Pressure Text Entry for Mobile Devices

David McCallum, Ed Mak, Pourang Irani, Sriram Subramanian

Department of Computer Science

University of Manitoba

University of Manitoba

Introduction

Current mobile devices assign multiple characters to each key, and multiple presses are required to select a desired letter. PressureText allows users to select a letter with a single press, by mapping each letter to a different amount of force.

MultiTap

- Lets users cycle through the letters on each key by pressing it multiple times
- Is commonly seen on mobile devices, especially cell phones
- Allows typing with fewer buttons, but the speed is significantly slower than a standard keyboard layout

Figure 1, MultiTap Keypad Layout

The Segmentation Problem

- It's difficult to type a second character on the same key
- MultiTap thinks the user is scanning to the next character
- The first character is typed after a 1.5 second delay (Nokia phone)
- Users can alternatively hit the 'next' key to instantly type the character, if available
- This slows the user's typing rate

Challenges with Pressure

- Pressure-based selection has error rates as high as 10 to 30 percent [1, 3]
- Users require at least one second to make a selection
- The small display area on mobile devices allows for only limited visual feedback

The PressureText Device

- Has 12 force sensitive buttons
- Buttons can detect 1024 distinct levels of force
- Detects forces between 0 and 1.5 N
- Push lightly to get the first letter on a key, and harder to get each successive letter
- Release slowly to move to the previous letter
- Release quickly to type the current letter

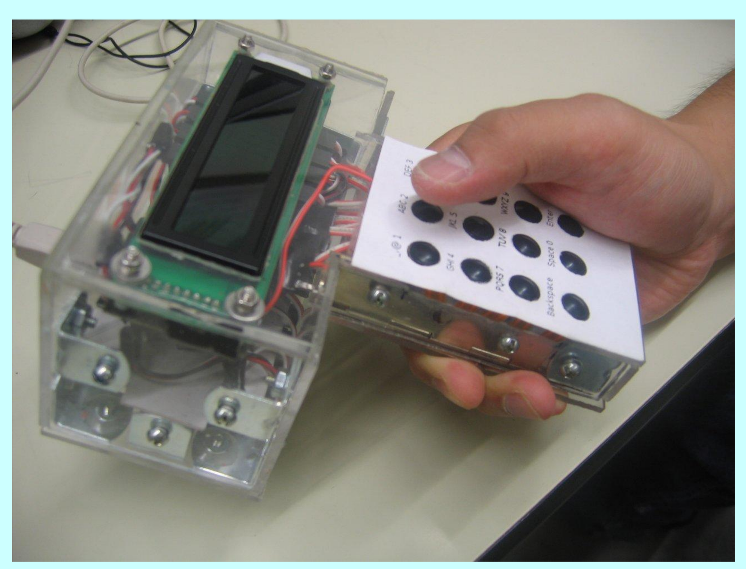

Figure 2, The PressureText Device

PQRS and WXYZ

- These buttons have four characters each
- In practice, users found it difficult to select the correct letter with four options
- We reduced the buttons to sense at most three distinct pressure inputs
- The first three characters are selected as normal
- To get to the last character, push hard to get to the third character (R or Y), then dwell there for 750 ms

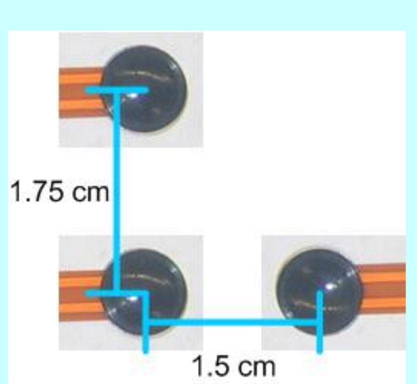

Figure 3, Keypad Button Spacing

Figure 4, PressureText Device With Typing

Debouncing

- Pressure values from each button are discretized into one level per character
- Debouncing prevents oscillation between two adjacent levels
- Expand the currently selected level by 10% by adding thresholds to each side
- To pass from one level to the next, the pressure must cross the currently selected level's threshold
- The effect is similar to the "Fisheye" function, which has been shown to outperform other discretization techniques and reduce error rates [1]

- Figure 5 shows debouncing in action
- The black triangle is the current amount of pressure detected by the button
- The dark blue lines show the thresholds
- The shaded area is the currently selected level and its threshold

1) Level 1 is selected. The pressure has passed into level 2, but remains in Level 1's threshold
2) Pressure passes beyond the threshold, Level 2 is selected
3) Pressure drops to Level 1, but remains in Level 2's threshold
4) Pressure drops below the threshold, and Level 1 is selected

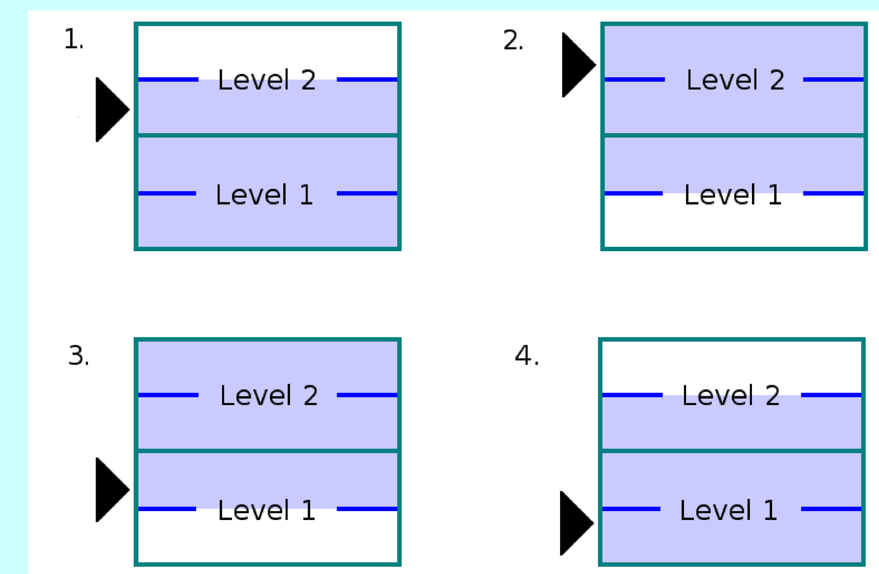

Figure 5, Debouncing Mechanism

The Experiment

Goal: Investigate the effectiveness of the pressure-based text entry system compared to MultiTap. Measures of interest are words per minute and error rate.

Subjects: Nine paid volunteers, three female and six male

- Five had experience with MultiTap, two used T9, and the remaining two did not have significant texting experience
- Three users were considered experts, as they had previously used the system

Design

- Used the Unconstrained Text-Entry Evaluation Paradigm (UTEEP) [2]
- Participants entered phrases that appeared on a secondary screen
- Used blocks of ten phrases each
- The backspace key was available to correct errors
- Primary instruction was to proceed quickly and accurately
- Used a within-subjects design, participants alternated between MultiTap and PressureText techniques over a period of three days

Words Per Minute

- Did not find a significant difference in speed based on technique (MultiTap vs PressureText)
- However, there was a significant speed increase for the expert users
- Performance improved with the number of blocks completed

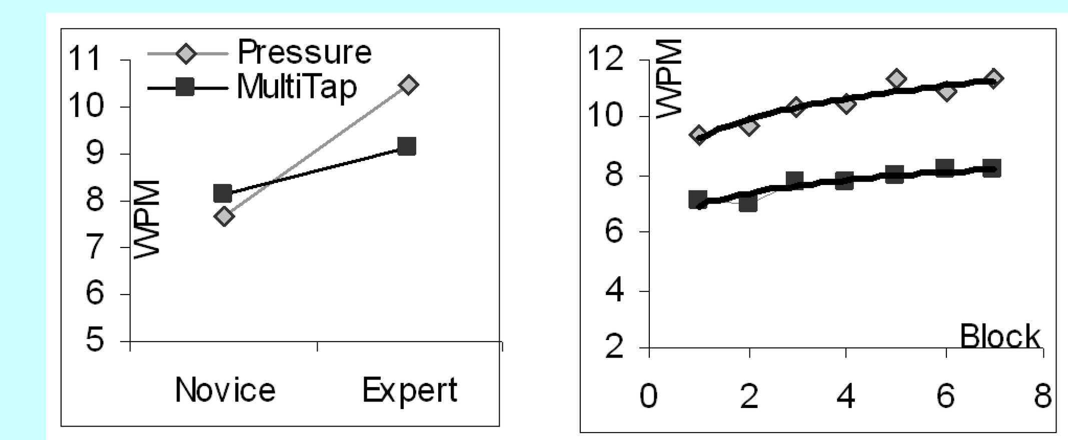

Figure 6, Words Per Minute

Error Rate

- Computed the average corrected error rate, where the user types an incorrect letter then fixes it, as recommended by [2]
- Characters in the middle of the discretization (for example B, E, H) had the highest number of errors in PressureText
- Average error rate for PressureText was 8.6%, compared to 2.7% in MultiTap
- However, there was a greater decrease in error rates for PressureText over multiple blocks
- The error rate reaches a plateau in MultiTap after block five, but performance with PressureText continues to improve with experience

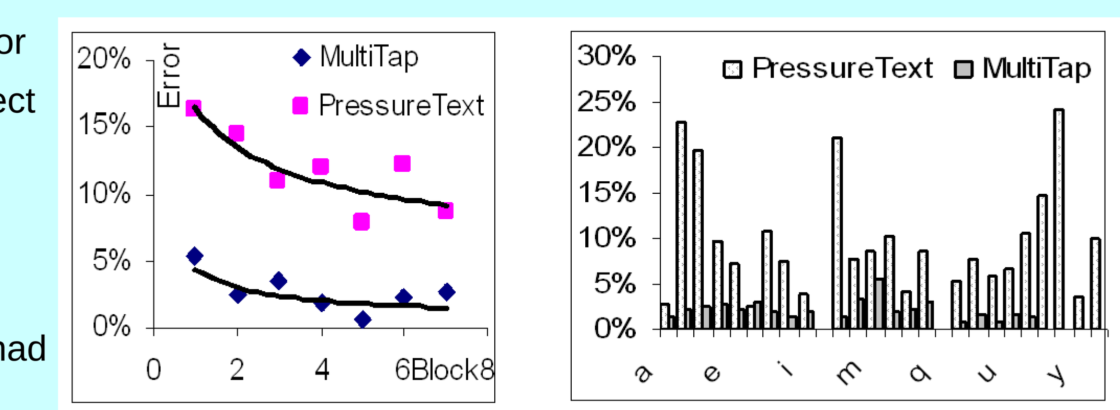

Figure 7, Error Rates

Discussion

- Pressure can assist in keypad text entry, but requires a high level of expertise to become beneficial
- We believe the error rate is primarily responsible for slowing user speed
- There was no tactile or auditory feedback to indicate the selected character
- Limited visual feedback was present (selected letter appears on the device display)
- Future work includes tweaking the discretization and adding vibrotactile feed back
- Pressure could also be used with T9 to disambiguate words

References

[1] Shi, K., Irani, P., Gustafson, S., Subramanian, S. (2008) PressureFish: a method to improve control of discrete pressure-based input. *Proc. CHI'08*, 1295-1298.

[2] MacKenzie, I. S., and Tanaka-Ishii, K. (Eds.) Text entry systems: Mobility, accessibility, universality, pp. 332. San Francisco.

[3] Ramos, G., Boulos, M., and Balakrishnan, R., Pressure widgets. *Proc. CHI 2004*, 487-494.