Mobile Brailler: Making Touch-Screen Typing Accessible to Visually Impaired Users

Nektarios Paisios* Alexander Rubinsteyn Lakshminarayanan Subramanian

ABSTRACT

For visually impaired users, existing touch-screen keyboards are cumbersome and time-consuming. We present several prototype methods of text entry on a modern touch screen mobile phone that are based on the Braille alphabet and thus are convenient for visually impaired users. We evaluate the strengths and weaknesses of our Braille-based methods through a user study with 15 participants. Our results indicate that a spatially-oriented method of entering Braille using a single finger was preferred since it balances simplicity with accuracy. We discuss how insights revealed by our user study can help us further refine and improve the preferred method.

Keywords

Alternative Text-Entry System, Braille Alphabet, Touch-screen, Mobile Phone, Visually Impaired.

Categories and Subject Descriptors

K.4.2 [**Computers and Society**]: Social Issues - Assistive Technologies for Persons with Disabilities; H.5.2 [**User Interfaces**]: User Centered Design, Prototyping.

General Terms

Algorithms, Design, Measurement, Experimentation

1. INTRODUCTION

The fast and widespread adoption of mobile phones which use only a touch screen for input has created several accessibility challenges for visually impaired individuals. These challenges include navigating a primarily visual user interface, finding numeric buttons to enter phone numbers and typing text on a purely visual keyboard. Improving interface navigability for visually impaired users has been dealt with in [6] through a set of gestures which allow the user to move around the UI controls and listen to their contents. Commercial implementations of this interface now exist [5]. However, concerning text entry, blind users are still forced to use the built-in on-screen keyboard. Even though this keyboard may be able to speak the character above which the user's finger is located, typing using this method is time consuming and error-prone. Letters on such keyboards are tiny and are placed too close to one another due to the constrained screen size and so the plethora of touch targets makes typing slow to the point of frustration. In previous work [4, 1, 3, 9, 7] researchers have tried to tackle this issue by proposing a set of new input methods, some of which use the Braille alphabet. In most cases, the new method of entering text was compared with a standard QWERTY implementation on a touch screen and found to be superior. However, the various Braille-based input methods have not been compared systematically and empirically so that their individual advantages and disadvantages could be analyzed in detail. In this paper, we put the Braille system of writing under our microscope and ask the question: Given that we have to design a Braille input method, what would be the best way that such a method would be implemented? We do this by proposing a set of 4 Braille input methods augmented with a small set of editing gestures. We compare our methods along with one existing method from the literature [10] in a usability study. By testing a set of diverse input methods, we are able to explore the solution space from different angles and elicit different feedback from different groups of users who might be attracted or served best by different methods. Even though this need to cater to the wide diversity of the blind population when typing on a touch screen has been previously recognized in the literature [8], this diversity has not been directly used to guide the design of future input methods. Further, instead of proposing a set of potentially numerous and separate input methods each one to be used by a different group of visually impaired individuals, the current work takes a different approach: We rely on the data collected during our user study to guide us in improving the Braille input method that our users preferred the most.

*Computer Science, New York University. Email:{nektar,alexr,lakshmi}@cs.nyu.edu

2. PREVIOUS WORK

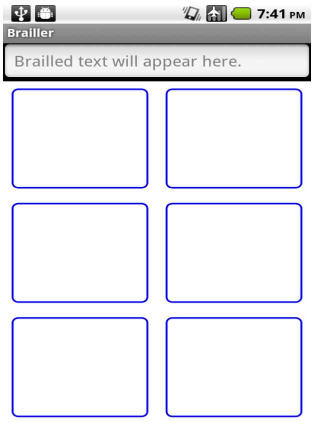
In [3, 9] typing Braille on a smartphone's touch screen similar to the way that Braille is entered on a mechanical Brailler was demonstrated. However, this method requires the use of at least 3 fingers from each hand, making holding the phone with the remaining fingers difficult and allowing for spurious touches. The use of a set of gestures in order to enter each Braille character row-by-row has been proposed in [7]. Despite the fact that this approach is similar to one of our Braille input methods discussed below 3.1, our method is different in the sense that it was designed to be used single-handed, a common usage scenario. Contrary to the method in [7] which includes gestures requiring 3 fingers, our corresponding method can be used on phones with narrower screens or on phones which do not support more than two simultaneous touches, a limitation which unfortunately is present on various devices. The diversity of the population of visually impaired users and its effects on the usage of touch screen input methods has been identified in [8]. The authors found that the age, the time of onset of blindness, as well as the cognitive and spatial abilities of the individual can play a role in the speed and accuracy of using various input methods that had been described previously in the literature. However, the authors did not try to design an improved input method but proposed that all text entry methods should be available in order to fulfill the needs of different users. Additionally, the authors did not compare different Braille input techniques in order to discover if the relative slowness determined in the Braille input method they used could be in any way remedied. In contrast, various Braille input techniques were described in [2]. However, their usage was not thoroughly evaluated and the relationships between the different methods not investigated to the point of deriving guidelines to assist in creating an improved Braille input method. A common weakness of most of the above solutions is that little emphasis was given on methods for editing or navigating already entered text.

3. SYSTEM DESIGN

This section describes the implementation of all our input methods and subsequently concentrates on the first method, with emphasis given to the algorithm used to improve its accuracy.

3.1 Input Methods

Figure 1: Screenshot of Interface



Braille is a system of reading and writing familiar to a large number of blind individuals. Every character of Braille is made up of a combination of 6 dots arranged in 3 rows. Instead of having to learn a new system of touches or swipes, our system aims to put Braille in the center, solving the memorization problem of other approaches. In our system the touch screen is used as the surface where Braille characters are entered, making our implementation able to run on most modern smart phones. To permit editing and review of entered text, we have also implemented a set of simple directional gestures which act as the space, backspace and arrow keys. The implemented input methods are described below:

1. One-Finger: Each Braille dot is selected spatially on a virtual 3 by 2 grid by tapping. Each time a dot is tapped, the software speaks the dot's number. After a specific short interval has passed without any screen touches, the entered dots are interpreted and the resultant character is typed. This method allows the user to effectively "sketch out" the character he or she wishes to enter using Braille. The interval before the entered dots are interpreted is calibrated to be short enough to make this input method appear natural to a blind Braillist, but long enough to allow a character with multiple dots to be entered without mistakes. Even though the dots are placed on a visual grid, our algorithm does not rely on the user having to touch the exact locations where dots are visually present, but can intelligently deduce the entered character by the overall shape of the tapped locations 3.2.

2. Split-Tap: To enable easy exploration, each dot is selected spatially as above but not by single taps. Instead the user moves a finger around the screen until he or she hears the desired dot number. Then, the user places a second finger on the screen to select it. After selecting all the dots making up the desired character, the user finishes entering the character by lifting up the first finger.

3. Two-Finger: To facilitate single handed text entry, we allow the ability to input a Braille character one row at a time. For each character the user taps each row of the Braille cell individually using the following gestures: If a row contains both dots, then two fingers are tapped. Otherwise the corresponding left or right dots are tapped. If a row has no dots, then a downward gesture is performed with both fingers. The remaining three fingers are used to hold the phone.

4. Thumb-Typing: To allow the user to hold the phone more securely with one hand, a method of typing only using the thumb is proposed. Given the popularity of using the thumb by many individuals to type on touch screens, it was hoped that this method would be deemed familiar by most. The Braille pattern is also entered row-by-row as in the previous method. The thumb is tapped in the vicinity of the top-left quadrant of the screen to indicate that the left dot in the current row is selected and it is tapped towards the top-right quadrant to indicate that the right dot is selected. To select both dots in a row, the thumb bends in a natural manner and taps in the bottom-half of the screen. To leave a row empty, the thumb swipes down.

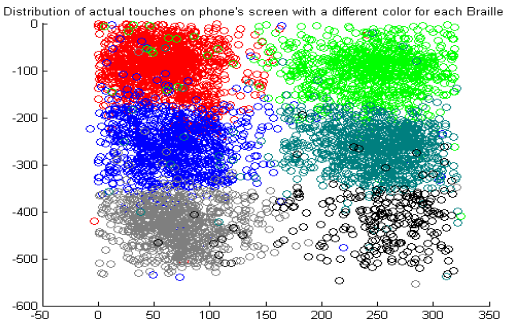
5. Nine-Digit: Implementing a combination of the methods detailed in [10, 1], the numbers 1 to 9 with their corresponding letters appear in a standard telephone keypad formation. The user chooses first a number and then one of its letters from a list arranged horizontally.

For all methods, a space character can be entered using a rightward swipe, whilst a leftward one is used for backspace. Similarly, moving the editing cursor left and right character-by-character through the entered text is enabled by swiping upwards and downwards respectively, except when using the Thumb-Typing method, where a two-finger up and down swipe is used instead.

3.2 The One-Finger Method: A Closer Look

After analyzing a set of touches for various Braille patterns, we realize that touches corresponding to each Braille dot were not always in the designated rectangluar region for that dot. This suggested that a better approach of interpreting

Figure 2: Distribution of Touches for each Dot



Braille dots from a list of touch locations should be devised. Our algorithm tries to find the Braille pattern which most closely resembles the shape of the touch locations. It enumerates all the possible Braille patterns whose dots centers are at the visible predetermined screen locations and finds the one that has the minimum Euclidian distance from all of the touch locations.

4. METHODOLOGY

We carried out a user study which included 15 visually impaired participants from the greater New York City area. Our age representation appears to include both younger and older individuals. Around two thirds of our participants were totally blind whilst the rest were legally blind. We observe that all of our participants have used at least a phone with a numeric keypad, even though 3 of them are using it without any accessibility support at all. Around half of our participants have experience with a touch screen phone and almost an equal number have used a dedicated mobile screen reader. What was surprising was that around the remaining half of our participants had only used the basic built-in text-to-speech support that comes with some phones. This support is very limited and is usually provided in order to assist in hands-free driving. The software was tested on an Android-based phone. For each of the participants, we gave them a short training time to familiarize themselves with each input method. Not all users managed to complete this training for all 5 input methods. The subjects were then given randomly chosen phrases from a standard phrase set, which they were asked to type. The users were asked to answer a questionnaire with the same quantitative and open-ended questions for each input method. The quantitative questions asked the users to rate on a scale of 1 to 4

Table 1: User Ratings of Input Methods

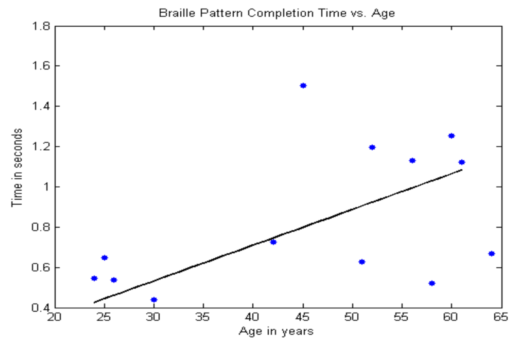
Method	Easy to learn	Likely to use
One-Finger (n=15)	3.6 ± 0.63	3.53 ± 0.92
Split-Tap (n=12)	3.17 ± 0.72	2.75 ± 0.97
Two-Finger (n=12)	2.58 ± 1	2.5 ± 1
Thumb-Typing (n=6)	2 ± 1.26	1.83 ± 1.33
Nine-Digit (n=15)	3.067 ± 0.97	2.47 ± 1.13

how easy each method was to use, how easy it was to learn, and how likely they would be to use it in the future. The open-ended questions prompted the users to give more details about what they liked and disliked about each method. At the end of each session, the person conducting the interview would enter detailed notes on what was deduced from the participant's expressed feelings and thought processes when employing the system.

5. RESULTS

Almost all of the participants who owned a touch screen phone found our system to be easier, more intuitive and more desirable for them to use than their phones' touch keyboard. Those participants who did not know Braille very well considered our system to be an excellent vehicle through which they could learn it quickly. The One-Finger method was judged to be the simplest, the most intuitive and the most preferred, followed by the Nine-Digit method. The One-Finger method was described as very natural and requiring no training, whilst a negative aspect of the Nine-Digit method was the difficulty of its two-step process for letter selection. An isolated group of users enjoyed the Two-Finger method very much and believed that it was a clever idea. However, most disliked its row-by-row input and the way it forced you to hold the phone. The Split-Tap method was perceived as being more accurate but much slower, causing frustration. The Thumb-Typing method was generally not understood or its gestures were found to be hard to perform. On a scale of 1 to 4, the following table lists the mean and standard deviations for each rating across each input method: For the One-Finger method, there appears

Figure 3: Completion Time Increases with Age



to be a correlation between the user's age and the average typing speed for each Braille pattern 3. Older users complete Braille patterns faster than younger ones. This indicates that a possible improvement to the One-Finger method would be to dynamically adjust the algorithmic parameters (such as the interpretation interval) to accommodate different ages.

6. DISCUSSION

The One-Finger method received the highest ratings because the users who knew Braille could grasp how it worked very quickly. Some of them even became comfortably proficient using it after less than an hour. Unlike the Two-Finger or the Thumb-Typing methods, which required a higher cognitive load and a longer learning curve, the One-Finger method proved to be a more natural way of entering Braille. Similarly, the Nine-Digit method, despite its familiar numpad design, proved somewhat less desirable compared to One-Finger due to its two-level hierarchy.

Concerning the benefits of the Split-Tap method, it was clear that specific users wanted to be able to cancel erroneously entered dots before completing the whole Braille pattern, whilst others wanted to be able to confirm each dot pressed. They felt that waiting until the whole pattern has been entered and interpreted, just to be subsequently able to delete it, was a waste of time. In spite of this, using the Split-Tap method for this purpose, was deemed undesirable as many participants rejected the split-tap gesture as too slow and cumbersome, whilst some found the whole Split-Tap method as too hard to learn. Performing this gesture while the first finger was towards the edges of the screen felt awkward, whilst the second tap would at times make some users accidentally lift up the first finger, registering the wrong Braille pattern. This indicates that for text entry input methods, the split-tap gesture might be inappropriate and should be avoided. Nevertheless, some form of input verification appears necessary, as some participants would continuously use backspace to ensure themselves that they had correctly typed something or in order to check their progress in the text. Ultimately, the participants wanted to have the ability that the Split-Tap method offers to confirm each Braille dot, but without having to perform a slow split-tap gesture each time. Designing such a method though would be hard, as using, for example, a double-tap gesture for confirming each dot, might turn a cumbersome method into an even more cumbersome one. At the end of the day, the seemingly simplistic One-Finger method offered a compromise for these users needs, as its pattern-matching algorithm would automatically compensate their desire for higher accuracy.

Most participants who knew Braille would conceptualize each Braille cell in terms of two columns of three dots. They had a hard time adapting to the row by row separation of the Two-Finger method, including those users who had enough vision. The widely known numerical ordering of the dots, which places dots 1-3 on the left column and dots 4-6 on a second column, seems to be creating this problem. Even users who understood the Two-Finger method conceptually, still had trouble switching to a different dot ordering required by the Two-Finger method. However, the few participants who familiarized themselves with this method quickly, found it extremely preferable. This indicates that training is needed for mastering this method but once learned the method might prove very useful. For this possibly laborious training to be undertaken by the user though, a sufficient motive should be present. The benefits of the input method, such as providing one-handed input or an enjoyable game-like approach of teaching yourself Braille, should outweigh the effort involved.

7. CONCLUSIONS AND FUTURE WORK

We have proposed 4 Braille-based text-entry techniques which use the Braille alphabet, one of which minimizes mistakes through pattern matching of user touches and all of which provide easy editing through swipe-based cursor manipulations. We have evaluated our system through a user study which compares our input methods with one another, as well as with an existing numpad-based technique. We have found that a balanced approach provided by the One-Finger method which combines ease of use with a good accuracy is most preferable. Other methods, which either enable a faster typing speed but impose more complicated gestures (Two-Finger method) or provide the ability to confirm each dot entered but are slower (Split-Tap method), are too cumbersome to be useful.

As future work, we plan to improve the One-Finger method to make it more resilient to noisy touches and more adaptive to users' idiosyncrasies. We envision implementing an adjustable interpretation interval and a dynamically resizing typing view. When a Braille pattern has more dots, the interpretation interval would lengthen to give you extra time to move around the screen and mentally construct the complex pattern. Similarly, when the dots are typed with a faster speed, the interval would shorten itself, to allow more experienced users to complete their typing without frustration. Finally, the typing view would resize dynamically based on the user's touch history.

8. REFERENCES

[1] M. Bonner, J. Brudvik, G. Abowd, and W. Edwards. No-Look Notes: Accessible eyes-free multi-touch text entry. *Pervasive Computing*, pages 409–426, 2010.

[2] H. Fard and B. Chuangjun. Braille-based text input for multi-touch screen mobile phones. 2011.

[3] B. Frey, C. Southern, and M. Romero. Brailletouch: mobile texting for the visually impaired. *Universal Access in Human-Computer Interaction. Context Diversity*, pages 19–25, 2011.

[4] T. Guerreiro, P. Lagoá, H. Nicolau, P. Santana, and J. Jorge. Mobile text-entry models for people with disabilities. In *Proceedings of the 15th European conference on Cognitive ergonomics: the ergonomics of cool interaction*, page 39. ACM, 2008.

[5] A. Inc. Apple - Accessibility - Voiceover - In Depth. `http://www.apple.com/accessibility/voiceover/`, June 2011.

[6] S. Kane, C. Jayant, J. Wobbrock, and R. Ladner. Freedom to roam: a study of mobile device adoption and accessibility for people with visual and motor disabilities. In *Proceedings of the 11th international ACM SIGACCESS conference on Computers and accessibility*, pages 115–122. ACM, 2009.

[7] S. Mascetti, C. Bernareggi, and M. Belotti. Typeinbraille: a braille-based typing application for touchscreen devices. In *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*, pages 295–296. ACM, 2011.

[8] J. Oliveira, T. Guerreiro, H. Nicolau, J. Jorge, and D. Gonçalves. Blind people and mobile touch-based text-entry: acknowledging the need for different flavors. In *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*, pages 179–186. ACM, 2011.

[9] M. Romero, B. Frey, C. Southern, and G. Abowd. Brailletouch: designing a mobile eyes-free soft keyboard. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, pages 707–709. ACM, 2011.

[10] J. Sánchez and F. Aguayo. Mobile messenger for the blind. In *Proceedings of the 9th conference on User interfaces for all*, pages 369–385. Springer-Verlag, 2006.