



# Text entry for the Blind on Smartwatches

## A study of Braille code input methods for a novel device

Mateus M. Luna<sup>1</sup> · Hugo A. D. Nascimento<sup>1</sup> · Aaron Quigley<sup>2</sup> · Fabrizzio Soares<sup>1</sup>

Accepted: 25 January 2022

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022

### Abstract

The interaction challenges presented by touch-screen enabled devices for blind and visually impaired people has been addressed extensively in the literature. However, the emergence of wrist-worn devices with small screens, the so-called smartwatches, exacerbates such challenges. This paper presents an evaluation of text entry methods for smartwatches, covering three existing approaches—Voice Input and two QWERTY methods: GBoard, and A4Keyboard—and three Braille input methods proposed by the authors. Nine blind users and a low-vision participant composed sentences with the six methods. Speed and error rates are presented and preference data are analyzed, showing the approval of two Braille methods and a positive reception to one of the QWERTY methods, followed by the Voice Input. Each modality has its own set of strengths and weaknesses, which suggests room for improvement before the widespread adoption of smartwatch text entry as a reliable and accessible method for blind people.

**Keywords** Smartwatches · Blind people · Braille · Text entry · Text input · Accessibility

## 1 Introduction

In 2018, the World Health Organization (WHO) estimated that there were 36 million blind people, in addition to 217 million with moderate to severe vision impairments (e.g., cataracts) [44]. This group of current and potential mobile technology users have faced both opportunities and challenges with the dominance of smartphones and their touch-sensitive screens. In simple terms, such devices lack tactile

feedback for specific buttons while the dynamic characteristics of the visual interface presents problems for people with vision impairment. This has motivated the development of screen readers and different interaction techniques, which have been studied extensively by researchers in accessibility and in smart devices, such as smartphones and smartwatches.

Smartwatches can be viewed as wrist-worn computers. Showing the time on the watch-face is now only a small aspect of what such devices offer, with notifications to health tracking being commonplace. These devices usually come with powerful hardware, a touchscreen, a speaker and a variety of sensors, along with the benefits associated with being continuously carried, while worn on the wrist, allowing for easy and immediate interaction [63].

When smartwatches emerged, they brought an appeal for supporting consumers daily routines, well being, and lifestyle. However, what was quickly realized is that new approaches had to be developed in order to provide better experiences for tasks such as text entry, due to their small screens [31]. Currently, many smartwatches rely only on voice recognition as a text input method, and some offer small QWERTY keyboards. Unlike smartphones, which increasingly offer alternative features for blind users, smartwatches generally focus on their

---

This work is dedicated to all the volunteers that helped developing and evaluating our ideas. We are extremely grateful for those who offered their time with such honesty and patience.

---

✉ Fabrizzio Soares  
fabrizzio@ufg.br

Hugo A. D. Nascimento  
hadn@ufg.br

Aaron Quigley  
a.quigley@unsw.edu.au

<sup>1</sup> Instituto de Informatica, Universidade Federal de Goias, Goiânia, GO, Brazil

<sup>2</sup> School of Computer Science and Engineering, University of New South Wales, Sydney, Australia

screen readers, such as Talkback and Voice Input, leaving text entry restricted primarily to voice input.

This work focuses on analyzing the challenges for blind people performing text entry. Voice input is the most common alternative, but it comes with issues of inadequate recognition and lack of privacy [45] [58]. It is possible to adapt current and existing solutions of on-screen QWERTY keyboards to the use of screen readers, such as Talkback and VoiceOver. However, with small screen sizes, as is common with smartwatches, we speculate it would be hard to adapt these.

An alternative approach might explore Braille text entry for smartwatches. Braille is a system based on cells of up to six dots per character (Grade 1), distributed over two columns. It is used by blind people, especially for reading via tactile sense, and it is adopted around the world not only for literacy but for daily information such as public signs and medicine in drug stores. Although its widely used, there are active discussions regarding the continued adoption of Braille. Reports show a decay on Braille literacy rate in recent years in the USA [57], where about 10% of the blind children are learning Braille, and less than 10% of the 1.3 million blind adults can read it. However, even with the arrival of alternative technologies that improve accessibility, Braille education retains its importance for blind people's independence [16].

From an understanding that different solutions must be provided for different kinds of users [42], here we explore three Braille input methods inspired by previous work for smartphones and evaluate their usage against other possible text input methods for smartwatches. Five Braille cell composition techniques were narrowed down to three by specialists in our pilot study [30]. In this paper, we describe an evaluation study of existing text entry methods on smartwatches, against our proposed methods, focused on blind people who are literate in Braille. A review of related research suggests this is the first work to provide an extensive evaluation of any text entry method on smartwatches targeting this demographic (Fig. 1).

The remainder of the paper is organized as follows: the following section describes our literature review, followed by our method presentation, where the prototypes and evaluation protocol are highlighted. The results section describes the data collected, followed by a discussion section with deeper analysis. Finally, we conclude with a description of the contributions along with suggestions for future work.

## 2 Related work

### 2.1 Text entry on smartwatches

Text entry on smartwatches has been highlighted as an open problem, especially due to small screen sizes. The area has



Fig. 1 Samples of proposed input methods

been considered with a systematic review of the literature in prior work [31]. The overview of related work focuses on existing methods, techniques and study designs which help to contextualize our proposed methods and results.

A strategy that is often adopted is to propose new layouts to fit bigger buttons on screen, separating character selection into two stages, where the first one reduces the amount of elements on screen, to guarantee a precise click on the second one. This was introduced in ZoomBoard [43], an interactive zooming keyboard, and similar concepts appear in SplitBoard [22], SwipeBoard [12], VirtualSlidingQWERTY [11], DriftBoard [50] and UniWatch [46].

Other research has analyzed different ways of interpreting a click action. For example ForceBoard [23], SwipeKey [49] and DualKey [21] provide more than one value to a single button on screen. On screen gestural input is also a theme explored in studies such as InvisiBoard [37] and the work by Nascimento et al. [39]. COMPASS [65] proposes a circular keyboard on Smartwatches, using a rotational bezel while WatchMI [64] employs pressure, twist and panning actions with a smartwatch face for input to a circular keyboard.

Finally, more recent studies try to address the problem by providing statistical word models, without modifying the QWERTY layout. Examples of work with this strategy include WatchWriter [19], the method analyzed in Turner et al. [60] and Velocitap [62]. The latest work presents the best text entry speed rates so far (up to 40.6 words per minute) with a corrected error rate of 3% which appears to inspire current industry adopted methods, such as Google's WearOS default QWERTY keyboard, that allows tracing gestures for word composition.

### 2.2 Blind text entry on touchscreen devices

Interaction with touchscreen devices by blind people has been considered from a variety of directions. The work by Kane [25] analyzes gestures on-screen, while Leporini [27] evaluate the usability and accessibility of the iPhone's

Voice Over and more recently, a study by Abdolrahmani [1], investigates situational-induced impairments and disabilities scenarios. Among many tasks, text entry receives particular attention due to its complexity.

Alternative multitap strategies, especially those inspired by old phone keypads were proposed in NavTouch [20] and No-look notes [9] with results inferior to subsequent research with QWERTY keyboards. Azenkot [5] exposes the advantages and challenges with speech input for blind people, especially when correcting mistakes. There is also discussion regarding better auto-complete with concurrent suggestions while typing [36], but these later ideas are yet to be evaluated. More recently, the usefulness of gesture input over the QWERTY layout has also been noted in a pilot study named AGTex [8], with promising early results.

The majority of proposals, however, come from researchers who suggest that Braille makes more sense for those literate on this system. Works dating from 2011 to 2015 regarding Braille text entry for smartphones are presented through a systematic review in [51]. Several interfaces were proposed, including tap and tracing methods (BrailleType [41], BrailleTouch [17] [55], EdgeBraille [35]), line by line composition (BrailleKey [56], TypeInBraille [34]) and different ways of automatically detecting which dot is being activated (B# [40], BrailleEasy [48]). Some methods required having both hands on-screen.

More recent research has explored fabrication and one-handed interaction. BrailleÉcran [52] has a tactile 3D printed film, overlaid on the smartphone. SingleTapBraille [3] eliminates the need for finding very specific locations on the screen, allowing information to be entered using a single finger, while an algorithm analyses sequential taps and their positions. The same authors later proposed BrailleEnter [4]: Braille cell is composed of six clicks anywhere on the screen, a press being responsible for activating a dot and a simple tap corresponding to an inactive dot. Alternatively, single-handed interaction is proposed on OneHand-Braille [15], which works similarly to TypeInBraille, where instead of using two-finger taps for selecting two dots, the user swipes over the screen. BrailleSketch [28] uses tracing over dots to compose the braille code, besides offering auto-correction of characters and long press for character confirmation. HybridBrailler [59] proposes a physical 3D printed case with keys on the back of the phone. Touch screen is used only for gestures associated with confirmation and other secondary tasks. The proposal is compared against an implementation of B# denominated OpenBraille.

Most of these studies provided some evaluation, and those available are summarized in Table 1. When multiple sessions were performed, excluding training, the first and last values are reported. Perkinput only informed the value of its latest session. *WPM* column stands for Words Per Minute, a speed measuring unit. The last column (*Both hands?*)

indicates if both hands are required on the device to use the keyboard.

There is also comparison work with QWERTY assisted by screen reader technology which we summarize in Table 2. It should be noted that, even though sessions coincide with the number of sessions practicing Braille input methods, most of the users in these studies had prior usage of QWERTY keyboards assisted either by Talkback or VoiceOver.

Some of these solutions have inspired the development of Braille keyboards for Android, such as SwiftBraille [2] and SoftBrailleKeyboard [13] and for iOS (default accessibility toolkit onscreen braille keyboard), which are now in daily use.

### 2.3 Smartwatches and text entry for blind people

Smartwatches have been used in several studies regarding assistance to blind people with audio and vibrotactile feedback. For instance, many studies focus on indoor and outdoor navigation, as in StepByWatch [18], which also employs extra sensors. Alternative use cases include face recognition via a smartwatch camera [10], virtual spatial map exploration supported by the worn device [6], and forms exploration oriented by vibrations and audio feedback [7]. Text entry on the device itself is less frequently considered, and Braille text entry is rarely noted.

TactBack [14] is a system to output braille cells for deaf-blind users. A sequence of up to three vibrations is emitted by a smartphone, paired with a smartwatch responsible for the other three vibrations. The method does not perform text input. The BrailleEasy [48] authors suggest that smartwatch input could work but no study for such is presented, which is necessary for small screens considering that it requires three-fingers tap.

HexaBraille [26] proposes a similar idea to one of our techniques; however, this work compares different dot button sizes on a round screen distribution. In addition, the evaluation is performed with sighted people in a smartphone attached to the user's arm, which we consider an unsuitable study protocol. Their conclusion goes in favor of a large 9mm button size that is close to our key dimensions shown below.

Finally, it is worth highlighting Dot Inc. [24] who are developing the Dot Watch™, a smartwatch effectively planned for blind users. Instead of a touchscreen, it provides a watch face with four refreshable Braille cells. The watch is paired with smartphones to receive notifications, translating to the braille line. The price is approximately \$300.00 and the product, while interesting, does not provide a method for text input.

**Table 1** Braille methods for Smartphones and their registered performance

Method	Session	Participants	WPM	Total Error Rate	Both hands?
BrailleType	1st	15	1.4	8.9%	No
BrailleKey	1st	5	1.8	5.71%	Yes
Perkininput	7th	8	6.0	15.7%	Yes
TypeInBraille <sup>1</sup>	1st	10	6.5	6.5%	No
EdgeBraille	1st	14	3.9	8.4%	No
EdgeBraille <sup>2</sup>	2nd	7	7.1	NA	No
BrailleEasy	1st	3	9.8	–	No
BrailleEasy	2nd	3	10.7	–	No
SingleTapBraille	1st	7	4.7	11.2%	No
BrailleTouch	1st	11	17.8	28.6%	Yes
TypeInBraille <sup>1</sup>	2nd	10	7.0	4%	No
BrailleEasy A <sup>3</sup>	1st	3	5.8	–	No
BrailleEasy A <sup>3</sup>	2nd	3	6.6	–	No
BrailleÉcran <sup>4</sup>	1st	10	2.5	2.6%	No
BrailleEnter	1st	2	11.5	14.1%	No
OneHandBraille	1st	6	3.54	25.61%	No
OneHandBraille	2nd	6	4.90	19.28%	No
BrailleSketch 1	1st	10	5.3	43.9%	No
BrailleSketch 1	5st	10	11.3	30.8%	No
BrailleSketch 2 <sup>5</sup>	1st	10	6.5	14.8%	No
BrailleSketch 2 <sup>5</sup>	5st	10	14.5	10.6%	No
HybridBrailler	1st	11	6.1	10.1%	Yes
OpenBraille	1st	11	5.5	16.5%	Yes

<sup>1</sup>Values approximated, as paper only presented bar plots. <sup>2</sup>Average the results of long term evaluation, that last two weeks. <sup>3</sup>This evaluation was performed with another set of phrases, by other participants, in Arabic language. <sup>4</sup>Evaluation values not published, but reported by authors. <sup>5</sup>The first evaluation is a 15 minute execution of as many phrases as possible, while the second consists of three phrases

**Table 2** QWERTY assisted by screen reader solutions compared on Braille input studies

Compared with	Session	Participants	WPM	Total Error Rate
BrailleType	1st	15	2.1	13%
BrailleKey <sup>1</sup>	1st	5	2.0	6.0%
Perkininput	7th	8	3.9	14.7%
TypeInBraille <sup>1</sup>	1st	10	6.8	6.0%
EdgeBraille	1st	14	3.6	10.5%
EdgeBraille <sup>2</sup>	2nd	7	6.2	NA
BrailleEasy	1st	3	9.1	–
BrailleEasy	2nd	3	8.6	–
SingleTapBraille	1st	7	3.7	20.5%

<sup>1</sup>Values approximated, as paper only presented bar plots.

<sup>2</sup>Average the results of long term evaluation, that last two weeks

The lack of alternatives to this specific context motivated our research. In a pilot, non-published study, we discussed six ideas for text entry on the smartwatch, mostly inspired by the ones reported on previous works for smartphones, named “Touch”, “Swipe”, “Connect”, “Serial”, “Perkins”

and “Pressure”. Aiming at an initial feedback, basic prototypes were evaluated by three blind participants. From the experiments, we eliminated the “Pressure” prototype, which was based on very sensitive inclination gestures collected via motion sensors. This result set the base to our following investigation published in [30]. That work focused on the very basic task of composing the Braille cell. The five remaining prototypes were improved and feedback features, such as haptic and sound with different intensities or combinations were proposed. They were then compared by seven specialists in an evaluation study that resulted in a qualitative analysis of which strategies might be most useful for blind users in a Braille text input. The insights identified in this review and analysis of related work has given rise to the method we now propose.

### 3 Method

This section presents our input techniques for evaluation, including an exposition of our proposed methods.

### 3.1 Proposed Braille input methods

Our previous work [30] evaluated five Braille input prototypes based on existing ideas for smartphones. Three of them have been considered acceptable for this next stage of evaluation namely “Touch”, “Swipe” and “Connect”. However, as they were initially designed for evaluating basic character inputs, they had to evolve to provide word and sentence composition and better interaction mechanisms, such as input confirmation and feedback, as suggested at the end of our previous study.

With the version used in this paper, each proposed input method provides a different strategy for composing the Braille cell, which is achieved by activating or deactivating dots. Upon confirmation of the dots selection, a smartwatch text-to-speech engine announces the chosen character and dots are reset to the original state. Confirming a cell where all dots are deactivated inserts a blank space.

**Touch:** Our most basic interaction method is inspired by BrailleType [41] and BrailleÉcran [52]. It is derived from the hypothesis that the small watch screen size could allow easy memorization of button position, especially after practice. One tap over a dot or its surrounding area toggles its activation. Confirmation of the desired Braille cell is performed by a double-tap on the middle of the screen. Double-tap mimics the rationale of the confirmation step of applications when using Talkback and VoiceOver. A mockup of this prototype is seen in Fig. 2a, where the darker dots are the activated ones and the circles indicate taps on-screen.

**Swipe:** This method attempts to minimize the need for target-precise clicks, using only directional gestures. The idea comes from previous work reporting that, for touch-screen devices, blind users would prefer gestures than buttons [25]. It is further hypothesized that by identifying corners, the user will have a sense of necessary paths to swipe. The user swipes in one of six directions to activate or deactivate a dot:

1. Bottom-right to top-left corner;
2. Middle-right to middle-left corner;
3. Top-right to bottom-left corner;
4. Bottom-left to top-right corner;
5. Middle-left to middle-right corner and
6. Top-left to bottom-right corner.

Confirmation of the Braille cell is a double-tap anywhere on-screen. Figure 2b demonstrates the logic of this method.

**Connect:** Allows connecting dots to compose the Braille cell. This interaction design is inspired by IPPITSU [61], BrailleSketch [28] and applications such as Swift-Braille [2]. The user swipes on screen, passing through the desired dots area sequentially. Confirmation on a Braille cell is performed by a short timeout (1.2 s), after the finger

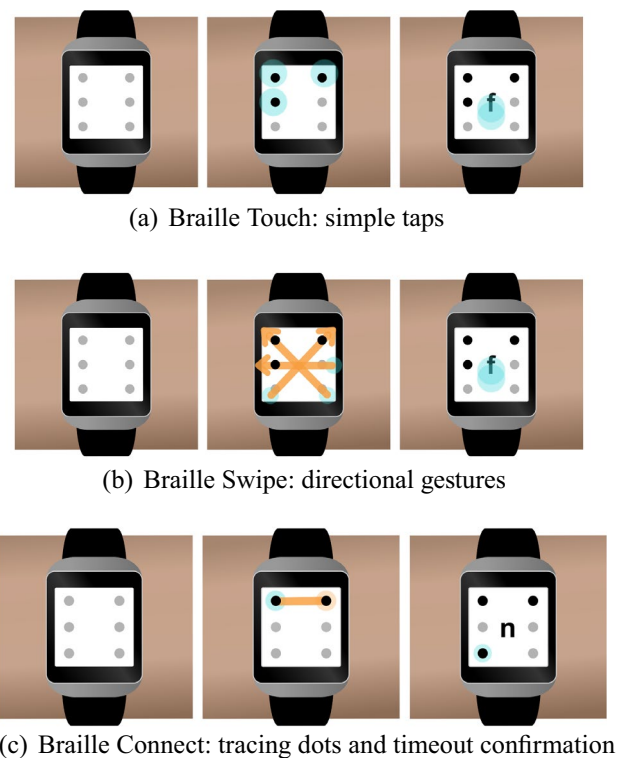


Fig. 2 Interaction logic of the proposed Braille methods

releases the screen. This confirmation allows a user to perform a click to activate dots left off-path, as long as it happens before the timeout finishes. If such a click occurs, the timeout is reset, meaning that one can compose the cell using only taps instead of tracing the path, if performed on time. Double-tap here is not necessary for confirmation, except for entering a blank space.

For all methods, each dot emits a feedback to keep user aware of what is being composed before character conclusion. When activating a dot, a dual tone generator is emitted, the same sound from an old phone keypad, and when deactivating a dot, no sound is emitted, but a vibration reinforces the interaction. These feedback strategies are also adopted based on our conclusions in [30].

New features were implemented to evaluate all the tasks proposed in the test, such as a long press of two seconds (longer than the system default long press), that sends the message to text input and finishes the keyboard, with spoken feedback of “Message sent.” However, some new features were disabled as will be explained further in our Study Protocol: two fingers left swipe for erasing, two fingers right swipe for navigating characters and long press to listen to the complete sentence or indicating activated buttons. The generic layout for the three methods is seen in Fig. 3a.

### 3.2 Existing text input applications for smartwatches

Presently, it is not possible to create third-party input methods for the Apple Watch. By default, the most recent watchOS offers voice input, handwriting input, and a list of preset reply messages that can be edited on the iPhone. Developers have been able to create a QWERTY app, but it does not work as an input method, instead it only generates text to be shared. The Samsung Galaxy Watch supports handwriting input, voice input, and a multitap keyboard by default. However, it does have a few third-party input methods available, typically paid ones.

Google WearOS (formerly Android Wear) offers Voice Input and a QWERTY keyboard. There are also a wider variety of third-party keyboard apps available. As a result, here we focus on this operating system and ecosystem for study. To compare with our proposed methods, however, these keyboards have to be usable with Talkback activated, as it is the way that blind people would make use of any WearOS Smartwatch. An empirical analysis of the existing keyboards on Play Store with Talkback activated is presented in Table 3.

The column *Writing interaction* indicates if typing characters by clicking on their keys (when available) is feasible. *Keys output before insertion* is related to the pronunciation of keys before inserting them, which is the standard interaction strategy for Talkback. MultiTap Wear Keyboard pronounces the first letter of its grouped keys correctly, but after trying to switch to other letters inside the same group, many wrong insertions would happen. Sometimes, Talkback is able to read a button on-screen but can not identify what it does, especially with icon buttons. If that happens to at least one of the buttons on the keyboard, it was marked “No” on column *Buttons with proper labels*.

After analysis of these methods, we concluded that GBoard, Google Voice Input, A4 Keyboard, and the SmartWatch Keyboard for WEAR OS are the only ones acceptable

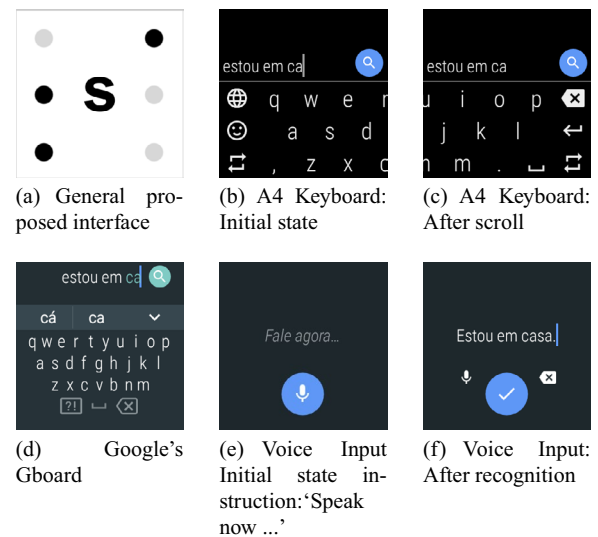


Fig. 3 Keyboards screenshots

for usage tests. The final one, however, has an interaction almost equivalent to the A4 Keyboard but lacks Portuguese support. As our study is performed with people who use the Portuguese language, we needed to exclude it from our tests. Below, we present in detail how these methods work and how some of their flaws were bypassed on the study to allow a minimum writing task to maintain internal validity in our study.

#### 3.2.1 A4 keyboard

The A4 Keyboard works similar to SplitBoard [22]. The QWERTY keyboard has large keys, so the complete layout is hidden on a horizontal scroll (Fig. 3b and c). The interaction on keys happens as follows: one click focuses on the button and then speaks out loud the character; a double click inserts it and speaks again. The icon buttons on the interface, however, were not labeled, so Talkback cannot identify them. To deal with this, in our experiments, every time one of these

**Table 3** Existing input methods for WearOS and their support for Talkback observed on empirical analysis

Keyboard app name	Writing interaction	Keys output before insertion	Buttons with proper labels
Google Keyboard (GBoard)	Yes	Yes	Yes
Google Voice Input	Yes	–	No
Google Handwriting	Partial	–	No
Keyboard for Wear OS (A4 Keyboard)	Yes	Yes	No
SmartWatch Keyboard for WEAR OS	Yes	Yes	No
FlickKey	No	No	No
MultiTap Wear Keyboard	Yes	Partial	Partial
Wear Keyboard	No	No	No
TouchOne	No	No	No

keys were hit, the researcher had to speak it out loud. We highlight here that scrolling with Talkback active must be performed with two fingers on the screen.

### 3.2.2 Google keyboard input (GBoard-Google QWERTY)

The default WearOS keyboard supports well Talkback. When pressing a finger over a button, the user immediately listens to its label/character, and then upon release, it is inserted. This way, the finger must be on the screen while the interface is being explored. A long press on some keys opens alternatives with accents on a popup, which is dismissed moving outside of it. A screenshot of its layout is seen in Fig. 3d.

### 3.2.3 Google voice input

Selecting Voice Input method on a text entry context, the screen in Fig. 3e is presented to the user. Talkback pronounces the “Speak now” sentence and emits a specific beep. The user can then say the desired sentence, and after a short period of silence, the processing is done, the second beep is emitted, and the transcript sentence is presented on text input as in Fig. 3e. The three buttons at the bottom do not contain labels for screen readers, so it is needed to speak out loud those too when a user is testing this method. For this technique to work, the smartwatch needs to be connected to the internet or paired with a smartphone connected to the internet, so that Google cloud service can be accessed.

## 4 Study protocol

We elaborated our study protocol to access users’ critical difficulties in using each method and compare their performance and preference. Each test lasted approximately 3 hours.

Users performed tests sat down in front of a table, in a quiet, private room. At first, a consent form printed in Braille is presented to the user, where privacy details are informed, as the experiment requires video recording. The user is declared free to abandon the experiment at any moment or to give up any of the tasks as he or she wishes. The consent is confirmed verbally on camera. Basic demographic information is collected, including onset blind age and discursive reports on their usage of Braille, computer, smartphones, and smartwatches. Participants are then questioned, on a 5-level Likert scale from None (1) to High (5), about their experience with the following: Braille System; QWERTY layout keyboards; and Voice Input methods (transcription). Following this we provide a demonstration with Talkback on, where users are free to navigate on some apps and familiarize themselves with the hardware.

Before actual testing, users are also asked to touch on the screen, indicating where does he or she expects each of the Braille dots to be. This is necessary, as we have seen in our previous study that different users can be more used to different dots’ arrangements. Some may prefer the reading order (1-2-3-4-5-6), other the writing order, as common when using slate and stylus (4-5-6-1-2-3). Also, depending on the user’s arm posture, it might be desired to have the screen rotated 90 degrees anti-clockwise. These details are then adjusted for customization on a settings page for the Braille keyboards.

The evaluation of the six methods is performed in a randomized order. For each one, the user is first introduced to its interaction logic. The instructor holds the participants finger, explaining exactly what to expect on screen and how to interact with their elements, as well as alerting the situations where Talkback will not work as it usually would. Some trial characters, such as letters “s”, “z”, “j”, “o” and white space are performed. This set of letters was chosen due to their diversity of Braille dots and distribution on QWERTY layouts.

Following this, the participant engages in actual writing, by entering four sentences (Fig. 4) plus a repetition of the first sentence at the end. The phrase selection process was aware of MacKenzie’s [32] set of phrases and the issue of having a representative set. The challenge here is that the participants understand Portuguese only, and despite the efforts, the researchers could not find a similar standardized corpus of phrases for this language. An attempt at translating some of the sentences from this well-known set was undertaken but their meaning felt lost and its features (such as rich correlation with the English language letter frequency) would not be the same. It was then agreed on the need for creating a custom set, aiming at sentences with approximate lengths and a considerable diversity of letters. These three phrases feature the seven most frequent letters in Portuguese vocabulary (a, e, o, s, r, i, d) [47]. It was further desired that sentences would look familiar for the participants in the context of short reply messaging,

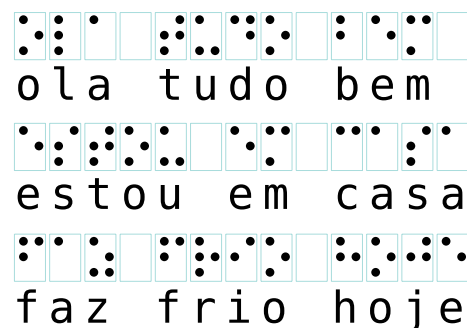


Fig. 4 Braille symbols for test sentences in *Brazilian Portuguese*, equivalents to “Hi, how are you?”, “I’m home” and “It’s cold today”

so that their experience could make sense of an interaction that so far was only performed on smartphones. It should be a phrase that would be coherent with the reality of blind people, without words that could cause confusion or misspellings.

There are not, however, any punctuation or accents (the word “ola” is actually written “olá”), as this would require for the QWERTY methods to interact with new layers of complexity in the interface, which was considered a negative for internal validity. For a similar reason, erasing was not allowed during the tests, as we were intended to record the raw error rate. Because of that, users were asked not to correct errors. In case any participant wishes to know the current typed sentence during tests, the instructor would inform them verbally, without the need for them to search for the text input field or to use long press on Braille keyboards. We choose to impose this limitation to prevent cases where the user would go over text input fields in QWERTY methods and accidentally replace or erase parts of the text. If also, by accident, any user toggles alternative layouts on QWERTY keyboards, the instructor would revert it to the original state and discount this later in the time analysis.

Finally, after each participant completes the four sentences, feedback is requested on the positive and negative aspects of the method. Further, the user must rate the method on a 5-level Likert score from (1) Too bad to (5) Very Good. Next, a RAW Task Load Index [38] - NASA-TLX questionnaire is applied, using a simplified 5-level Likert scale for each question. The six questions from this workload assessment tool are:

- *Mental Demand* How mentally demanding was the task?
- *Physical Demand* How physically demanding was the task?
- *Temporal Demand* How hurried or rushed was the pace of the task?
- *Performance* How successful were you in accomplishing what you were asked to do?
- *Effort* How hard did you have to work to accomplish your level of performance?
- *Frustration* How insecure, discouraged, irritated, stressed, and annoyed were you?

Finally, after all the input methods are evaluated, participants are asked to rank them by preference, and then provide any suggestions regarding improvements. We conclude by asking, which, if any, methods would be used if the user had such a device and whether it would be in their interest to purchase a smartwatch if any of the Braille keyboards were available.

#### 4.1 Evaluation metrics for obtained text data

All test processes were recorded on a smartphone camera for subsequent data processing. With Braille methods, logs were also available to complement the acquired information.

The Words per Minute (WPM) metric was used for typing speed, as it is widely accepted on text entry evaluation [32], including the convention of five character length for a word:

$$WPM = \frac{|T| - 1}{S} \times 60 \times \frac{1}{5}, \quad (1)$$

where  $T$  is the final transcribed phrase, and  $|T|$  is the length of it. The  $S$  term is seconds, measured from the entry of the first character to the entry of the last. The “60” is seconds per minute and the “1/5” is words per character [33].

For text entry accuracy, we calculated values for Minimum String Distance (MSD), also known as Levenshtein distance, that provides a notion of how many characters editing would be necessary to fit the resulting sentences into the proposed ones. The algorithm is well known in statistics and has been used for text entry error analysis widely since the proposal by Soukoreff and MacKenzie for doing so [53], particularly when no correction is allowed, as in our protocol. From the same authors, we use an implementation offered in JAVA to calculate the distance, its source code is available online [54]. To present these values as error rates in percentage instead of character numbers, Eqn. 2 is used,

$$Error\ Rate = \frac{MSD(P, T)}{\max(|P|, |T|)} \times 100\%, \quad (2)$$

where  $|P|$  and  $|T|$  are the lengths in characters of the presented and transcribed phrases, respectively.

As we opted for not allowing the use of capital letters, punctuation or accents, we normalize the results from voice input before calculating any metrics. So for example, a phrase transcript as “Olá, tudo bem?” was considered “*ola tudo bem.*”

#### 4.2 Participants

Our study was conducted with ten participants, detailed in Table 4, that we refer here as P1, P2, P3...P10, to preserve anonymity. Of these participants, nine are blind, and one is visually impaired—P7 reported having between 10% and 15% of vision, being able to distinguish letters when having surfaces extremely close to the eyes, which was not allowed. Six of the participants were born blind, and the other four started losing sight in their twenties. The group averages in age 39.8 years (Standard Deviation 4.83, Minimum 21, Maximum 50) and is composed of seven people who identify as men and three as women.



**Table 4** Participants collected information

#	Age	Age Blindness	Age Literacy	Professional Activity	Braille experience <sup>1</sup>	QWERTY Experience <sup>1</sup>	Voice input experience <sup>1</sup>
P1	48	0	7	Public agent	5	5	4
P2	39	21	21	Public management technician	3	4	3
P3	46	22	7	Professor	5	5	5
P4	41	29	33	Professor	3	5	5
P5	50	0	8	Professor	5	4	2
P6	38	0	21	Public agent	2	4	4
P7	40	0	5	Psychologist	4	5	3
P8	38	19	20	Psychologist	2	4	2
P9	37	0	9	Public agent	5	5	5
P10	21	0	9	Undergrad Computer Science student	5	5	4

<sup>1</sup>Self-declared in a Likert Scale of 1 to 5

All participants learned Braille as a child or a few years after their blindness developed. They all use screen readers intensively on smartphones, where typing is mostly performed with QWERTY, sometimes replaced by voice input for quick reply messages. All participants use Android's Talkback, but P1 and P3, who use iPhone's VoiceOver. Moreover, only P2 had previous contact with smartwatches, but without accessibility features nor any method to input text. Participants (mean) reported experiences with Braille system (3.9), QWERTY layout (4.6), and Voice Input text entry (3.7) on a 5 level Likert Scale.

### 4.3 Hardware and software

Our prototypes were tested on an ASUS ZenWatch2 005R. This smartwatch has a built-in speaker with Google WearOS 2.0. The overall watch face is  $49.6 \times 40.7$ mm, but the screen itself, or the clickable area is a square touchscreen asymmetrically positioned over the face, with  $320 \times 320$  pixels ( $\sim 42.5\%$  screen-to-body ratio). In Braille methods, this means around  $9.66 \times 9.66$ mm dots clickable area. The fact that there is not a tactile cue to indicate the transition from the clickable area is a particularity reported as an issue by early testers of our research. One user suggested to cover the non-clickable area with a layer of "scotch tape." The solution is cheap and sufficient to differentiate texture and was used during tests (Fig. 1). We highlight that this responds to an issue which is not readily apparent in other existing watch models.

The three third-party non-Braille methods were used as provided from the Google Play Store for wearable devices, without any modification. They were all downloaded and installed by August of 2019. The three Braille input methods have their source code published on GitHub [29] in their latest version, as used in the tests.

## 5 Results

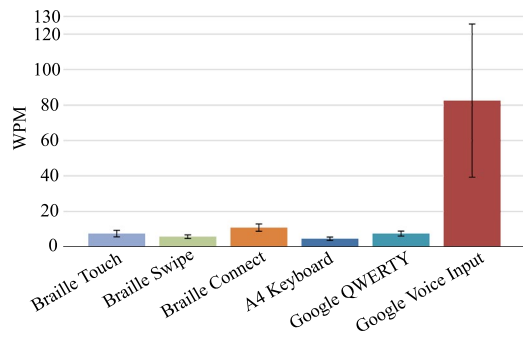
Among the ten participants, only P3 and P7 gave up completing some method, both of them could not work with A4 Keyboard and Braille Swipe after training steps. In these situations, any evaluation regarding the method received the worst score possible, except for the preference order, which was still asked. The data shown ahead for text entry speed and error rate consider values without any penalties for them. Only P3 opted for using the screen rotated  $90^\circ$ , while P8 and P9 chose the writing order (4-5-6-1-2-3) for the arrangement of the dots. The following are our quantitative and qualitative results.

### 5.1 Text entry speed

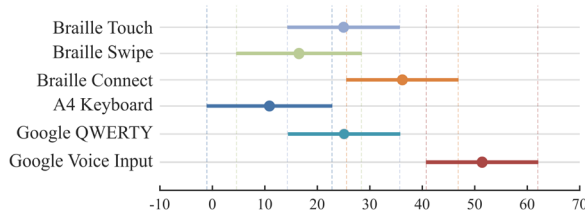
The results for speed evaluation is seen in Fig. 5a.

As expected, voice input provides the fastest entry rate with 82.53 WPM average (Standard Deviation:  $\sigma = 44.96$ , min 16.69, max 264.95) which is much faster than any of the typing methods. We note that the recognition itself lasted less than two seconds most of the time, but the longer duration for some users was due to their difficulty in finding the send button and activating it, even with instructors speaking out loud its label. A few users did not have any sentence recognized at first trial, needing to scan the interface for the re-trial button.

Connect was the fastest among the Braille methods, (Mean:  $\bar{x} = 10.89$  WPM,  $\sigma = 0.75$ , min 7.32, max 17.81) as might be expected from its lack of a confirmation step. Touch is in third place with 7.51 WPM ( $\sigma = 2.91$ , min 3.41, max 12.14), which is much closer to Google QWERTY with 7.23 WPM ( $\sigma = 0.82$ , min 3.59, max 9.73). This result demonstrates the viability of the QWERTY method, which was considered surprising even



(a) Mean values of Text Entry Speed in WPM.



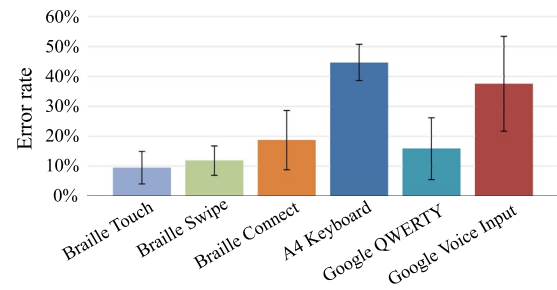
(b) Multiple comparison with Bonferroni for Kruskal-Wallis test.

**Fig. 5** Results for obtained speed values

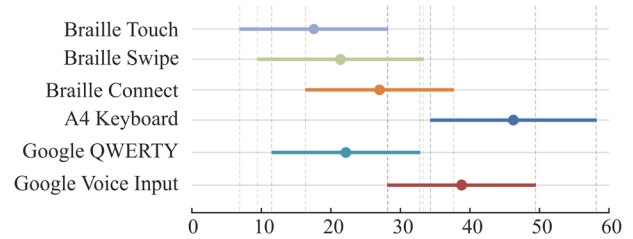
by some participants who believed that the small letters would make the interaction unfeasible.

Braille Swipe ( $\bar{x} = 5.78$  WPM,  $\sigma = 0.41$ , min 3.41, max 8.25) and A4 Keyboard ( $\bar{x} = 4.19$  WPM,  $\sigma = 0.80$ , min 2.76, max 7.44) are the slower methods, also featuring two withdrawals. The first divided participants between those who could easily perform the swipes and those who could barely activate one dot. We discuss this disparity later. A4 Keyboard was considerably slower due to the time that users would waste searching for keys. The position where the scroll would stop varies according to the intensity applied, so one could believe to be in the end of the keyboard when it was still in the middle.

only the data from Voice Input provided a normal distribution, verified via a *Lilliefors test*; thus, we could not apply standard One-way ANOVA. A *Kruskal-Wallis* test was applied ( $Chisquare = 36.51, p = 7.50e^{-7}, df = 5$ ) instead, meaning a statistical difference between the methods. By performing multiple comparisons with Bonferroni correction, Fig. 5b is generated, where we can have a more in-depth analysis. The lack of intersection between methods indicate, for instance, that Voice Input has significant statistical difference against all methods ( $p < 0.05$ ), except for Braille Connect. A4 Keyboard only has a significant statistical difference when compared with speed values for Braille Connect and Voice Input.



(a) Mean values for Text Entry Error Rate percent obtained from MSD.



(b) Multiple comparison with Bonferroni for Kruskal-Wallis test.

**Fig. 6** Results for obtained error rate values

## 5.2 Text entry accuracy

Figure 6a has the mean value for Error rate calculated with Formula 2 from the MSD obtained with each method.

The A4 Keyboard resulted in extremely high error rates ( $\bar{x} = 44.63\%$ ,  $\sigma = 8.76\%$ , min 35.64%, max 58.45%), particularly due to incorrect insertions of characters while the users scrolled among keys. With Talkback activated, scroll is performed when the user swipes two fingers at the same time over the screen. However, if the user starts this movement by touching with one finger a few milliseconds before the other, the movement can be interpreted by the system as a double touch. Furthermore, in some cases, participants confused a Talkback character's utterance with the confirmation of insertion of that character, leading to believe that the symbol was inserted when, in fact, it was not.

Notably, voice input also features poor error rate ( $\bar{x} = 37.52\%$ ,  $\sigma = 25.56\%$ , min 0%, max 67.47%). This happens less because of bad text processing and more as a consequence of a common use case for these users: Talkback's pronunciation would be transcript mixed with the user sentence, or completely take over it. This happens mostly because, as seen in Fig. 3e, the initial state of the voice input screen contains a "Speak now..." label that is pronounced automatically by the screen reader when the text appears. If the user does not begin speaking fast enough, it seems that its voice loses priority over the screen reader. A couple of participants reported this as a known issue among them. Despite that, one user could adapt and obtain a 0% error rate.

Connect ended up with a poor error rate ( $\bar{x} = 18.75\%$ ,  $\sigma = 16.00\%$ , min 1.92%, max 57.58%). The high standard deviation is a consequence of P5, in particular, who had more difficulties with this method. Despite that, it is part of its nature that, the timeout and trace strategy are susceptible to errors. For example, when composing the letter “o”, a participant must swipe through dots 1, 3 and 5. In this situation, he or she might accidentally pass over the dot 2 while tracing from 1 to 3, if not enough distance is applied to the center of the Braille cell. Another common situation is trying to compose a letter only by tapping and lifting fingers within the timeout, but not being able to reach a dot on time. These would insert the letter corresponding to the dots activated so far, clean all the dots, and later insert the one related to the dot remaining. Being unable to perform any dot’s correction after insertion, users notably needed to focus more on this method.

Google QWERTY results were also affected by some participants (P5, P7), but the majority were able to tackle its issues ( $\bar{x} = 15.90\%$ ,  $\sigma = 16.69\%$ , min 3.87%, max 44.34%). One example of a common error is inserting a letter that the participant heard from the screen reader as being the correct one, when actually it was not—the Talkback’s pronunciation of the letter “u” resembles “o” for some users, as these phonemes sound similar in Portuguese. Another recurrent situation is a key being pressed for a time long enough to be interpreted as a long press. For some letters, this would cause the opening of a popup layer with a list of variations for that symbol, making it hard for the user to leave this context without a wrong insertion.

Braille Swipe and Braille Touch were the more accurate methods, due to their nature of confirmation previous to insertion. Braille Swipe ( $\bar{x} = 11.89\%$ ,  $\sigma = 7.07\%$ , min 3.84%, max 25.53%) should prevent common cases of dot activation while exploring screen, but sometimes users performed diagonal gestures to close to the middle, causing a dot 3 activation, for example, to result on a dot 2 activation. Finally, Braille Touch ( $\bar{x} = 9.53\%$ ,  $\sigma = 8.56\%$ , min 0%, max 23.85%) demands mostly a good spatial understanding of where buttons are on screen, which was easily achieved and could be improved over time. Participants P4 and P5 completed all phrases without errors using Braille Touch.

We apply a *Kruskal-Wallis* test ( $Chisquare = 21.1, p = 8.00e^{-4}, df = 5$ ), resulting in significant statistical difference and again generating Fig. 6b for multiple comparisons. Braille Connect, Braille Swipe,

and Google QWERTY have significant statistical differences only with A4 Keyboard. While, Braille Connect and Google Voice Input have no groups with a significant statistical difference for the error values.

### 5.3 Task load index

The NASA-TLX questionnaire helps to analyze the user-perceived demands and the results of the typing tasks. It is composed of six questions aforementioned, where all but the *Performance* one are negative impact metrics, thus the results mean that the lower, the better. However, to help visualize the results as a group, in this section, we transformed the *Performance* values from maximization to minimization, also indicating that the lower, the better.

The results for *Kruskal-Wallis* test are summarized in Table 5, where it can be seen that *Temporal Demand* and *Performance* were the only to not achieve some significant statistical difference ( $p < 0.05$ ). Figure 7 shows, for example, the apparent low demand and effort necessary for using voice input, but also some level of frustration compared to other methods. Physical Demand shows us how gestures of Braille Swipe could be heavier than others, and also some effort on using Google QWERTY when compared to Braille Touch, for example. There is no significant impact on the Braille system’s usage on Mental Demand, and the A4 Keyboard is conclusively the keyboard that provided the worst experience.

### 5.4 Users preference

Figure 8 provides a view of how each method was scored at the end of its evaluation. Table 6 shows values for Mean ( $\bar{x}$ ), Median ( $\tilde{x}$ ) and Mode (*Mo*), for each method. Except for A4 Keyboard, all methods had values of at least 3.5 mean. No methods scored a mean above 4.3, which may suggest to the need for improvements even for the highest-ranked method.

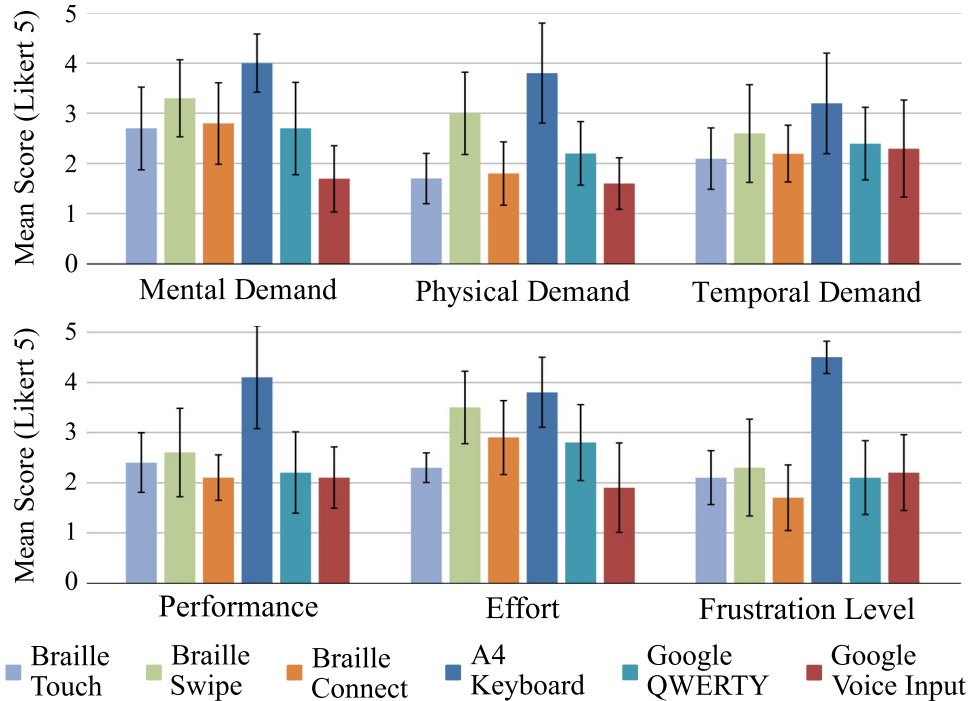
The *Kruskal-Wallis* test results in some statistical significance ( $Chisquare = 29.97, p = 1.49e^{-5}, df = 5$ ). Multiple comparisons with Bonferroni correction demonstrates that A4 Keyboard is significantly less approved than every method excepting for Swipe, but we cannot affirm that there are sufficient difference among the other methods.

Considering these values, it seems to exist a preference for Braille Touch and Connect methods among our participants, but analyzing the statistics and final preference

**Table 5** NASA-TLX results for *Kruskal-Wallis* test

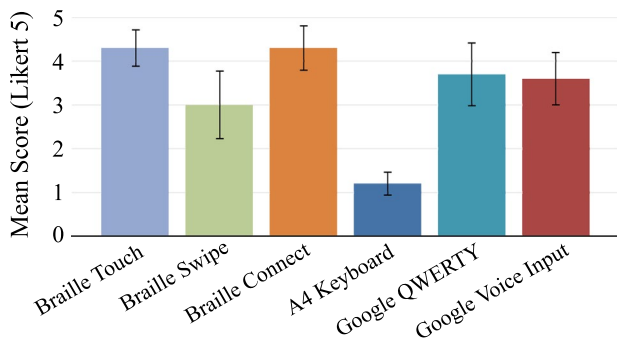
	Mental demand	Physical demand	Temporal demand	Performance	Effort	Frustration level
$\chi^2$	15.45	16.86	3.26	10.25	17.17	23.14
$p$	0.0086	0.0048	0.6604	0.0684	0.0042	0.0003
$df$	5	5	5	5	5	5

**Fig. 7** Mean values for each question from NASA-TLX

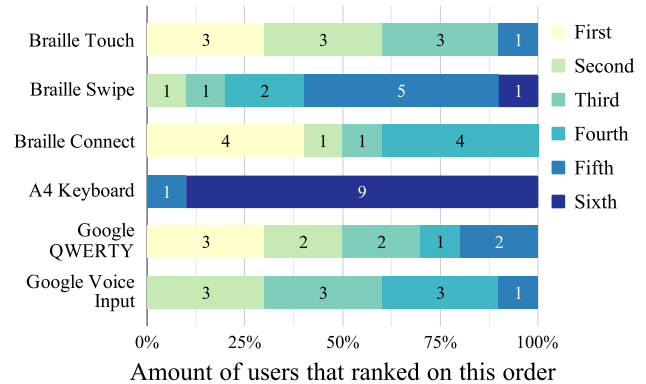


**Table 6** Score values obtained for each method

	Braille touch	Braille swipe	Braille connect	A4 keyboard	Google QWERTY	Google voice
$\bar{x}$	4.3	3.0	4.3	1.2	3.7	3.6
$\hat{x}$	4.0	3.5	4.0	1.0	4.0	3.5
$Mo$	4.0	4.0	5.0	1.0	5.0	3.0



**Fig. 8** How much do you like the method? 5 level Likert scale, mean of values



**Fig. 9** Amount of participant preference ranking for each method

order offered by the end of the test (Fig. 9), Google QWERTY still has a proper approval of some participants. Voice input is never the first option, but it often features among second or third as it is seen as a good complementary method.

**5.5 Users interest in the methods**

Our last questions on the protocol aimed to grasp the user interest in the usage of any of the methods. All ten participants affirmed they would use both Braille and Voice Input methods in case they owned a smartwatch. Only participant P9 affirmed that she would not type with any of the

QWERTY methods. Tackling specifically their opinion on purchasing the device, nine participants affirmed their plans to buy a smartwatch in case some of the Braille methods were available.

## 6 Discussion

Text entry on smartwatches is an emerging issue for blind people. A discussion that this work does not dive in deeply is whether there is real interest on this type of device, even though participants demonstrated it. In this sense, a comparison of Braille input methods, and existing methods, such as QWERTY keyboards and Voice input, is very important.

Below, we present our analysis of the results regarding each of the studied input modalities, to summarize the understanding acquired by this research.

### 6.1 Braille input

Our proposal of Braille input for smartwatches has shown they are not only feasible as a text input method but also desirable by most participants. In this sense, we have learned different lessons from each of the three proposed methods.

Braille Touch is the most basic interaction proposed, and that is positively received by users, ending the tests as the most preferred. With a few minutes of practice, one could memorize the position of each dot and quickly dominate it. The main issue reported by participants was the impossibility of checking which dots are active if for a moment a wrong activation seems to have happened. In our application, this is possible by performing a long press, when the typed sentence is spoken, and the dot state is reported. However, we left that feature disabled for fair competition with the QWERTY methods, where the user should not reach the input field. Due to that, if, in any case, the user felt that he or she had activated a wrong dot, it would need to test it again by observing if a sound is emitted (activation) or only a vibration (deactivation). Even for the participants that felt more insecure with the distribution of points, we believe practice or other tactile cues could improve the experience. There is even the possibility of keeping dot number announcement on for first time users. The double click for confirmation was questioned by some users, who believe a single click on the middle should be sufficient.

Braille Connect offers fast input but with the cost of more possibilities of errors. One of its main disadvantages is that once a dot is wrongly activated, there is no way to deactivate it. Overall, it is a method for users who are more experienced, as there is a pressure over time composition or spatial awareness that is not present on other methods. Still, even when users committed errors, they tend to get satisfied by

typing faster with this method, which leads to a high score in Fig. 8.

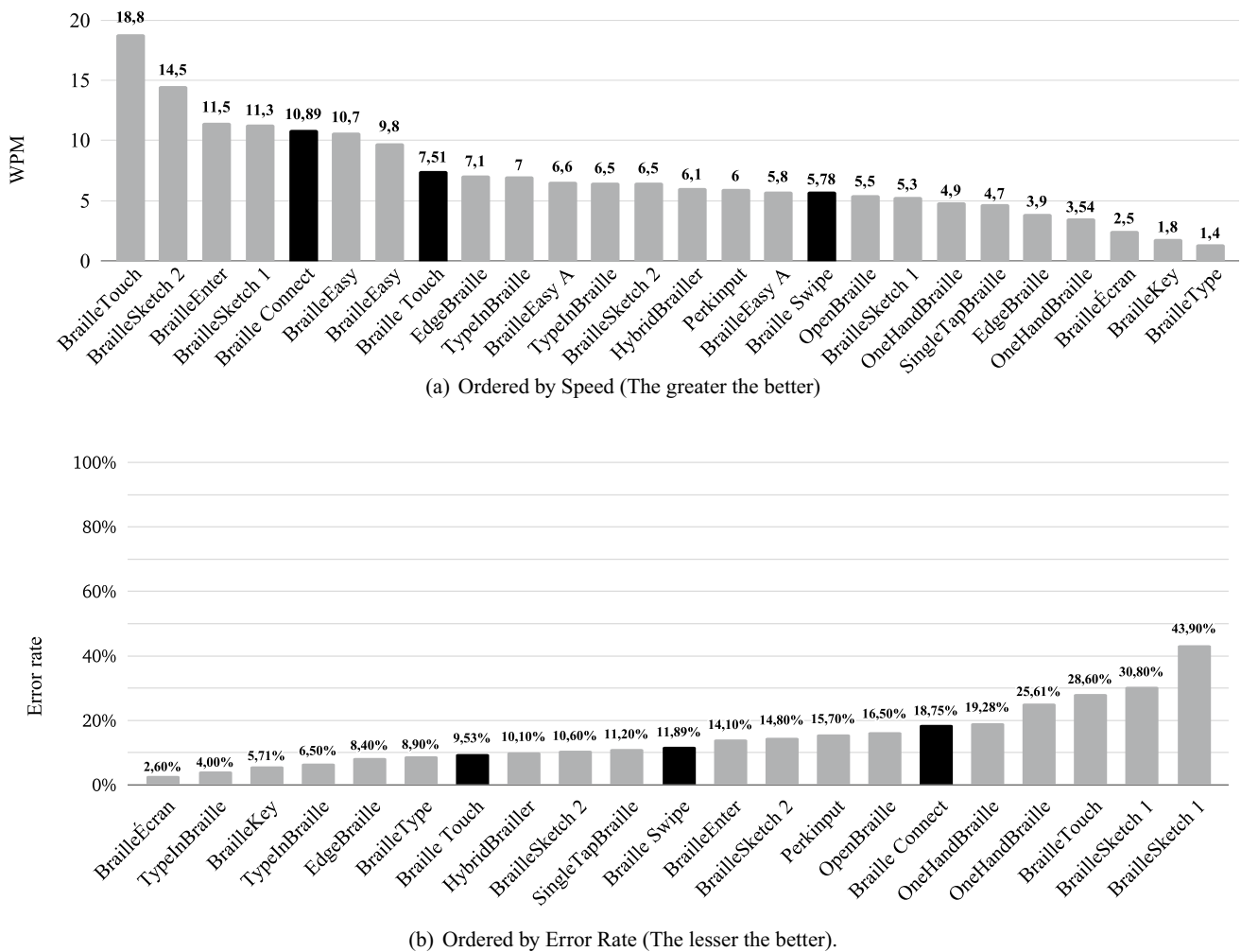
Braille Swipe can provide a safe insertion of letters, without the need to precisely hit a dot region, but its interaction is not for everybody. Swipe gestures on a small screen, which was supposed to be a simple task, ended up being surprisingly hard for some users. The participants that faced this issue tended to swipe with their fingers heavily over the screen, apparently applying pressure in a slow drag movement, sometimes even twisting the watch position. This happens more often when direction involved pushing instead of pulling the fingers, which might be related to the friction created. Even with instructions regarding how to perform this, participants P3 and P7 could not activate a single dot, which resulted in the two withdrawals. It is not clear to us how Android gestures library handle this internally, and it is hard to reach conclusions as while some had this issue, other users could perform every gesture quickly without any problem. Even so, not a single participant preferred this method over at least one of the other Braille methods.

Finally, it is worth noting that compared to existing Braille methods for smartphones proposed in related work, the results achieved, especially by Braille Touch, are encouraging. We analyzed the results from Table 1 and ordered them by speed on Fig. 10a, adding our smartwatch results. The speed rates obtained by Braille Touch and Braille Connect are superior to most of the methods listed for smartphones, e.g., multitouch ones, except for BrailleTouch (17.8 WPM) [17], BrailleSketch (14.5 WPM) [28] and BrailleEnter (14.5, 11.3) [4]. However, observing other variables such as number of sessions, number of participants and the need for both hands on device, we can argue that the results are encouraging as noted.

Error rates, however, give mixed results, negatively for Braille Connect and positively for Touch. As it is common in text entry studies, slower methods tend to be safer, and that can be seen in Fig. 10b, where the previous studies for smartphones and ours are ordered by error rate. There is, of course, ingenuity on any comparison, as a test with the same protocol would be required. Most of these studies, for example, tested the correction of errors. We hypothesize, however, that the small screen may benefit users. Having less space to explore, easier memorization of the dot disposal may happen, which could make the Braille methods suitable in scenarios for use with smartwatches.

### 6.2 QWERTY keyboard input

One of our initial hypotheses that motivated the development of Braille alternatives is that QWERTY input on such a small screen would be unfeasible. The results of Google QWERTY prove us wrong, showing that the user experience with QWERTY keyboards and screen readers was enough



**Fig. 10** Existing Braille methods for smartphones and our methods for smartwatches

for a short period of adaptation. This good reception to a layout that is familiar has been reported in the previous study for touch screen devices [25]. Yes, there are challenges here to be addressed, faced by users with larger fingers or any motor coordination, but our research cannot present data related to these specifics.

In comparison with the results commented on related work, we have a better speed entry rate than the ones reported for smartphones (see Table 2) against our 7.23 WPM, excepting only to BrailleEasy participants, that achieved 9.1 WPM, but even then, in a study with only three participants. Regarding the error rate, the value of 15.9% is high but not much distant from the results seen on smartphones neither. A reason that we speculate could favor speed and disfavor error rate is the fact that the small screen size provides a lesser spatial area to explore. Thus, with concentration, users could navigate quickly through keys in a layout that they have evident expertise. There is still much room

for improvement here when compared to the values already possible for sighted people.

Not much commenting is necessary regarding the A4 Keyboard. Not only it was not well developed to be read by a screen reader, but it also proposes an interaction that becomes harsh for blind users. We highlight, however, that this was one of the third-party keyboards tested that had minimum accessibility support, pronouncing at least its characters. Even worse, it is more usable than others, which is something that exposes the poor accessibility quality in this platform.

One convention of our study that privileges QWERTY keyboards over Braille is the lack of numbers and special characters. For Braille methods these would require merely insertion of other symbols, while for any of the QWERTY methods those add extra layers of complexity, switching to new layouts. We avoided allowing users to check the written sentence on the text input field, to prevent the error of

selecting part of the phrase and accidentally replace it with new text.

### 6.3 Voice input

Most smartwatches available on the market, when providing any text input, do it by offering voice input mechanisms. It is clear that this technology has evolved and matured, becoming a handy tool for users. Nevertheless, we decided on testing this method to compare the interaction experience. As in all other methods, the user needs to complete the typing task by reaching a “Send button,” which took some time for users and was even assisted by instructors, as the current application for Google Voice Input does not provide proper labels to buttons. This was a lesser issue compared to the conflict between detecting the user voice and Talkback’s synthesized speech. We believe, however, that this error could be easily overcome either by users, by initializing their speech before Talkback’s feedback, or by developers, handling the input source. This was reported by participants as an open problem, which they face daily. One participant said that she tends to decrease the accessibility volume before using this method on the smartphone so that the same error does not happen.

Even so, it is the most practical method and effortless (see Fig. 7 results for Effort), and should be preferred in many situations. Some issues as privacy while speaking are discussed in other studies [5, 45], and we firmly believe that recognition challenges may still exist for some languages, something reported by most participants. Indeed, we suggest that a hybrid solution, with speech and typing strategies is still necessary.

### 6.4 The importance of specialists feedback

This project followed an incremental strategy of implementation of methods and features, as it strongly relied on specialists’ feedback. It may be obvious to mention, but we believe it is important to stress out, that a study coordinated by sighted people will not come up with real solutions without the participation of experts, no matter how much literature review is performed.

our first published work led us to the decisions regarding the study protocol presented in Sect. 4. With the analysis carried out by specialists, we were able to realize the need for providing a tactile cue to indicate where the touchable screen actually begins in our smartwatch. We also learned the importance of offering both writing and reading layouts of the Braille dots distribution in our prototype, which also received new features, such as character confirmation and space insertion, based on double taps instead of swipes, something also suggested in the early study. Henceforth, we could see how important it is to perform a more in-depth

study before we could tackle text entry processes and even compare our ideas with existing methods, as we do here. In the end, it was a necessary step to prepare our testing workflow and narrow down our options to the three methods used in this work.

This incremental work made us more confident to raise the discussion with new findings and propose more mature interactions as the complexity of the tasks evaluated increased.

### 6.5 Adoption of text entry on smartwatches for blind people

We believe that adoption of smartwatches by a blind user involves a profound market investigation and also research on new accessibility for other tasks that could be useful. This study was developed in a large city in the middle of a huge country, where smartwatches are yet to penetrate the consumer market. Considering the related work mentioned and amenities such as having arms free of holding a device, we believe blind people will have an interest in using a smartwatch in future. Again, this is a supposition but is something we feel from the discussion with participants. In fact, a few months after completing the tests, participant P1 purchased an Apple Watch, reinforcing his desire to have this type of device.

If we assume that this adoption would happen, offering text entry techniques is a necessity. Voice input is a very effective solution nowadays, but as shown, its error rate creates limitation in its adoption, because of excessive corrections might be necessary on an extra method and in some way there is some lack of awareness. If sighted users can use traditional keyboards with efficient tracing features, then blind people should have their accessible alternatives too.

### 6.6 Challenges and limitations

This project faces some methodological limitations. A concerning one is the fact that we had few users to perform our evaluations. This happens mainly due to the specificity of our public (people who are blind and literate in Braille) as well as their availability and willingness to engage in experimental sessions of considerable duration and complexity.

Finding ten participants apart from the seven consulted in the previous study, with these characteristics was challenging. In our experiments, some participants reported that because of the growing number of screen reader tools, Braille literacy level is decaying more and more. The participants were kind enough to offer their time and opinions without any financial return.

It should be clarified that each test involved driving a participant from and to work or home, and ensuring a private and neutral environment for tests, making it rare the

case of more than one test performed in a day. That said, we acknowledge that the diversity of users capabilities and limitations requires a much broader analysis. With those ten users, we were not able to make more in-depth comparisons regarding age or literacy level effects, for example. Although, this means that our results do not represent completely the universe that we are studying, it provides us important insights for future works.

The large amount of methods evaluated and the context of our participants also resulted in test design decisions that certainly limit the impact of our analysis, such as a reduced set of phrases and single session tests. We believe, however, that the results obtained still provide a second delimitation of options (being the first the pilot study previously published [30]), that shall be useful for the next step on this research.

## 7 Conclusion

In this work, we propose a novel approach for Braille text in smartwatches based on different types of text entry methods, and compare to existing input methods on WearOS. To construct our approaches, the initial prototypes were discussed and evaluated by blind users, who supported the selection of the three proposed text entry methods, which are referred as Braille Touch, Braille Swipe and Braille Connect.

although all methods still need some improvement, the proposed braille input methods presented outstanding results when compared to existing solutions on smartphones, proving the feasibility of such task which could be considered challenging at first. Braille Touch and Braille Connect presented the best Speed and Error Rate relation and a very good user preference; however, Google QWERTY keyboard and Google voice input method also presented potential as input methods, even though exposing some of its issues.

This work concludes that the proposed methods have potential to be used as text entry method. We believe there is valuable contribution here to any future research in this area, and also proof that there is a lot to be studied and proposed in order to offer accessible text entry methods for blind people on smartwatches.

## 8 Future work

For future work, we plan to investigate the features left unexplored here, such as character correction, punctuation, and even auto completion. Some Braille composing strategies recently published as BrailleEnter [4] were not investigated here. Also, screen size and shapes (square, rounded) open another field of inquiry.

Another step is comparing the existing solutions for smartphone (Braille or not) against the tested methods on smartwatches. This could be complemented by a broader study, on the execution of daily tasks by blind people with smartwatches.

**Acknowledgments** The authors would like to thank our participants for the valuable contribution, dedication, patience and time. We hope our findings inspire work that help address your needs. Moreover, authors would like to thank CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior—Brazil) [CAPES Finance Code #001] and [TED 7371 - CAPES # 3/2018 - Accessibility Tools -*Ferramentas de acessibilidade*] for the financial support.

## Declarations

**Conflict of interest** On behalf of all authors, the corresponding author states that there is no conflict of interest in this work.

## References

1. Abdolrahmani, A., Kuber, R., Hurst, A.: An empirical investigation of the situationally-induced impairments experienced by blind mobile device users. In: Proceedings of the 13th Web for All Conference, W4A '16, pp. 21:1–21:8. ACM, New York, NY, USA (2016). <https://doi.org/10.1145/2899475.2899482>. <http://doi.acm.org/10.1145/2899475.2899482>
2. Albanna, M.M.: Swiftbraille - connect the dots. (2017). <http://en.swiftbraille.com/>. [Last accessed December 24, 2018]
3. Alnfai, M., Sampalli, S.: 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, pp. 161–169. ACM, New York, NY, USA (2016). <https://doi.org/10.1145/2982142.2982161>
4. Alnfai, M., Sampalli, S.: BrailleEnter: a touch screen Braille text entry method for the blind. *Proc. Comput. Sci.* **109**, 257–264 (2017)
5. Azenkot, S., Lee, N.B.: 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, pp. 11:1–11:8. ACM, New York, NY, USA (2013). <https://doi.org/10.1145/2513383.2513440>
6. Bardot, S., Serrano, M., Jouffrais, C.: From tactile to virtual: using a smartwatch to improve spatial map exploration for visually impaired users. In: Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI '16, pp. 100–111. ACM, New York, NY, USA (2016). <https://doi.org/10.1145/2935334.2935342>
7. Billah, S.M., Ashok, V., Ramakrishnan, I.: Write-it-yourself with the aid of smartwatches: a wizard-of-Oz experiment with blind people. In: 23rd International Conference on Intelligent User Interfaces, IUI '18, pp. 427–431. ACM, New York, NY, USA (2018). <https://doi.org/10.1145/3172944.3173005>
8. Billah, S.M., Ko, Y.J., Ashok, V., Bi, X., Ramakrishnan, I.: Accessible gesture typing for non-visual text entry on smartphones. *Proc. 2019 CHI Conf. Human Fact. Comput. Syst.* **2019**, 1–12 (2019). <https://doi.org/10.1145/3290605.3300606>
9. Bonner, M.N., Brudvik, J.T., Abowd, G.D., Edwards, W.K.: No-look notes: accessible eyes-free multi-touch text entry. In: International Conference on Pervasive Computing, pp. 409–426. Springer (2010). [https://doi.org/10.1007/978-3-642-12654-3\\_24](https://doi.org/10.1007/978-3-642-12654-3_24)



10. Britto Neto, L.d.S., Maike, V.R.M.L., Koch, F.L., Baranauskas, M.C.C., Rocha, A.d.R., Goldenstein, S.K.: A wearable face recognition system built into a smartwatch and the visually impaired user. In: Proceedings of the 17th International Conference on Enterprise Information Systems, pp. 5–12. SCITEPRESS - Science and Technology Publications, Barcelona, Spain (2015). <https://doi.org/10.5220/0005370200050012>
11. Cha, J.M., Choi, E., Lim, J.: Virtual sliding QWERTY: a new text entry method for smartwatches using Tap-N-Drag. *Appl. Ergon.* **51**, 263–272 (2015)
12. Chen, X.A., Grossman, T., Fitzmaurice, G.: Swipeboard: a text entry technique for ultra-small interfaces that supports novice to expert transitions. In: Proceedings of the 27th annual ACM symposium on User interface software and technology, pp. 615–620. ACM Press (2014). <https://doi.org/10.1145/2642918.2647354>
13. Dalton, D.: Softbraillekeyboard (2016). [https://docs.google.com/document/d/1OX6yaFjkK61VfROdU9uQNxjEZ\\_QNwxiMnUY\\_1ECnYRQ/pub](https://docs.google.com/document/d/1OX6yaFjkK61VfROdU9uQNxjEZ_QNwxiMnUY_1ECnYRQ/pub). [Last accessed December 24, 2018]
14. Dhar, A., Nittala, A., Yadav, K.: TactBack: VibroTactile Braille output using smartphone and smartwatch for visually impaired. In: Proceedings of the 13th Web for All Conference, W4A '16, pp. 13:1–13:2. ACM, New York, NY, USA (2016). <https://doi.org/10.1145/2899475.2899514>. <http://doi.acm.org/10.1145/2899475.2899514>
15. Dobosz, K., Szuścik, M.: OneHandBraille: An Alternative Virtual Keyboard for Blind People. In: Man-Machine Interactions 5, Advances in Intelligent Systems and Computing, pp. 62–71. Springer, Cham (2017). [https://doi.org/10.1007/978-3-319-67792-7\\_7](https://doi.org/10.1007/978-3-319-67792-7_7). [https://link.springer.com/chapter/10.1007/978-3-319-67792-7\\_7](https://link.springer.com/chapter/10.1007/978-3-319-67792-7_7)
16. Frey, B., Rosier, K., Southern, C., Romero, M.: From texting app to Braille literacy. In: CHI '12 Extended Abstracts on Human Factors in Computing Systems, CHI EA '12, pp. 2495–2500. ACM, New York, NY, USA (2012). <https://doi.org/10.1145/2212776.2223825>
17. Frey, B., Southern, C., Romero, M.: BrailleTouch: mobile texting for the visually impaired. In: C. Stephanidis (ed.) Universal Access in Human-Computer Interaction. Context Diversity, vol. 6767, pp. 19–25. Springer Berlin Heidelberg, Berlin, Heidelberg (2011). [https://doi.org/10.1007/978-3-642-21666-4\\_3](https://doi.org/10.1007/978-3-642-21666-4_3)
18. Gaggi, O., Palazzi, C.E., Ciman, M., Bujari, A.: StepByWatch: a smartwatch-based enhanced navigation system for visually impaired users. In: 2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC), pp. 1–5. IEEE, Las Vegas, NV (2018). <https://doi.org/10.1109/CCNC.2018.8319311>
19. Gordon, M., Ouyang, T., Zhai, S.: WatchWriter: tap and gesture typing on a smartwatch miniature keyboard with statistical decoding. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, pp. 3817–3821. ACM Press (2016). <https://doi.org/10.1145/2858036.2858242>. <http://dl.acm.org/citation.cfm?doid=2858036.2858242>
20. Guerreiro, T., LagoÁ, P., Nicolau, H., Gonçalves, D., Jorge, J.A.: From tapping to touching: making touch screens accessible to blind users. *IEEE MultiMedia* **15**(4), 48–50 (2008)
21. Gupta, A., Balakrishnan, R.: DualKey: miniature screen text entry via finger identification. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, pp. 59–70. ACM Press (2016). <https://doi.org/10.1145/2858036.2858052>. <http://dl.acm.org/citation.cfm?doid=2858036.2858052>
22. Hong, J., Heo, S., Isokoski, P., Lee, G.: SplitBoard: A Simple Split Soft Keyboard for Wristwatch-sized Touch Screens. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, pp. 1233–1236. ACM Press (2015). <https://doi.org/10.1145/2702123.2702273>. <http://dl.acm.org/citation.cfm?doid=2702123.2702273>
23. Hsiu, M.C., Huang, D.Y., Chen, C.A., Lin, Y.C., Hung, Y.p., Yang, D.N., Chen, M.: ForceBoard: using force as input technique on size-limited soft keyboard. In: Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct, pp. 599–604. ACM Press (2016). <https://doi.org/10.1145/2957265.2961827>. <http://dl.acm.org/citation.cfm?doid=2957265.2961827>
24. Inc., D.: Dot smartwatch (2018). <https://dotincorp.com/>. [Last accessed December 24, 2018]
25. Kane, S.K., Wobbrock, J.O., Ladner, R.E.: Usable gestures for blind people: understanding preference and performance. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '11, pp. 413–422. ACM, New York, NY, USA (2011). <https://doi.org/10.1145/1978942.1979001>. 00206
26. Lee, D., Lee, S., Yi, J., Kang, H., Oakley, I.: HexaBraille: touch-based text entry method on smart watches for blind users. *International Conference on Ergonomics* pp. 509–509 (2017). <http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE07301939>
27. Leporini, B., Buzzi, M.C., Buzzi, M.: Interacting with mobile devices via VoiceOver: usability and accessibility issues. In: Proceedings of the 24th Australian Computer-Human Interaction Conference on - OzCHI '12, pp. 339–348. ACM Press, Melbourne, Australia (2012). <https://doi.org/10.1145/2414536.2414591>. <http://dl.acm.org/citation.cfm?doid=2414536.2414591>
28. Li, M., Fan, M., Truong, K.N.: BrailleSketch: a gesture-based text input method for people with visual impairments. In: Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility - ASSETS '17, pp. 12–21. ACM Press, Baltimore, Maryland, USA (2017). <https://doi.org/10.1145/3132525.3132528>. <http://dl.acm.org/citation.cfm?doid=3132525.3132528>
29. Luna, M.M.: Wearable braille input proposals on github (2019). <https://github.com/pixellab-ufg/WearableBraille>. [Last accessed April 11, 2021]
30. Luna, M.M., de M. N. Soares, F.A.A., Nascimento, H.A.D., Quigley, A.: Braille text entry on smartwatches: an evaluation of methods for composing the Braille cell. In: Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems, EICS '19, pp. 19:1–19:6. ACM, New York, NY, USA (2019). <https://doi.org/10.1145/3319499.3328233>. <http://doi.acm.org/10.1145/3319499.3328233>
31. Luna, M.M., Soares, F.A.A.M.N., Nascimento, H.A.D., Siqueira, J., Souza, E.F., Nascimento, T.H., Costa, R.M.: Text entry on smartwatches: a systematic review of literature. 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC) **02**, 272–277 (2018). <https://doi.org/10.1109/COMPSAC.2018.10242>
32. MacKenzie, I.S., Soukoreff, R.W.: Phrase sets for evaluating text entry techniques. In: CHI '03 Extended Abstracts on Human Factors in Computing Systems, CHI EA '03, pp. 754–755. ACM, New York, NY, USA (2003). <https://doi.org/10.1145/765891.765971>. <http://doi.acm.org/10.1145/765891.765971>
33. MacKenzie, I.S., Tanaka-Ishii, K.: Text Entry Systems: Mobility, Accessibility, Universality. Morgan Kaufmann, Burlington (2010)
34. Mascetti, S., Bernareggi, C., Belotti, M.: TypeInBraille: quick eyes-free typing on smartphones. In: Computers Helping People with Special Needs, Lecture Notes in Computer Science, pp. 615–622. Springer, Berlin, Heidelberg (2012). [https://doi.org/10.1007/978-3-642-31534-3\\_90](https://doi.org/10.1007/978-3-642-31534-3_90). [https://link.springer.com/chapter/10.1007/978-3-642-31534-3\\_90](https://link.springer.com/chapter/10.1007/978-3-642-31534-3_90). 00023
35. Mattheiss, E., Regal, G., Schrammel, J., Garschall, M., Tscheligi, M.: Dots and letters: Accessible braille-based text input for visually impaired people on mobile touchscreen devices. In: K. Miesenberger, D. Fels, D. Archambault, P. Penaz, W. Zagler (eds.) Computers Helping People with Special Needs, *Lecture Notes in Computer Science*, vol. 8547, pp. 650–657. Springer

- International Publishing (2014). [https://doi.org/10.1007/978-3-319-08596-8\\_100](https://doi.org/10.1007/978-3-319-08596-8_100)
36. Montague, K., Guerreiro, T.: Towards inviscid text-entry for blind people through non-visual word prediction interfaces. In: CHI Workshop on Inviscid Text-Entry and Beyond, p. 5 (2016)
  37. Mottelson, A., Larsen, C., Lyderik, M., Strohmeier, P., Knibbe, J.: Invisiboard: maximizing display and input space with a full screen text entry method for smartwatches. In: Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services, pp. 53–59. ACM Press (2016). <https://doi.org/10.1145/2935334.2935360>. <http://dl.acm.org/citation.cfm?doi=2935334.2935360>
  38. NASA, N.A.S.A.: The official nasa task load index (2019). <https://humansystems.arc.nasa.gov/groups/TLXI/>. [Last accessed May 12, 2019]
  39. Nascimento, T.H., Soares, F.A.A.M.N., Irani, P.P., Oliveira, L.L.G.d., Soares, A.d.S.: Method for text entry in smartwatches using continuous gesture recognition. In: 2017 IEEE 41st annual Computer software and applications conference (COMPSAC), pp. 549–554. IEEE (2017). <https://doi.org/10.1109/COMPSAC.2017.168>. <http://ieeexplore.ieee.org/document/8029988/>
  40. Nicolau, H., Montague, K., Guerreiro, T., Guerreiro, J.a., Hanson, V.L.: B#: Chord-based correction for multitouch braille input. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1705–1708. ACM, New York, NY, USA (2014). <https://doi.org/10.1145/2556288.2557269>. <http://doi.acm.org/10.1145/2556288.2557269>
  41. Oliveira, J., Guerreiro, T., Nicolau, H., Jorge, J., Gonçaves, D.: Brailletype: Unleashing braille over touch screen mobile phones. In: Proceedings of the 13th IFIP TC 13 International Conference on Human-computer Interaction - Volume Part I, INTERACT'11, pp. 100–107. Springer-Verlag, Berlin, Heidelberg (2011). <http://dl.acm.org/citation.cfm?id=2042053.2042066>
  42. Oliveira, J.a., Guerreiro, T., Nicolau, H., Jorge, J., Gonçalves, D.: 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, pp. 179–186. ACM, New York, NY, USA (2011). <https://doi.org/10.1145/2049536.2049569>. <http://doi.acm.org/10.1145/2049536.2049569>
  43. Oney, S., Harrison, C., Ogan, A., Wiese, J.: ZoomBoard: a diminutive qwerty soft keyboard using iterative zooming for ultra-small devices. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, p. 2799. ACM Press (2013). <https://doi.org/10.1145/2470654.2481387>. <http://dl.acm.org/citation.cfm?doi=2470654.2481387>
  44. Organization, W.H.: Visual impairment and blindness (2018). <https://www.who.int/en/news-room/fact-sheets/detail/blindness-and-visual-impairment>. [Last accessed 2019-04-14]
  45. Pizza, S., Brown, B., McMillan, D., Lampinen, A.: Smartwatch in Vivo. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, CHI '16, pp. 5456–5469. ACM, New York, NY, USA (2016). <https://doi.org/10.1145/2858036.2858522>. <http://doi.acm.org/10.1145/2858036.2858522>
  46. Poirier, F., Belatar, M.: UniWatch: a soft keyboard for text entry on smartWatches using 3 keys. In: Kurosu, M. (ed.) Human-Computer Interaction. Interaction Platforms and Techniques, vol. 9732, pp. 341–349. Springer International Publishing, Cham (2016)
  47. Quaresma, P.: Frequency analysis of the portuguese language (2008). <https://www.mat.uc.pt/~pedro/cientificos/Cripto/CISUC-TR200803.pdf>. [Last accessed January 22, 2020]
  48. Šepić, B., Ghanem, A., Vogel, S.: BrailleEasy: One-handed Braille Keyboard for Smartphones. *Stud. Health Technol. Inform.* **217**, 1030–1035 (2015)
  49. Shao, Y.F., Chang-Ogimoto, M., Pointner, R., Lin, Y.C., Wu, C.T., Chen, M.: SwipeKey: a swipe-based keyboard design for smartwatches. In: Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services, pp. 60–71. ACM Press (2016). <https://doi.org/10.1145/2935334.2935336>. <http://dl.acm.org/citation.cfm?doi=2935334.2935336>
  50. Shibata, T., Afergan, D., Kong, D., Yuksel, B.F., MacKenzie, I.S., Jacob, R.J.: DriftBoard: a panning-based text entry technique for ultra-small touchscreens. In: Proceedings of the 29th Annual Symposium on User Interface Software and Technology, pp. 575–582. ACM Press (2016). <https://doi.org/10.1145/2984511.2984591>. <http://dl.acm.org/citation.cfm?doi=2984511.2984591>
  51. Siqueira, J., Soares, F.A.A.M.N., Ferreira, D.J., Silva, C.R.G., Berretta, L.d.O., Ferreira, C.B.R., Felix, I.M., Soares, A.d.S., Costa, R.M.d., Luna, M.M.: Braille text entry on smartphones: a systematic review of the literature. In: 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), vol. 2, pp. 521–526 (2016). <https://doi.org/10.1109/COMPSAC.2016.74>. <https://ieeexplore.ieee.org/document/7552266>
  52. Siqueira, J., Soares, F.A.A.M.N., Silva, C.R.G., Berretta, L.d.O., Ferreira, C.B.R., Felix, I.M., Luna, M.M.: Brailleécran: A braille approach to text entry on smartphones. In: 2016 IEEE 40th Annual Computer Software and Applications Conference (COMPSAC), vol. 2, pp. 608–609 (2016). <https://doi.org/10.1109/COMPSAC.2016.5>. <https://ieeexplore.ieee.org/document/7552286>
  53. Soukoreff, R.W., MacKenzie, I.S.: Measuring errors in text entry tasks: An application of the levenshtein string distance statistic. In: CHI '01 Extended Abstracts on Human Factors in Computing Systems, CHI EA '01, pp. 319–320. ACM, New York, NY, USA (2001). <https://doi.org/10.1145/634067.634256>. <http://doi.acm.org/10.1145/634067.634256>
  54. Soukoreff, R.W., MacKenzie, I.S.: Minimum string distance (msd) download web page (2001). <https://www.yorku.ca/mack/MSD/MSD.html>. [Last accessed February 24, 2019]
  55. Southern, C., Clawson, J., Frey, B., Abowd, G., Romero, M.: 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, pp. 317–326. ACM (2012)
  56. Subash, N.S., Nambiar, S., Kumar, V.: BrailleKey: An alternative Braille text inputs system: Comparative study of an innovative simplified text input system for the visually impaired. In: 2012 4th International Conference on Intelligent Human Computer Interaction (IHCI), pp. 1–4 (2012). <https://doi.org/10.1109/IHCI.2012.6481790>
  57. of The Blind, N.F.: The braille literacy crisisin america (2009). [https://www.nfb.org/images/nfb/documents/pdf/braille\\_literacy\\_report\\_web.pdf](https://www.nfb.org/images/nfb/documents/pdf/braille_literacy_report_web.pdf). [Last accessed June 18, 2018]
  58. Timmermann, J., Heuten, W., Boll, S.: Input methods for the Borg-RPE-scale on smartwatches. In: 2015 9th International Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth), pp. 80–83 (2015). <https://doi.org/10.4108/icst.pervasivehealth.2015.259220>. <https://eudl.eu/doi/10.4108/icst.pervasivehealth.2015.259220>
  59. Trindade, D., Rodrigues, A., Guerreiro, T., Nicolau, H.: Hybrid-Brailier: Combining physical and gestural interaction for mobile Braille input and editing. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18. ACM Press, Montreal QC, Canada (2018). <https://doi.org/10.1145/3173574.3173601>. <https://dl.acm.org/doi/10.1145/3173574.3173601>
  60. Turner, C.J., Chaparro, B.S., He, J.: Text input on a smartwatch QWERTY keyboard: tap vs. trace. *Int. J. Human-Comput. Interact.* (2016). <https://doi.org/10.1080/10447318.2016.1223265>
  61. Ushida, K., Sekine, Y., Hasegawa, S.: IPPITSU: A one-stroke text entry method for touch panels using Braille system. In: 2014

- IEEE 3rd Global Conference on Consumer Electronics (GCCE), pp. 374–375 (2014). <https://doi.org/10.1109/GCCE.2014.7031257>. <https://ieeexplore.ieee.org/document/7031257>
62. Vertanen, K., Memmi, H., Emge, J., Reyat, S., Kristensson, P.O.: VelociTap: investigating fast mobile text entry using sentence-based decoding of touchscreen keyboard input. In: Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, pp. 659–668. ACM Press (2015). <https://doi.org/10.1145/2702123.2702135>. <http://dl.acm.org/citation.cfm?doid=2702123.2702135>
  63. Visuri, A., Sarsenbayeva, Z., van Berkel, N., Goncalves, J., Rawasizadeh, R., Kostakos, V., Ferreira, D.: Quantifying sources and types of smartwatch usage sessions. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, CHI '17, pp. 3569–3581. ACM, New York, NY, USA (2017). <https://doi.org/10.1145/3025453.3025817>
  64. Yeo, H.S., Lee, J., Bianchi, A., Quigley, A.: WatchMI: applications of watch movement input on unmodified smartwatches. In: Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct, pp. 594–598. ACM Press (2016). <https://doi.org/10.1145/2957265.2961825>. <http://dl.acm.org/citation.cfm?doid=2957265.2961825>
  65. Yi, X., Yu, C., Xu, W., Bi, X., Shi, Y.: COMPASS: rotational keyboard on non-touch smartwatches. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, pp. 705–715. ACM Press (2017). <https://doi.org/10.1145/3025453.3025454>. <http://dl.acm.org/citation.cfm?doid=3025453.3025454>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.