Hybrid-Brailler: Combining Physical and Gestural Interaction for Mobile Braille Input and Editing

Daniel Trindade[1], André Rodrigues[2], Tiago Guerreiro[2], Hugo Nicolau[1]
[1]INESC-ID, Instituto Superior Técnico, Universidade de Lisboa
[2]LASIGE, Faculdade de Ciências, Universidade de Lisboa
daniel.trindade@tecnico.ulisboa.pt, afrodrigues@fc.ul.pt, tjvg@di.fc.ul.pt, hman@inesc-id.pt

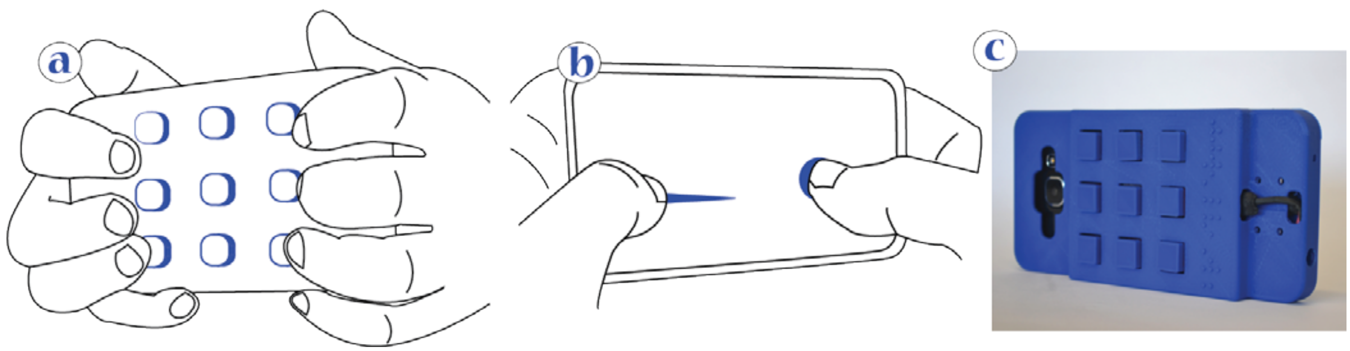

Figure 1. Hybrid-Brailler, a Braille input and editing system that combines physical and gestural interaction. (a) The system consists of 9 physical buttons used for Braille input. Notice that the 3x3 layout enables usage on both portrait and landscape modes. (b) Representation of editing gestures using bimanual interaction. Directional gestures move the caret while touching with a second thumb starts text selection. (c) 3D-printed case that encloses all hardware and can be attached to mobile devices.

ABSTRACT
Braille input enables fast nonvisual entry speeds on mobile touchscreen devices. Yet, the lack of tactile cues commonly results in typing errors, which are hard to correct. We propose Hybrid-Brailler, an input solution that combines physical and gestural interaction to provide fast and accurate Braille input. We use the back of the device for physical chorded input while freeing the touchscreen for gestural interaction. Gestures are used in editing operations, such as caret movement, text selection, and clipboard control, enhancing the overall text entry experience. We conducted two user studies to assess both input and editing performance. Results show that Hybrid-Brailler supports fast entry rates as its virtual counterpart, while significantly increasing input accuracy. Regarding editing performance, when compared with the mainstream technique, Hybrid-Brailler shows performance benefits of 21% in speed and increased editing accuracy. We finish with lessons learned for designing future nonvisual input and editing techniques.

Author Keywords
Blind; Braille; Mobile; Text Entry; Editing; Touchscreen.

CHI 2018, April 21–26, 2018, Montreal, QC, Canada

ACM ISBN 978-1-4503-5620-6/18/04...$15.00
https://doi.org/10.1145/3173574.3173601

ACM Classification Keywords
H.5.2. Information interfaces and presentation (e.g., HCI): User Interfaces - Input devices and strategies (e.g., mouse, touchscreen); K.4. Computers and Society: Social Issues – Assistive technologies for persons with disabilities.

INTRODUCTION
Mobile touchscreen devices such as smartphones have been widely adopted by blind people [1]. While these devices support text input through accessibility services, providing fast text entry remains a challenge [2,19,25]. The lack of tactile and kinesthetic feedback on a touchscreen forces blind users to drag their finger across a flat keyboard, while receiving auditory feedback, to find the intended key, resulting in a slow typing process [19].

To address this issue, multitouch Braille chording approaches have been proposed [2,15,25]. These are highly effective in improving input speed; however, they are characterized by a decrease in typing accuracy when compared to traditional touch-based input. The difference is more evident when comparing physical and virtual Braille keyboards, where error rates can double [25]. Despite these challenges, touch-based keyboards offer a major advantage over their physical counterparts: they are software-based, thus, offering greater input expressiveness.

In this paper, we present Hybrid-Brailler, an input solution that combines physical and gestural interaction to provide fast, accurate, and flexible nonvisual input. We use the back of the device for physical chorded input while freeing the touchscreen for gestural interaction. Such multimodal approach opens design opportunities for novel input tasks that have been largely unexplored in mobile research, especially for subpopulations alike blind people. One such task is text editing. We leverage Hybrid-Brailler in editing operations and provide a gesture set for caret movement, text selection, and clipboard control.

To evaluate Hybrid-Brailler, we conducted two user studies. The first focused on comparing Hybrid-Brailler with a touch-based and a physical Braille input method. Results show that Hybrid-Brailler was significantly more accurate than its virtual counterpart while maintaining entry speed. The second user study compared non-visual text editing performance between Hybrid-Brailler and Android's accessibility service. Overall, results show that nonvisual editing tasks are highly demanding. Still, Hybrid-Brailler was significantly faster and more accurate than Android's Explore-by-Touch. Caret movement operations were the easiest to perform with average success rate of 85% with Hybrid-Brailler. Clipboard tasks were the hardest to complete, mostly due to inaccurate placement of the cursor. Participants had additional issues with Explore-by-Touch in selecting the intended clipboard operation.

The contribution of this paper is three-fold. First, the design and implementation of a novel Braille input method that combines physical and gestural interfaces. Second, an evaluation of text entry performance, which compares three Braille-based input methods. Finally, the first empirical assessment of text editing performance using Explore-by-Touch and a comparison with our proposed solution, which demonstrate the challenges that blind users face in these tasks and the potential of Hybrid-Brailler for text editing as well text entry. We finish with lessons learned that should be of interest to keyboard designers that aim to combine physical and gestural interaction for nonvisual text input.

RELATED WORK

In this section, we discuss previous work in three topics: first, we analyze related research on physical text input for blind people and the (dis)advantages of such methods. Second, we discuss touchscreen input research focusing on Braille-based and alternative text entry methods. Finally, we describe previous attempts to improve text editing operations and the existing gap in non-visual techniques.

Physical Text Input for Blind Users

Before the advent of mobile touchscreen devices, physical keypads were the *de facto* hardware. Most users resorted to the 12-button keypad using multi-tap or T9's predictive text input [6]. Alternatively, there were chording keyboards. Twiddler, an eyes-free one-handed mobile keyboard, was probably the most successful, enabling entry rates much faster than multi-tap [12].

Braille-based physical keyboards are still in use today whether through the standard Perkins Brailler[1] or the more recent electronic Braille note takers. A traditional Braille keyboard consists of 7 main keys paired with each of the six dots of the Braille code and a space key. To input text, users simultaneously press the intended set of keys, entering a character as a chord. Braille is a powerful language, making it possible to represent alphabet letters, accentuated letters, punctuation, numbers, mathematical symbols, or even musical notes. These keyboards are consistently faster and more accurate than their touch-based counterparts [20,25].

However, these assistive technologies are bulky and lack portability, forcing blind users to miss on the opportunities provided by current smartphones. Hybrid-Brailler aims to fill this gap by integrating and combining physical Braille keyboards and touchscreen devices.

Touchscreen Mobile Input for Blind Users

Today's touchscreen devices support non-visual text input via built-in screen readers such as VoiceOver[2] and Talkback[3]. They enable users to explore the keyboard by dragging their finger and have the keys read aloud as they touch them. While the visual layout of the QWERTY keyboard is identical to that presented to sighted users, entry rates are slower for visually impaired users [19]. To address this problem several non-visual text entry methods for touchscreen devices have been proposed [8,16,21,25].

Yfantidis and Evreinov [29] proposed one of the first touchscreen input methods for blind people, which consists of a pie menu with eight options and three levels. Users can select letters by performing directional gestures and dwelling over characters. NavTouch [9] also uses a gestural approach to navigate through the alphabet using four directions. Vertical navigation is used to decrease navigation time using vowels as shortcuts to the intended letter. No-Look Notes [3] is a text entry method with large virtual keys that uses an alphabetical character grouping identical to multi-tap approaches. Users need to select two keys to enter one character.

In the last decade, there have been several attempts to bring Braille input to touch-based devices. BrailleType [21] splits the screen in six areas representing the Braille cell. Users enter characters by selecting the regions that correspond to the raised dots in the character. Although accurate, this is a slow input method. TypeInBraille [15] is another entry method that uses gestures to input characters based on its Braille encoding. BrailleTouch [25] introduced multitouch Braille chording where the screen faces away from the user while holding with both hands in a landscape orientation. The method allows for fast entry rates, around 25 words per minute but is highly error prone. Similarly, Perkinput [2] uses a multitouch approach but improves accuracy by proposing a novel finger tracking technique. More recently,

[1] http://www.perkinsproducts.org/about-perkins-braillers

[2] https://www.apple.com/accessibility/iphone/vision/

[3] https://support.google.com/accessibility/android/answer/6006598

Nicolau et al. proposed B# [17], a spellchecker for multitouch Braille input that leverages chording information to improve input accuracy.

Overall, since the advent of touchscreen mobile devices, numerous input methods have been proposed for blind people. Braille chording approaches have been particularly effective in improving typing speed at the cost of accuracy. Hybrid-Brailler differs from previous research by augmenting current mobile devices with an integrated chording physical keyboard and freeing the touchscreen for simultaneous gestural interaction.

Mobile Text Editing for Blind Users

Along with text entry, built-in accessibility services enable common editing operations such as caret positioning and movement, text selection, and clipboard operations (i.e. copy, cut, and paste). This is achieved through contextual menus and gestural interaction. For example, with iOS, users move the caret by performing vertical gestures. They can adjust the navigation granularity (character, word, line) through the 'rotor' menu, which is a two-finger rotational gesture. Clipboard and selection operations are also accessible via the 'rotor' menu. Android's Talkback has a similar approach but the contextual menu is opened via the inverted 'L' gesture. Caret movement can be achieved by selecting movement granularity (up gesture) and pressing the volume up/down or through horizontal gestures.

Although editing operations are virtually accessible, previous research has shown that blind people face several issues when attempting to perform them. The severity of these issues is demonstrated by some users that prefer to clear the entire text field and rewrite the original text rather than edit it [1,4]. Azenkot and Lee [1] conducted an observational study on how blind people used speech input and an on-screen keyboard. Although users were 5 times faster using speech, they spent 80% of their time on editing operations. The experience was described as "frustrating". Caret movement, text selection, and error detection were some of the most troublesome operations.

Some authors have proposed alternative editing techniques for sighted people [5,7,22,26], usually through gestural interaction. However, these do not directly apply to blind users, since proposed interfaces are visually demanding and gesture input ability differs significantly from that of sighted people [10].

Although text entry has been an active field in the past decade, interaction techniques for text editing have been neglected, particularly nonvisual text editing techniques. Hybrid-Brailler leverages the expressiveness of touch interaction to enable multiple editing operations, while using a physical keyboard for text entry. Also, we are the first to report a quantitative evaluation with blind users on nonvisual text editing.

COMBINING PHYSICAL AND TOUCH INTERACTION

The main goal of this work is the development of a mobile Braille text entry system that combines the advantages of physical and touchscreen interaction. Our design augments the back of the device with a physical keyboard while freeing the touchscreen for gestural interaction. Thus, users can accurately type on the back of the device and use the touchscreen to perform editing operations. In this section, we present our design concept for both text input and editing, as well as the built prototype in reproducible detail.

Text-Input Design

A Braille character is represented by combinations of dots on a 3 by 2 matrix. A traditional Braille keyboard consists of 7 main keys displayed horizontally paired with each of the six dots of the Braille code and a space key. To input text, users simultaneously press the intended set of keys entering a character as a chord.

The design of Hybrid-Brailler draws inspiration from the traditional Braille writing system; however, rather than using the touchscreen for multitouch text input [2,25], the back of the device features a physical Braille keyboard.

We first prototyped and preliminary tested several possible configurations before building a functional keyboard (Figure 1-c). We were interested in exploring button sizes, layouts, and whether users would prefer to type in portrait or landscape orientation. Initial results did not show a clear preference for either mode. Some users mentioned that it could vary depending on the situation. For example, for quick input they would use the device in portrait orientation and for data intensive tasks they would turn the device for a more comfortable landscape orientation.

Thus, in our final design, we added three additional buttons to the traditional Braille cell, resulting in a 3 x 3 grid. This design has the advantage of being usable either in portrait or landscape orientations. Users can type by operating the 6 edge buttons while the remaining 3 middle buttons can be used as function keys. We decided to use those keys for the backspace, space, and edit mode operations. Users can personalize the pairing between these 3 keys and functions in the first time they use the system.

Text Editing Design

The Braille keyboard on the back of the mobile device frees the touchscreen for thumb input. This section describes the design of gestural interaction for editing operations.

Figure 2. Editing operations and matching gestures. (a) Caret movement using horizontal directional gestures. (b) Text selection using caret movement and touching the screen with a second thumb. (c) After selection, users cycle through the clipboard operations using vertical directional gestures and keeping the other thumb on the screen.

Edit operations start by pressing the edit mode key. This key works as a toggle; once in edit mode, users can perform caret movements, selections, and clipboard operations. The edit mode starts with character granularity; that is, all edit operations are at character-level. Users can start word granularity by pressing the edit mode key and button 2 (middle finger, left hand). Pressing edit mode and button 1 (index finger, left hand) returns to character granularity.

Although we could have designed our system without the edit mode activation, preliminary experiments showed that blind users often accidentally touched the screen while typing, resulting in unwanted actions and interruptions. Thus, to increase confidence in the system, we decided to have a specific key where users actively engage in editing operations. Moreover, previous research has shown that blind users prefer to use a mode key to reduce potential conflict between actions [10]. Still, users can enter characters while in edit mode.

The editing gestures are shown in Figure 2. There are three categories of editing operations:

Caret movement. The caret can be moved in two directions (left or right) by sliding either thumb in the corresponding direction. Our design goal was to minimize the number of gestures and keep them simple. Thus, we avoided symbols or complex forms [10] and adopted directional gestures that could be performed anywhere on the screen.

Text selection. To start a text selection, we leverage bimanual interaction. A selection is initiated by holding one of the thumbs on the screen and performing caret movements (i.e. directional gestures) with the other thumb. Selections always start from the current position of the caret. It is possible to adjust the movement granularity as described above. To reset the selection, users can lift the "holding" thumb.

Clipboard control. We support three clipboard operations: copy, cut, and paste. These operations are accessed via a menu. Unlike previous work, we intentionally avoided shapes or symbols [7] as blind users may have limited knowledge of print writing [10]. After a selection, users can cycle through the clipboard options by holding the selection thumb and performing a vertical gesture (i.e. up or down) with the other hand.

Hardware

Figure 3 shows the Hybrid-Brailler prototype. The Braille keyboard is composed of a custom-made PCB with an Arduino Mega 2560 microcontroller and Bluetooth module (Sparkfun Bluetooth Mate Silver). The 9 physical buttons are off-the-shelf 12x12x7.3mm pushbutton switches with removable flat caps. The microcontroller is powered via USB cable, which is connected to the mobile device. In order to hide all electronics, we placed the keyboard inside a 3D printed case to fit 5-inch touchscreen devices.

Software

The software running on the microcontroller sends button events to an Android app via Bluetooth. We used the Amarino[4] library to establish the Bluetooth connection. The Android app manages and decodes all button and gesture events and performs the text input/editing operations. Auditory feedback is given upon character and word input.

Gesture recognizers were specifically designed to cope with blind users' abilities. Because blind people may perform gestures at a different speed than sighted people, developed recognizers enable users to perform slower gestures [10]. Directional gestures consist of a minimum of 8.5mm. The direction (vertical or horizontal) is computed as the movement axis with the greater Euclidean distance between

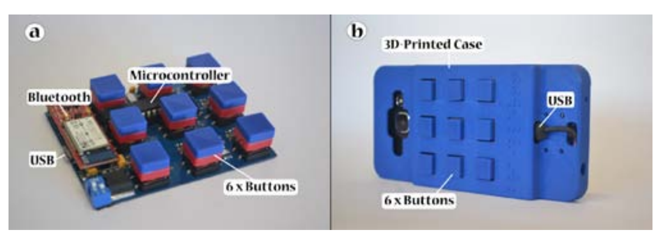

Figure 3. Hybrid-Brailler prototype. (a) PCB components. (b) 3D-printed case attached to a mobile device.

[4] http://www.amarino-toolkit.net/

the touch down and touch up coordinates. Gestures slower than 8.5mm per second are discarded, as these were probably unintentional touches.

STUDY 1: TEXT ENTRY

The purpose of this user study was to assess text input performance with Hybrid-Brailler. Particularly, we aimed to compare input speed and accuracy with a full-size physical Braille keyboard and a virtual keyboard. Although we did not have empirical evidence, we expected Hybrid-Brailler to outperform a state-of-the-art virtual Braille keyboard.

Apparatus

We used the Hybrid-Brailler prototype, previously described, connected to a Samsung Galaxy J5 (2015). The mobile device features a 5-inch capacitive touchscreen with multitouch support, running Android 5.1. All audio feedback was given using VocalizerEx, female language pack. As the virtual Braille keyboard, we used the opensource OpenBraille input method [17]. All user interactions were logged by each keyboard application for later analysis. As a baseline condition, we used a Perkins typewritter, which features a full-size Braille keyboard. In this condition, completion times and transcribed sentences were manually recorded by the experimenter.

Participants

We recruited eleven blind participants, 7 males, from a local training institution for visually impaired people. Participants' age ranged from 25 to 58 (M=42, SD=12.5) years old, and all participants were legally blind. All of them knew how to write with a Perkins Braille typewriter and needed screen readers to use a mobile device. Three participants had never used a smartphone while the remaining owned such a device for at least a month.

Procedure

The evaluation was set up as a within-subjects design where all participants experienced all three conditions. At the beginning, participants were told that the overall purpose of study was to investigate how text entry performance is affected by type of Braille keyboard. Following this, with the help of the experimenter, participants filled in a questionnaire about demographics, Braille knowledge, and mobile phone usage. We then demonstrated each input keyboard and participants were given warm-up trials (10 minutes per keyboard). They were encouraged to ask questions and allay all doubts.

After training, participants were instructed to complete 5 trials. Each trial contained a sentence, chosen randomly from a text entry corpus, to avoid order effects. The corpus used a proverbs dataset (using participants' native language) and was built following a similar methodology to MacKenzie and Soukoreff [14]. We chose proverbs because they are easy to memorize. The corpus has 451 sentences, with an average of 8 words per sentence, 4 characters per word, and a .98 correlation with the language character frequency. The order in which the keyboard conditions were undertaken was randomly selected.

The experimenter started by reading the target sentence aloud and asking for the participant to repeat it. Upon successfully repeating the sentence, they were asked to type as accurately and quickly as possible. Participants were instructed to type using Grade 1 Braille since some participants were not proficient with Grade 2 Braille. We used an unconstrained text entry protocol [28], where participants were free to correct any errors they encountered. Automatic correction/completion and cursor movement operations were not used during this study.

It was made clear to all participants that we were testing the keyboards and not their writing skills. Upon finishing each sentence, the device was handed to the experimenter to load the next random sentence and continue with the evaluation. The session ended with a debriefing questionnaire about each keyboard and overall preference. The entire procedure took between 60 to 75 minutes.

Dependent Measures

Performance was measured by analyzing trials' input stream [28]. We report on words per minute, total error rate, uncorrected error rate, and corrected error rate. Qualitative data was also gathered at the end of the experiment by debriefing each participant.

Design and Analysis

The study used a within-subjects design with *keyboard* as an independent variable. For each condition, participants completed 5 trials (440 words, average of 1760 characters): 5 *trials* x 3 *keyboards* x 11 *participants* = 165 *trials*.

We used a mixed-effects model analysis of variance [11] with a fixed effect of keyboard; trial was included as a nested factor within keyboard. Also, participant was modeled as a random effect to account for correlated measurements within subjects. Post-hoc pairwise comparisons were applied with Bonferroni corrections. Mixed-effects models can, unlike traditional repeated measures ANOVA, accommodate unbalanced data such as ours, where we had 3 participants that were unable to use one of the keyboards (see Results section). We applied Shapiro-Wilk tests to all observed measures. Words per minute, uncorrected error rates, and corrected error rates did not follow a normal distribution (Shapiro-Wilk $p<.001$). Thus, we used the Aligned Rank Transform procedure [27], which enables the use of repeated measures parametric tests after alignment and ranking.

Results

In this section, we analyze input performance of 2 mobile keyboards and a baseline condition (Perkins Braille typewriter) regarding speed and accuracy. Three of the 11 participants were not able to complete the warm-up trials with OpenBraille. Particularly, after 10 minutes, these participants were not comfortable in writing simple words and asked to skip the keyboard condition.

To assess input speed, we used the words per minute (WPM) measure [13]. Participants typed an average of 14.2

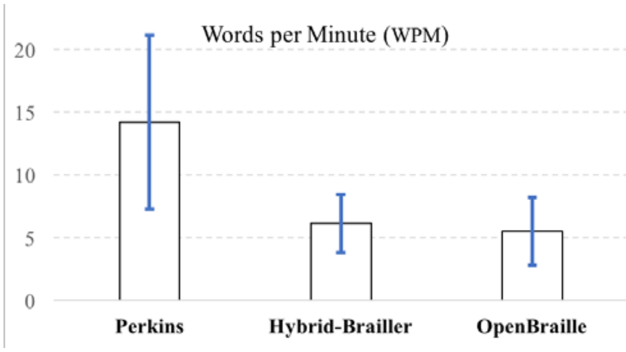

Figure 4. Words per minute across all keyboard conditions. Higher is better. Error bars denote 95% confidence intervals.

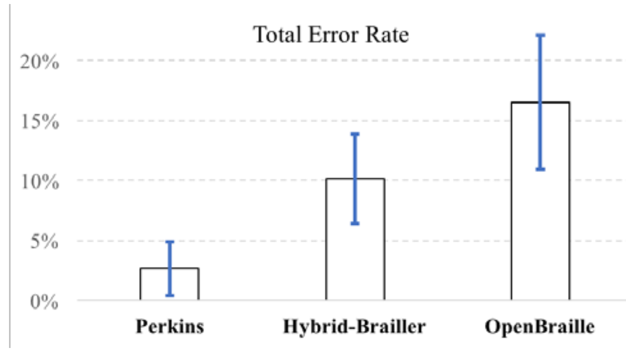

Figure 5. Total error rate across all keyboard conditions. Lower is better. Error bars denote 95% confidence intervals.

WPM (SD=12), 6.1 WPM (SD=3.9), and 5.5 WPM (SD=3.9) with Perkins, Hybrid-Brailler, and OpenBraille, respectively (Figure 4). We found a statistically significant main effect ($F_{1,2}$=56.783, *p*<.001) with differences between the baseline condition and both Hybrid-Brailler (*p*<.01) and OpenBraille (*p*<.001). Although the physical keyboard of Hybrid-Brailler allowed participants to achieve higher input rates, the difference was not statistically significant to its virtual counterpart (*p*=.443).

In order to analyze input accuracy, we calculated: 1) uncorrected error rates – erroneous characters in the final transcribed sentence, 2) corrected error rates – deleted characters that were erroneous, and 3) total error rates – erroneous characters that were entered (even those that were corrected) [28].

Overall uncorrected error rates were low, with averages between 2.5% and 4.8%. As with previous research, participants tend to correct most errors when given the chance [18], which result in high quality transcribed sentences. Participants achieved mean uncorrected error rates of 2.5% (SD=3.8) with Perkins, 2.7% (SD=3.9) with Hybrid-Brailler, and 4.8% (SD=4.5) with OpenBraille. Although both physical keyboards show lower error rates, due to the high variance, we did not find a statistically significant difference ($F_{1,2}$=2.349, *p*=.1).

Corrected error rates illustrate the amount of effective fixing and allows to answer the question "of all deleted characters, what percentage were erroneous?" High rate means that most deleted character were errors and should have been corrected. Participants achieved average corrected error rates of 83% (SD=29), 92% (SD=13), and 74% (SD=18) with Perkins, Hybrid-Brailler, and OpenBraille, respectively. Results show a significant main effect ($F_{1,2}$=5.008, *p*<.05), with significant differences between Hybrid-Brailler and OpenBraille (*p*<.05). This suggests that the physical keyboard increased participants' awareness of entering incorrect characters, while with the virtual keyboard participants deleted several characters, including correct characters, to fix the error. The baseline condition did not reveal significant differences, mostly due to the small number of fixes (4 deletes in 1765 characters). Notice that these fixes had to be done manually on the sheet of paper by rubbing on the respective raised dots.

Regarding total error rates (Figure 5), the percentage of all erroneous characters that were entered, participants achieved an average of 2.7% (SD=3.8%) with Perkins, 10.1% (SD=6.3) with Hybrid-Brailler, and 16.5% (SD=8) with OpenBraille. We found a significant main effect ($F_{1,2}$=18.926, *p*<.001), with significant differences between all keyboards. Overall, the most accurate keyboard, as expected, was the Perkins typewriter followed by Hybrid-Brailler. OpenBraille was the least accurate keyboard.

Regarding overall preference, results were similar throughout the different keyboards. Three participants preferred the Perkins Braille typewriter, due to its speed and accuracy. Two of three participants that preferred the Perkins had never used a smartphone. The remaining 8 participants were equally split between Hybrid-Brailler and OpenBraille. Those who preferred Hybrid-Brailler highlighted the tradeoffs between portability/mobility and accuracy. They liked having a mobile keyboard with tactile cues. On the other hand, the main reason for participants preferring OpenBraille over Hybrid-Brailler is related to the bulkiness of the later. Although participants agreed that it was easier to type with Hybrid-Brailler, they would not be comfortable using such a bulky device daily. This finding relates to the importance of social acceptability of assistive technologies [23]. In future iterations of the prototype, it is crucial to miniaturize it by building a slim, flexible, and ergonomic case that is perceived as an improvement over the original device rather than a "special" aid.

Discussion

Unsurprisingly, both mobile keyboards were significantly slower than the full-sized Braille typewriter. Results are in line with previous work on novice smartphone users where they start with entry rates around 4 WPM and just after 13 practice sessions achieve typing speeds of 16 WPM [2]. Thus, we expect mobile Braille keyboards to improve at similar rates with practice.

Nonetheless, the physical buttons of Hybrid-Brailler did not make a significant difference in terms of speed when compared to OpenBraille. The difference between input methods was only visible in terms of accuracy and ease of use. Using Hybrid-Brailler resulted in significantly less errors. When errors occurred, Hybrid-Brailler was significantly more effective than OpenBraille in dealing with them. Additionally, three participants were unable to use OpenBraille, highlighting the advantages of having tactile cues for inclusive text entry.

STUDY 2: TEXT EDITING

Non-visual text editing is fairly unexplored in the literature. To the best of our knowledge, we present the first controlled user study of blind people performing text editing operations on a mobile touchscreen device. Moreover, we aimed to validate our design concept of combining physical and virtual keyboards for editing tasks. Since current virtual Braille keyboards do not support edit operations, we compared its performance with the Explore-by-Touch + QWERTY keyboard approach.

Apparatus

The apparatus was identical to Study 1, except that participants were asked to perform editing tasks rather than solely text entry tasks. Also, instead of a virtual Braille keyboard, which does not support editing operations, we used Google's keyboard and Talkback/Explore-by-Touch.

The editing tasks used in the experiment are listed in Table 1 and were inspired by Fuccella et al. work [7]. We reduced the number of tasks from 15 to 7 to avoid the evaluation sessions from being too time consuming, while including different editing situations to let participants exploit all the gestures available in both conditions. Although each task contains more than one type of editing operation, we divided the task set into subsets according to the dominant type of operation, to help in the analysis of results. Similarly to [7], there are 3 subsets: first, keyboard use (tasks 2 and 4), where participants needed to delete existing text or add new words. Second, caret movement (tasks 1, 3, and 7), which consisted in navigating through larger areas of text to insert or delete characters. Third, text selection and clipboard use (tasks 5 and 6), where participants had to move words within a text area.

Participants

Participants of this user study were recruited from Study 1 accordingly to availability. The group was composed by a total of 11 participants, eight from the previous study. Ages ranged from 25 to 58 (M=40, SD=11.6) years old. None of the participants reported being able to perform editing operations, such as caret movement, selection or clipboard operations. However, they mentioned that they used the backspace to delete text as their editing procedure.

Procedure

At the start of the study, participants were told that its purpose was to investigate how different text editing systems perform in terms of speed and errors. We then explained the experimental setup and each editing system. For each one, we demonstrated how to move the cursor, perform selections, and clipboard operations. Participants were given warm-up trials for ten minutes per keyboard. They were encouraged to practice the different gestures and respective operations.

During the evaluation, participants were instructed to perform editing tasks as quickly and accurately as possible. Participants started with erroneous passages and needed to perform edit operations. Presented sentences were read aloud before they started the trial and the experimenter described the operations needed to correct the sentence. All user interactions were logged and the screen was captured. Order of tasks was randomly chosen and conditions were counterbalanced. At the end of the experiment, we debriefed participants and asked which system they would prefer if they had to make a choice. Similarly to Study 1, the entire procedure took between 60 to 75 minutes.

Task	Title	Instruction	Presented form	Correct form
1	Delete character	Delete the X character in the sentence	one two *thrXee* four five	one two three four five
2	Delete word	Delete the X characters in the sentence	one two three four *XXXXX* five	one two three four five
3	Insert character	Insert a space in the sentence	one two *threefour* five	one two three four five
4	Insert word	Insert the correct word in the sentence	one three four five	one two three four five
5	Move word	Move a word to restore the correct order	one three *two* four five	one two three four five
6	Move line	Move a line to restore the correct order	one one one one three three three three *two two two two* four four four four five five five five	one one one one two two two two three three three three four four four four five five five five
7	Correct errors	Correct the misspelled words.	Heroes of the sea, noble *paople*, valiant and immortal *X* nation, rise once again the splendor of *Spain*.	Heroes of the sea, noble people, valiant and immortal nation, rise once again the splendor of Portugal.

Table 1. The editing tasks used in Study 2.

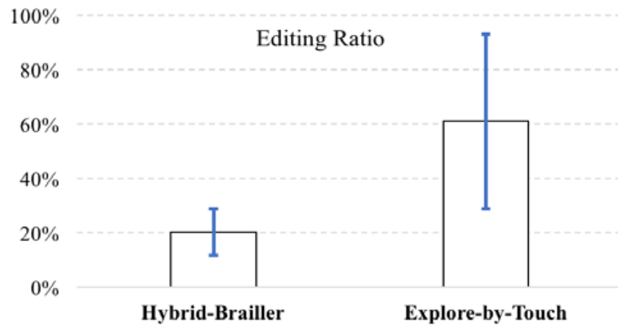

Figure 6. Editing ratio for both editing techniques. Lower is better. Error bars denote 95% confidence intervals.

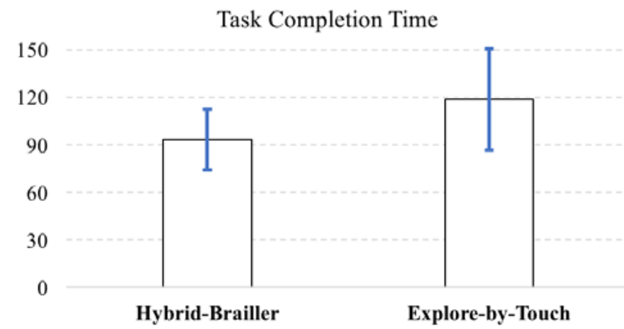

Figure 7. Task completion time (seconds) for both editing techniques. Lower is better. Error bars denote 95% confidence intervals.

Dependent Measures

We assessed performance during text edit tasks by several quantitative measures: success rate, task time, editing ratio, number of edit events, and selection changes. Qualitative data was also gathered by debriefing each participant.

Design and Analysis

The study used a within-subjects design with one independent variable: edit keyboard. For each condition, participants completed 7 trials: 7 *trials* x 2 *keyboards* x 11 *participants* = 154 *trials*. We applied Shapiro-Wilk tests to all observed measures and conducted parametric statistical analysis, using t-tests, for normally-distributed dependent variables, or non-parametric tests (Wilcoxon) otherwise.

Results

In this section, we examine the aggregated results before exploring the operation-specific results in more depth.

Aggregated Results

One of the 11 participants was not able to complete the warm-up trials with Explore-by-Touch. Although the participant wrote text daily with an Android smartphone, he was often disoriented while navigating written sentences and had difficulties performing the 'L' gesture for clipboard operations. After 10 minutes of practice, he asked whether he could skip the Explore-by-Touch condition. Thus, we discarded the participant from further timing analysis.

Overall, success rate (percentage of transcribed sentences that were exactly the same as the required sentences), was low for both conditions ($t_{(10)}$=1.170, *p*=.269). Hybrid-Brailler achieved an average success rate of 48% (SD=19.5) while Explore-by-Touch obtained 39% (SD=32.6).

In order to further analyze editing accuracy, we calculated the editing ratio. This measure indicates how far participants are from finishing the editing tasks considering the original sentences. Editing ratio is calculated using the Minimum String Distance (MSD) measure [24], which quantifies the similarity between two sentences:

$$Editing\ Ratio = \frac{MSD(required, transcribed)}{MSD(required, original)}\ x\ 100$$

Figure 6 shows the editing ratio for both editing techniques. Overall, Hybrid-Brailler (M=20.2% SD=14.5) outperformed Explore-by-Touch (M=61% SD=54.1%) by three-fold, resulting in a statistically significant difference ($t_{(10)}$=2.597, *p*<.05). The results suggest that participants were, on average, three times closer to finish all editing tasks when using Hybrid-Brailler and completed on average 80% of each editing task. This overwhelming difference can be mostly explained by the difficulties participants felt when using Explore-by-Touch, particularly in selection and clipboard operations. Average number of initiated selections was 0.2 with Explore-by-Touch (vs. 1.4 with Hybrid-Brailler), suggesting that participants were unable to start selections when intended.

Task completion times are shown in Figure 7. Hybrid-Brailler exhibited faster editing times with an average of 93.2 seconds (SD=32.6) while Explore-by-Touch obtained 118.3 seconds (SD=51.5). A paired t-test showed that the difference between editing techniques was statistically significant ($t_{(9)}$=2.276, *p*<.05). Again, most of the differences are related to clipboard operations (Figure 8). Although both techniques had similar number of editing events ($M_{Hybrid\text{-}Brailler}$=55, $M_{Explore\text{-}by\text{-}Touch}$=53), participants were not able to perform intended selections and clipboard operations, spending more time in secondary actions such as menu navigation.

These results were confirmed by participants' comments, referring that Hybrid-Brailler was easier to understand and perform intended editing actions. Indeed, 10 out of 11 participants preferred to use Hybrid-Brailler, with a 95% adjusted-Wald binomial confidence interval ranging from 60% to 99%, a lower limit above the two-choice chance expectation of 50%.

Operation-specific Results

The editing tasks (Table 1) can be grouped into three categories according to the dominant operation: keyboard, caret movement, and clipboard. Notice that each category contains more than one type of operation rather than being purely typing, caret movement or clipboard. Still, this clustering will help in analyzing results.

Keyboard dominated tasks. Tasks 2 and 4 were classified as keyboard dominated, as most operations are related to

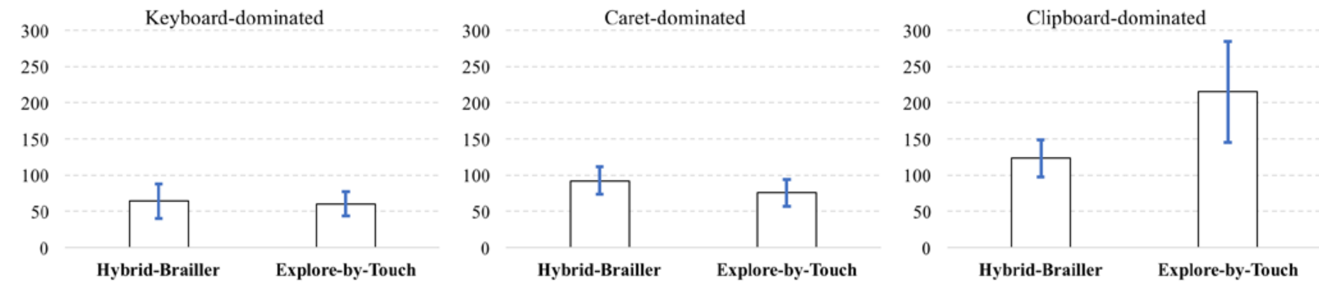

Figure 8. Average time to complete tasks (in seconds). From left to right, keyboard-dominated tasks, caret-dominated tasks, and clipboard-dominated tasks. Notice that the highest difference is on clipboard-dominated tasks. Lower is better.

enter characters, either alphanumeric or backspaces. Participants took, on average, 64 seconds (SD=40) and 60 seconds (SD=25.5) with Hybrid-Brailler and Explore-by-Touch, respectively (Figure 8 - left). This difference did not show to be statistically significant ($t_{(9)}$=.500, p=.631). On the other hand, Hybrid-Brailler allowed participants to get closer to complete the editing tasks ($M_{Hybrid\text{-}Brailler}$=20%, $M_{Explore\text{-}by\text{-}Touch}$=60%), which resulted in a significant difference ($t_{(10)}$=2.900, p<.05). Most errors were related with blank spaces, either forgetting to delete a single blank space between words (e.g. *one two three four five*) or missing a space (e.g. *onetwo three four five*).

Caret movement dominated tasks. We classified tasks 1, 3, and 7 as caret movement dominated. Being able to effectively move the caret is essential to most editing tasks. Caret-dominated tasks achieved the highest success rate. Hybrid-Brailler obtained 84.8% (SD=23) while Explore-by-Touch achieved 54.5% (SD=37), which resulted in a statistically significant difference (Z=2.060, p<.05). Still, participants took longer to perform caret-dominated tasks with Hybrid-Brailler ($M_{Hybrid\text{-}Brailler}$=92s, $M_{Explore\text{-}by\text{-}Touch}$=75s). This result can be explained as participants made use of volume keys to move the caret in the Explore-by-Touch condition, while having to perform directional gestures with Hybrid-Brailler. Nevertheless, this difference did not yield statistically significant results ($t_{(9)}$=.920, p=.385). Although we could have designed Hybrid-Brailler to make use of the physical keyboard for caret movement, we decided to clearly separate the roles of the keyboard – entry – and touchscreen –edit. Also, this enable users to be on edit mode and still use the keyboard to input text.

Selection and clipboard use dominated tasks. Tasks 5 and 6 were classified as clipboard dominated tasks. Overall, these were the tasks where participants experienced more difficulties. Regarding Explore-by-touch, average editing ratio was 55.6% (SD=38.6) and task time was 215 seconds (SD=106). On the other hand, Hybrid-Brailler achieved higher editing accuracy (M=31% SD=14) with smaller task times (M=122s SD=43). We found statistically significant differences for both measures: editing ratio ($t_{(10)}$=2.574, p<.05) and task time ($t_{(10)}$=2.656, p<.05). Regarding Hybrid-Brailler, although participants could select the intended text, most errors were related to caret placement, resulting in paste operations in the wrong position (e.g. *one one one three two two two two three …* - see Table 1, task 6). As for Explore-by-Touch, participants had additional difficulties in operating the context menu, either in performing the 'L' gesture or navigating clipboard options.

DISCUSSION

In this section, we describe major results, lessons learned and avenues for future research on nonvisual text entry, and limitations of our work.

Summary of Major Results

Regarding text input, participants achieved similar entry rates with both physical (M=6.1 WPM) and virtual (M=5.5 WPM) versions of the mobile Braille keyboard. Nevertheless, Hybrid-Brailler had a 6.4% improvement over OpenBraille with an average total error rate of 10%. Participants naturally corrected the overwhelming majority of errors, with high-quality transcribed sentences, resulting in 2.7% and 4.8% uncorrected error rate with Hybrid-Brailler and OpenBraille, respectively. Still, Hybrid-Brailler was significantly more effective (M=18%) in correcting errors than its virtual counterpart.

Study 2 revealed that editing performance was significantly higher with Hybrid-Brailler in terms of speed and accuracy. The most significant differences were found in editing accuracy with Hybrid-Brailler outperforming Android's Explore-by-Touch by three-fold. These differences were mainly due to difficulties felt during selection and clipboard operations, where participants struggled operating editing menus. On the other hand, caret movement tasks were the easiest to perform since participants could use the physical volume buttons to navigate the text.

Finally, the overwhelming majority of participants (10 out of 11) preferred Hybrid-Brailler over Explore-by-Touch due to its speed and ease of use in editing operations.

Lessons Learned

Nonvisual editing is demanding. Nonvisual editing is still time consuming and ineffective. Average success rate, i.e. edit tasks completed without errors, was 48% with Hybrid-Brailler and 39% with Explore-by-Touch. Future research should go beyond text entry and provide effective methods for nonvisual error correction and text manipulation.

Unawareness of caret position within text area. Although participants did not show major difficulties in caret movement, with either editing technique, most editing errors were related to lack of awareness of surrounding text. This issue resulted in clipboard operations (e.g. paste, select) done at the wrong text position. This result may be related to how Android reads the caret text, which is dependent of caret's movement direction rather than its absolute position. For example, words are equally read whether the caret is at the start or end positions, which is confusing and inconsistent with desktop screen readers.

Leverage physical buttons or directional gestures. When using Explore-by-Touch, participant could press the volume buttons to move the caret across a text area. As a result, caret movement was the easiest editing operation. Hybrid-Brailler obtained similar results, using horizontal gestures. Overall, this ease of use needs to be extended to other edit operations, such as text selection, cut, copy, and paste.

Simplify clipboard operations. One of the main issues revealed by Explore-by-Touch was related to opening the clipboard menu, through the 'L' gesture and navigating its options. Participants' final comments confirmed this result, mentioning that it was too confusing to select text, cut it, and paste it. For each of these operations, users had to navigate through multiple menus. Indeed, there is a need for better nonvisual selection and clipboard operations.

Leverage bimanual interaction. Hybrid-Braille integrated caret movement and text selection by leveraging bimanual interaction. One hand was used to activate text-selection while the other moved the caret. Results show that this approach was naturally easier and faster to use rather than navigating through multiple menu levels. Participants' comments reinforced this idea; Explore-by-Touch clipboard menus are slow and hard to operate. Moreover, this approach can co-exist with other editing gestures and be added to current accessibility services.

Awareness of errors. The overall lack of editing accuracy, shows a need for novel nonvisual feedback techniques that enable users to quickly detect errors in a text area, while being able to easily position the caret and correct them.

Standard editing task set and metrics. Similarly to text entry evaluations, where researchers have proposed standard phrase sets [14], methods, and metrics [28], input researchers need to work towards similar goals for text editing evaluations. The choice of tasks should consider statistics on frequency of editing actions that occur in everyday situations. Evaluations should be realistic and use unrestricted procedures, allowing for errors in transcribed sentences. Such evaluations need to go beyond task times and include new error metrics and analysis tools.

Limitations

Our participant sample was limited to blind people without experience performing text editing tasks on mobile devices. Although we report on a specific segment of the population, they represent an important user group when designing easy to use techniques. While we acknowledge that performance and experienced challenges can be significantly different for expert users, the derive lessons learned may still apply. Further research should replicate the user studies reported in the paper with proficient blind users.

In study 2, we asked participants to perform editing tasks on sentences already written on the text area. Although this procedure guarantees internal validity, performance and overall error awareness may be different when users edit their own sentences.

Finally, we investigated typing and editing performance of different input techniques. In order to avoid introducing confounding variables, both editing techniques responded similarly to user actions (e.g. caret movement, selection). Although improvements could be made to feedback, we choose to have comparable experimental conditions. Still, designers should devise novel feedback mechanisms that improve awareness of editing actions.

CONCLUSION

We have presented Hybrid-Brailler, a system that combines physical and gestural interaction for Braille input. The prototype can be attached to mobile touchscreen devices to enhance their typing experience. Hybrid-Brailler allows blind users to leverage physical buttons, on the back of the device, to input Braille characters while freeing the touchscreen for editing operations. Results show that Hybrid-Brailler is significantly more accurate than its virtual counterpart while maintaining the same entry rate in typing tasks. Moreover, in a performance comparison to the default editing technique of Android 5.1, Hybrid-Brailler was significantly faster and three times more accurate.

Further work is needed to improve our prototype by building slim and ergonomic components. Additionally, future research should focus on improving touchscreen-based editing techniques, particularly text selection and clipboard operations. Devising novel nonvisual feedback mechanisms will likely play a crucial role in improving error detection and editing awareness.

Overall, users acknowledged the benefit of augmenting current "flat surfaces" of mobile devices with tactile cues, particularly in data entry tasks. Such approach can be extended to other tasks beyond text entry and be leveraged as an enhancement method for mobile devices. Do-It-Yourself movements, 3D-printed technologies, and modular phones will play a crucial role in promoting this culture of inclusion and personalized computing.

ACKNOWLEDGEMENTS

We thank Fundação Raquel and Martin Sain in Lisbon (Portugal) and all participants. This work was partially supported by Fundação para a Ciência e Tecnologia (FCT) through scholarship SFRH/BD/103935/2014, INESC-ID research unit UID/CEC/50021/2013, and LASIGE research unit UID/CEC/00408/2013.

REFERENCES

1. Shiri Azenkot and Nicole B. Lee. 2013. Exploring the use of speech input by blind people on mobile devices. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility - ASSETS '13*, 1–8. https://doi.org/10.1145/2513383.2513440
2. Shiri Azenkot, Jacob O. Wobbrock, Sanjana Prasain, and Richard E. Ladner. 2012. Input finger detection for nonvisual touch screen text entry in Perkinput. In *Proceedings of Graphics Interface*, 121–129. https://doi.org/2305276.2305297
3. Matthew N. Bonner, Jeremy T. Brudvik, Gregory D. Abowd, and W. Keith Edwards. 2010. No-look notes: Accessible eyes-free multi-touch text entry. *Lecture Notes in Computer Science* 6030 LNCS: 409–426. https://doi.org/10.1007/978-3-642-12654-3_24
4. Maria Claudia Buzzi, Marina Buzzi, Barbara Leporini, and Amaury Trujillo. 2014. Designing a text entry multimodal keypad for blind users of touchscreen mobile phones. In *Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility* (ASSETS '14), 131–136. https://doi.org/10.1145/2661334.2661354
5. Chen Chen, Simon T Perrault, Shengdong Zhao, and Wei Tsang Ooi. 2014. BezelCopy: an efficient cross-application copy-paste technique for touchscreen smartphones. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, 185–192. https://doi.org/10.1145/2598153.2598162
6. Mark D Dunlop and Andrew Crossan. 2000. Predictive text entry methods for mobile phones. *Personal Technologies* 4, 2: 134–143. https://doi.org/10.1007/BF01324120
7. Vittorio Fuccella, Poika Isokoski, and Benoit Martin. 2013. Gestures and Widgets: Performance in Text Editing on Multi-touch Capable Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '13), 2785–2794. https://doi.org/10.1145/2470654.2481385
8. William Grussenmeyer and Eelke Folmer. 2017. Accessible Touchscreen Technology for People with Visual Impairments: A Survey. *ACM Transactions on Accessible Computing (TACCESS)* 9, 2: 6:1--6:31. https://doi.org/10.1145/3022701
9. T Guerreiro, P Lagoá, H Nicolau, D Gonçalves, and J A Jorge. 2008. From tapping to touching: Making touch screens accessible to blind users. *IEEE MultiMedia*: 48–50.
10. Shaun K Kane, Jacob O Wobbrock, and Richard E Ladner. 2011. Usable Gestures for Blind People: Understanding Preference and Performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '11), 413–422. https://doi.org/10.1145/1978942.1979001
11. R C Littell, P R Henry, and C B Ammerman. 1998. Statistical analysis of repeated measures data using SAS procedures. *Journal of animal science* 76, 4: 1216–1231. https://doi.org/10.2527/1998.7641216x
12. Kent Lyons, Thad Starner, Daniel Plaisted, James Fusia, Amanda Lyons, Aaron Drew, and E W Looney. 2004. Twiddler Typing: One-handed Chording Text Entry for Mobile Phones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 671–678. https://doi.org/10.1145/985692.985777
13. I S MacKenzie and R W Soukoreff. 2002. Text entry for mobile computing: Models and methods, theory and practice. *Human--Computer Interaction* 17, 2: 147–198. https://doi.org/10.1080/07370024.2002.9667313
14. I S MacKenzie and R W Soukoreff. 2003. Phrase sets for evaluating text entry techniques. In *Extended abstracts of the SIGCHI Conference on Human Factors in Computing Systems*, 754–755. https://doi.org/10.1145/765891.765971
15. Sergio Mascetti, Cristian Bernareggi, and Matteo Belotti. 2011. TypeInBraille: A Braille-based Typing Application for Touchscreen Devices. In *Proceedings of the 13th International ACM SIGACCESS Conference on Computers and Accessibility*, 295. https://doi.org/10.1145/2049536.2049614
16. Kyle Montague, Hugo Nicolau, and Vicki Hanson. 2014. Motor-Impaired Touchscreen Interactions in the Wild. In *16th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS)*.
17. Hugo Nicolau, Kyle Montague, Tiago Guerreiro, João Guerreiro, and Vicki L Hanson. 2014. B#: Chord-based Correction for Multitouch Braille Input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI'14), 1705–1708. https://doi.org/10.1145/2556288.2557269
18. Hugo Nicolau, Kyle Montague, Tiago Guerreiro, André Rodrigues, and Vicki L Hanson. 2015. Typing Performance of Blind Users: An Analysis of Touch Behaviors, Learning Effect, and In-Situ Usage. In *ACM SIGACCESS Conference on Computers and Accessibility*. https://doi.org/10.1145/2700648.2809861
19. Hugo Nicolau, Kyle Montague, Tiago Guerreiro, André Rodrigues, and Vicki L Hanson. 2017. Investigating Laboratory and Everyday Typing Performance of Blind Users. *ACM Transactions on Accessible Computing (TACCESS)* 10, 1: 4:1--4:26. https://doi.org/10.1145/3046785
20. João Oliveira, Tiago Guerreiro, Hugo Nicolau, Joaquim Jorge, and Daniel Gonçalves. 2011. 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 - ASSETS '11, 179. https://doi.org/10.1145/2049536.2049569

21. João Oliveira, Tiago Guerreiro, Hugo Nicolau, Joaquim Jorge, and Daniel Gonçalves. 2011. BrailleType: unleashing braille over touch screen mobile phones. *Lecture Notes in Computer Science* 6946 LNCS, PART 1: 100–107. https://doi.org/10.1007/978-3-642-23774-4_10
22. Jean-baptiste Scheibel, Cyril Pierson, Benoît Martin, Nathan Godard, Vittorio Fuccella, and Poika Isokoski. 2013. Virtual Stick in Caret Positioning on Touch Screens. In *Proceedings of the 25th Conference on L'Interaction Homme-Machine* (IHM '13), 107:107--107:114. https://doi.org/10.1145/2534903.2534918
23. Kristen Shinohara and Jacob O Wobbrock. 2016. Self-conscious or self-confident? A diary study conceptualizing the social accessibility of assistive technology. *ACM Transactions on Accessible Computing (TACCESS)* 8, 2: 5. https://doi.org/10.1145/2827857
24. R William Soukoreff and I Scott MacKenzie. 2003. Metrics for text entry research: an evaluation of MSD and KSPC, and a new unified error metric. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, 113–120. https://doi.org/10.1145/642611.642632
25. Caleb Southern, James Clawson, Brian Frey, Gregory Abowd, and Mario Romero. 2012. An Evaluation of BrailleTouch: Mobile Touchscreen Text Entry for the Visually Impaired. In *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services* (MobileHCI '12), 317–326. https://doi.org/10.1145/2371574.2371623
26. Kenji Suzuki, Kazumasa Okabe, Ryuuki Sakamoto, and Daisuke Sakamoto. 2015. Fix and Slide: Caret Navigation with Movable Background. In *Adjunct Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, 79–80. https://doi.org/10.1145/2815585.2815728
27. Jacob O Wobbrock, Leah Findlater, Darren Gergle, and James J Higgins. 2011. The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 143–146. https://doi.org/10.1145/1978942.1978963
28. Jacob O Wobbrock and Brad A Myers. 2006. Analyzing the input stream for character- level errors in unconstrained text entry evaluations. *ACM Trans. Comput.-Hum. Interact.* 13, 4: 458–489. https://doi.org/10.1145/1188816.1188819
29. Georgios Yfantidis and Grigori Evreinov. 2004. Adaptive blind interaction technique for touchscreens. *Universal Access in the Information Society* 4, 4: 344–353. https://doi.org/10.1007/s10209-004-0109-7