Dots and Letters: Accessible Braille-Based Text Input for Visually Impaired People on Mobile Touchscreen Devices

Elke Mattheiss[1], Georg Regal[1], Johann Schrammel[1], Markus Garschall[1,3], and Manfred Tscheligi[2,3]

[1] CURE – Center for Usability Research & Engineering, Vienna, Austria
{mattheiss,regal,schrammel,garschall}@cure.at
[2] ICT&S, University of Salzburg, Salzburg, Austria
manfred.tscheligi@sbg.ac.at
[3] AIT Austrian Institute of Technology GmbH, Vienna, Austria
{markus.garschall,manfred.tscheligi}@ait.ac.at

Abstract. Tailored text input methods for visually impaired and blind users are needed on touchscreen devices to support their accessibility. Therefore, we developed a new Braille-based text input method named EdgeBraille, which allows entering Braille characters by swiping one finger along the edges of the touchscreen. The approach was compared with the current standard method of a talking keyboard, first in a short-term lab study (14 participants) and then during two weeks of daily training (7 participants). Overall EdgeBraille was perceived well by the users. In terms of user performance we found no significant differences between the two methods. Based on the evaluation results and the feedback of our participants, we discuss advantages and disadvantages of Braille-based methods in general and EdgeBraille in particular, as well as possibilities for improvements.

Keywords: Text Input Method, Touchscreen, Mobile Devices, Braille, Visually Impaired and Blind Users.

1 Introduction

Many visually impaired and blind (VIB) people are intense technology users and use mobile devices regularly. With the emerging era of smartphones, touchscreen devices without keypads are becoming increasingly common, thus the improvement of touchscreen accessibility is an important issue to address.

An important aspect of interaction with smartphones is the ability to enter text, for example in order to write short messages or emails. The common way of making soft keyboards accessible for VIB users is based on the "talking fingertip technique" [1]. VoiceOver for iOS and TalkBack for Android are commercial products using this technique, which allows the device to read onscreen elements (such as letters of the keyboard) to the users, when they touch them with their fingers. A disadvantage of using a soft keyboard for entering text on touch devices is that the entire alphabet has

K. Miesenberger et al. (Eds.): ICCHP 2014, Part I, LNCS 8547, pp. 650–657, 2014.

to fit on the screen, which hampers the selection process. This is why a number of alternative solutions for non-visual text input have been presented in the scientific literature (e.g. pie-menu-based [2], gesture-based [3] and Braille-based text input [4]), aiming to reduce the number of elements on the screen.

In this paper, we present the design of a new Braille-based text input method for touchscreens named EdgeBraille and reflect on Braille-based methods in general as well as possibilities for improvement.

2 Related Work

Braille-based text input for blind people, which in the basic version uses six dots for one letter, was presented as a promising possibility in accessibility research. Previous Braille-based input methods can be differentiated into those who allow entering a Braille letter in one single step, and those who split the entry into several steps.

With TypeInBraille [4] three steps are needed to enter one Braille letter. For each row, users can select no, one or both dots. Perkinput [5] supports entering a Braille letter in one or two steps. For small screens such as smartphones, characters can be entered in two steps with three fingers. When using two small screens or one larger screen such as a tablet, two hands can be used simultaneously in order to input both columns of a Braille character in a single step.

Input of Braille letters in one step is used in BrailleType [6] and BrailleTouch [7]. Both approaches use six targets on the screen that represent the six dots of a Braille character. In the BrailleType system, target dots can be selected successively by touching them. BrailleTouch uses a multi-touch paradigm. Therefore the mobile phone is used with the screen facing away from the user, with three fingers of each hand resting over one of the six targets (three on each side).

Table 1. Summary of evaluation results reported by previous work on Braille-based methods (N = the number of participants, # = the number of sessions, wpm = the measured words per minute). BrailleTouch [7] reported the performance values captured in the last of 5 sessions. For the remaining methods the reported value is the average over all sessions conducted. TypeInBraille [8] calculated the error rate by dividing all errors through the length of the text. The rest of the methods used metrics proposed by [9]. BrailleType [6] reported the old MSD error-rate, Perkinput [5] the uncorrected error-rate and BrailleTouch [7] the total error-rate.

Method	N	#	wpm	error rate
TypeInBraille [8]	7	1	6.30	3.00 %
VoiceOver			5.20	4.00 %
BrailleType [6]	13	1	1.49	7.00 %
VoiceOver			2.10	14.12 %
Perkinput [5]	8	7	6.05	3.52 %
VoiceOver			3.99	6.43 %
BrailleTouch [7] Expert performance	6	5	23.20	14.50 %
BrailleTouch [7] Moderate performance	3	5	21.00	33.10 %
BrailleTouch [7] Poor performance	2	5	9.40	39.30 %

Considering previous work related to Braille-based text input, our goal was to create a new Braille-based method, which allows entering a Braille letter fast and in one single step. For comparison Table 1 provides an overview of evaluation results of the different Braille-based input methods and the values reported for VoiceOver. Furthermore the method should be convenient and usable with one finger, following findings by Paisios [10], who reports that one-finger interaction was rated best by blind users.

Inspired by previous research concerning text input with edge-supported Graffiti strokes [3], we developed EdgeBraille. We use a screen layout similar to BrailleType [6] and BrailleTouch [7], but the input paradigm differs significantly. BrailleType uses a multi-touch approach, which is indeed fast but requires the users to hold the phone with both hands. With EdgeBraille users can hold the phone with one hand and they only need one finger to enter the Braille letter. BrailleTouch uses tap-based input of each dot at a time, which is not very fast and requires accurate tapping. We expect a swipe-based approach along the display's edges to be faster and easier. Our design approach is described in detail in the following section.

3 Prototype Design of EdgeBraille

The structure of the screen relates to the structure of a Braille letter. The top two and bottom two dots (diameter: 12mm) of the Braille letter are placed in the corners of the display. Two points halfway along the side edges of the screen are used for the two middle dots.

A letter is entered by sketching an arbitrary sequence of Braille dots. Each dot can be activated by moving the finger on the dot, and revisiting a dot deactivates it. Examples of different strategies to write a letter can be found in Fig. 1.

Fig. 1. Possibilities to enter the letter "N" with EdgeBraille. (a) Selecting the dots using the shortest path (b) Selecting the dots column wise circling around dot no. 2 (c) Selecting the dots by first activating dot no. 2 and then deactivating it on the way back from dot no. 3.

A space is entered by tapping anywhere on the screen but not on a dot. To prevent accidentally deactivating a dot, the zone for deactivation is approximately 1.7 mm smaller than the zone for activation. When activating and deactivating, vibro-tactile feedback is provided and different sound files are played for activation and deactivation.

For users with residual vision also visual feedback is provided, as the dots are highlighted green when activated and grey when deactivated. When the finger is lifted, the activated dots are registered and the letter is spoken to the user and displayed on the screen. To prevent accidentally writing a letter, a threshold of 75 milliseconds for lifting the finger was defined.

The edges of the touchscreen are used as guardrails by marking them with a mechanical frame. Thus, we assume that this will ease orientation for the VIB people and speed up the process. This assumption is supported by Kane et al. [11], who reported that blind people preferred gestures that used screen edges and corners. For the short-term evaluation we built a cardboard frame to provide physical guidance. For the longer-term evaluation we used a commercial cover (Griffin GB01902 Survivor Cover) as shown in Fig. 1.

4 Short-Term Evaluation

In a short-term evaluation we compared the participants' performance and opinion towards EdgeBraille along with Android's talking keyboard method TalkBack. We did this for two reasons: first talking keyboards still are the standard accessible method for text input on touchscreens provided by mobile operating systems and secondly, the Braille-based approaches presented in the related work were not available for comparison at the time we conducted our research. For the short-term evaluation we used a HTC Desire S with Android 2.2.3.

For the evaluation 14 VIB participants (9 male, 5 female) with a mean age of 33.00 years (standard deviation = 12.22) were invited to participate. All participants were able to read and write Braille letters and five of them already had a lot of experience with touch-based mobile phones and the talking fingertip method. They were provided with a short description and five minutes of training with each method. To assess the training effect throughout usage, participants had to enter 16 two-word texts successively with each input method. Participants were told to enter the text as quickly and accurately as possible and they could not correct incorrectly entered characters. We measured the words per minute (wpm = number of correct characters per minute divided by five) and MSD error rate [9] for the methods and compared the beginning (i.e. first four tasks) with the end (i.e. last four tasks) of the test, to analyze the training effect.

The results show that EdgeBraille achieves the same performance as TalkBack at the end of the test (wpm: EdgeBraille=3.97±1.00, TalkBack=3.64±1.35, $F_{1,13}=0.793$, $p=.389$; error rate (in percent): EdgeBraille=8.43±5.21, TalkBack=10.58±9.99, $F_{1,13}=0.46$, $p=.512$). Regarding the training progress, it is not surprising that the wpm rates are significantly higher ($p<.000$) in the last four tasks (mean=2.94±1.03) than in the first four tasks (mean=3.81±1.18). The further data analysis showed a significant increase for EdgeBraille ($t_{13}=-6.14$, $p<.000$), as well as for TalkBack ($t_{13}=-2.76$, $p=.016$). The results are also shown in Fig. 2 (left figure). Regarding the participants' preference, we found that EdgeBraille was preferred by eight users, while TalkBack was only preferred by four (and two were indecisive).

5 Longer-Term Evaluation

To understand how the users' performance and opinions evolve over time, a two-week evaluation with a subset of seven users of the first evaluation was conducted. Five men and two women with a mean age of 38.86 years (standard deviation = 14.29) participated; two of them were users of the talking fingertip method.

For the longer-term evaluation we used the iPhone 4 (with iOS 5.1.1), which was configured to work exactly in the same way as TalkBack in the short-term evaluation. Participants had to enter given texts (92 to 99 characters per method) in a specific sequence every day with EdgeBraille and VoiceOver. The input was logged to ensure participants conducted all training sessions. At the beginning, the 8th and the 15th day of the study, a lab session was organized to assess the participants' performance and opinion.

At the end of the two-week training, participants were able to enter text at an average with 7.17 (±2.14) wpm with EdgeBraille and 6.29 wpm (±2.60) with VoiceOver ($F_{1,6}=1.92$, $p=.215$). The data analysis shows a significant trainings effect ($F_{2,5}=12.76$, $p=.011$), and a significant difference in wpm between the first and the third test for EdgeBraille ($t_6=-3.72$, $p=.010$) as well as VoiceOver ($t_6=-4.86$, $p=.003$). Regarding the preferences, we allowed the participants to state multiple preferences. We found a clear preference for VoiceOver in the first session (five participants preferred Voice-Over, one EdgeBraille, and two were indecisive). This changed in the second session, where EdgeBraille was the most preferred input method (five preferred EdgeBraille, two VoiceOver and one was indecisive). In the last session the preference for Edge-Braille and VoiceOver was balanced (four preferred EdgeBraille, four VoiceOver, and one was indecisive).

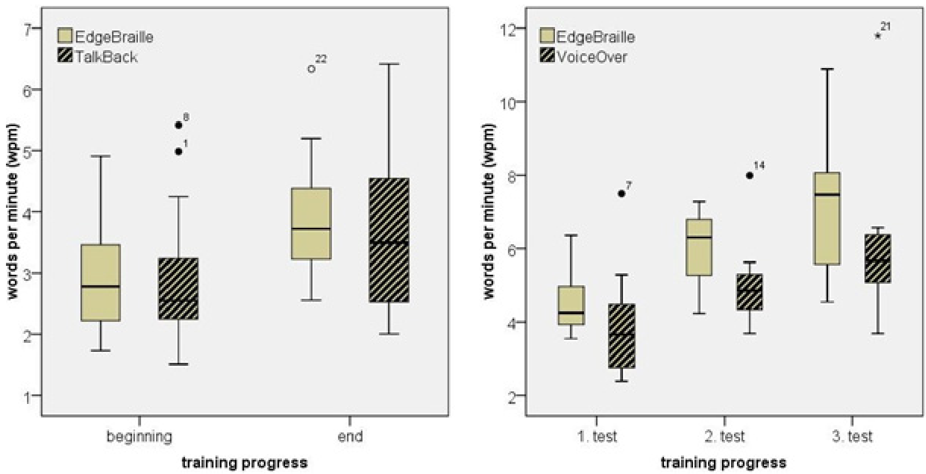

Fig. 2. Text input rate (wpm) in short-term (left) and longer-term (right) evaluation

6 Discussion

With regard to performance measures, our results show that there is no difference between EdgeBraille and the talking keyboard approaches. Also, looking at reported results from related work, EdgeBraille has a comparable input speed as TypeInBraille and Perkinput, but seems to be faster than BrailleType and slower than BrailleTouch.

In our research, participants stated that Braille-based methods are especially suitable for people who do not know the QWERTY keyboard layout very well. On the other hand for people familiar with the keyboard it is easy to find specific letters. Two disadvantages of Braille-based methods compared to keyboard-based methods are the need to know all Braille characters by heart and that it is not clear, which characters exist, as they cannot be directly accessed with the talking fingertip technique. Regarding the prototypical implementation of EdgeBraille, participants were missing some control characters such as delete, enter, cursor back and cursor forward. These could be implemented by assigning unused Braille combinations, although those are not standardized and therefore could decrease learnability.

However, our participants appreciated that with EdgeBraille there are fewer elements on the screen compared to talking keyboard and that the elements are larger. Due to the lower number of target elements in Braille-based approaches (typically six elements) compared to keyboard implementations (typically more than 26 elements) the Braille-based interfaces can be designed much smaller. This concurs with the feedback obtained from participants, that they would prefer a smaller version of EdgeBraille and expected it to be faster than the full-screen version.

Another aspect worth noticing is that the currently used 6-point Braille version could be extended to improve text input performance. By the implementation of Grade-2 Braille, contractions and abbreviations could be entered instead of whole words.

Based on these insights, EdgeBraille offers specific possibilities of further improvement of Braille-based methods. EdgeBraille could be used in scaled down versions, which do not occupy the whole screen’s real estate. Therefore it could better be integrated with applications, because text input is no goal in its own but typically used in combination with other interface elements. This applies also to TypeInBraille and BrailleType but not to Perkinput and BrailleTouch as the size of the interface is directly related to the user’s hand size and could not be scaled down to very small configurations.

Finally, all Braille-based methods discussed in this paper – including EdgeBraille – use 6-point Braille. Though it seems that for text written with a mobile phone 6-point Braille is sufficient, participants of our research activities call for the 8-point version to have a greater repertoire of characters. EdgeBraille (as well as Perkinput, TypeInBraille and BrailleType) could be easily extended to an 8-point version. In the case of BrailleTouch an extension to 8-point would be problematic, as users will en-counter difficulties in handling the device in a stable manner.

7 Improvements and Future Work

To analyze the identified improvement potential – smaller size and 8-point Braille – we developed a version of EdgeBraille allowing input of 8-point Braille, which was scalable to different sizes. We created two versions of 8-point EdgeBraille, one scaled by the factor 0.5 (occupying a quarter of the screen, see Fig. 3) and one scaled by factor 0.3 (occupying a ninth of the screen). These smaller versions could seamlessly be used as alternative text input method instead of the talking keyboard, by integrating it with typical smart phone use cases (e.g. writing emails).

To provide tactile feedback, we used regular screen protection foil where the area occupied by 8-point EdgeBraille was cut out, to create a perceptible edge for guidance of the input finger. This approach for providing tactile feedback is similar to the one presented by [12]. The authors found that that haptic structures can serve as additional feedback in non-visual situations (demonstrated for an in-vehicle application).

In the 8-point version of EdgeBraille we also extended the range of functions. We added the possibility to delete characters and to search for unknown characters, by assigning unused Braille combinations. For example, to delete a character the unassigned dot 7 (down, left corner) was used. Moreover the text written so far could be spoken to the user by the text to speech engine. A double tap anywhere on the screen triggers the text to speech engine. By these means the text input method could be used in a more realistic manner than before.

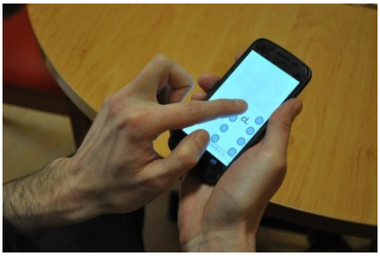

Fig. 3. 8-point Braille version of EdgeBraille (scale 0.5), guidance with display protection foil

A first proof-of-concept user study with 7 participants revealed that users are able to enter text including special characters using the 8-point version of EdgeBraille in both sizes (0.5 and 0.3). Input speed and error rate differed widely depending on the experience with Braille-based input in general and knowledge of 8-point Braille in particular. However, from the initial results we see that the 8-point version of EdgeBraille is a promising approach for entering special characters, which is cumbersome to do with the talking keyboard approaches.

Moreover we could show, that it is possible to use a scaled down version of EdgeBraille with a perceptible edge provided by cut out screen protection foil. All participants stated that the guidance by screen protection foil was helpful, although two participants stated that in real life they would only use it if possible without foil, as the edges provided by the foil may be distracting when performing other tasks than text input. Regarding size the 0.5 version was perceived well, the 0.3 version was perceived as too small by five participants. Providing the user the possibility to tailor the size of the input element to their preferences might be a suitable option.

In future work we plan to examine the optimal size that balances speed and error, and analyse text input performance of the 8-point EdgeBraille approach in detail.

8 Conclusions

In this paper we presented a new Braille-based text entry method and discussed different approaches of text entry for VIB people on touchscreen devices. Braille-based text entry mechanisms are an important possibility to complement current text input paradigms based on talking keyboard.

Overall EdgeBraille was perceived well by the users, possesses favourable handling characteristics, and performed comparable to talking keyboard. Especially when considering the improvements, EdgeBraille has potential to become a convenient form of text input for Braille literate users.

References

1. Vanderheiden, G.C.: Use of audio-haptic interface techniques to allow nonvisual access to touchscreen appliances. In: 40th Annual Meeting of the Human Factors and Ergonomics Society (1996)
2. Yfantidis, G., Evreinov, G.: Adaptive blind interaction technique for touchscreens. Universal Access in the Information Society 4(4), 328–337 (2006)
3. Wobbrock, J.O., Myers, B.A., Kembel, J.A.: EdgeWrite: A stylus-based text entry method designed for high accuracy and stability of motion. In: Proc. UIST 2003, pp. 61–70 (2003)
4. Mascetti, S., Bernareggi, C., Belotti, M.: TypeInBraille: A Braille-based typing application for touchscreen devices. In: Proc. ASSETS 2011, pp. 295–296 (2011)
5. Azenkot, S., Wobbrock, O., Prasain, S., Ladner, R.: Input finger detection for nonvisual touch screen text entry in Perkinput. In: Proc. GI 2012, pp. 121–129 (2012)
6. Oliveira, J., Guerreiro, T., Nicolau, H., Jorge, J., Gonçalves, D.: Blind people and mobile touch-based text-entry: Acknowledging the need for different flavors. In: Proc. ASSETS 2011, pp. 179–186 (2011)
7. Southern, C., Clawson, J., Frey, B., Abowd, G., Romero, M.: An evaluation of BrailleTouch: Mobile touchscreen text entry for the visually impaired. In: Proc. MobileHCI 2012, 317–326 (2012)
8. Mascetti, S., Bernareggi, C., Gerino, A.: TypeInBraille: Quick typing on smartphones by blind users. Rapporto interno N. 39 11 RT, Uni. of Milano
9. Soukoreff, R.W., MacKenzie, I.S.: Metrics for text entry research: An evaluation of MSD and KSPC, and a new unified error metric. In: Proc. CHI 2003, pp. 113–120 (2003)
10. Paisios, N.: Mobile accessibility tools for the visually impaired. PHD Thesis (2012), `http://cs.nyu.edu/web/Research/Theses/nektariosp.pdf` (retrieved online September 19, 2012)
11. Kane, S., Wobrock, J., Ladner, R.: Usable gestures for blind people: Understanding Preference and Performance. In: Proc. CHI 2011, pp. 413–422 (2011)
12. Zimmermann, S., Rümelin, S., Butz, A.: I Feel it in my Fingers: Haptic Guidance on Touch Surfaces. In: Proc. TEI 2014, pp. 9–12 (2014)