Pressure-Based Text Entry for Mobile Devices

Stephen A. Brewster and Michael Hughes
Glasgow Interactive Systems Group, Department of Computing Science
University of Glasgow, Glasgow, G12 8QQ, UK
+44 141 330 4966

stephen@dcs.gla.ac.uk www.dcs.gla.ac.uk/~stephen

ABSTRACT

This paper describes the design and evaluation of a touchscreen-based pressure keyboard to investigate the possibilities of pressure as a new method of input for mobile devices. A soft press on the touchscreen generated a lowercase letter, a hard press an uppercase one. The aim was to improve input performance when entering mixed-case text, or shifted characters often used for emoticons, etc. An experiment compared two different forms of pressure input (Dwell and Quick Release) against a standard shift key keyboard, with users both sitting and walking. Results showed that Quick Release was the fastest for input of mixed case text with Dwell being the most accurate, even when users were mobile. The results demonstrate that pressure input can outperform a standard shift-key keyboard design for mobile text entry.

Categories and Subject Descriptors

H5.2. [**User Interfaces**] *Haptic I/O.*

General Terms

Human Factors.

Keywords

Pressure input, keyboard, text entry, multimodal interaction.

1. INTRODUCTION

Many new mobile devices now use touchscreens rather than physical keyboards (the Apple iPhone is a current example). These add display flexibility (there is no need for a physical keyboard so the screen can be made bigger) at the expense of text entry (the lack of tactile feedback reduces typing performance [3]). One problem comes in targeting small buttons and widgets on a touchscreen. Hitting a small button with a large finger is error prone, especially when there is no tactile feedback. This gets worse for uppercase or punctuation characters where the shift key and a character key must be pressed, giving two chances of error. The aim of our work is to facilitate the entry of shifted characters by reducing the number of keys the user has to target and press from two to one, potentially reducing errors and time.

The use of shifted characters is far less common than lowercase ones, but people do still need them. In some cases there are software solutions. For example, the Apple iPhone capitalises letters after full stops and some proper nouns that it recognises. This saves the user having to press the shift key. However, if users want to type whole words in capitals they may need to press the shift key for each one (the iPhone does have a setting to allow Caps Lock by hitting the Shift key twice). If users want to type emoticons (such as :-) on many phones they have to use the shift key for the punctuation characters. The use of these characters is less than for lowercase letters, but there is still a need for them to be entered. Our aim here is to see if we can use pressure input to improve text entry performance and make any of the shifted characters easier to use.

2. BACKGROUND

Humans have very precise control over pressure, especially at the fingertips. It is key in tasks such as picking up objects, drawing or playing a musical instrument. Pressure for interaction has been studied in the context of graphics tablets [9] and mice [2] but not for touchscreen mobile devices where there could be many benefits from a richer form of dynamic input.

Research into touch (skin-based) interaction for mobile devices has grown over recent years due to the limitations of screen size and the fact that audio is not always appropriate for output. There are several different sub-modalities within touch. Tactile feedback (commonly via vibrotactile stimulation of the skin) is the best understood and used in HCI but pressure is part of the same sensory-motor system and could be used for mobile input.

Srinivasan and Chen [11] studied force using the index finger. Participants had to control the force applied to a sensor under a range of different conditions (including an anesthetized fingertip to examine the effect of removing tactile feedback). They suggest that pressure interfaces need to have a force resolution of at least 0.01N to make full use of human capabilities. Mizobuchi *et al.* [8] suggest that ranges of 0-3N are comfortable and controllable and users can reliably apply around 5-6 levels of pressure [8, 9].

Studies of touchscreen text entry have shown the difficulties of typing using the QWERTY keyboard layout and have proposed different layouts to combat this [5, 10]. However, QWERTY is still the standard for Roman characters. Hoggan *et al.* showed that text entry is poorer on touchscreens than physical keyboards [3], partly due to the missing tactile feedback from the keys. They added tactile feedback using the phone's vibration motor, with text entry performance rising close to the level of real buttons. An alternative would be to look at how key presses could be minimized by novel keyboard designs. We propose the use of different levels of pressure to select different characters. For example, a harder press might select the shifted version of a key, with a hard press on 'a' selecting 'A'. This reduces the number of keystrokes needed, and thus potentially the errors and time incurred as the user no longer has to move to the shift key and back to select a shifted character. Our design does not require a change to the standard QWERTY layout, as it can be hard to persuade users to change from the layout they know, even if performance is demonstrably better.

Research has shown that pressure can be a useful addition to interactions. Ramos *et al.* have done some of the key work in HCI on pressure input using a graphics tablets and a stylus. They

MobileHCI'09, September 15 - 18, 2009, Bonn, Germany.
ACM 978-1-60558-281-8.

looked at how pressure might be used in applications such as video editing and proposed a set of 'pressure widgets' [9] for tasks such as zooming and selection based on pressure. However, they did not develop a widget for text entry or test finger-based pressure input. Tang *et al.* [12] developed a chord keyboard that used pressure for text input. It used three keys which could resolve 3 pressure levels, giving a vocabulary of 27 characters. Their work showed pressure was possible for text input, but their error rate was high (18% after three sets of trials). We used this work as inspiration for our own, but simplified ours to use a standard QWERTY layout and only two levels of pressure. Holleis *et al.* [4] added touch sensors to a mobile phone pad to allow users to preview content by touching a key (in effect giving an extra pressure level). Their qualitative study showed people generally liked the touch feature. Irani *et al.* [2] added pressure to a mouse for desktop interactions. Their results showed that users were slower when they had to press harder, and that a click selection technique was faster than a dwell, although dwell was the most accurate. We built on these techniques, using them for text entry on a mobile device.

All of these previous studies were done in static situations. The movements of walking or the bumping of a train may have a serious impact on the amount of force people can consistently apply and may reduce the number of usable pressure levels for real-world mobile interactions. The aim of the study presented here is to investigate how pressure might be used for text entry and if it is still usable when on the move.

3. EXPERIMENT

To test the usefulness of pressure as a mobile input technique we designed a pressure-based keyboard where a soft press generated a lowercase letter and a harder press an uppercase one. This removed the need for a movement to the shift key to change case, potentially reducing targeting errors and time when typing. We used the Nokia N800 Internet Tablet (europe.nokia.com /phones/n800), a small touchscreen device normally operated with a finger or stylus. It is possible to read the pressure values generated from its resistive touchscreen, meaning that we can use pressure for input without making any modifications to the standard device (pressure is not used in the normal interaction with this device). The level of force can be read in device specific values (0-255), where 0 is no contact, 255 very light contact and 1 maximum force.

We ran a pilot study to assess the levels of pressure that users could apply with this device. Pressure levels used in other studies were taken as a starting point, but they had been based on stylus or mouse use and we were using fingertip. Twelve participants made 24 selections of buttons at 3 different pressure levels. The results showed that selections could be made effectively with values of >=65 for a hard press and values of <65 for a soft press.

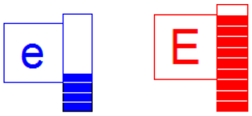

Figure 1: The graphical pressure meter.

3.1 Pressure Interaction Design

We implemented the two best pressure techniques from Ramos *et al.* [9]: Dwell and Quick Release. A press with a value >=65 generated an uppercase letter. With the Dwell technique the user had to apply force for 0.5s before a selection was made. Audio feedback was given when pressure had been applied for the appropriate duration. For Quick Release the user pressed a key with the appropriate pressure and released immediately. In this case, a different sound was played to confirm whether an upper or lowercase letter had been typed. A dynamic graphical representation of the current pressure level (and the case of the letter that would be chosen) was given with a pressure meter that popped-up beside the key being pressed (Figure 1).

3.2 Design and Procedure

The Independent Variables were: keyboard type (3 levels: Standard, Dwell and Quick release) and mobility (2 levels: sitting and walking), leading to six experimental conditions. We used a within-subjects design, with all participants using all of the keyboard types walking and sitting in a counterbalanced order. We logged all keyboard activity, measuring input times and error rates. NASA TLX workload ratings were taken after each condition. We used 12 new participants, all students from the University. Nine were male and three female. All were familiar with text entry on phones, but novices with the N800 and pressure input.

The keyboard layout we used can be seen in Figure 2 (in the Standard keyboard condition shift keys were added and pressure values from the screen were ignored). Keys were 6mm^2, with a 1mm gap (similar to the standard N800 keyboard). Participants typed 6 randomly chosen phrases from MacKenzie and Soukoreff's phrase set [7] in each condition. The phrase set does not contain uppercase letters, so we capitalized the first letter of each word (this was not ideal as it is does not represent the normal distribution of uppercase letters but was the simplest way to generate enough mixed case text for testing purposes. Future tests will use a more standard distribution of upper and lowercase characters). The phrase to be typed was shown at the top of the window, with the user's text beneath it. When the phrase had been typed the user pressed 'Enter' to move to the next one. The phrase had to be correct before the user could move on. Users entered six phrases giving around 170 key presses in each condition, keeping the experiment to 35 minutes.

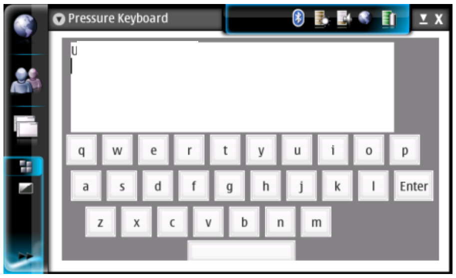

Figure 2: Screenshot of the pressure keyboard.

Participants walked standard 'figure-of-eight' loops around obstacles in the mobile condition [1] to simulate the movements caused by use of mobile devices on the move. The experiment was conducted on an empty floor of an office building. In the sitting condition users held the device in hand and did not rest it on a surface. Our hypotheses were:

> *H1:* Average time to make selections would be fastest with Quick Release. Ramos *et al.* showed this to be a fast technique in their work on graphics tablets.
>
> *H2:* Error rates would be lowest with Dwell. Again, Ramos *et al.* [9] showed this technique to have a low error rate in their experiments.

H3: Time and errors would be higher when mobile. The movements of walking would induce more errors as the device would be harder to keep stable and for pressure to be applied consistently.

H4: Subjective workload of the pressure conditions would be higher than Standard. Participants are more used to standard shift key interactions.

3.3 Timing Results

An overall two-factor ANOVA was used to compare mean times per condition for each keyboard type when sitting and walking (Figure 3). Results showed there was a significant main effect for keyboard type ($F_{2,66}=80.737$, $p<0.001$), but not for mobility ($F_{1,66}=1.779$, $p=0.186$) and no interaction ($F_{2,66}=1.307$, $p=0.277$). *Post hoc* Tukey HSD tests showed that Dwell was slower than Quick and Standard ($p<0.05$) and Standard was slower than Quick ($p<0.05$). The results for keyboard type confirmed H1, but H3 was not accepted as there was no effect of mobility on text entry speed.

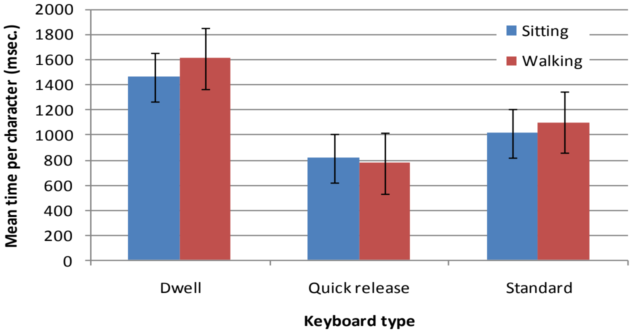

Figure 3: Mean times to enter a character (Standard Error bars are shown on all graphs).

The Words per Minute (WPM) values for each keyboard were: Dwell (sitting: 8.2, walking 7.4), Quick (14.7, 15.5) and Standard (11.8, 10.9). The slow performance of Dwell is no surprise as participants had to stay on a key for 0.5s to make a selection, and matches the results of Ramos [9] and Irani [2]. The results show that the Quick release keyboard was faster than the Standard one, meaning that if speed is important then pressure is one way to achieve it. The results suggest that mobility did not affect the time to make input in any condition, showing that input speed with pressure is robust to the movements caused by walking.

An important issue is the time taken to enter uppercase characters, as no movement to the shift key is needed with the pressure keyboards. A two-factor ANOVA compared the mean times to enter upper and lowercase characters for each keyboard type (Figure 4). As before, there was a significant main effect for keyboard type ($F_{2,66}=77.28$, $p<0.001$), also for letter case ($F_{1,66}=353.8$, $p<0.001$) and a significant interaction between them ($F_{2,66}=61.22$, $p<0.001$). A Tukey test showed uppercase letters took significantly longer to enter than lowercase ones ($p<0.05$). This is particularly obvious in the Standard condition, where the difference is most extreme due to the movements to the Shift key. The interaction occurred because for the Dwell keyboard the difference between upper and lowercase times was small, especially when compared to the Standard keyboard.

These results suggest that Quick release is effective when upper or mixed case text must be entered quickly, but text entry rate on lowercase letters alone was lower than the Standard keyboard. The mean time to enter an uppercase letter with the Quick keyboard was 1.17s and 1.88s with Standard, for lowercase it was 0.4s and 0.2s respectively. The benefit of Quick release comes from eliminating the need to move to the shift key, but at a time cost of pressing accurately at the lower pressure level.

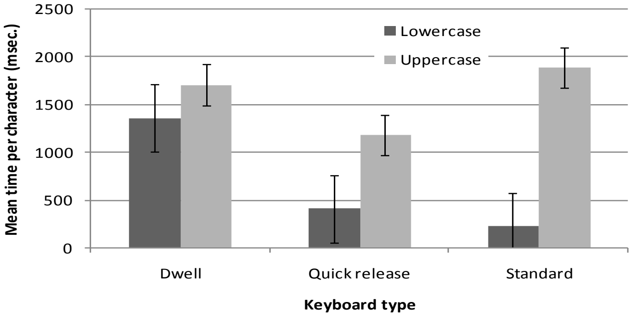

Figure 4: Upper and lowercase time results for each keyboard.

3.4 Error Results

An overall two-factor ANOVA compared the error rates (number of errors/total number of characters entered) of the different keyboards when sitting and walking (Figure 5). Results showed there was a significant main effect for keyboard type ($F_{2,66}=24.39$, $p<0.001$), for mobility ($F_{1,66}= 7.751$, $p=0.006$) but no interaction. Tukey tests showed that Dwell had a significantly lower error rate than Quick and Standard ($p<0.05$), with no difference between Quick and Standard. The error rate when walking was significantly higher than when sitting ($p<0.05$). The results for keyboard type confirmed H2 and partially H3, as error rates were higher when mobile.

The results again confirm those found by Ramos *et al.*, but show that Dwell is also effective in mobile settings, reducing errors from a mean of 5.9 for Quick, 4.8 for Standard to 2.8 for Dwell. The movement of the device and user did not make it harder to apply the appropriate level of force and suggests that if accurate text entry is needed Dwell is the best technique to use.

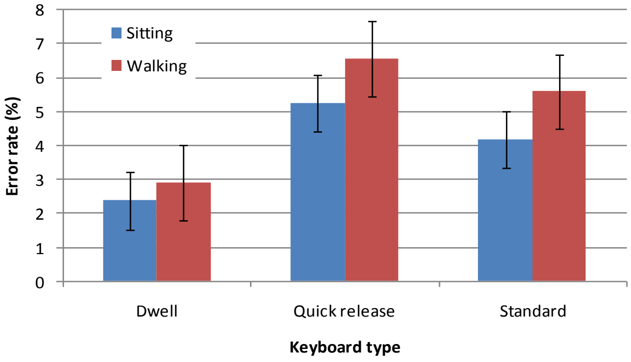

Figure 5: Error rates for the different keyboard types.

We also wanted to know if the error rate of uppercase characters (number of uppercase errors/total number of uppercase characters) was different to lowercase. A two-factor ANOVA compared error rates on upper and lowercase characters with the different keyboard types (Figure 6). As before, there was a significant main effect for keyboard type ($F_{2,66}=15.73$, $p<0.001$), also for letter case ($F_{1,66}= 34.78$, $p<0.001$) and an interaction between them ($F_{2,66}=3.03$, $p=0.05$). Uppercase characters had a significantly lower error rate than lowercase ($p<0.05$). The interaction occurred as the difference in error rate was much smaller for Dwell than for the other two conditions.

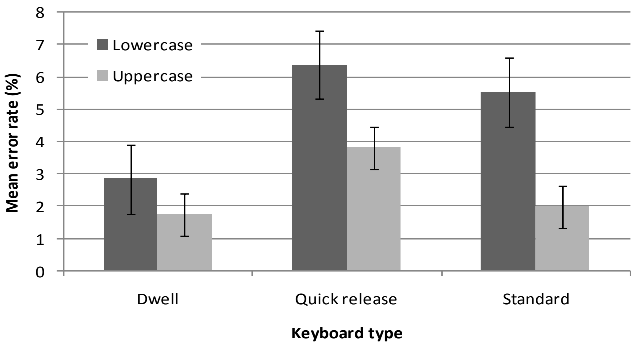

Figure 6: Error rates for upper and lowercase characters.

We had expected there would be more errors with uppercase letters (more pressure had to be applied, or a shift key pressed), but this was not the case. One reason could be that users were more careful when they entered them. The results for Standard, in particular, show that users spent much more time on the uppercase letters, so in this case trading accuracy for time. A longer term study would be needed to see if this behaviour continued in real world use.

3.5 Subjective Workload

An overall two-factor ANOVA compared mean NASA TLX Overall Workload scores for the keyboard types when sitting and walking. Results showed a significant main effect for keyboard type ($F_{2,66}$=7.447, p=0.001), for mobility ($F_{1,66}$= 23.27, p<0.001) but no interaction. Tukey tests showed that Standard had a significantly lower workload than Quick or Dwell (p<0.05), and sitting a lower workload than walking (p<0.05) confirming H4.

The detailed results showed that Quick had a significantly lower perceived Performance level (p<0.05) and higher frustration level (p<0.05) than the other two conditions, probably due to its error rate; its higher text entry rate did not overcome the frustration caused by the high error rate when the user was forced to enter the correct text before moving to the next phrase.

4. DISCUSSION AND CONCLUSIONS

The results showed that pressure-based text entry was effective using a fingertip on a mobile touchscreen device. The Quick Release keyboard was faster and the Dwell keyboard caused fewer errors than a standard one with a shift key. The main speed benefit for Quick release came with upper or mixed case text, for lowercase the standard keyboard was still faster. Dwell was particularly effective when users were mobile as the error rate only increased slightly over use when sitting. Pressure performance remained robust when users were mobile. We had anticipated that the movements of walking would make it harder to apply pressure consistently, with the device and user both moving. This turned out not to be the case, suggesting that pressure could be a useful method of interaction for mobile users.

Putting our results in context, MacKenzie and Zhang [6] found error rates of ~4% for novices using a stylus on a soft keyboard with 6.4 mm keys, close to our Standard/sitting keyboard condition. The overall WPM on our keyboards was quite low. MacKenzie and Zhang found rates of 19 WPM, compared to our 11.8. A direct comparison is tricky as the differences between finger and stylus input are not clear. However, one reason could be our device as drawing the pressure meter was quite slow due to its graphics capabilities (animated cursors were used to display the popup pressure meter) and the complex programming needed to get pressure values in real time. We have redesigned the software to go faster so these practical problems can be removed in further research.

If speed and error rates can be optimized we may be able to create a pressure keyboard that would be better in both regards than a standard one. One potential way to do this would be to reduce the dwell time, perhaps to 0.25s or less. This might retain the performance of Dwell but get closer to the speed of Quick release. Investigating the trade-off between dwell time and accuracy will be the focus of our next study on pressure based input.

5. ACKNOWLEDGMENTS

This work was supported by the EPSRC-funded GAIME project (EP/F023405).

6. REFERENCES

[1] Brewster, S.A., Lumsden, J., Bell, M., Hall, M. and Tasker, S. *Multimodal 'Eyes-Free' Interaction Techniques for Wearable Devices*. In Proceedings of ACM CHI 2003, ACM Press, Addison-Wesley, 463-480.

[2] Cechanowicz, J., Irani, P. and Subramanian, S. *Augmenting the Mouse with Pressure Sensitive Input.* In Proceedings of ACM CHI 2007, ACM Press, Addison Wesley, 1385-1394.

[3] Hoggan, E., Brewster, S.A. and Johnston, J. *Investigating the Effectiveness of Tactile Feedback for Mobile Touchscreens*. In Proceedings of ACM CHI2008, ACM Press Addison Wesley, 1573-1582.

[4] Holleis, P., Huhtala, J. and Häkkilä, J. *Studying applications for touch-enabled mobile phone keypads.* In Proceedings of TEI 2008, ACM Press,15-18.

[5] Mackenzie, I.S., Zhang, S. and Soukoreff, R.W. Text Entry Using Soft Keyboards. *Behaviour and Information Technology*, 1999, 18 (4), 235 - 244.

[6] MacKenzie, I.S. and Zhang, S.X. An empirical investigation of the novice experience with soft keyboards. *Behaviour and Information Technology*, 2001, 20 (6), 411-418.

[7] MacKenzie, S. and Soukoreff, W. *Phrase Sets for Evaluating Text Entry Techniques*. In Extended Abstracts of ACM CHI 2003, ACM Press, 754-755.

[8] Mizobuchi, S., Terasaki, S., Keski-Jaskari, T., Nousiainen, J., Ryynanen, M. and Silfverberg, M. *Making an impression: force-controlled pen input for handheld devices.* In Vol II Proceedings of ACM CHI 2005, ACM Press, 1661-1664.

[9] Ramos, G., Boulos, M. and Balakrishnan, R. *Pressure Widgets.* In Proceedings of ACM CHI 2004, ACM Press, Addison-Wesley, 487-494.

[10] Sears, A., Revis, D., Swatski, J., Crittenden, R. and Shneiderman, B. Investigating Touchscreen Typing: The Effect of Keyboard Size on Typing Speed. *Behaviour and Information Technology*, 1993, 12 (1), 17-22.

[11] Srinivasan, M. and Chen, J. *Human performance in controlling normal forces of contact with rigid objects*. Winter Annual Meeting of the ASME, 1993, 49. 119-125.

[12] Tang, H., Beebe, D. and Kramer, A. A multilevel input system with force-sensitive elements. *International Journal of Human-Computer Studies*, 2001, (52), 495-507.