SBraille: A New Braille Input Method for Mobile Devices

Soonyong Lee[1], Ji Su Park[2], and Jin Gon Shon[1](✉)

[1] Department of e-Learning, Graduate School, Korea National Open University, Seoul, South Korea
{hayanhosu, jgshon}@knou.ac.kr

[2] National Center of Excellence in Software, Chungnam National University, Daejeon, South Korea
bluejisu@cnu.ac.kr

Abstract. There are not many accessibility technologies to input braille in the mobile devices for the visually impaired. We have studied three conventional methods to input braille in mobile devices and, as a result of our study, we propose SBraille, a new braille input method using swiping gesture for mobile devices. In order to input a 6-dot Braille letter using SBraille, six dots are divided into three parts with two dots. Each part can be one of the patterns: '●○', '○●', '●●', and '○○'. The first two patterns can be represented by tapping left side and right side, respectively. The last two patterns by swiping upward and swiping downward, respectively. The most important advantage of the SBraille method is that the visually impaired can input braille with only one hand, so that they can use the other hand for another use, e.g., holding the cane. Other advantages of SBraille and its comparative analysis with other braille input methods (Braille Keyboard, ThaiBraille, and MBraille) are also explained in the paper.

Keywords: Braille · Braille input methods · Mobile braille
Touchscreen · Soft braille keyboard

1 Introduction

The touch screen technology of mobile devices brings a lot of benefits to the visually impaired (VI). This technology makes the requirement of physical keyboards to interact with mobile devices no longer necessary. Moreover, the VI easily accesses information with a screen-readable function that is built into the mobile device [1]. However, most mobile devices are not designed for the VI. That is the reason why it is imperative to develop a better electronic accessibility tool to meet their standard needs. There is a limit on the technology for typing a letter using a mobile device by the VI [2]. For the VI, there is the braille input method for mobile devices, which uses the electronic braille keyboard or the conventional soft braille keyboard. The former is using an external keyboard, and the price is too expensive [3]. The latter needs improvement because of inefficiency in use by the VI. The disadvantages of the latter are: The VI are difficult to access easily because they need to find an object's location with their fingers

© Springer Nature Singapore Pte Ltd. 2018
J. J. Park et al. (eds.), *Advances in Computer Science and Ubiquitous Computing*, Lecture Notes in Electrical Engineering 474,
https://doi.org/10.1007/978-981-10-7605-3_87

when typing braille dots using a fixed object or a baseline on the screen of a mobile device [4]. Most of the VI use assistance tools, such as a white cane or the sight of guide dogs in the daily life. Therefore, the use of both hands to input braille letters is inconvenient. In addition, the input methods using multiple fingers make the VI tired because the rest of the fingers not involved for input braille dot at this moment should be prepared for the next input. In order to improve these shortcomings effectively, we propose a new braille input method, SBraille, which is based on the swipe gesture function provided by most mobile devices.

2 Braille and Its Representations

2.1 Braille

Braille is a system of tactile writing and reading used by the VI who rely on contacts and hearing. They touch the bumps or dots that represent characters using their fingertips [4]. One cell of braille consists of six dots and these dots are placed within two columns, and each column has three dots vertically as shown in Fig. 1. In Fig. 1, the black dot represents a bump of braille. Figure 2 shows the order to read braille dots. They read only the number, in which the bump is, in order. Figure 1 shows the English alphabet letter 'a', and it can be read as '①' [5].

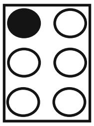


Fig. 1. English 'a'

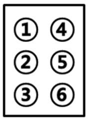


Fig. 2. The order to read braille

2.2 Conventional Braille Input Methods for Mobile Devices

We have researched three braille keyboard systems for mobile devices (Braille Keyboard, ThaiBraille, and MBraille) that are internal software keyboards based on the touch screen technology.

Firstly, Braille Keyboard has UI with the same shape as the six braille dots, as shown in Fig. 3. To type a braille letter, the VI should know the exact location of the object representing each dot of a braille letter, and it is very difficult for the VI to find the location with their fingers [6]. Secondly, ThaiBraille has a vertical baseline of typing located in the center of the screen on mobile devices as shown in Fig. 4. If the VI touch the left area of the baseline with one finger, '●○' is entered. If they touch the right area, '○●' is entered. In addition, if they touch the screen with two fingers at the same time, '●●' is entered, and touch with three fingers simultaneously, '○○' is entered. The VI should use both hands constantly because this method requires up to three fingers [7]. Lastly, MBraille provides the six-dot braille shape as shown in Fig. 5, and the VI should touch effective dot(s) with six fingers to input a braille letter. This method requires both hands simultaneously, three fingers on each hand. While the VI

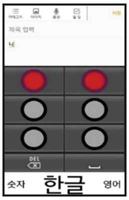


Fig. 3. Braille Keyboard

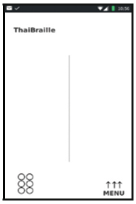


Fig. 4. ThaiBraille

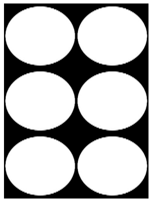


Fig. 5. MBraille

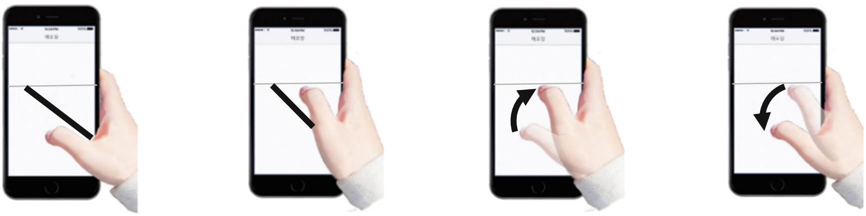
enter one dot, tired because the rest of five fingers not involved for input braille dot at this moment should be prepared for the next input [3].

3 SBraille

SBraille is a mobile braille input method, which uses not only a tap gesture function but also a swipe gesture function on the touch screen of mobile devices. This method requires only one hand. In order to input braille dots, it requires only one finger, the thumb of the using hand. A baseline or touching positions for entering braille do not need to be fixed. The VI creates their own baseline using swipe gesture from the position they are holding the mobile device. The input action is centered around its baseline. The direction of the swipe gesture is determined by comparing the first coordinates of the start of the swipe and the last coordinates of the stop of the swipe.

3.1 Input Method of SBraille

To start SBraille, the user needs to set up a braille baseline with the following three steps. First, one hand holds the mobile device. Second, the user swipes up from the bottom of the screen to the diagonal direction above the touch screen. Third, SBraille creates the baseline based on necessary calculations including the direction and length of the swipe. When holding the mobile device, each person's thumb position is different. Hence, it creates the baseline customized to the user. In Fig. 6, the thick black line represents the baseline for the right-handed person (also applicable to the left-handed).



ⓐ left-tap(●○) ⓑ right-tap(○●) ⓒ swipe-up(●●) ⓓ swipe-down(○○)

Fig. 6. Input method of SBraille

SBraille provides four types of input methods: (1) Touch the left side of the area which is divided by the baseline, as Fig. 6ⓐ shows '●○' are entered. (2) Touch the right side of the area, as Fig. 6ⓑ shows '○●' are entered. (3) Swipe up, as Fig. 6ⓒ shows '●●' are entered. (4) Swipe down, as Fig. 6ⓓ shows '○○' are entered.

This method will input two braille dots in one row at a time. It takes three gestures to input six dots of braille on average. For example, one can input the English alphabet letter 'd' (①④⑤) by using the SBraille input method. They would enter '●●' (①④) by swiping up at one time. Then they would enter '○●' (⑤) by the right-tap action. Finally, they would enter '○○' as the action while swiping down.

3.2 Implementation of SBraille

The SBraille mobile keyboard software has been developed in an Android-based smartphone. The language used is Java. The implementation details are as follows: (1) Set up a braille baseline. (2) Convert from the input action to the braille code, and the braille codes are combined. (3) Classify the braille type in the braille set. (4) Decide whether to combine with the previous braille dots. (5) Convert from input code to a character after applying the braille attribute and braille rules. (6) Display a character on the screen.

Figure 7ⓐ (set up a baseline) shows the step is setting a baseline for entering the braille dots. SBraille verifies whether there is a stored baseline setting value. If the saved setting value does not exist, a new baseline will be created by the swipe gesture. Or if the saved settings value exists, the baseline can be restarted with the previously saved settings value.

Figure 7ⓑ (input braille dots) shows the step to enter the braille dots. Input action is three times on average. SBraille determines what action is entered, and converts it to the corresponding braille code. Due to two braille dots being entered in one row at one time, the third time six braille dots will be entered. If less than six braille dots are entered, the action can be continued until six braille dots are entered.

Figure 7ⓒ (representation of a character) shows the step to express the braille code of six braille dots for a character. SBraille searches a received braille code from a braille set and classifies the braille code by what was entered. After determining whether the braille code is to be combined with the previously entered braille code, it converts into a character by using a braille rule and the braille properties. The braille characters can be expressed as English, Korean, or numbers.

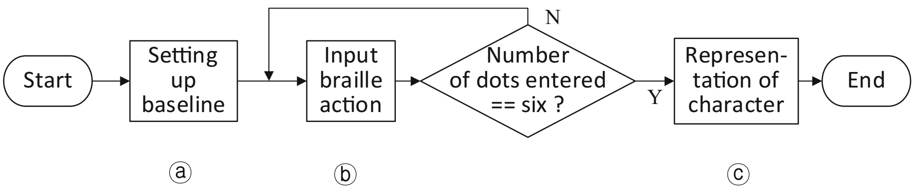


Fig. 7. Flow chart of SBraille

4 Comparative Analysis for Verification

In order to determine and verify the effectiveness of the proposed SBraille braille input method, we have conducted a comparative analysis with conventional braille input methods. For input dots, the program measures the average distance of a finger's movement to input dots. The letter is compared and applied to the frequency of consonants and vowels [8].

The average was calculated distance between the touchscreen and the finger of holding up for the next input action. Korean braille is only possible with a Braille Keyboard and SBraille. Therefore, we separated the gestures into entering Korean braille and entering English & numbers braille, which gave us comparative analysis. For the Korean braille, we compared using the Braille Keyboard and SBraille. In the English braille & numbers, we compared the ThaiBraille, MBraille, and SBraille. Below is the comparative analysis of the criteria and a description of the results shown in Table 1.

Table 1ⓐ, the number of hands the VI used when entering braille text and (number of fingers) are shown. The Braille Keyboard and SBraille use one hand and one finger in the Korean braille, two hands (three fingers) in ThaiBraille, two hands (six fingers) in MBraille, and one hand (one finger) in SBraille in the English & numbers braille.

Table 1ⓑ, the average number of input action time applied to the frequency of use characters is shown. In the Korean braille, Braille Keyboard was less than SBraille. In the English & numbers braille, MBraille was the least and SBraille was less than the ThaiBraille.

Table 1ⓒ, the grand sum of the average moving distance of a finger for input braille dots is shown. It sums up the three average moving distances, the average distance of moving the finger the first time for preparing the input braille dots, the average distance of moving the finger after the input action and for next input, and the average distance of moving the finger for preparing the input braille dots. The total sum of the average travel distances of SBraille is the smallest in all cases shown.

Table 1. Comparative analysis of braille input method for mobile devices

Braille language	Input method	ⓐ	ⓑ (times)	ⓒ (mm)
Korean	Braille Keyboard	One-hand (one)	2.79	54.97
	SBraille	One-hand (one)	6.03	45.50
English & Number	ThaiBraille	Two-hand (three)	19.66	54.44
	MBraille	Two-hand (six)	10.10	62.30
	SBraille	One-hand (one)	11.55	42.02

5 Conclusion

In this paper, we have designed and implemented a new braille input method, SBraille, for mobile devices that takes advantage of the swipe gesture function. This method allows the visually impaired (VI) to input braille text on a mobile device efficiently.

SBraille creates a customized baseline to input braille text with a swipe gesture of the thumb on the mobile touchscreen. The braille dots are entered in the area that created by the baseline. The braille dots are entered and converted into a braille character, again transformed into the character, and represented on the screen of the mobile device.

In order to compare and verify the effectiveness of SBraille, we have done a comparative analysis of the number of hands and fingers, the average number of times of the input action and the average moved distance of the fingers used to create the braille character. SBraille and Braille Keyboard used one hand and one finger, the average number of times of the input action of braille dots in Braille Keyboard was smaller than SBraille in the case of Korean braille. In the case of English & number braille, MBraille was the smallest. In addition, the average moved distance of the fingers to input braille text in SBraille was the smallest in the all of the cases.

It is the most important contribution of SBraille that the VI can use only one hand to enter braille text. It means that they can hold a cane or the leash of a guide dog at the same time while they enter braille text. Since this method uses only the thumb, the other fingers need not be held up to prepare the next input. A swipe gesture takes place within a scope that allows the thumb to be moved comfortably. Since the VI can enter a braille in the area divided by the baselines the create, there is no need to locate the object or baseline into a fixed position to be entered by the user. Additionally, it has a feedback function that announces with a TTS voice service or vibration as to whether the VI correctly input a braille text. As a result, the proposed new braille input method, SBraille, is more efficient than all other conventional input methods.

References

1. William, G., Eelke, F.: Accessible touchscreen technology for people with visual impairments: a survey. J. ACM Trans. Access. Comput. (TACCESS) **10**(1), 256–290 (2016)
2. Nektarios, P.: Mobile accessibility tools for the visually impaired. Ph.D. thesis, NY University, NY (2012)
3. Caleb, S., James, C., Brian, F., Gregory, A., Mario, R.: An evaluation of brailletouch: mobile touchscreen text entry for the visually impaired. In: 14th International Conference on Human-Computer Interaction with Mobile Devices and Services Companion (MobileHCI), pp. 155–156. ACM, New York (2012)
4. Mrim, A., Srinivas, S.: SingleTapBraille: developing a text entry method based on braille patterns using a single tap. In: 11th International Conference on Future Networks Communications (FNC), pp. 248–255. Elsevier B.V., Amsterdam (2016)
5. Nahar, L., Jaafar, A., Ahamed, E., Kaish, A.B.: Design of a Braille learning application for visually impaired students in Bangladesh. J. RESNA Assist. Technol. **27**(3), 172–182 (2015)
6. Korea Blind Family Association. http://www.kbfa.or.kr/bbs/board.php?bo_table=0403-&wr_id=102. Accessed 5 Feb 2017
7. Sergio, M., Cristian, B., Matteo, B.: TypeInBraille: a braille-based typing application for touchscreen devices. In: 13th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS 2011), pp. 295–296. ACM, New York (2011)
8. Hong, W.P.: Study on the ASCII code in the side of the transmission efficiency in data communications. J. Korea Inst. Electron. Commun. Sci. (Electron. Commun.) **6**(5), 657–664 (2011)