

# Designing a Text Entry Multimodal Keypad for Blind Users of Touchscreen Mobile Phones

Maria Claudia Buzzi<sup>1</sup>, Marina Buzzi<sup>1</sup>, Barbara Leporini<sup>2</sup>, Amaury Trujillo<sup>1</sup>

<sup>1</sup>IIT-CNR; <sup>2</sup>ISTI-CNR

Via Moruzzi

Pisa, Italy

{claudia.buzzi, marina.buzzi, amaury.trujillo}@iit.cnr.it; barbara.leporini@isti.cnr.it

## ABSTRACT

In this report, we share our experience and observations on the challenges blind people face with text entry on touch-based mobile phones, particularly from the perspective of one of the authors, who is blind. To better understand these issues we developed and tested *Multimodal Text Input Touchscreen Keypad* (MTITK), an audio-tactile text entry prototype based on multitap, which relies on a telephone keypad layout organized into five key groups with distinct audio-tactile feedback. Users explore the screen to identify the current selected key, tap to enter text, and gesture to edit it, while receiving the corresponding voice, audio, and tactile feedback; no additional equipment is necessary in our software-only approach. We implemented a prototype on Android and tested its usability with visually impaired participants; they welcomed its multimodality and the familiar layout, but also expressed the need to increase vibration pattern differentiation and refine the character selection mechanism.

## Categories and Subject Descriptors

H.5.2. [Information Interfaces and Presentation]: User Interfaces – *input devices and strategies*; K.4.2. [Computers and Society]: Social Issues – *assistive technologies for persons with disabilities*.

## Keywords

Accessibility, blindness, text entry, mobile devices, multimodal interfaces

## 1. INTRODUCTION

Smartphones are everywhere in our society: we use them not only to communicate with others, but also to entertain ourselves and organize our daily lives. However, the transition from hardware keys to virtual keys on a touchscreen has made their use more difficult for people who are unable to see. Cost is another detrimental factor for blind users, both for the smartphone itself and the need for additional costly equipment. For instance, external Braille keyboards are expensive and cumbersome to transport [6]. Moreover, despite the awareness of more accessible devices, some blind users prefer to buy cheaper options [7]. Even if costs were low, some users would not want to be dependent on additional physical objects in order to use their cellphones [8].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org).

ASSETS '14, October 20 - 22, 2014, Rochester, NY, USA.

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM 978-1-4503-2720-6/14/10...\$15.00.

<http://dx.doi.org/10.1145/2661334.2661354>

To tackle these difficulties, mobile manufacturers have incorporated accessibility suits into their products, mainly based on voice interaction, such as Apple's VoiceOver or Android's TalkBack. Still, voice interaction has its complications: it is not always accurate, it is difficult to use in noisy environments [10], and can be undesirable due to privacy or etiquette concerns [7]. Despite voice interaction flaws, mobile text entry through automatic speech recognition (ASR) and Text To Speech (TTS), remains the mode preferred by blind users [1], but it can raise privacy issues. Thus, an alternative text-entry method is worthy of investigation.

## 2. A PERSONAL PERSPECTIVE

This research was inspired by a previous study on haptic differentiation of user interface (UI) elements on mobile touchscreens [2], and by the personal experience of one of the authors, who is blind. She uses two smartphones on a daily basis: a Nokia N95 and an iPhone 4S. The Nokia N95, released in 2007, has a slider form factor with a physical number keypad, it runs Symbian OS (v. 9.2), and the author uses TALKS as screen reader, a third party solution sold by Nuance. The iPhone 4S, released in 2011, is a touchscreen phone with no physical keyboard, and runs iOS (v. 7.1) which includes the VoiceOver screen reader. Each phone is used for a different set of tasks. In the past, she had used Nokia phones with Symbian, but it should be noted that this mobile platform was discontinued in 2011 due to a remarkable loss of market share. The author also occasionally uses an iPod Touch and an iPad, but she finds the latter cumbersome because of its larger size. In addition, she has briefly tested both Windows and Android phones, but still considers the iOS platform more accessible.

The older phone model, Nokia N95, is mostly used for traditional cellphone functionality: short text messages and voice calls. Thanks to the Nokia N95 physical keypad and the memorization of its character mapping, the author is able to type messages and dial numbers (if not present in the contacts list) more easily and comfortably than with the iPhone 4S. Besides, audio feedback without earphones is also more comfortable with the Nokia N95 especially when editing in a noisy situations. The iPhone 4S has the speakers on the bottom, so she has to hold the phone horizontally and bring its bottom side closer to her ear, which is impractical and awkward to do. In addition, the author experiences a longer battery life on the Nokia N95, due in part to a smaller screen size and lack of mobile Internet access. For these reasons, she feels more confident using the Nokia N95 as the main means of mobile communication, especially in case of emergency.

The newer iPhone 4S is used as a more portable alternative to a laptop for information and entertainment: news and ePub books reading through TTS, as well as music, audio and video podcast listening. However, the author does not like and avoids browsing the Web with VoiceOver, as she feels overwhelmed by too much

feedback on page structure and navigation. Although she has some issues with text input, she also uses her iPhone 4S to communicate via online social networks, instant messaging, and especially e-mail. Because she finds the Nokia N95 more comfortable for making and receiving calls, she rarely uses her iPhone 4S for these tasks although it can be carried out in a more usable way. For example, when using an automated service over the phone, it is difficult for her to type the desired option unless she uses earphones or the speakerphone, since the phone's proximity sensor disables the screen while the device is held close to the ear. Another issue she has with the iPhone 4S, for privacy reasons, is the automatic voice announcement of the caller's identity on incoming calls.

For text input on the iPhone 4S, the author mostly uses the virtual QWERTY keyboard. She only uses Siri's speech recognition to write when she is alone and in a quiet environment. Although the author thinks that Siri's speech recognition is fairly accurate, she still has to resort to the virtual keyboard to correct and edit the inserted text. After having briefly used iOS 7 new character drawing method, the author thinks it is promising but cumbersome since some characters (e.g., letters *a* and *d*) are difficult to draw accurately enough to be recognized easily. She also likes this method's use of gestures to edit text—two-finger swipe to the right for space and two-finger swipe to the left to delete the last character for instance—and would like to use specific gestures for different text granularity, such as word or phrase deletion.

Regarding the author's use of the iOS virtual keyboard, despite being her most used text input method she has several issues with character insertion and text editing. For instance, she has accuracy problems when inserting characters while traveling by train or car given the small size of the keys, as a slight finger movement may cause the insertion of an unwanted character. Moreover, the use of the phone in horizontal mode for a modest increase in the size of the keys is not much better. Another difficulty with insertion arises from the similar sounds of adjacent letters, such as *m* and *n*; although iOS offers a phonetic pronunciation—such as Alpha, Bravo and Charlie for letters A, B and C—the author finds its use impractical in the long run. She would also like to have direct access to frequently used punctuation marks (comma, period, question mark), and avoid changing the keyboard's set of characters to enter them.

Text editing with the virtual keyboard presents even more challenges. For instance, text deletion with the delete key only allows one to cancel one character at a time, and she needs to pay attention when pressing this key several times to delete more than one character. Text navigation and revision at the word and phrase level is also difficult: first, she has to explore and find the given text field, then she has to move the text cursor to the desired position, next she has to return to the virtual keyboard and find the desired key. Sometimes she prefers to clear the text field and rewrite the text from the beginning rather than editing it. Finally, the author would also like a gesture to activate the field's main actions (e.g., go, search, send and next), instead of having to search for the specific key or risk pressing it by accident.

To study additional details on blind user interaction, and collect other elements for better understanding of sightless text-entry via touchscreen, we developed the MTITK prototype, a software-only approach for multimodal text entry for blind users -- which is modeled following the standard multitap text entry system of telephone keypads.

### 3. MOBILE TEXT ENTRY ACCESSIBILITY

Before the advent of touchscreen smartphones, people mainly used their phones to communicate via phone calls or short text messages (SMS) using multitap on physical keys. Blind users adapted earlier cellphones in three ways: using special purpose devices, through accessible features in commodity cellphones, or without any assisting technology. Accessible mobile text entry development focused on Braille-based alternatives, word prediction, and the use of screen reader software [4]. Today, touchscreen smart devices are ubiquitous, but the inherent accessibility issues with such interfaces have made their adoption by blind users difficult [5]. They have a smooth surface, which offers no distinction between display space and controls. Prior approaches regarding the use of touchscreen by blind users can be classified into three general categories: software-only, hardware-only and hybrid approaches [8].

Software-only approaches, such as TalkBack and VoiceOver, do not modify the device's hardware but adapt the underlying software. They can greatly increase accessibility but are limited to the specific device's hardware features. Hardware-only approaches do not modify the underlying software; they rely on placing physical tools between the screen and the user. For instance, special plastic cases or screen overlays can delimit screen borders; however, smartphones are not aware of such overlays and do not adapt to their presence or absence [4, 8]. Hybrid solutions combine custom software and hardware. For example Touchplates [8] are low-cost plastic overlays that interact with the underlying device software. Different overlays are available for varying interaction needs (e.g., keyboards, keypads, geographical maps), making it a flexible solution. Still, users would need to carry different overlays with them, which can be impractical.

Since MTITK mobile text entry approach is software-only, in the following we focus and expand on this sub-category.

#### 3.1 Software-only Mobile Text Entry Approaches

Several software-only approaches for mobile eyes-free text entry on touchscreen have been proposed, and we can group them as follows: multitouch character encoding, character drawing, and constructive methods.

##### 3.1.1 Multitouch Character Encoding

This approach relies on coded forms of characters, based on the number of fingers (pointers) the user has on the screen in a given moment. The most common encoding in use is Braille, in which a rectangular cell with six dots with a binary state (raised or not), offers 64 ( $2^6$ ) possible patterns. For example, BrailleTouch is a mobile text entry solution which works on phone-size devices using both hands [13]; it was tested and well received by proficient Braille participants. A non-Braille example is DigiTaps [1], which uses encoding based on four gestures and minimal audio feedback, but it is limited to number entry only. These solutions rely on learning coded forms for every supported character, which could be a demanding cognitive task for users. Besides, despite knowledge of the encoded Braille alphabet, blind people may have low confidence in writing or reading it [4].

##### 3.1.2 Character Drawing

In this approach, each character is input through the use of a pointer (finger or stylus), with which users draw directly on the screen. For instance, one study [14] tested eyes-free shorthand handwriting based on *Graffiti*, a commercial recognition system

by Palm Inc. However, none of the participants were blind. Also, beginning with iOS 7, Apple offers a new accessibility feature for text entry called *Handwriting*, but we could not find any related formal study at the moment of writing. Even though research confirms that blind users are able to draw and make gestures on touchscreens [9], it is very difficult for them to learn and use handwriting [13].

### 3.1.3 Constructive Methods

In this kind of method users perform several actions to insert a character; multitap is the most famous example for mobile phones. Qwerty keyboard exploration, through touch and TTS feedback, is also a widely-used constructive text entry approach for the blind, as it is included by default in VoiceOver and TalkBack. Nevertheless, it is time-consuming and is considered to be of low usability [10]. TypeInBraille [11] is both a constructive and a multitouch character encoding method. It uses a three-step process to enter Braille characters, with four gestures for each row in a Braille cell. It is intended to be easy to learn, but it requires more actions to enter text compared to similar solutions. Another example, included in the commercial accessible suit Mobile Speak, features a touch entry mode based on the standard keypad and multitap [3]. It uses taps, gestures and a physical command key. Because of the lack of feedback before input and TTS based input confirmation, users need to memorize not only the keypad character assignment and ordering, but also the arrangement of keys; this requires several trial sessions of and is error-prone.

## 4. MULTIMODAL TEXT INPUT TOUCHSCREEN KEYPAD

MTITK originated from the idea of defining UI areas through haptic feedback in touchscreen cellphones using their integrated motor [2], and from the problems the blind author has with text entry on these devices, mainly in noisy environments.

### 4.1 Layout and Key Groups

MTITK layout is based on the common 12-button (4×3 grid) telephone layout. It is defined by the International Telecommunications Union (ITU-T E.161), and recommended by the International Organization for Standardization (ISO 9955-8). The names and characters used to identify keys on the keypad (from left to right and top to bottom) are: *one* (1), *two* (2), *three* (3), *four* (4), *five* (5), *six* (6), *seven* (7), *eight* (8), *nine* (9), *star* (\*), *zero* (0) and *pound* (#).

We classified the keys into five groups, each with a distinct audio-tactile feedback, so as to identify a particular key based on its feedback and relative position to other keys or to the screen. We started by categorizing the keys into two subsets: the first comprises the odd and even numeric keys (from *one* to *nine*); the second subset is the bottom row (*star*, *zero* and *pound* keys). Each subset is further divided differently. The first subset of keys is grouped based on compass points: cardinal points (keys *two*, *four*, *six* and *eight*), intercardinal points (keys *one*, *three*, *seven* and *nine*), and a center (key *five*). The bottom row subset is grouped based on key identification characters: zero (homonymous key) and symbols (keys *star* and *pound*). With this classification, illustrated in Fig. 1, no keys with the same audio-tactile feedback are next to each other.

### 4.2 Character Ordering, Assignment and Insertion

The keypad character ordering and assignment used in MTITK is based on the standard for the 12-key telephone keypad of the

European Telecommunications Standards Institute (ETSI ES 202 130). At the time of writing, we only support the required character repertoire and essential letters for English, French, Italian and Spanish. All the letters are assigned to keys *two* to *nine*. The assignment of special characters is not indicated in the standard, so we distributed them among keys *one*, *star*, *zero* and *pound*. Only the *new line* character is inserted through a gesture.

MTITK also has a two-state (*down* and *up*) shift status, controlled with a set of gestures, which toggles the set of characters available. If the shift status is *down*, letters will be lower case, and a subset of special characters will be available. Otherwise, letters will be upper case, and a different subset of special characters will be available. The space and numeric characters are selectable at the same amount of taps on both *down* and *up* shift statuses.

Character insertion is a two-step process. First the user explores the keypad with one finger (the primary pointer); audio-tactile feedback will be provided according to the key group of the active key (i.e., the key currently under the primary pointer). Once the desired key is active, without raising or changing the active key, the user taps the required number of times anywhere on the screen to select the given character, and finally, the device will announce the entered character name through TTS and a short vibration will confirm character insertion.

### 4.3 Gestures and Additional Actions

Additional gestures – simple continuous strokes drawn on the UI – allow the user to perform text editing (change shift status and delete or read text), and insert the new line character, or the current text field command (e.g., search, send, and go). When a gesture is recognized by the application and executable within the current editor context, the user is notified through a short vibration pattern (that is the same for all gestures) and the corresponding TTS feedback (e.g., “deleted all text”). We based the gestures we used on those used by TalkBack’s default settings, which consist of single right-angled swipes. However, we also extended and attached granularity semantics to changes in direction or shape of gestures. For instance, a swipe left deletes the last character, a swipe left then up deletes the last word, a swipe left then down deletes the last phrase, and a swipe left then down then right clears the text field. Another example are the gestures for the shift status: a chevron for shift up and a chevron inverted for shift down, both made from left to right.

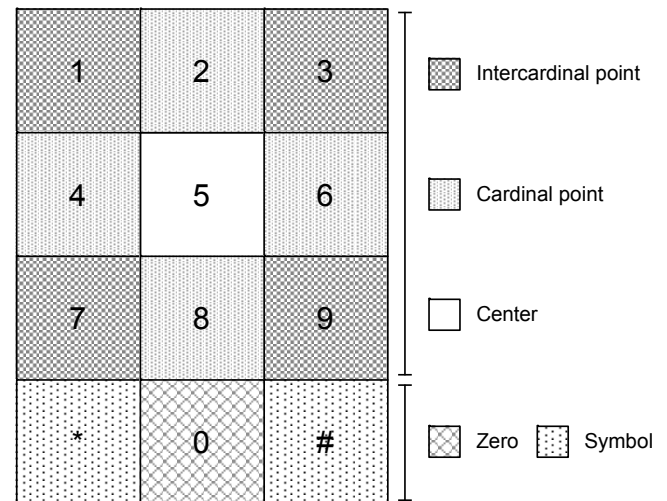


Fig. 1. Keypad layout and key groups

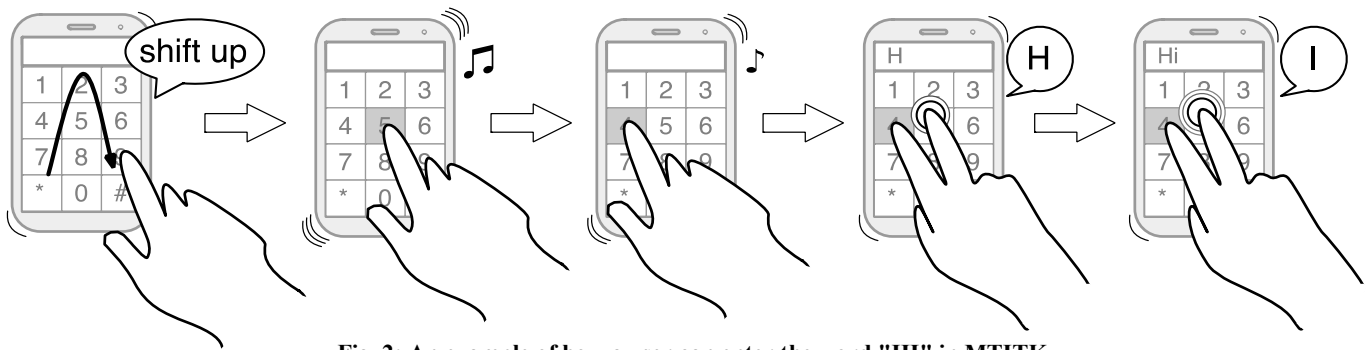


Fig. 2: An example of how a user can enter the word "HI" in MTITK.

#### 4.4 Multimodal Interaction

MTITK has four interaction modes: finger multi-touch via the screen for input, speech, audio, and tactile vibrations for output and feedback. Speech is used for character names upon insertion; on deletion or reading of the last inserted character, word, sentence or all of the text; and on the field command name. Audio-tactile feedback is used cross-modally (both modes convey the same meaning) for the exploration and identification of the keys. In all of the previous cases, a short vibration pattern confirms the recognition and execution of the given action. Figure 2 illustrates how this multimodal approach can be used to enter text accurately. Moreover, it would be possible to locate keys in noisy environments using primarily haptic feedback, even when the audio (including speech) is not clearly heard.

#### 5. PROTOTYPE IMPLEMENTATION

To test the use of MTITK as well as identify its usability issues, we designed and developed an initial prototype. It was implemented in Android; in order to ease development regarding the application programming interface (API) and new accessibility features, the minimum supported Android version is 4.2 (API level 17). For speech synthesis, we used Google TTS. During development and testing, we used three Android touchscreen smartphone models: Galaxy Nexus, Nexus 4 and Nexus 5. All of them have a slate form factor and share similar physical specifications. For gesture recognition we used the Android Gesture package, which offers similarity comparison of a given gesture and a gesture reference library based on their respective Euclidean and Cosine distances.

#### 6. USABILITY TESTING

##### 6.1 Preliminary Assessment

In an early assessment of MTITK, two blind participants, one female (the blind co-author) and one male, evaluated our initial prototype through a usability test. Both participants are proficient users of electronic devices, including computers and smartphones (both of them use iPhones). During the test we used the think-aloud protocol, in which the participants expressed their thoughts while freely using the interface or performing specific tasks.

First, we explained the scope of MTITK, its features and how to use it. Then we handed both participants our development devices (Galaxy Nexus and Nexus 4) and asked them to freely explore the keypad's interface. Once they became familiar with the layout, the distinct keypad groups and audio-tactile feedback, we finally asked them to perform specific text entry tasks (e.g., write capital letters, read the last character, delete the last entered word), in both a quiet and a noisy environment (we asked three people to have a normal conversation around them). At that point, we also

observed the way the participants held the devices, how they drew the gestures and which fingers they used for character insertion.

MTITK was well received by both participants. They could accurately identify keys through audio-tactile feedback, even in the noisy environment. They also appreciated the use of the standard keypad layout. We noted that the participant author with the secondary cellphone (with physical keys) was better at remembering character assignment and ordering. Both participants held the device in one hand and used the other (dominant) hand, for text entry. In the beginning, participants used their index finger for keypad exploration and the middle finger for tapping. Later they also used the middle finger for exploration and the index and ring fingers for tapping. We also noticed some difficulties performing gestures with steep angles, as the participants were more likely to draw rounded strokes.

##### 6.2 Pilot Test

Based on previous observations and the feedback received from the participants during the preliminary assessment of MTITK, we improved the prototype, mainly increasing the input and feedback timings. We then proceeded to carry out an informal pilot test, with a larger, heterogeneous group of participants. We first asked them to complete a questionnaire about their use of mobile touchscreen devices and how they perform text entry on them. Then we demonstrated MTITK and asked them to test it.

###### 6.2.1 Participants

Fourteen visually impaired participants were recruited from three different local centers for blind and low-vision individuals. The group was composed of 6 females and 8 males, with ages ranging from 26 to 70 years (median age 44 years); 3 of the participants had severe low vision, 3 were blind from birth, 3 blind from adolescence, and 5 became blind as adults. Participants have been compensated with a flash memory card.

###### 6.2.2 Touchscreen mobile use and text entry

All of the participants have frequently used mobile phones with a physical keypad, and most of them use this kind of mobile phone exclusively or in combination with touch-based devices. With the exception of one participant, all of them have used touchscreen phones, of which four use these devices exclusively. Almost all of participants reported they remembered the standard letter mapping of 12-key phone keypad fairly well, one reported recalling it quite well, and another one reported remembering more or less.

Most of the participants who have touchscreen phones frequently or always use the virtual QWERTY keyboard. Few participants use command and text dictation (e.g., Siri for iOS); most of them never or rarely use this feature. Only one of the participants frequently uses an alternate text entry method, based on Braille.

We asked the participants about the level of difficulty they experience while performing certain text entry tasks on the QWERTY virtual keyboard (Table 1). It should be noted that two of the participants answered “I don’t know” in all of the tasks: one of them does not use this virtual keyboard; the other had not used a touch-based phone before. The two top difficult tasks, as reported by the participants, are the deletion of one or more words and the placement of the cursor. We also noticed that a considerable number of participants did not know how to select text or use autocomplete on their touch-based mobile devices.

**Table 1. Number of participants by self-reported difficulty level on text entry and editing tasks on the QWERTY virtual keyboard. Scale: VE (Very easy), EA (Easy), NE (Neutral), DI (Difficult), VD (Very difficult) and DK (I don’t know)**

Task	VE	EA	NE	DI	VD	DK
Identify keys	4	5	1	1	1	2
Autocomplete	1	1	4	1	0	7
Delete last character	5	5	2	0	0	2
Delete one or more words	2	1	0	5	4	2
Place the cursor	1	2	2	4	3	2
Select text	1	0	0	2	2	9
Read inserted text	6	4	1	1	0	2

### 6.2.3 Procedure

We followed a procedure similar to the one used in the initial assessment, in which we explained the scope of MTITK, its features, and how to use it. In this test we used three Nexus 5 mobile phones with Android 4.4 as operating system. This multitouch model features a capacitive screen of 4.95 inches and weighs 130g. It has only three physical buttons: the power button on the upper right side, and volume up and down on the upper left side. No tactile boundaries were used to delimit the display area on the front of the devices. All voice feedback was given using Google TTS, Italian language pack.

In the first phase of the test, we asked the participants to freely explore the prototype’s layout, the different pattern vibrations and audio feedback for each group of keys, to practice the letter mapping, and the gestures for editing actions. Next, we asked the participants to perform three tasks: write a common four-letter word, write a word with all the vowels, and compose a telephone number (using the numeric input of MTITK). The tasks were performed with vocal feedback alone, and with vocal and haptic feedback, in both a noisy and a quiet environment. We used the think-aloud protocol during the test in a quiet environment; for the noisy environment we asked them to give us feedback after task completion. In both cases, we observed participants’ actions while they performed each of the tasks.

### 6.2.4 Results

Reception of the prototype varied from mixed to positive. All of the participants preferred both voice and haptic feedback compared to voice feedback alone. However, in the case of noisy environments, half of the users expressed difficulty in correctly identifying the keys based on haptic feedback; the other half of the participants could identify the different vibration patterns after using the prototype for a few minutes, but needed to focus more on exploration of the keypad.

Most of the participants welcomed the use of the familiar keypad layout. However, in the beginning some of the users had difficulty correctly remembering standard keypad letter mapping, although most of them had previously reported a good recall of it. Likewise, part of the group had difficulty remembering the editing gestures they had practiced in the first phase of the test. Also, when trying to enter text some participants instinctively lifted their finger from the screen and proceeded to tap the keys as they are used to doing on a physical keypad. As with the preliminary test, we observed difficulty in performing certain gesture figures, especially those with right and steep angles. We noticed that the age from which the participant has been visually impaired seems to affect form of gestures; people blind from an early age appeared to have more difficulty.

Nearly all of the participants liked the idea of using the mobile phone vibration motor to identify areas of the keypad UI, but expressed that it should be easier to differentiate the current vibration patterns. In addition, they also expressed their interest in using MTITK as an alternate text entry method for a more private and discrete interaction in scenarios where using dictation or TTS would be inconvenient. Furthermore, MTITK was most appreciated for composing telephone numbers in a noisy environment. Being a simpler task than letter entry, the participants indicated they could correctly identify the key number based on the haptic feedback and position of their finger on the touchscreen.

## 7. OBSERVATIONS AND LESSONS LEARNED

Despite the ongoing development and early stage of MTITK, we share the following observations and lessons learned so far during its design, implementation and testing.

- Among the participants, iOS is by far the most popular mobile platform. Of the 16 participants, only one does not regularly use a touchscreen device, 11 use an iPhone, and 4 use an Android phone. Further, 8 participants also use other mobile touchscreen devices, such as music players (mainly iPod) and tablets (mostly iPad). The main reason for the preference of iOS, as explained by the participants, is the considerable set of accessible features on the built-in screen reader VoiceOver, although significant issues have been identified regarding its usability [10]. While both VoiceOver and TalkBack were initially released in 2009 for their respective mobile platforms, VoiceOver is based on its desktop counterpart, released almost a decade ago; consequently, it has a bigger and more mature set of features than TalkBack.
- Due to the open nature of Android it is possible to change or extend many of its system-wise features, such as input method editors and TTS engines, whether with third-party solutions or customized alternatives. Only a more limited customization of these features is possible in iOS, if at all. Moreover, the core Android platform and TalkBack are open source, so it is possible to have a finer degree of customization and understanding of the software. We chose Android over iOS because of this openness and the possibility of developing software on multiple platforms, as well as the usually lower cost of the devices. Nonetheless, the Android platform is much more fragmented than iOS. Several manufacturers make a plethora of models, with different form factors, electronic components and features. This can hinder the implementation and testing of a unified user experience of publicly released software. In

addition, as pointed out earlier, iOS is the most popular choice for an accessible experience on mobile touchscreen devices at the moment. Still, we think Android is more suited for academic research and development on mobile device accessibility software.

- Individual attributes and abilities have a significant impact on learning different text entry methods and on their performance [12], but so do previous experiences and technical aptitudes of users. Both participants in our preliminary test were technologically experienced and proficient, as they aimed to better understand the devices they use. Although we obtained valuable feedback from these participants, we also overlooked the different needs and aptitudes of less experienced users. For instance, during the pilot test less experienced participants had more difficulty remembering the keypad character mapping, performing gestures and remembering the editing commands. When designing for universal access, it is crucial to have early feedback of both proficient and novice users.
- For the preliminary design of UIs, we suggest wireframing with engraved and high reliefs. An expensive embossing printer is not necessary since these wireframes can be made on plastic, cardboard or other cheap material, with silicone glue or wax sticks. It is a convenient and timesaving way to explain the layout, disposition and functioning of UI elements or other graphical elements to blind or visually impaired people and obtain their feedback. Another option is to use cut cardboard, as we did in our pilot test, in order to have a complementary representation (besides verbal) of the shape of the gestures.
- A substantial proportion of visually impaired mobile phones users do not know all of the accessibility features of their devices. For instance, some participants talking among themselves realized they did not know a given gesture, feature, or option of their phones. Moreover, after the pilot test one of the participants asked the authors for help configuring and understanding his Android device, as he realized he only knew and used its basic accessibility functions.
- Awareness of existing solutions and current studies on accessibility motivates visually impaired people to adopt new technology. Several participants acquired their touchscreen phones after participating in a previous study. Some of them, after our pilot test, expressed an interest in upgrading their touchscreen mobile phones or acquiring one for the first time.

## 8. CONCLUSION

In this report, we have described the challenges that visually impaired people face in touch-based mobile text entry, and we have shared our experience while developing and studying a multimodal software-only solution to overcome these challenges. We have described our approach, the ideas received, and possible improvements. In addition to continuing our study, we think there is still much work to do on the use of multitouch gestures by blind people on handheld devices, based on our usability tests and available literature. For this reason, in future work we will also study how visually impaired people perform gestures on mobile touchscreen phones. In spite of the early stage of the project, we hope the lessons and observations we presented will be helpful to other researchers and entities interested in improving touch-based mobile devices for the visually impaired.

## 9. REFERENCES

- [1] Azenkot, S. and Lee, N.B. Exploring the Use of Speech Input by Blind People on Mobile Devices. In Proc. 15th International ACM SIGACCESS Conference on Computers and Accessibility, ACM (2013), 11:11-11:18.
- [2] Buzzi, M.C., Buzzi, M., Leporini, B., and Paratore, M.T., 2013. Vibro-Tactile Enrichment Improves Blind User Interaction with Mobile Touchscreens. In Human-Computer Interaction--Interact 2013 Springer, 641-648.
- [3] Code Factory, 2012. Mobile Speak and Mobile Magnifier for Symbian/S60 Phones Code Factory, S.L, 08221 Terrassa (Barcelona).
- [4] Guerreiro, T., 2012. User-Sensitive Mobile Interfaces: Accounting for Individual Differences amongst the Blind IST - Technical University of Lisbon, R. Alves Redol, 9, 1000-029, Lisboa, Portugal.
- [5] Guerreiro, T., Jorge, J., and Gonçalves, D., 2011. Exploring the Non-Visual Acquisition of Targets on Touch Phones and Tablets.
- [6] Guerreiro, T., Lagoá, P., Nicolau, H., Santana, P., and Jorge, J. Mobile text-entry models for people with disabilities. In Proc. 15th European conference on Cognitive ergonomics, (2008), 39.
- [7] Kane, S.K., Jayant, C., Wobbrock, J.O., and Ladner, R.E. Freedom to roam: a study of mobile device adoption and accessibility for people with visual and motor disabilities. In Proc. 11th international ACM SIGACCESS, (2009).
- [8] Kane, S.K., Morris, M.R., and Wobbrock, J.O. Touchplates: low-cost tactile overlays for visually impaired touch screen users. In Proc. 15th International ACM SIGACCESS, (2013), 22.
- [9] Kane, S.K., Wobbrock, J.O., and Ladner, R.E. Usable gestures for blind people: understanding preference and performance. In Proc. SIGCHI Conference, (2011).
- [10] Leporini, B., Buzzi, M.C., and Buzzi, M. Interacting with mobile devices via VoiceOver: usability and accessibility issues. In Proc. 24th Australian Computer-Human Interaction Conference, (2012), 339-348.
- [11] Mascetti, S., Bernareggi, C., and Belotti, M. TypeInBraille: a braille-based typing application for touchscreen devices. In Proc. 13th international ACM SIGACCESS, (2011).
- [12] Oliveira, J., Guerreiro, T., Nicolau, H., Jorge, J., and Gonçalves, D. Blind People and Mobile Touch-based Text-entry: Acknowledging the Need for Different Flavors. In Proc. 13th International ACM SIGACCESS Conference on Computers and Accessibility, ACM (2011), 179-186.
- [13] Plimmer, B., Crossan, A., Brewster, S.A., and Blagojevic, R. Multimodal collaborative handwriting training for visually-impaired people. In Proc. SIGCHI Conference, (2008), 393-402.
- [14] Tinwala, H. and MacKenzie, I.S. Eyes-free text entry on a touchscreen phone. In Proc. Science and Technology for Humanity (TIC-STH), IEEE (2009), 83-88.