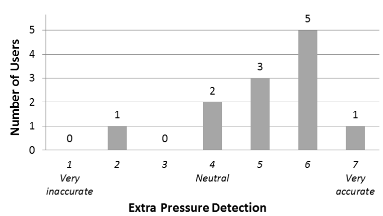

Figure 17. User feedback on how accurate they thought the pressure detection simulation was during the pressure-based condition on a seven-point Likert scale.

Overall Rating

A Wilcoxon Signed-Rank test identified significance with respect to participants' overall rating of the two techniques (z = -2.27, p < .05). Average ratings for the conventional and the new techniques were 3.75 and 5.50, respectively. See Figure 18. Results showed that most users (83%) favoured the new technique over the conventional one.

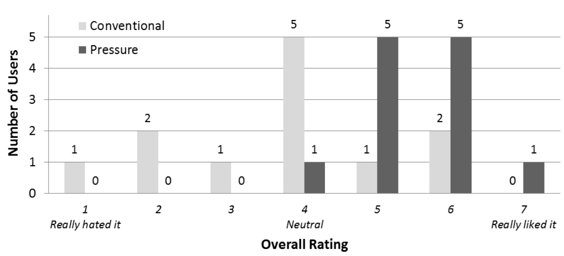

Figure 18. User feedback on how much users liked the examined techniques on a seven-point Likert scale.

Discussion

Results show that the new technique improves overall text entry performance significantly, both in terms of speed and accuracy. On average, entry speed increased by 9% and error rate decreased by 25% with the new pressure-based technique compared to the conventional technique. The 22% decrease in the corrective operations also provides indirect evidence in that users had to fix fewer mistakes with the new technique. No significant effect of technique was identified on the accepted prediction rate. This is not unexpected as both techniques allow users to accept predictions by the same method – by tapping on the *Space* key.

However, there was a significant effect of technique on the percentage of rejected and ignored predictions. This means users rejected (and ignored) more predictions with the new technique. As far as we can tell, most of these rejected predictions were instances where the prediction was not the desired word. We also did not observe a significant effect of technique on the sum of the mental preparation and physical movement times. We see this as corroborating evidence that this factor did not contribute significantly to the observed differences. Therefore, we believe that the main difference was that users accepted fewer predicted words incorrectly with the new technique. This reduced the overall error percentage and error fixing time. The decrease in corrective operations also supports this observation.

Most users found text entry with the new technique easier than with the conventional one. Most also thought that their entry speed was higher and a majority believed to make fewer errors with the new technique. This means that the new technique is perceived as "faster" and "more accurate". Therefore, it is not surprising that most users favoured the new technique over the conventional one.

Hybrid Pressure Detection Simulation

The results for the rate at which the hybrid technique relied on the pressure detection simulation criteria here support the observation from the first study that there are three distinct behaviours. We found again that a single criterion is not adequate for all users. This again highlights the utility of the hybrid pressure detection approach. The percentage of detections of extra pressure via the touch-point-based approach was larger (8% vs. 31% in the last study). Thus, we can state that in text entry almost one third of all extra pressure taps are best detected through a touch-point-based approach. Besides, user feedback data revealed that most users found the new technique to be "accurate". The hybrid method also sped up text entry significantly. The average tap times for regular and extra pressure were 117 and 390 ms, faster than Quick-release's 200 and 400 ms and Dwell's 1400 and 1700 ms (Brewster and Hughes, 2009), respectively.

CONCLUSION AND FUTURE WORK

This article presented a new pressure detection technique that combines the existing time- and touch-point-based approaches to detect pressure on standard touchscreens. Results of two independent user studies showed that the new hybrid technique distinguishes reliably between (at least) two pressure levels: regular with about 1 N, and extra with about 3 N. We then presented a new pressure-based predictive text entry technique that used the new pressure detection approach to enable users to bypass incorrect predictions by applying more pressure on the next key. Results of a study showed that when inputting short English phrases containing 10% non-dictionary words, the new technique increased entry speed by 9% and reduced errors by 25% compared to the conventional technique. Besides, user feedback data showed that most users (75%) found the hybrid pressure detection technique accurate and most (83%) favour the pressure-based predictive text entry technique.

We believe that the new hybrid method can detect more than two pressure levels and intend to investigate this in the future. We also plan on evaluating our technique in landscape mode with two-hand text entry, and in mobile settings, such as when walking or on a train.

REFERENCES

Arif, A. S. A survey on mobile text entry handedness: how do users input text on handheld devices while nomadic? IHCI '12, IEEE (2012), 266-271.

Arif, A. S. and Stuerzlinger, W. Analysis of text entry performance metrics. TIC-STH '09, IEEE (2009), 100-105.

Benko, H., Wilson, A. D., and Baudisch, P. Precise selection techniques for multi-touch screens. CHI '06, ACM (2006), 1263-1272.

Boring, S., Ledo, D., Chen, X. A., Marquardt, N., Tang, A., and Greenberg, S. The fat thumb: using the thumb's contact size for single-handed mobile interaction. MobileHCI '12, ACM (2012), 39-48.

Brewster, S. A., Hughes, M. Pressure-based text entry for mobile devices. Mobile HCI '2009, ACM (2009), 1-4.

Buxton, B. Multi-touch systems I have known and loved. Microsoft Research. 2012. Feb. 21, 2013. http://www.billbuxton.com/multitouchOverview.html

Buxton, W., Hill, R., and Rowley, P. Issues and techniques in touch-sensitive tablet input. SIGGRAPH '85, ACM (1985), 215-224.

Cechanowicz, J., Irani, P., and Subramanian, S. Augmenting the mouse with pressure sensitive input. CHI '07, ACM (2007), 1385-1394.

Davies, M. Word frequency data from the Corpus of Contemporary American English (COCA). 2011. Dec. 24, 2011. http://www.wordfrequency.info

Dawes, J. Do data characteristics change according to the number of scale points used ? An experiment using 5-point, 7-point and 10-point scales. International Journal of Market Research 50, 1, 2008.

Dietz, P. H., Eidelson, B. Westhues, J., and Bathiche, S. A practical pressure sensitive computer keyboard. UIST '09. ACM (2009), 55-58.

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

Forlines, C., Shen, C., and Buxton, B. Glimpse: a novel input model for multi-level devices. Ext. Abstracts CHI '05, ACM (2005),1375-1378.

Graham-Rowe, D. Mobile touch screens could soon feel the pressure. MIT Technology Review, (2010).

Heo, S. and Lee, G. Forcetap: extending the input vocabulary of mobile touch screens by adding tap gestures. MobileHCI '11. ACM (2011), 113-122.

Herot, C. and Weinzapfel, G. One-point touch input of vector information for computer displays. SIGGRAPH '78, ACM (1978), 210-216.

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. 2013. Mar. 8, 2013. http://shar.es/jNIP8

Hwang, S. and Wohn, K-Y. PseudoButton: enabling pressure-sensitive interaction by repurposing microphone on mobile device. Ext. Abstracts CHI '12. ACM (2012), 1565-1570.

Jong, S. d., Kirkali, D., Schraffenberger, H., Jillissen, J., Rooij, A. d., and Terpstra, A. One-press control: a tactile input method for pressure-sensitive computer keyboards. Ext. Abstracts CHI '10, ACM (2010), 4261-4266.

Levenshtein, V. I. Binary codes capable of correcting deletions, insertions and reversals. Soviet Physics-Doklady 10, (1966), 707-710.

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

MacKenzie, I. S., Kober, H., Smith, D., Jones, T., and Skepner, E. LetterWise: prefix-based disambiguation for mobile text input. UIST '01, ACM (2001), 111-120.

McCallum, D. C., Mak, E. Irani, P., and Subramanian, S. PressureText: pressure input for mobile phone text entry. Ext. Abstracts CHI '09, ACM (2009), 4519-4524.

Mizobuchi, S., Terasaki, S., Keski-Jaskari, T., Nousiainen, J., Ryynanen, M., and Silfverberg, M. Making an impression: force-controlled pen input for handheld devices. Ext. Abstracts CHI '05, ACM (2005), 1661-1664.

Nurmi, M. User Interface. US Patent 0256807. Oct. 15, 2009.

Parhi, P., Karlson, A. K., and Bederson, B. B. Target size study for one-handed thumb use on small touchscreen devices. MobileHCI '06. ACM (2006), 203-210.

Raisamo, R. Evaluating different touched-based interaction techniques in a public information kiosk. OzCHI '99, ACM (1999), 169-171.

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

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.

Srinivasan, M. and Chen, J. Human performance in controlling normal forces of contact with rigid objects. ASME Winter Annual Meeting, 49 (1993). 119-125.

Synaptics ForcePad™: a New Dimension in Capacitive Input: Pressurepressure. http://www.synaptics.com/solutions/products/forcepad

Tang, H., Beebe, D. J., and Kramer, A. F. A Multilevel input system with force-sensitive elements. Int. J. Hum.-Comput. Stud. 54, 4 (2001), 495-507.

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.

Watanabe, Y., Makino, Y., Sato, K., and Maeno, T. Contact force and finger angles estimation for touch panel by detecting transmitted light on fingernail. EuroHaptics '12, Springer-Verlag (2012), 601-612.

Zeleznik, R. Miller, T., and Forsberg, A. Pop through mouse button interactions. UIST '01, ACM (2001), 195-196.