Journal of Ubiquitous Systems & Pervasive Networks
Volume 9, No. 1 (2017) pp. 23-31

Improved SingleTapBraille: Developing a single tap text entry method based on Grade 1 and 2 braille encoding

Mrim Alnfiai[a*], Srinivas Sampalli[b]

Faculty of Computer Science, Dalhousie University, Halifax, B3H 1W5, Canada

Abstract

Touchscreen technology has brought about significant improvements for both normal sighted and visually impaired people. Visually impaired people tend to use touchscreen devices because these devices support a screen reader function, providing a cheaper, smaller alternative to screen reader machines. However, most of the available touchscreen keyboards are still largely inaccessible to blind and visually impaired people because they require the user to find an object location on a touchscreen in order to interact with an application. In this paper, we describe SingleTapBraille, a novel nonvisual text input approach for touchscreen devices. With SingleTapBraille, a user enters characters including text, numbers, and punctuations by tapping anywhere on the screen with one finger or a thumb several times based on braille patterns. This paper presents our initial keyboard design to enter Grade 1 and our explorative evaluation of SingleTapBraille conducted with braille instructors and visually impaired users. It also presents the implementation of Grade 2 and an initial evaluation of the improved SingleTapBraille keyboard with a blind user.

Keywords: *Accessibility; text entry; blindness; braille; touchscreens; mobile devices; single-touch interaction techniques; speech output*

1. Introduction

Mobile technology has undergone remarkable developments, allowing users to perform an ever-increasing range of different tasks. Touchscreen technology, which has shown the most dramatic technological advances in recent years, is available for numerous items, including household and medical devices, but it is most commonly used for smartphone devices. Touchscreens have brought numerous benefits for users, allowing them to interact simply and quickly with the touchscreens. Touchscreen technology also eliminates the requirements of external keyboards for interacting with the computer. As a result, mobile devices have become an essential part of most normal sighted people's lives. Notably, blind and visually impaired people have also widely adopted these devices and depend heavily on them. This is due to the notable improvements in the screen reader, which is built into the mobile device and is cheaper than screen reader machines. Nonetheless, touchscreen mobile devices present serious accessibility challenges to people with no or low vision, simply because touchscreen device design is based on visual interaction. The main difficulties that face blind and visually impaired individuals relate to interaction procedures associated with touchscreen mobile phones, which is incredibly difficult without vision. Thus, visually impaired individuals have difficulty finding the letters, numbers, or punctuation required to enter information on a touchscreen and to easily send messages. A number of recent studies have proposed novel methods for overcoming the challenges of the built-in on-screen keyboard [1][2][3]. In this article, we analyzed the input methods designed to assist blind and visually impaired individuals in entering information on mobile phone touchscreens. Defining the advantages and disadvantages of the existing input methods can help us overcome some limitations and propose a new input method that may better aid blind individuals in finding the exact locations of letters and other characters.

We decided to tackle this issue because it is a problem that affects a very significant portion of our population. According to the American Foundation for the Blind, there are approximately 10 million blind or partly blind people in the United States [4]. It estimates that the number of adults with vision loss could double over the coming 30 years [4]. Therefore, it is essential to create a low cost and efficient keyboard for accessible text entry.

The primary objective of this research is the development of a novel application to enable visually impaired individuals to enter text efficiently by eliminating looking for a specific object location on a touchscreen and eliminating the need to switch between numbers, letters, and punctuation modes. This new text entry method will be more effective than existing ones because it will use a single touch several times in combination with VoiceOver service in order to facilitate error detection and editing. This research also aims to address the main issue confronting visually impaired users: entering information on a touchscreen while using one hand to hold the mobile device and a thumb to tap braille dots on a screen [5][6]. In addition, the novel application allows users to enter Grade 1 and Grade 2

[*] Corresponding author. Tel.: +19023297444
E-mail: mrim@dal.ca

DOI: 10.5383/JUSPN.09.01.003

braille and also enables users to perform other text features, including switching from capital to lower case letters, in a short time. Our approach also enables users to edit text and insert punctuation. We also use VoiceOver service to allow users to identify the written letters and words. The keyboard proposed here is easy to use and does not require users to carry extra equipment to be able to interact with the system. Crucially, it is easy to learn because it is based on braille, which is the primary way in which blind persons read and write.

By achieving these objectives, we can achieve the overall goal of minimizing the accessibility challenges that blind people face when using touchscreen devices. Perhaps the most central limitation for people with vision loss is their inability to find the position of a particular character on a touchscreen keyboard. In light of this, the critical motivation for this project is to assist blind and visually impaired people to select letters without finding exact, difficult locations, allowing them to tap anywhere on the screen to insert an intended letter. Our proposed keyboard will save them time by letting them tap braille letter dots on any accessible part on a screen and swiping a particular gesture to edit the typed text. The system was developed using a Nexus 5 mobile device with an Android 4.3 platform. The app is based on an android operating system and built using the android studio program.

In our previous paper [20], we introduced our initial algorithm and discussed the strengths of the SingleTapBraille keyboard over other existing braille keyboards. In the first version of SingleTapBraille, we only implemented Grade 1 braille. We explained the novel interaction method, which is using a single finger to type on a screen based on braille code. we evaluated the keyboard with three participants (two teachers and one blind user adept at writing in braille. We presented participants' opinions about the keyboard, and their suggestions for improvements. We also reported the keyboard's strengths and limitations based on the participants' experiences when using the application.

This article has been expanded from its original presentation at The 11th International Conference on Future Networks and Communications [20]. As noted, our previous article introduced the implementation of Grade 1 braille in the SingleTapBraille application and described an explorative evaluation. This expanded article contains additional sections describing Grade 2 or contracted braille and its purpose. We also explain the implementation of Grade 2 braille, which includes abbreviations and contractions [16], allowing us to improve the speed of the SingleTapBraille keyboard. In addition, we report an initial evaluation about the improved keyboard after implementing Grade 2 braille.

The paper is organized as follows: Section 2 gives a brief introduction of Grade 1 and 2 braille coding. Section 3 provides an overview of existing text entry methods using braille. It describes the purpose and scope of each input method and analyzes each method's advantages and disadvantages. Section 4 describes the SingleTapBraille algorithms and the design of a single tap keyboard for Grade 1 and 2 braille. It also addresses the advantages of the proposed approach. Section 5 presents the preliminary evaluation regarding the performance of SingleTapBraille keyboard. Section 6 presents conclusion and explains possible future work.

2. Braille Code

Braille is a tactile writing and reading system used by people who have low or no vision. They use their fingertips to touch bumps or raised dots that represent characters. Braille machines are used to stamp bumps on paper to represent letters of the alphabet, numbers, and punctuation marks. The braille cell contains six dots, which are organized in two columns. These dots are numbered in a particular order according to their position in the braille cell from top to bottom and from left to right [7]. The first column has 1, 2 and 3 dots and the second column has 4,5 and 6 dots (as shown in Figure 1). Thus, there are 64 unique braille patterns codes. Since blind readers' needs vary, three different braille Grades were developed, which are Grade 1, 2 and 3. In this paper, we will only describe and implement Grade 1 and 2 because they are the most commonly used.

2.1. Grade 1

In Grade 1, Characters consist of some combination of these six dots. Each braille pattern in Grade 1 represents only one character. Grade 1 is used by novice blind users who are just starting to learn basic braille. Grade 1 is used to print some books and magazines, but these documents are very large because Grade 1 requires a fair amount of space. Figure 1 shows the braille alphabet where black dots indicate a bump position.

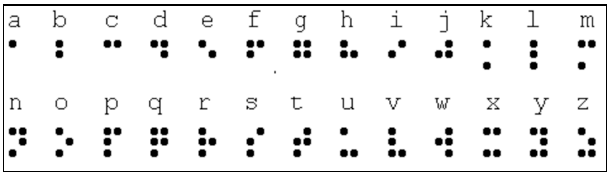


Fig. 1. Braille Alphabet

Fig. 2. Braille Numbers

The numbers 0-9 have the same shapes as the first ten letters of the alphabet in braille [8] (Figure 2). As such, the letter "a" sign represents number 1, the letter "b" sign represents number 2, and the letter sequence "a-j-c" represents the number 193.

2.2 Grade 2

Grade 2 braille is used as an alternative to Grade 1 braille to save space in large documents [16], and is used by people who are experts in braille. The main purpose of Grade 2 braille is to save space in printed braille documents because it allows for a shortened form of a word, for example letter "c" indicates the word "can". It is also faster to read and write using Grade 2 braille than Grade 1 braille, thus most printed materials like books and magazines are written with Grade 2 braille. There are many braille code combinations that have been used to represent the shortened form of common words [16].

1. Grade 2 represents a group of common letters in particular braille patterns, for example, it includes common suffixes or prefixes in individual braille signs (as shown in Table 1)

Table 1. Prefixes and suffixes forms in Grade 2

Words	code	Words	code	Words	code
ar		ing		ow	
dis		ed		sh	

2. It represents entire commonly used words into single braille character contractions (as shown in Table 2)

Table 2. Contraction and shorten words forms in Grade 2

Words	code	Words	code	Words	code
but	b	it	x	people	p
can	c	just	j	quite	q
do	d	knowledge	k	rather	r
every	e	like	l	for	□
from	f	more	m	so	s
go	g	not	n	still	□
have	h	this	□	that	t
in	□	out	□	will	w
us	u	you	y	the	□

3. It shortens words by removing most or all of the vowels in them (as shown in Table 3)

Table 3. Removing the vowels in words in Grade 2 braille

Words	Contractions	Words	Contractions
about	ab	after	af
above	abv	afternoon	afn
according	ac	afterward	afw
across	acr	again	ag
always	alw	braille	brl
blind	bl	deceive	dcv
either	ei	good	gd
friend	fr	great	grt

The abbreviations of words in Grade 2 braille are not restricted to semantics, and can be used for nouns or verbs [16]. For example, in Grade 2, the abbreviation of (can) is (c) and it can be typed as a verb or a noun. Sometimes, users should spell words to avoid confusion.

3. RELATED WORK

Apple and Android platforms have tried to improve the accessibility of touchscreen mobile devices by providing VoiceOver service in iOS devices and Talkback service in Android devices, respectively. However, typing text using the QWERTY keyboard with the accessibility service is difficult, especially when the user is holding the device and therefore typing with one hand [2]. The QWERTY keyboard has three layers including alphabet, number and punctuation layers, and requires the user to move his/her finger on a screen, with the VoiceOver service reading aloud the letter that is located below the user's finger. Thus, the user has to move his/her finger over a screen until he/she finds the intended letter. He/she also has to find a particular object location to switch between layers. The main advantage of the QWERTY keyboard is that it allows users to type and edit written text, with the VoiceOver service helping users ensure that they enter the correct text. However, this approach is extremely time consuming: the user has to go through all the letters and listen to them until he can find the desired letter [2]. It is also error-prone because the keyboard letters are located very close to each other, leaving only a tiny spot for each letter due to small screen size [2]. Other applications offer other accessibility tools, such as speech recognition, which enables people with vision loss to enter text quickly and easily; however, these tools are not preferable because most participants in a previous evaluation study [18] felt that it is awkward to speak to a device. another reason that makes voice recognition less preferable is that people can not speak out their private information in public [18] [19]. This situation might cause privacy challenges for blind users. In short, the standard keyboard on mobile devices is not satisfactory.

Many studies have proposed different nonvisual touchscreen text entry methods to overcome the challenges faced by people with low or no vision [9] [10] [11]. Previous work by Frey et al. [9] developed a BrailleTouch keyboard that allows blind people to use a multi-touch method to input text on a touchscreen based on braille patterns using two hands. The main strengths of this keyboard is that it is easy to understand and easy to learn. The main drawback is that it requires two hands, which makes it difficult for a user to hold the phone and type using another hand at the same time. It is very difficult to use two hands and three fingers to type while walking. To make it easier to interact with this proposed system, the user can hold the phone with the screen facing away from them in landscape mode, which allows people around the user to see what has been written on a screen. Besides the privacy issues that result from this, users are required to tap accurately on the location of braille dots because it uses fixed soft keys representing the six dots in a braille character. The braille touch can not be integrated with an application to be used for a specific goal like email because it requires several fingers to tap on the screen, eliminating the possibility of using part of a screen to enter text.

In 2012, Oliveira et al. [10] proposed a BrailleType keyboard using six large keys on a touchscreen to represent braille dots. It uses VoiceOver to help users tap the intended dots by speaking out the dot number when they touch a key. The main strengths of this keyboard is that it is easy to understand and easy to learn. However, the possibility of tapping the wrong dot is high because it is hard to find the exact location of each braille dot. In addition, the proposed keyboard cannot be integrated with an application like email because there is not enough space on a screen.

Azenkot [11] developed the Perkinput keyboard system, which allows users to touch a screen using three fingers, each finger representing a dot in braille. On a small screen, the user has to tap twice using three fingers in each tap to represent a letter. On a tablet, users can use two hands and three fingers from each hand to input a letter in one step. The most significant advantages of the Perkinput keyboard is that it overcomes navigation problems by removing the fixed soft keys on the touchscreen, such that it does not require the users to tap on a particular object location. It is also easy to learn and use because it is based mainly on braille coding. One of the major disadvantages of this keyboard is that a user must register reference points anywhere on the screen that are used to detect which fingers were used to perform each touch. A user should set his/her finger with a 3-point "long press" to create the reference points. However, users might forget the location of the reference points, leading to a high uncorrected error rate. In addition, it can not be integrated for a specific goal like email because it requires several fingers to tap on the screen, eliminating the possibility of using part of the screen to enter text.

EdgeBraille is yet another accessibility keyboard, developed by Mattheiss et al. [12]. The keyboard represents the braille dots 1, 4, 5 and 6 in the corners of the screen and dots 2 and 3 on the long edges of the screen, and requires users to swipe one finger along the screen's edges to activate braille dots. Users have to move their finger to draw a continuous line on braille dots in order to represent a letter. The author also developed another version of EdgeBraille that permits users to enter eight dots for braille. The two added points allow users to perform other functions like deleting and searching. This keyboard has several advantages including providing vibrio-tactile feedback and using sound files to inform the users about the activation and deactivation of braille dots. In addition, it uses VoiceOver to read out the written text. The main strength of the

EdgeBraille is that it uses the screen's edges and corners as the locations for braille dots, making the dots easy for the blind user to find. In terms of weaknesses, EdgeBraille is not highly accurate and it is not easy to learn because of its drawing mechanism: some of the braille dots are separate, requiring users to draw a line to connect them. This might confuse user, and it requires that the users memorize other shapes of braille. It is also slow because it provides a threshold of 75millisconds for lifting the finger from each activated dot. It allows only for entering letters and thus does not enable users to enter numbers, symbols and punctuation. It does not provide control characters for editing tasks, such as delete, enter, and backspace. As with other existing methods, it does not integrate the proposed keyboard with applications.

Mascetti et al., have addressed the editing limitation for visually impaired people by developing the TypeInBraille Keyboard, which requires users to enter braille letters using gestures [13]. Users enter the three rows of each braille pattern from the top to the bottom. The first gesture represents the 1st and 4th dots, the second gesture represents the 2nd and 5th dots, and the last gesture represents the 3rd and 6th dots. This input approach splits the touchscreen into left and right portions, and operates using four gestures: A tap on the left part of the screen to activate the left dot and inactivate the right dot in a row, a tap on the right part of the screen to activate the right dot in the first row and inactivate the left dot in a row, a tap with two fingers to activate both the left and right dots, a triple tap to inactivate both left and right dots in a row. It also uses other defined gestures to allow users to select numbers, move the cursor, and perform typing and editing operations, such as delete, undo, and capitalization. The most important advantage of this keyboard is that it supports both typing and editing tasks. Furthermore, it supports voice feedback, though users can use this keyboard without using voice feedback in order to protect private information. However, it is not easy to understand and learn because it is difficult to memorize all the defined gestures in order to perform a task. A major weakness is that users are required to switch between writing, exploration and selection modes, which can be time-consuming for visually impaired users. In addition, it requires users to use two hands to make some of the gestures that require three fingers. Users cannot type while moving because it is difficult to hold a mobile phone and a guide dog or a cane at the same time. Lastly, it does not integrate the proposed keyboard with applications such as email.

Similar to TypeInBraille is the Eyedroid keyboard, which uses five defined gesture patterns on the touchscreen to enable visually impaired people to enter text based on the braille patterns [14]. The registered gestures used to enter braille dots on the touchscreen are: Swipe left-to-right to activate the left dot and inactivate the right dot, swipe right-to-left to activate the right dot and inactivate the left dot, tap the screen with one finger to inactivate both dots, and swipe from bottom to top to activate one dot. This keyboard has several advantages. First, it overcomes the problem of navigation because it does not require the user to find an object in a specific location. It also uses voice feedback to assist users in interacting with the keyboard. However, it is not easy to understand because it requires users to memorize the defined gesture functions. Another limitation of the Eyedroid keyboard is that it does not support editing or typing capital and small letters.

Most of the existing braille keyboards only support Grade 1 and not Grade 2, except the AccessBraille keyboard [17], which uses a multi-touch approach and presents braille dots as six columns on a touchscreen to allow blind users to use both hands to type on a small screen. This keyboard has the same drawbacks as the BrailleTouch keyboard [9].

4. The SingleTapBraille Text Entry Method

The proposed research seeks to overcome the limitations of existing input methods based on braille. Our main goal is to develop a new text entry method that eliminates the requirement that users find specific locations in order to input letters, allowing information to be entered in a way that is fast and effective using a single thumb or finger. This is in accordance with Paisios [6], who found that using one finger to interact with a touchscreen is the best, most accessible way for blind users.

Similarly, a previous study on travelers conducted by Karlson, Bederson, and Contreras-Vidal (2004) [15] [21] found that they usually hold bags with one hand and hold the mobile phone using the other hand. This situation is similar to the usual scenario for blind and visually impaired people, as they hold a cane or a guide dog with one hand and use the other hand to interact with a mobile phone. The main objective of the present study is to identify the best features for users who are restricted to using only one hand to perform tasks on their mobile device. A preliminary finding of the study was that the majority of users would prefer to use the thumb to interact with touchscreens. The description of our input method is described in the following section in greater detail.

4.1 Typing and editing on a touchscreen

The system developed based on braille patterns, each with two columns and three rows. Our input method allows users to enter dots from left to right, which is generally the way blind users read braille. The first tap means a dot in the left column. Each braille dot is activated individually using a single tap on a touchscreen. For example, if the user taps on any part of the screen once, this will represent the letter "a" and it means the user activated the first dot in braille. When the user stops touching the screen for a short interval, the braille pattern is interpreted and the resulting letter is typed and the software reads out the letter. With SingleTapBraille, users can hold the mobile phone with one hand because they need only one finger or their thumb to enter braille dots on the touchscreen.

4.1.1 Grade 1

In our approach, we analyze the relationship between active dots in each braille character based on their coordinates and the distance between each dot and the following one(s). For example, if the user taps two taps below each other on a screen, this will represent the letter "b" and it means the user activated the first dot and second dot in braille. The software will analyze the relationship between these two taps and based on our algorithm will identify the intended letter, number or punctuation mark, and show it on the screen. The main concept behinds our new input method is the clustering of activated braille dots for each character. When the user stops touching the screen for a short interval, our algorithm clusters the entered taps and interprets them based on the number of dots. The algorithm processes each dot's coordinates, as well as the space between each dot and the following one. The algorithms that have been used to analyze the relationship between dots are described in Figure 3 and Figure 4.

The factors that indicate the relationship between braille dots are:

1. The value of coordinates (X, Y) for each dot.
2. The distance between dots, D, which means a defined value representing the distance between two dots.
3. The number of dots in each character.

The error parameter is a specific value that is used as a threshold for whether two floating point numbers are in a vertical or horizontal line. If the difference between x values of two taps is smaller than the error value, the algorithm will accept it because it's very likely that the user's taps are in the same vertical line.

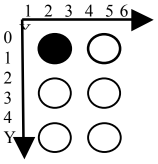
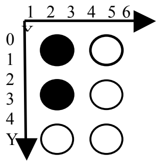
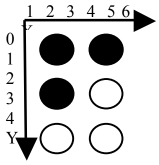
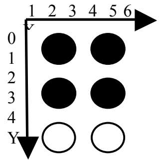
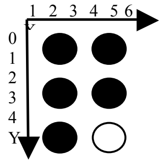
Character has one dot	Characters have two dots	Characters have three dots	Characters have four dots	Character has five dots
Letter a, number 1  One tap anywhere in a mobile screen	Letter b, number 2  |X1-X2|<error; D <= a specific value	Letter f, number 6  |X1-X2|<error; |Y1-Y3|<error; D <= a value	Letter g, number 7  |X1-X2|<error; |X3-X4|< error ; X2<X3; |Y1-Y3|< error; |Y2-Y4|<error ; Y3<Y2, D <= a value	Letter q  |X1-X2|<error; |X3-X2|< error; |X4-X5|< error; |Y1-Y4|<error; |Y2-Y5|< error; D <= a specific value

Fig. 3. Sample of categorization of braille characters based on number of dots

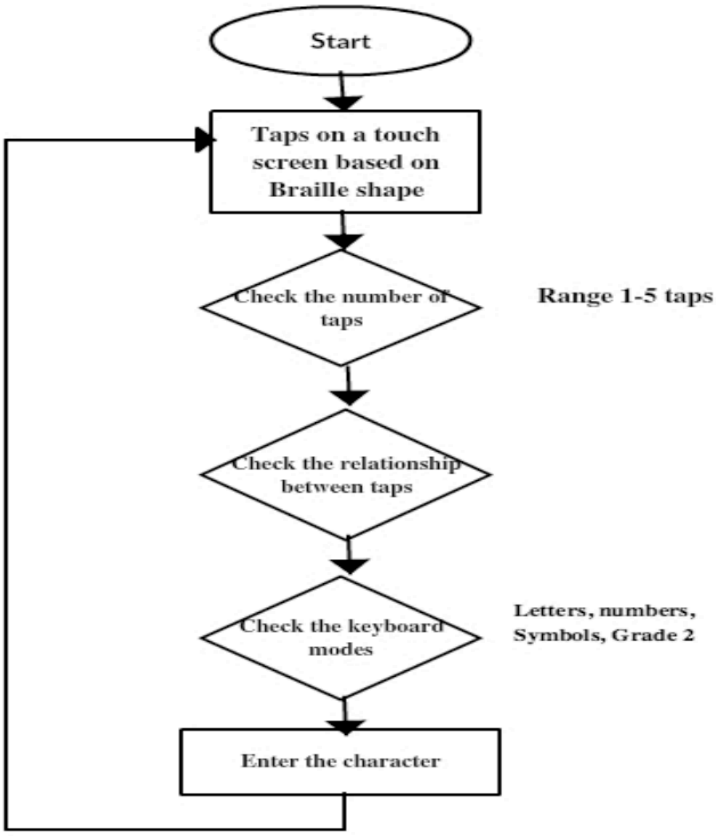


Fig. 4. Algorithm flowchart

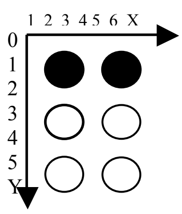
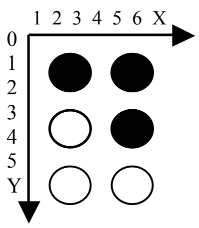
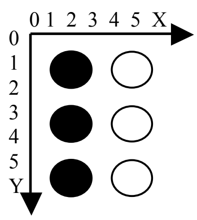
can	do	knowledge
Letter c  |Y1-Y2|<error	Letter d  |Y1-Y2|<error; |X2-X3|<error	Letter K  |X1-X2|<error; D > gap

Fig.5. Example of whole words abbreviations in Grade 2

As seen in Figures 3 and 5, each braille dot location is identified by both an x and y coordinate. Our algorithm used these coordinates to generate an equation that reflects the relationship between the the dot locations in each braille code. These equations are represented below each braille code in Figures 3 and 5. Once the pattern of braille code is recognized, the algorithm will present the corresponding number.

4.1.2 Grade 2

The main purpose for implementing Grade 2 braille is to improve the keyboard speed. As indicated, when using Grade 2 braille, contractions and abbreviations can be typed instead of whole words.

The implementation of Grade 2 follows the same approach of Grade 1 algorithm. The algorithm will analyze the braille sign of each contraction based on number of dots and the relationship between these dots in a single braille sign.

Users select the Grade 2 braille mode and they type the braille sign of a word that they want to type. Our algorithm will check which word this braille sign indicates and will spell out the complete word (as shown Figure 5). At the same time, our application's speak function will speak out the intended word. For example, if a user inserts the letter k braille sign in Grade 2 mode, our keyboard will type out "knowledge" and it will speak out "knowledge".

However, the algorithm follows other steps to type words that their abbreviations have two letters or more (see Table 3). Users tap the letters' braille codes and after inserting the abbreviation, our algorithm will find out the corresponding words and insert the complete words. At the same time, the speak function will read out the interpreted word. For example, a user types letter "a" and "b" and then adds space, the algorithm will check a corresponding word for the abbreviation "ab" and it will insert and speak "about".

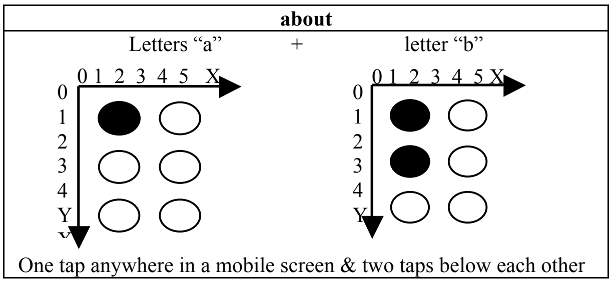


Fig. 6. Sample of typing shorten words forms in Grade 2

Typing steps for Grade 2 abbreviations that contain two letters or more in the SingleTapBraille:

1. Typing the letters based on the braille shape of each letter (as shown Figure 6).
2. Finding out the corresponding word
3. Replacing the abbreviation with the entire words
4. Reading out the inserted word

4.2 Algorithm advantages

The main advantage of our algorithm is that it does not restrict users by requiring them to find a specific object location on a screen, instead allowing them to tap the braille dots anywhere on the screen based on braille patterns. It does not have a button or other type of control that requires the user to precisely tap or click specific spots; it is gesture driven and has a corresponding voice output. Users do not have to tap on a specific location to activate a particular braille dot. After the user taps using one finger (or thumb) to activate braille dots, the software will analyze the relationship between those dots and represent the interpreted characters. The application uses audio feedback after a short interval, allowing the user to confirm that he/she entered the intended letter and understands the gestures being performed. As a result of letting users confirm what they are doing, the error rate will be significantly reduced. Furthermore, users can use one hand to hold the mobile phone and tap on the screen using only a thumb. Users can also operate our input method on a tablet using one hand to hold the tablet and another hand to enter text. Importantly, our application can be used both by stationary people and those on the move.

There are some typing operations, such as space and backspace, that the braille keyboard does not present using braille dots, because they have specific buttons on the braille machine that perform these functions. The SingleTapBraille method uses an assortment of flicking motions on the mobile screen in order to carry out certain functions, as detailed in Table 4.

The main advantage of this approach is that when the user performs a gesture, the software will speak out the gesture function's name. The SingleTapBraille method activates the text-to-speech function to speak out the registered interaction gestures that are used to facilitate typing, as shown in Table 4.

In order to change the keyboard mode, users can swipe vertically in a loop to activate a particular mode including lower case, upper case, numbers, symbols and Grade 2 modes. The first swipe will activate the first keyboard mode and TalkBack service will read out the keyboard mode. Then, if the users swipe vertically twice, they will obtain the second keyboard mode. For example, when the user performs the first vertically swipe from bottom to top, the app will speak "Upper case". In case the user does not want the upper case mode, he/she can swipe again vertically to select the lower case mode instead. Performing a vertical swipe from top to bottom will open the symbols mode. A second swipe from top to bottom will open the Grade 2 braille mode. They can also swipe horizontally to edit their typing (see Table 4).

Table 4. Braille keyboard interaction techniques and their purposes

Interaction techniques	purposes
Swipe from bottom to top	Upper case, number mode
Swipe from top to bottom	Lower case, symbols mode
Swipe from left to right	Add space
Swipe from right to left	Backspace
Swipe from top to bottom	Grade 2 braille

In braille, there are no special characters that represent capital or small letters [8]. In order to enter a capital letter using braille, the user should activate dot number 6 in the pattern just before the letter that is meant to be capitalized [8]. None of the existing input methods have been able to help blind and visually impaired individuals type capital letters. In our method, users can simply swipe their finger from top to bottom to type capital letters. When the user swipes from top to bottom, the software will speak out the gesture function. The software also begins any sentence with a capital letter and then automatically shifts back to small letters. Swiping from the top down will allow the user to write a small letter.

As mentioned above, numbers follow the same order and have the same shapes as the first ten letters of the alphabet in braille [8]. Consequently, the letter "a" sign represents number 1, letter "b" sign represents number 2, and letters "a-j-c" signs represents the number 193.

Using our input method, users can perform a defined gesture (a swipe from top to bottom) in order to enter numbers on a touchscreen.
Grade 2 braille also follows the same braille patterns order and shapes. A single braille sign in Grade 2 indicates a word. Users can insert a braille sign and the algorithm will replace it with the matched word.

4.3 Screen Layout

The SingleTapBraille keyboard is designed for users who know braille and can read and write using braille, and also have experience with touchscreen devices. Specifically, this keyboard is intended to improve the speed of typing and editing on a mobile screen while minimizing the error rate.
The SingleTapBraille application has two fundamental purposes: typing, and editing. The application interface is entirely speech-based and it has an edit text on the top of the screen to show the written text. It displays written text for people who have low vision, but it does not provide other visual feedback.

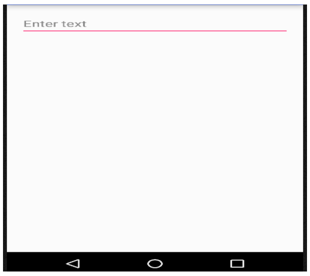


Fig. 7. SingleTapBraille user interface

5. Preliminary evaluation

5.1 Evaluation of the first version the SingleTapBraille

To understand the end users' opinions about the developed keyboard and its interaction techniques, we first conducted a preliminary evaluation to examine the performance of the keyboard that has only Grade 1 braille (our first prototype).
The performance of the SingleTapBraille keyboard was evaluated by three participants: two teachers who train blind users how to write and read braille, and one blind user who is an expert in using touchscreen technology and fluent in braille. The participants ranged in age from 40 to 52 years old. Following an introduction to the keyboard and how to use it, the participants were asked to use the keyboard to enter the entire alphabet, as well as the numbers 0 to 10. At the end of the experiment, each participant had an interview about their experience and some suggestion for improvements.
Evaluation by the participants suggested SingleTapBraille to be a promising text entry tool that might encourage a new generation to learn braille and avoid dependency on voice for interacting with touchscreens. The most consistent feedback was to improve the algorithm's accuracy for some letters in order to reduce the typing error rate; for example, there was difficulty distinguishing between 'r' and 'v', which are similar to each other The third participant was a blind person who is an expert in using touchscreen technology and fluent in braille. He learned how to operate the SingleTapBraille keyboard in only a few minutes. He liked the idea of eliminating the need of locating objects on touchscreens. He also found using specific gestures to perform some tasks (e.g., switching between the number, letter, and punctuation keyboards) helpful. The blind user encouraged us to integrate our proposed keyboard in an SMS application, which is a common tool for quick communication.

In the interview with the braille teachers and blind user, we learned that there are different ways that people with vision impairments can enter letters based on braille. Some of them who used the slate (a solid piece with slight depressions spaced in representing six braille dots) and stylus device prefer to use one hand to enter text. The depressions in the slate are used to assist users to create a dot in the correct position. Other blind people use two hands, as with the Perkins braille device. They usually used the right hand to enter the first column in braille and the second hand to enter the second column. In addition, the braille teachers and blind user stated that most people who are born blind prefer to use their index finger to read and interact with a touch screen.

5.2 Evaluation of Grade 2 encoding in SingleTapBraille

To address the main limitation of the first prototype, we implemented the second type of braille (Grade 2) to develop a complete prototype and we evaluated the second version of the keyboard to obtain the participants' feedback about the feature that has been added, and also to guide the final evaluation study design.
After implementing Grade 2 braille, the improved keyboard was evaluated with a blind participant to determine whether the implementation of Grade 2 braille improved the accessibility and overall performance of the keyboard. During the evaluation, we explained the improved SingleTapBraille and how can a user open the Grade 2 braille and insert words. Then, we asked the blind participant to type several words for 5 minutes as a training session. The participant was then required to type specific words that have abbreviations or signs in Grade 2 braille. Some of these words can be represented by only the first letter of each word, for example, "can", "you", "do", "like", "people", and "rather". Other words can be represented using the first two letters in each word like "about", "after", "also", "again", "blind" and "according". After typing these words, we conducted an interview with the participant about the strengths and limitations of the improved keyboard. He stated that this keyboard is a very useful text entry tool. The most valuable features in this keyboard after integrating Grade 2 braille is enabling a user to type words in a short time using Grade 2 abbreviations and without a need to insert each letter in a word. He said "I like it because it is much quicker than spelling the word right out because it is a shortcut for us". He also liked that the keyboard provides both Grade 1 and 2 levels. He stated "I like the way it allows us to have both options Grade 1 and 2 by swiping. It is much easier and we are not guessing where we are swiping and typing braille dots." He also stated that "I could use Grade 1 to type but I prefer Grade 2 braille because it is much faster for me than the other one."
He also likes the way that the keyboard speaks and types out the whole word. In Grade 2 braille, only abbreviations of words can be typed, which might cause confusion. In addition, the participant pointed to a limitation with the keyboard after adding Grade 2. He found that in case the user types a wrong abbreviation, the keyboard will insert the whole word and the user has to delete the entire word letter by letter. Thus he suggested we allow the users in Grade 2 braille to delete entire words and not each letter. Implementing this suggestion will definitely increase the speed of the keyboard.
In general, the results of the initial evaluation of the improved keyboard confirm that the single tap approach is a promising direction for further exploration and development of an accessible touchscreen keyboard for the blind users.
This SingleTapBraille concept can be implemented in any mobile platform, including iOS, Android, and Windows phones. The components of designing the user interface of this approach are available across all platform versions. In the

implemented approach, we used only essential gestures to create a simple interface including tap and swipe gestures that are easy to implement in other platforms. This approach does not require any element that is restricted to any specific platform release, and can be easily implemented across platform versions.

6. Conclusion and future work

This paper introduced an accessible braille keyboard that enables blind people to enter text on touchscreens more easily than with existing alternatives. This approach addresses the key difficulties that blind people face when they enter text, including their inability to locate an exact object position on a touchscreen. Instead, users of this keyboard can call forth letters and other symbols via repetition of a single tap on the touch screen, making the use of multiple fingers and the location of specific spots unnecessary. Because the evaluation is not the major contribution in this paper, we will evaluate SingleTapBraille keyboard with blind users via both qualitative and quantities methods.

We are presently designing a full study to examine SingleTapBraille using quantitative and qualitative methods. The main objective of this research will be to test SingleTapBraille's accessibility, learnability and usability. This evaluation study will also aim to determine the strengths and limitations of our input method, and to to obtain users' feedback regarding our keyboard. In addition, we are planning to assess the efficiency of our input method by measuring total typing time, characters and words per minute, as well as evaluating the effectiveness of our improved input method by measuring the number and types of errors. We are also planning to compare our keyboard speed with the AccessBraille keyboard that supports Grade 2 braille.

Acknowledgements

We thank the Canadian National Institute for Blind, and especially the study volunteers. We also gratefully acknowledge support from Taif university and the Saudi Arabian Cultural Bureau in Canada.

References

[1] Kane S., Jayant C., Wobbrock J., Ladner, R. 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, 2009;115–122. https://doi.org/10.1145/1639642.1639663

[2] Nicolau H, Guerreiro T, Jorge J, Gon D. Proficient blind users and mobile text-entry. In Proc. of the 28th Annual European Conference on Cognitive Ergonomics, ECCE '10, New York, NY, USA, ACM, 2010;19–22. https://doi.org/10.1145/1962300.1962307

[3] Ladner R, Kane S, Wobbrock J. Usable gestures for blind people: understanding preference and performance. In Proceedings of the 2011 annual conference on Human factors in computing systems. ACM, 2011.

[4] Caute A, Woolf C. Using voice recognition software to improve communicative writing and social participation in an individual with severe acquired dysgraphia: An experimental single-case therapy study. *Aphasiology, 30*(2-3), 2016; 245-268.

[5] Bits. Disruptions: Visually Impaired Turn to Smartphones to See Their World. [serial online] 2013 Jan 5 [cited 2016 Feb14]; Available from: URL: http://bits.blogs.nytimes.com/2013/09/29/disruptions-guided-by-touch-screens-blind-turn-to-smartphones-for-sight/?_r=0

[6] Paisios N. Mobile accessibility tools for the visually impaired. *PHD Thesis*. 2012, Available from: URL: http://cs.nyu.edu/web/Research/Theses /nektariosp.pdf (retrieved online September 19, 2012).

[7] CNIB. About the Braille System. [serial online] 2001 Mar; Available from: URL: Retrieved from http://www.cnib.ca/en/living/braille/ braille-system/Pages/default.aspx

[8] Pierce B. Braille--what is it? What does it mean to the blind? *The World Under My Fingers: Personal Reflections on Braille*. [serial online] 1995 [cited 2016 Mar 14]; 15(1), Available from: URL: Retrieved from https://nfb.org/images/nfb/publications/fr/fr15/issue1

[9] Frey B., Southern C, Romero M. BrailleTouch: Mobile Texting for the Visually impaired. *Proc. HCII '11*. Springer, 2011;19-25. https://doi.org/10.1007/978-3-642-21666-4_3

[10] Oliveira J, Guerreiro T, Nicolau H, Jorge J, Gonçalves D. BrailleType: Unleashing Braille over touch screen mobile phones. *INTERACT* '11. Heidelberg: Springer, 2011 100-107.

[11] Azenkot S, Ladner R, Wobbrock J, Borning A, Landay J, Gina-Anne L. Eyes-Free Input on Mobile Devices, *ProQuest Dissertations and Theses*. 2014.

[12] Mattheiss E, Georg R, Johann S, Markus G, Schrammel J, Garschall M, Tscheligi M. EdgeBraille: Braille-based text input for touch devicesnull. *Journal of Assistive Technologies*, *9*(3), 2015; 147–158. https://doi.org/10.1108/JAT-10-2014-0028

[13] Mascetti S, Bernareggi C, Belotti M. TypeInBraille: A Braille-based typing application for touchscreen devices. *In: Proc. ASSETS 2011*, 2011;295–296. https://doi.org/10.1145/2049536.2049614

[14] Shabnam M, Govindarajan S. Braille-Coded Gesture Patterns for Touch- Screens: A Character Input Method for Differently Enabled Persons using Mobile Devices. *International Conference on Communication, Computing and Information Technology (ICCCMIT-2014).*

[15] Karlson K, Bederson B, Contreras-Vidal L, 2006. Understanding Single- handed Mobile Device Interaction. University of Maryland. HCIL-2006-02. Martin, G.L., 1988. Configuring a numeric keypad for a touch screen. Ergonomics 31, 945–953.

[16] Royal National Institute of Blind People. (2016, February 27). Contracted (Grade 2) braille explained. Retrieved fromhttp://www.rnib.org.uk/braille-andmoon%E2%80 %93- tactile -codes-braille-codes/contracted-grade-2 braille- explained

[17] Ludi S, Timbrook M, Chester P. 2014. AccessBraille: tablet-based braille entry. In *Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility* (ASSETS '14). ACM, New York, NY, USA, 341-342. https://doi.org/10.1145/2661334.2661421

[18] Alnfiai M, Sampalli S. (2016). An Evaluation of SingleTapBraille Keyboard: A Text Entry Method that Utilizes Braille Patterns on Touchscreen Devices. In Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '16). ACM, New York, NY, USA, 161-169. http://doi.org/10.1145/2982142.2982161

[19] Yuan-Fu Shao, Masatoshi Chang-Ogimoto, Reinhard Pointner, Yu-Chih Lin, Chen-Ting Wu, and Mike Chen. 2016. SwipeKey: a swipe-based keyboard design for smartwatches. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services* (MobileHCI '16). ACM, New York, NY, USA, 60-71. DOI: http://doi.org/10.1145/2935334.2935336

[20] Alnfiai, M., Sampalli, S. (2016). SingleTapBraille: Developing a text entry method based on braille patterns using a single tap. *The 11th International Conference on Future Networks and Communications (FNC 2016)*, *Procedia, Computer Science,* 248-255. ISSN 1877- 0509, http://doi.org/10.1016/j.procs.2016.08.038

[21] Strohrmann C, Seiter J, Tröster G. (2014). Feedback Provision on Running Technique with a Smartphone. *Journal of Ubiquitous Systems & Pervasive Networks. Volume 5, No. 1 (2014) pp. 25-31.*