Investigating Laboratory and Everyday Typing Performance of Blind Users

HUGO NICOLAU, INESC-ID, Instituto Superior Técnico da Universidade de Lisboa
KYLE MONTAGUE, Newcastle University
TIAGO GUERREIRO, LaSIGE, Faculdade de Ciências da Universidade de Lisboa
ANDRÉ RODRIGUES, LaSIGE, Faculdade de Ciências da Universidade de Lisboa
VICKI L. HANSON, Rochester Institute of Technology and University of Dundee

Over the last decade there have been numerous studies on touchscreen typing by blind people. However, there are no reports about blind users' everyday typing performance and how it relates to laboratory settings. We conducted a longitudinal study involving five participants to investigate how blind users *truly* type on their smartphones. For twelve weeks, we collected field data, coupled with eight weekly laboratory sessions. This paper provides a thorough analysis of everyday typing data and its relationship with controlled laboratory assessments. We improve state-of-the-art techniques to obtain intent from field data, and provide insights on real-world performance. Our findings show that users improve over time, even though it is at a slow rate. Substitutions are the most common type of error and have a significant impact on entry rates in both field and laboratory settings. Results show that participants are 1.3-2 times faster when typing during everyday tasks. On the other hand, they are less accurate. We finished by deriving some implications that should inform the design of future virtual keyboard for non-visual input. Moreover, findings should be of interest to keyboard designers and researchers looking to conduct field studies to understand everyday input performance.

• Human-centered computing → Accessibility → Empirical studies in accessibility • Human-centered computing → Accessibility → Accessibility systems and tools • Human-centered computing → Human computer interaction (HCI) → Interaction techniques → Text input.

Additional Key Words and Phrases: Blind, text-entry, input, laboratory, in-the-wild, everyday, touchscreen, mobile, behavior, performance, errors, longitudinal.

ACM Reference Format:

Hugo Nicolau, Kyle Montague, Tiago Guerreiro, André Rodrigues, and Vicki L. Hanson, 2016. Investigating laboratory and everyday typing performance of blind users. *ACM Trans. Access. Comput.* . x, x, Article xx (date), xx_pages.
DOI:http://dx.doi.org/10.1145/0000000.0000000

This work was partially supported by: RIT's Center for Accessibility and Inclusion Research, RCUK Hub: Inclusion in the Digital Economy (EP/G066019/1), and FCT (SFRH/BD/103935/2014, UID/CEC/00408/2013, UID/CEC/50021/2013).

Author's addresses: Hugo Nicolau, Av. Prof. Doutor Cavaco Silva, 2744-016 Porto Salvo, Portugal; Kyle Montague, Open Lab, Newcastle University, Newcastle Upon Tyne, United Kingdom, NE1 8HW; Tiago Guerreiro, Departamento de Informática, Faculdade de Ciências, Universidade de Lisboa, Campo Grande, Edifício C6, 1749-016 Lisboa, Portugal; André Rodrigues, Departamento de Informática, Faculdade de Ciências, Universidade de Lisboa, Campo Grande, Edifício C6, 1749-016 Lisboa, Portugal; Vicki Hanson, IST, GCCIS, Rochester Institute of Technology, Rochester, NY, 14623, USA.

Permission to make digital or hardcopies of part or all 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 show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credits permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

© 2016 ACM xxx

DOI:http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Text input is one of the most common tasks in mobile interaction: from text messaging and web browsing to emailing and social networks. Currently, blind users are able to enter text on their touchscreen devices using accessibility services, such as Android's Explore by Touch[1] or Apple's Voice Over[2]. Previous laboratory studies have shown that blind users achieve lower typing rates than sighted users and make more errors [Oliveira et al. 2011, Azenkot et al. 2012]. Most prior solutions that attempted to tackle these problems used familiar keyboard layouts [Guerreiro et al. 2008, Bonner et al. 2010] and Braille-based approaches [Azenkot et al. 2011, Mascetti et al. 2012, Southern et al. 2012, Nicolau et al. 2014].

While text input has been studied for years, research has been limited to laboratory studies. Furthermore, most studies rely on a single laboratory session, producing a snapshot of typing performance (e.g. [Southern et al. 2012, Oliveira et al. 2011, Rodrigues et al. 2016]). Understanding how input performance changes over time and how people *truly* type with their mobile devices remains an open question. Performance data is usually collected in laboratory settings by instructing participants to copy a number of sentences and measuring speed and errors [Azenkot et al. 2011, Oliveira et al. 2011, Nicolau et al. 2015]. While this procedure is valuable to guarantee internal consistency, it can miss several challenges encountered in the real-world. However, collecting and analyzing field data can be difficult, since it is not as controlled as data from a laboratory study. Difficulties include not knowing what users intended to type and having to collect data from different applications.

In contrast with previous work, our goal is to understand the real-world learning experience of novice blind users by analyzing their everyday typing performance. To our knowledge, there are no previous reports of blind users' text-entry performance from smartphone use. We conducted a twelve-week field study with five novice smartphone participants and compared their real-world performance with controlled typing tasks. Results allowed us to answer questions such as: *What is the everyday mobile typing performance of blind users? How does everyday performance relate to laboratory performance? What are the most common types of errors? Do participants maintain the same typing behaviors in real world?*

Our findings have implications for the design of touchscreen keyboards and input techniques for blind and visually impaired users. Based on typing data, our results show that substitutions are the most common error type both in laboratory and field settings. Participants' performance significantly improved over time, both in terms of errors and speed. We also show why improvements occur by examining hit positions, movement time, movement paths, and pausing behaviors. Correction strategies were consistent among users, but required a significant amount of time. Results also show that laboratory data provides a skewed view of input performance. Particularly, *input speed* is on average 1.5 times faster during every day typing tasks. On the other hand, *uncorrected error rates* are 2.5 times higher.

The contributions of this article include: (1) an understanding of mobile typing performance (speed and errors) of blind users in laboratory and real-world settings; (2) a mobile service that collects and analyzes everyday text-entry data; (3) an

[1] http://developer.android.com/design/patterns/accessibility.html

[2] http://www.apple.com/accessibility/osx/voiceover/

analysis of touch exploration behaviors in text-entry tasks; and (4) a report on the learning experience of blind users, particularly how input performance and behaviors changed over a 12-week period. The findings herein presented should be of interest to mobile keyboard designers and accessibility researchers looking to gain from quantitative insights into blind users' text-entry performance with touch devices.

This article extends our prior work in characterizing the typing performance of blind users with mobile devices [Nicolau et al. 2015]. In that paper, we provided an analysis of unconstrained text-entry tasks in laboratory settings during an 8-week period. We proposed using touch movement measures to better understand text input behaviors. This extended article complements our body of knowledge by going beyond controlled laboratorial assessments, reporting on 12 weeks of field data. We include a technical description that improves state-of-the-art techniques to analyze everyday typing data of blind users and present real-world typing performance. We also include an extended analysis of related work, namely on the challenges of analyzing real-world data, and an extended discussion section that reflects upon laboratory and everyday performance of blind users.

2. RELATED WORK

In this section, we discuss previous work on non-visual input methods, text-entry evaluation measures, and methodologies to conduct field studies on input research.

2.1 Text-Entry and Visual Impairments

Today's mainstream touchscreen devices support non-visual text input via the built-in screen readers e.g. VoiceOver and Talkback. They enable users to explore the keyboard with their finger and have the keys read aloud as they touch them. While the visual layout of the QWERTY keyboard is identical to that presented to sighted users, input rates are much slower for visually impaired people [Oliveira et al. 2011]. To address this problem other research has proposed novel interfaces for non-visual text-entry on mobile touchscreen devices, including new keyboard layouts [Yfantidis and Evreinov 2006, Guerreiro et al. 2008, Bonner et al. 2010] and alternative methods of inputting text [Tinwala and MacKenzie 2010, Oliveira et al. 2011, Mascetti et al. 2012, Southern et al. 2012, Azenkot et al. 2012, Nicolau et al. 2015].

Yfantidis and Evreinov (2006) proposed a new input method consisting of a pie menu with eight options and three depth levels. Users could select characters by performing a gesture in one of the eight directions of the layout. Dwelling on a character after a gesture was used to access alternative characters. NavTouch also used a gestural approach [Guerreiro et al. 2008], allowing blind users to navigate through the alphabet using four directions. Horizontal gestures would navigate the alphabet sequentially, and vertical gestures would navigate between vowels, which served as shortcuts to reach the intended letter. Bonner et al. (2010) presented No-Look Notes, a keyboard with large targets that used an alphabetical character-grouping scheme (similar to keypad-based multitap approaches). The layout consisted in a pie menu with eight options. Split-tapping a segment sent the user to a new screen with that segment's characters, ordered alphabetically from top to bottom.

In the past few years, Braille-based approaches have been proposed to allow blind people to input text on their mobile touchscreen devices. For example, BrailleTouch [Southern et al. 2012] or PerkInput [Azenkot et al. 2012] are both multitouch Braille chording approaches have that have shown to be very effective in improving input speed. Both methods enable users to input chords on their touchscreen devices, similarly to what they do on a traditional Perkins Brailler. More recently, Nicolau et al. (2014) proposed a correction system for such methods to decrease input errors.

Despite much work in the field of non-visual input, research has been restricted to performance comparisons of input techniques. In these studies, performance is often measured in terms of words per minute and errors in a single laboratory session. The literature lacks an understanding of everyday typing performance of blind users.

2.2 Text-Entry Measures

Text-entry performance is most often measured in terms of speed and errors, using metrics such as words per minute (WPM) and minimum string distance (MSD) error rates [Soukoreff and MacKenzie 2003]. Character-level errors are also commonly used to assess the most common types of errors (i.e. omissions, insertions, or substitution) and which characters are more troublesome [MacKenzie and Soukoreff 2002, Wobbrock and Myers 2006]. The methodology proposed by Wobbrock and Myers (2006) is the current state of the art for text-entry performance assessment. The authors introduced the input stream taxonomy to support unconstrained text-entry evaluations. This approach allows participants to make corrections to their typing and automatically capture both uncorrected and corrected error rates. Knowing the target sentence (i.e. intent) and using this analysis, it is possible to capture character-level errors and identify corrective behaviors.

In addition to speed and error measures, other authors have been using touch-based metrics to better understand typing behaviors. Findlater et al. (2011) evaluated the typing performances of expert sighted typists on large touch surfaces. Through an analysis of touchscreen measures, they identified individual differences in key centroids and hit point deviations (i.e. x and y offsets of touch gestures with regards to individual keys). Later, they proposed personalized keyboards that could adapt to individual typing patterns and improve entry rates [Findlater and Wobbrock 2012]. Guerreiro et al. (2015) applied similar touch measures to investigate tablet text-entry behaviors of blind users with one- and two-handed input. While the text input performance metrics revealed no statistical difference between conditions, using the x, y offsets of the initial touch down positions, the authors uncovered that users landed closer to intended keys with two-handed input. Furthermore, when measuring movement distances of non-visual exploration, participants using two hands performed more efficient paths through the keyboard. The authors leveraged the fact that non-visual touchscreen interactions result in gestures with periods of continuous movement and traces through the interface, opposed to the discrete point interactions of sighted users.

While using movement measures is uncommon when analyzing text input, they are well established within cursor movement research. MacKenzie et al. (2001), proposed seven accuracy measurements to understand users' behaviors with pointing devices. Included in these were path analysis measurements, such as target re-entries, task axis crossing, movement direction and orthogonal direction change. The authors also proposed continuous measures such as movement variability, errors and offsets. Hwang et al. (2004) believed analysis of submovements within pointing device selections could reveal new insights into the challenges faced by motor-impaired users. To understand individual differences between motor-impaired users' cursor movements, the authors proposed analyzing the number and duration of pauses, verification times, submovements within the intended target, target slips, and velocity profile of movements.

In this paper, we extend on existing text-input analysis techniques and propose the inclusion of discrete and continuous touch movement measurements to better understand touchscreen text input behaviors of blind users.

2.3 In-the-Wild User Studies

Unlike laboratory evaluations, real-world data lacks information about user intent [Hurst et al. 2008, Gajos et al. 2012, Montague et al. 2014, Rodrigues et al. 2015]. In a laboratory study, participants are given target sentences and instructed to copy them as "quickly and accurately as possible" [Wobbrock 2007]. Each sentence corresponds to a trial; this ensures experimental control and makes computing entry speeds and errors straightforward [Wobbrock and Myers 2006]. However, everyday data contains a continuous input stream and no information about whether that was the user's intent. Thus, computing text-entry speed and errors is a much more complex task. A possible solution to this problem is to prompt users with target sentences at random times. The Dynamic Keyboard evaluation used such an approach where participants were asked to provide typing samples throughout the day [Trewin 2004]. Others have used similar approaches where researchers have some control over target sentences; however, tasks are still artificially created and may not reveal everyday typing performance.

Hurst et al. (2008) investigated everyday pointing performance of individuals with motor impairments. The initial phase of their work required participants to complete baseline calibrations using the IDA [Koester et al. 2005] software suite. Afterwards, participants were free use the system and play games, or use other applications such as word processing. The authors used application interaction models to infer user intent from mouse input, allowing them to calculate measurements of pointing performance. More recently, Montague et al. (2014) used a similar approach to understand "in-the-wild" touchscreen performance of motor-impaired participants using a Sudoku stimulus application. Results enabled them to create novel touch models tailored to individual abilities.

Regarding everyday text-entry performance, it has been fairly ignored in the past. An exception is the work by Evans and Wobbrock (2012), which proposes an approach similar to ours to compute input errors using an online spellchecker. The authors focused on users with motor impairments. There were no restrictions regarding application use, and input performance was based on their everyday usage without the need for artificially created tasks. We extend Evans and Wobbrock (2012) work by: 1) implementing a mobile data collection tool; 2) developing novel techniques to obtain intent, tailored to the specific typing behaviors of blind users; and 3) validating the proposed approach.

3. PERFORMANCE FROM EVERYDAY TYPING DATA

In this section we describe our methodology to get intent from everyday typing data. Particularly, we explain each of the steps involved in the process from collecting text-entry data to segmenting trials and calculate typing speed and errors.

3.1 Collecting Everyday Data

Previous work has stressed the importance of studying technology use in the real-world, particularly when exploring accessibility challenges [Anthony et al. 2013, Montague et al. 2014, Naftali and Findlater 2014]. We developed TinyBlackBox (TBB) [Montague et al. 2015], a system-wide data collection tool that runs as a background Accessibility Service in Android 4.0+ devices. Once installed and activated, TBB will continuously run in the background of the operating system, capturing the user's device interactions system-wide. TBB scrapes application data, including page layouts and interface elements – these are represented in a DOM tree structure, revealing information about the nesting of interface elements. TBB also records all interface interactions, e.g. clicks and swipes made within applications.

In addition to recognizing interface "clicks" and keystroke events, TBB provides overwritten touchscreen drivers. This enables the tool to receive the sub-gesture touch *begin*, *move* and *end* interactions, as typically recorded for touch modeling, and gesture analysis [Froehlich et al. 2007]. We ensure the security of user data by encrypting the log files locally on the device before they are transmitted using HTTPS protocols. Moreover, password fields are never captured and participants can turn off the logging service at any time.

We have integrated TBB with Google Cloud storage to aggregate log data from multiple participants while the study is live. TBB will attempt to synchronize with the cloud storage when the device has an active WiFi connection, at least 40% battery remaining and the device is inactive or charging. Prior to uploading log files, they are compressed for network performance and to minimize cloud storage costs. In addition to transmitting users' log files, TBB periodically pings the cloud storage servers with a status report. We use this to verify that TBB is functioning correctly and that the participants are using the devices regularly – reducing the need to conduct field assessments of the devices and software.

3.2 Segmenting Trials

Our logging software captures a continuous input stream of events (e.g. operating system events, touch events, and screen update events) and screen information (i.e. DOM trees). The first step is to segment this stream into trials.

Finding the first and last keystrokes of a trial can be challenging in everyday text-entry data. To this end, we perform a series of segmenting steps, which originate new trials. First, we use unlock/lock actions to segment the data stream into individual device sessions. Second, within each session, segmentation occurs when users change focus of text field. Third, end-of-sentence punctuations, new line characters, symbols, and characters not part of the language are used to segment sentences within each text field. Identifying pauses is the fourth step in segmentation. Because users can pause for different periods of time, we computed an average time between keystrokes for each participant and week, using laboratory data (see Section 4). Pause thresholds were 3 standard deviations to each of these means.

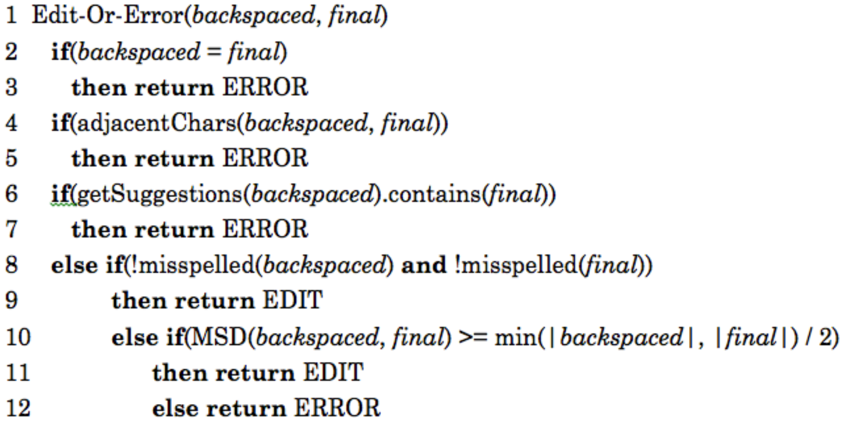
Segmenting pauses can occur in the middle of words. In these cases, the partial word is maintained just for input speed calculations. However, the same word is included in the next trial to prevent errors from being counted twice. Finally, after segmenting the input stream, trials with less than 5 transcribed characters are discarded, as they result in inaccurate input measures.

3.3 Distinguish Errors from Edits

In laboratory studies, all backspaces are regarded as error corrections since participants are aiming to match a required sentence. In field studies, backspaces may consist of error corrections or they may indicate “changes of mind”. As highlighted by Evans and Wobbrock (2012), we must distinguish between errors and edits. Backspaces for edits should be filtered because they do not represent typing errors.

To distinguish between errors and edits, backspaced text is compared to the text entered in its place, word by word. In case users stop backspacing mid-word, the partial word is extended up to the nearest space (with the re-entered text) to make a complete word. If the backspaced text is different from the re-entered text, then we need to check whether it was an edit or error correction.

Previous work assumed that most backspaces take place after the whole word is written, which is not the case for non-visual input. From empirical observations, blind users tend to correct errors as soon as they hear the auditory feedback for entered characters. Moreover, most errors are substitutions of adjacent characters [Nicolau et al. 2015]. Thus, we propose an algorithm that considers blind users’ typing behaviors. The pseudo-code to distinguish between errors and edits is given in Figure 1.



```
Edit-Or-Error(backspaced, final)
  if(backspaced = final)
    then return ERROR
  if(adjacentChars(backspaced, final))
    then return ERROR
  if(getSuggestions(backspaced).contains(final))
    then return ERROR
  else if(!misspelled(backspaced) and !misspelled(final))
        then return EDIT
        else if(MSD(backspaced, final) >= min(|backspaced|, |final|) / 2)
            then return EDIT
            else return ERROR
```

Figure 1. Algorithm to distinguish between an edit and an error.

If the backspaced text is not the same as the text that replaced it, then we check whether *each* re-entered character is adjacent to the backspaced character. If so, backspaces are classified as error corrections (Figure 2.a). Otherwise, we use Hunspell[3] to get spelling suggestions for the backspaced word. Similarly to Evans and Wobbrock (2012), if the final word is suggested then backspaces are classified as error corrections (Figure 2.c).

Otherwise, we need to find whether the backspaced text and the final text are two different intended words. If both correspond to correctly spelled words, then we classify backspaces as edits (Figure 2.b), as users changed their minds to input a

[3] http://hunspell.sourceforge.net/

different string. If there is a misspelling, then our best guess to distinguish between errors and edits is the similarity between the backspaced text and the text that replaced it. Backspaces are classified as edits when the minimum string distance between the backspaced and re-entered text is more than half the length of the strings. In this case, we consider that there are significant differences between words, thus there is a high probability of it being an edit; that is, users were trying to enter a different word in the first place (Figure 2.d).

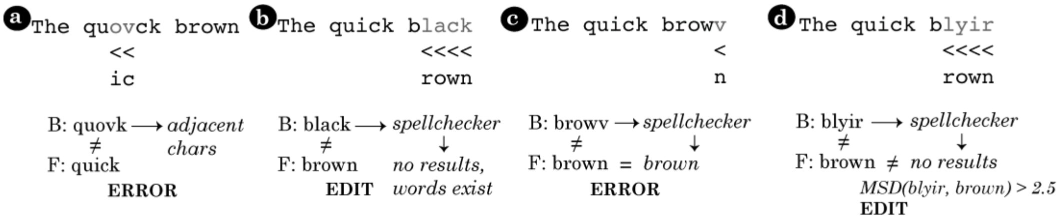


Figure 2. Four input streams and how to distinguish errors from edits. a) backspaces are classified as errors, since all character corrections are adjacent; b) the backspaced text is a different and valid word, showing a change in mind/intent from black to brown; c) the spellchecker return a suggestion that is equal to the re-entered text, meaning that backspaces were errors; d) spellchecker does not return results, but words are significantly different, thus we assume it is an edit.

3.4 Calculating Typing Speed and Error Rates

Computing input speed is straightforward once a trial is segmented. Conversely, error rates are much harder. Key performance measures include uncorrected, corrected, and total error rates [Wobbrock and Myers 2006].

Uncorrected errors are those remaining in the transcribed sentence. Corrected errors consist of backspaced characters that were erroneous, and total error rate represents the percentage of erroneous characters that were entered, including those that were corrected. All these measures are computed by comparing the input stream to the required/target sentence.

However, in everyday data there is no required sentence. To measure error rates, each transcribed word is checked against a spellchecker [Evans and Wobbrock 2012]. We use Hunspell due to its popularity and reproducibility purposes. The lexicon contained about 44,000 words from the OpenOffice open-source project. If the transcribed word is found, it is considered correct. Otherwise, the top suggested word is taken as the intended word.

Words that did not return any spelling suggestions are marked for manual review. This is useful for words that do not exist in the lexicon, such as abbreviations. The research team can then add them to the lexicon or mark them as errors. At the end, the average minimum string distance from words containing known errors is applied to all stored words with unknown errors.

4. LONGITUDINAL USER STUDY

We believe that an analysis of real-world performance is key to expose the *true* challenges faced by novice blind users. Our ultimate goal is to identify new opportunities to reduce the learning overhead and support better non-visual input on mobile touchscreen devices. In order to achieve these goals, we conducted a longitudinal study that comprised a 12-week field deployment and laboratory evaluations during the first 8 weeks. Laboratory results provide a baseline of participants' text-entry performance. Participants were each provided with a mobile device preloaded with our data collection tool and asked to use the device as their primary phone. Due to the ethically sensitive nature of the research, no participants were asked to consent to their everyday data being shared beyond the research group and as such supporting data cannot be made openly available.

4.1 Participants

We recruited five blind participants, four males, from a local training institution. Participants' age ranged from 23 to 55 (M=37.2, SD=15.2) years, and all participants were legally blind as defined within our IRB approved recruitment criteria. They were experienced desktop screen reader users. However, none had prior experience with touchscreen screen readers.

We recruited novice users for three reasons: 1) these are the majority of available participants in our home country as most blind people do not own a smartphone; 2) it was an unique opportunity to understand how performance evolves over time, since these users are still in a learning state; 3) novice users are usually more willing to participate in field deployments due to the novelty factor.

4.2 Procedure

Participants received basic training on how to use an Android device, particularly *Explore by Touch*. We helped participants transferring all contact information from their feature phones. Since we were interested in understanding natural typing performance from everyday use, we did not force usage protocols. Participants were informed that usage was being recorded, namely used applications, touchscreen interactions, and text-entry data.

In addition to real-world data, we conducted controlled weekly laboratory experiments. Participants performed 20 minutes of text-entry trials and were asked to type as quickly and accurately as possible.

We created a Portuguese text-entry corpus from news articles, using the methodology proposed by MacKenzie and Soukoreff (2003). The frequency of letters in the resulting corpus had a correlation with the language of 0.97. Each trial contained one sentence comprised of five words; each word with an average size of five characters. The application randomly selected the sentences for the session to avoid order effects. The experimental application started the trial by reading the target sentence aloud via the device's Text-to-Speech engine. After each sentence, participants pressed the return key twice to advance to the next trial. We used an unconstrained text-entry protocol [MacKenzie et al. 2001], where participants were free to correct errors. To ensure that participants would not practice the trials outside the laboratory evaluations, the application was installed on the participants' device at the beginning of each session, and uninstalled at the end. Automatic correction and cursor movement operations were not used during the trials.

Our study was carried out in Portuguese. Because of this, there are a number of letters that are uncommon in the written language, and therefore do not appear within our trial sentences (e.g. W and Y). Subsequently, these keys contain no examples of intended interactions within our evaluation.

4.3 Apparatus

Participants were each provided with a *Samsung S3 Mini* touchscreen smartphone, running Android 4.1 operating system. We enabled the *Talkback* screen reader and pre-installed our data collection service, TinyBlackBox (TBB) [Montague et al. 2015]. TBB was designed to constantly run in the background, capturing users' interactions with the device. This approach enabled us to capture text-entry performance throughout the 12-week period.

The S3 Mini default input method was Samsung's own Android QWERTY keyboard. Although visually the keys have both horizontal and vertical spacing, when *Talkback* is enabled and the participants touch the screen, they receive feedback for the nearest key to their touch point. However, when moving from one key to another, the key with current focus occupies the spacing. This means that target boundaries can grow and shrink based on the exploration paths. S3 Mini's default keyboard was used throughout our study, both in laboratory evaluations and real-world settings.

4.4 Dependent Measures

Text-entry performance was measured by analyzing trials' input stream [Wobbrock and Myers 2006]. We report on words per minute (WPM), total error rates, uncorrected error rates, and corrected error rates. Moreover, we investigate character-level errors and types of errors (substitutions – incorrect characters, insertions – added characters, and omissions – omitted characters). Touch exploration behaviors were measured using x, y positions and variability (hit point deviations), movement time, movement distances, Path Length to Task Axis length ratio (PL/TA), count and duration of pauses within the movements [Hwang et al. 2004, Keates and Trewin 2005, MacKenzie et al. 2001], and visited keys.

4.5 Design and Analysis

We performed Shapiro-Wilk tests on all dependent measures. For normally distributed values we used a mixed-effects model analysis of variance [McCulloch and Neuhaus 2001]. Mixed-effects models extend repeated measures models, such as ANOVAs, to allow unequal number of repetitions for unbalanced data such as ours, in which we have different numbers of trials per week for each participant. We modeled *Week* and *Data Type* (lab, wild) as fixed effects. Trial was included as a nested factor within both factors. Participant was modeled as a random effect. For the laboratory data, *Participant* and the interaction between *Participant* and *Real-World Usage Time* were modeled as random effects to account for correlated measurements within subjects over time

For the measures that were not normally distributed, we used the nonparametric *Align Rank Transform* procedure [Wobbrock et al. 2011] and then used the mixed-effects model terms previously described for further analysis.

5. LABORATORY TYPING RESULTS

In this section, we aim to characterize novice blind users' text-entry performance and learning when using *Explore by Touch*. We analyze input speed, accuracy, and character-level errors over an eight-week period in laboratory settings. Finally, we characterize users' touch exploration behaviors and provide insights on how and why input performance changes over time.

5.1 Everyday Usage

Participants used their mobile devices for a variety of text-entry tasks, including adding contacts (14%), messaging (6%, e.g. *WhatsApp*), and writing SMS (67%). To control for device usage when analyzing participants' laboratory performance, Table I and Table II summarize the number of characters entered and time spent using a virtual keyboard per participant, respectively. Overall, participants entered a total of 32,764 characters over eight weeks. They spent a total of 51 hours entering text. Generally, the number of characters entered is directly related with time spent typing. However, there is a high variance in usage results both between participants and weeks. For instance, while P2 and P3 were particularly active in the fourth week, others such as P4 were more active in the last two weeks. P5 was the least active with an average usage of 12.5 minutes (SD=20) per week. On the other hand P2 and P4 spent on average 125 (SD=110) and 111 (SD=65) minutes typing per week.

Table I. Characters entered in-situ. Columns represent weeks.

	1	2	3	4	5	6	7	8
P1	245	405	555	678	799	133	732	1292
P2	1283	648	1548	5396	1248	411	2120	208
P3	75	697	579	1115	310	1205	1	447
P4	1002	1022	566	601	2435	603	2578	1099
P5	32	45	22	21	12	24	189	383

Table II. Time spent typing in-situ (minutes).

	1	2	3	4	5	6	7	8
P1	66.2	62	46.6	54.6	101	26.7	46.5	85.9
P2	180	53.6	98.7	383	92.8	29.8	149	12.3
P3	1.78	85.8	99.1	170	40.7	131	0	57.7
P4	160	196	43	36.5	127	36.5	201	91
P5	5.25	3.7	7.4	1.5	0.45	1.17	15.2	65.3

5.2 Laboratory Typing Performance

In total, participants produced 11,560 characters from which 1,323 were backspaces, resulting in 10,237 transcribed characters. In this section we analyze input performance regarding speed and accuracy. To assess input speed, we used the words per minute (WPM) measure calculated as (length of transcribed text – 1) * (60 seconds / trial time in seconds) / (5 characters per word).

Slow learning rate. Participants improved on average 2.4 wpm (SD=.36) from week 1 with 1.6 wpm (SD=.23) to 4 wpm (SD=.35) after eight weeks. We found a significant effect of *Week* on *WPM* [$F_{1,7}$=12.329, p<.001] as all participants improved over time. Nevertheless, considering that they were familiar with QWERTY keyboards, learning rates are still low with an average improvement of 0.3 wpm per week.

Still improving after eight weeks. Figure 3 shows WPM graphed over eight weeks. We can see that participants are still improving input speeds at the end of the user study. Fitting power laws [Wobbrock 2007] to entry rates and extrapolating to twice the weeks gives an average entry speed of 5 wpm in week 16th.

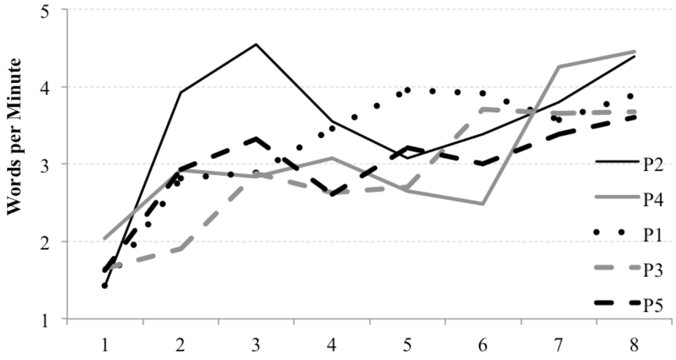


Figure 3. Words per minute over 8 weeks.

External factors can negatively influence performance. We also notice that P2 and P4 have atypical changes in performance in weeks 4 and 7, respectively. When debriefing P2 about this sudden drop in performance, she mentioned perceiving the speech feedback being slower while typing after installing a 3rd party app, *WhatsApp*. In fact, this is a known issue with this particular application. Although we are not able to confirm that speech feedback changed, we can show that both number of pauses and duration of pauses during movement, increased from week 3 to week 5, while movement speed and distance traveled decreased in the same time period (see Section 5.4). This suggests that external factors had an influence in this participant's typing behavior (e.g., other apps or emotional issues).

In-situ usage improves performance. Regarding P4, the abrupt increase in input speed is most likely related with the increase of usage in week 7 (Table I and Table II). After debriefing P4 in that week, he mentioned that he was finally using his phone to the fullest, particularly sending and receiving text messages. He stated "*… the phone is finally fully accessible to me, I can send SMS, I can send text messages via Skype, I can send all the messages that I want*". Therefore, we believe the sudden increase in input speed is due to his increase in usage of messaging applications. In fact, we found a significant medium size effect between *Input Speed* and *In-Situ Usage time* [Pearson's $r_{(290)}$=.353, p<.001].

To analyze input accuracy, we calculated: 1) uncorrected - erroneous characters in the final transcribed sentence, 2) corrected - erased characters that were erroneous, and 3) total error rates - erroneous characters that were entered (even those that were corrected) [Wobbrock and Myers 2006].

Total error rates tend to 7.4%. P2 achieved the highest total error rate of 45% on week 1 and finished the user study with the lowest rate of 5.4% by week 8. Overall participants started with an average total error rate of 26% (SD=11.7%) and finished with 7.4% (SD=1.7%) [$F_{1,7}$=4.176, p<.001]. Moreover, Figure 4 shows that error rates start to stabilize around that value.

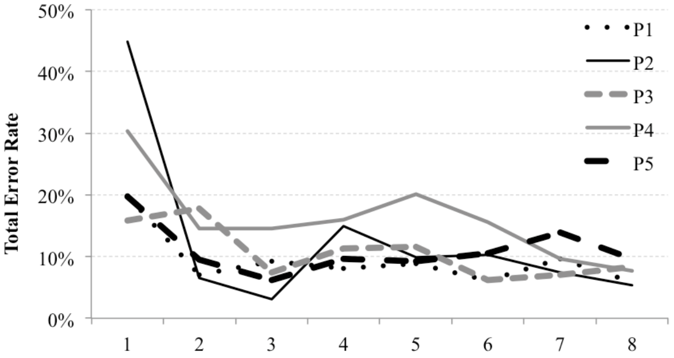


Figure 4. Total error rate over 8 weeks.

Errors are usually corrected. Table III shows the uncorrected error rates for each participants and week. Overall, when given the chance, users tend to correct most errors, resulting in high quality transcribed sentences. This goes in line with previous findings for sighted users [Soukoreff and MacKenzie 2003]. For instance, P1 and P2 had the lowest uncorrected error rates with 0% and 0.3% by week 8. On average, participants left only 1.6% (SD=1.4%) errors in the transcribed sentences by week 8, which resulted in a significant effect of *Week* [$F_{1,7}$=2.306, *p*<.05].

Table III. Uncorrected error rates (%). Columns represent weeks.

	1	2	3	4	5	6	7	8
P1	4	0.4	1.9	1.4	2.3	0.3	2.6	0
P2	1	1	0.3	0	0	0	1.5	0.3
P3	7.6	8.5	3.4	4.1	0.5	2.8	1.9	2.5
P4	20	4.7	5.2	6.3	7.8	3.2	3.2	1.9
P5	11	5.6	4.3	5.3	5.3	2.3	5.1	3.3

Table IV. Corrected error rate (%). Higher is better.

	1	2	3	4	5	6	7	8
P1	74	77	63	89	81	81	77	91
P2	87	55	73	89	84	91	85	68
P3	62	50	41	72	50	46	71	57
P4	69	81	69	68	71	56	62	60
P5	86	100	60	50	92	86	89	88

23-39% of deletions were inefficient. Corrected error rates (Table IV) illustrates the amount of effective *"fixing"* and allows to answer the question "of the erased characters, what percentage were erroneous?" High rate means that most of erased characters were errors and should have been corrected. Participants achieved average corrected error rates between 61% (SD=12%, week 3) and 77% (SD=11%, week 7), which means that 23% to 39% of deleted characters had been correctly entered. This occurs because errors are not immediately recognized. For instance, when phonetically similar characters are entered (e.g. N→M), users only notice that mistake when the word is read aloud. To fix the error, several characters, including correct characters, are usually deleted. A detailed inspection of logs files shows that editing operations, such as cursor movement, were never used. Average corrected error rate per week is 73%, which remains fairly constant throughout the eight weeks [$F_{1,7}$=.98, p=.447].

13% of time is spent correcting errors. The time spent correcting errors is subsumed by input speed (see Section 5.2.1); however, such analysis does not provide insights on the cost of such corrections. Examining correcting actions shows that participants spent on average 32% (SD=17%, MIN=19% [P5], MAX=65% [P2]) of their time correcting text in the first week. Performance significantly improved over time and by week 8 only 13% (SD=1.8%) of time was spent in this task [$F_{1,7}$=4.806, p<.001].

5.3 Character-Level Errors

In this section, we present a fine-grained analysis by categorizing types of input errors: insertions, substitutions, and omissions [MacKenzie and Soukoreff 2002]. We report aggregate measures, which represent the method's accuracy over all entered characters, but also at the level of individual letters [Wobbrock and Myers 2006]. These findings can aid designers in addressing specific types of errors and characters.

Substitutions are the most common type of error. Figure 5 illustrates the types of errors over the eight-week period. Substitution errors were consistently higher than insertions and omissions. Although there was a significant decrease in substitution error rates over time, from 24% (SD=12%) to 6% (SD=1%) [$F_{1,7}$=3.518, p<.005], they still remain significantly higher than the remaining types of errors [$F_{2,8}$=125.321, p<.001]. In fact, substitution error rate is higher than omissions and insertions combined. This result holds true for all participants.

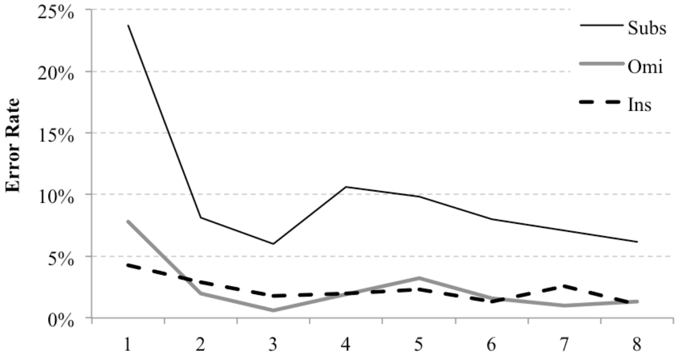


Figure 5. Total error rate for each type of error.

Similar substitution rates across keys. Overall, participants had similar error rates across all intended keys. No row, column, or side patterns emerged from weekly data. Moreover, keys near edges had similar accuracy rates to those in the center.

No clear substitution pattern. To analyze the most common substitution errors, we created confusion matrices. In week 8, some of the most common substitutions were Q→E (33%), B→H (17%), P→O (9%), P→L (4%), R→T (4%). Unlike sighted users that experience substitution patterns towards a predominant direction [Findlater et al. 2011, Nicolau and Jorge 2012], blind users' patterns are less clear.

This is most likely related with the differences between visual and auditory feedback when acquiring keys. Further discussion on this topic is available in Section 5.3.

Adjacent phonetically similar characters promote substitutions. Since feedback is solely auditory, phonetically similar characters have the potential to be confused when blind users are exploring the keyboard. In the Portuguese language, particularly when using Android's Text-to-Speech engine, there are three cases prone to confusions: I-E, O-U, and M-N. For I-E substitution error rates are constantly low over time (0-1%) and inexistent from week 5. Regarding O-U substitutions, error rates are slightly higher with 8.5% in week 1 and decreasing to 0.5% in week 8. Finally, concerning M-N substitutions, error rates remain between 3% (weeks 1-3) and 6.5% (week 5) across the eight-week time period. Indeed, in week 8, error rates are still 4.5%. No other adjacent pair of letters obtained such a consistently high (and symmetrical) error rate over time. These results suggest that phonetically similar letters that are close together have higher probability of being substituted.

68% of omission errors are left uncorrected. Omission error rates decreased 6.5% from week 1 (M=8% SD=6%) to week 8 (M=1% SD=0.7%) [$F_{1,7}$=3.858, p<.005]. Unlike substitutions, the majority of omission errors are not corrected. On average 68% (SD=14%) of errors are left uncorrected. These errors are usually described as cognitive errors [Kristensson 2009]. A common explanation is misspellings or users forgetting to type certain letters. However, leaving errors uncorrected may also be related with (lack of) feedback after an attempt to enter a character, confirming that an input action had a consequence. This option seems less likely since users received feedback after each character entry. Although omissions only account for 2.4% of errors (Figure 5), they are the least likely to be corrected.

5.4 Touch Exploration Behaviors

In this section we provide new insights on participants' touch exploration behaviors. We examine the three stages that compose a key selection: touching the screen, moving the finger to find the intended key, and lifting the finger. For this analysis, we removed outlying points where the entered key (on lift) was more than one key distance away from the intended key in either x or y direction to account for transposition or misspelling errors.

It is noteworthy that before touching the screen and landing on a key, users do not receive any feedback. Unlike sighted users, which aim towards a visual stimulus, blind users solely resort on their spatial model of the keyboard and some physical affordances (e.g. device size).

Users land on intended keys nearly half the times. By week 8, 48% (SD=12%) of key presses landed within the boundaries of intended targets. This number may seem low, but it is not unexpected given that participants did not receive any auditory feedback until this point. Nevertheless, performance significant increased from week 1 (M=27%, SD=15) to week 8 [$F_{7,28}$=5.222, p<.01], showing that users gain a better spatial model of the keyboard. We found that at week 8, 91% (SD=5%) of the times, participants land either inside the intended key or an adjacent key. Also, landing on the correct row (M=78%, SD=7%) is easier than landing on the correct column (M=59%, SD=11%) [$F_{1,4}$=27.611, p<.01], which is not surprising given that rows make larger targets than columns.

Keys near physical edges are easier hit. Throughout the eight-week period, keys that were positioned on physical edges were easier to land on. For instance, in week 8, participants correctly landed on characters A and Q in 75% and 71% of times, respectively. On the other hand, characters such as B or M were only correctly hit 14% and 16%, respectively. The space bar consistently outperformed the remaining keys (week 8, M=99%), most likely due to a combination of its positioning (on the bottom edge) and width (five times larger).

Emergent keyboard is shifted towards the bottom and most key overlaps are horizontal. We examined the emerging key shapes and sizes using hit points. Figure 6 illustrates the emergent keyboard for week 8; that is, the keyboard layout that results from participants' touches. In week 1, the key sizes are larger and shifted towards the center of the screen, where users started their exploration, which resulted in larger overlaps between keys. By week 8, participants are able to land nearer to keys; however, there are still significant overlaps, mostly horizontally. Characters M and N are particularly interesting, since they present the largest overlap (Figure 6). Also, we can see that hit points tend to occur below the center of the intended target.

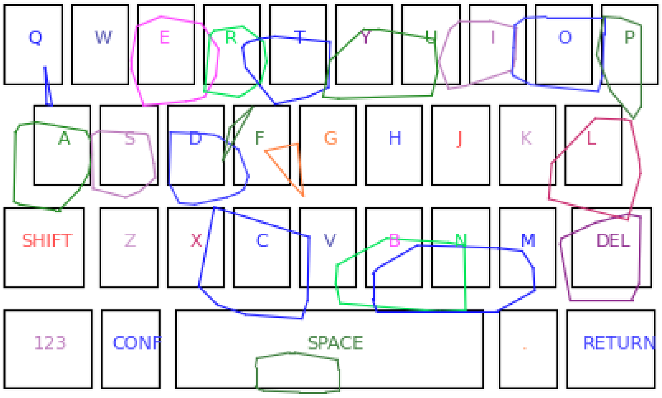


Figure 6. Polygons encompass hit points within a standard deviation of key centroid.

Previous research has investigated text-entry performance by blind users. However, results tend to focus on performance measures, such as time and errors. In the following analysis, we aim to establish why performance improvements occur by conducting a thorough analysis of touch exploration behaviors.

Users visit on average one extra key. In the first week, the average number of visited keys per keystroke was 4.9 (SD=1.9). Participants significantly improved their performance achieving an average of 2 visited keys (SD=0.3) by week 8 [$F_{7,28}$=5.133, p<.001]. Similarly, the number of target re-entries (entering the same target for the second time) also improved from 6.6 (SD=3.2) to 0.8 (SD=0.3) [$F_{7,28}$=7.498, p<.001]. This corresponds to an average of 49 traveled pixels (SD=11), where 60% of movement is done in the x-axis, which is consistent with previous results where users are more likely to land on the intended row and then perform horizontal movements.

Users learn how to perform more efficient explorations. In order to understand exploration efficiency, we calculated the Path Length (movement distance) to Task Axis length (Euclidean distance between hit point and center of target) ratio. Participants significantly improved over time from 3.6 (SD=1.3) to 0.95

(SD=0.15) [$F_{7,28}$=6.033, p<.001]. Notice that we obtained an average ratio below 1 because the Task Axis length is the distance to the center of the target. Users only require traveling to the edge of the target in order to select the key.

Keystroke time is on average 1.9 seconds. In line with previous touch measures, movement times also improved from 4.1 seconds (SD=1.4) to 1.9 seconds (SD=0.3) [$F_{7,28}$=5.424, p<.001]. This value may seem high, but it is expected since users need to wait for auditory feedback to confirm which letter they are touching. As a consequence, entry times are directly related to speech rate and delay. Figure 7 illustrates P1's dwell times in week 1 and 8. Longer pauses are clearly visible in the first week. Also, because feedback is received when entering keys, pauses often occur near their edges.

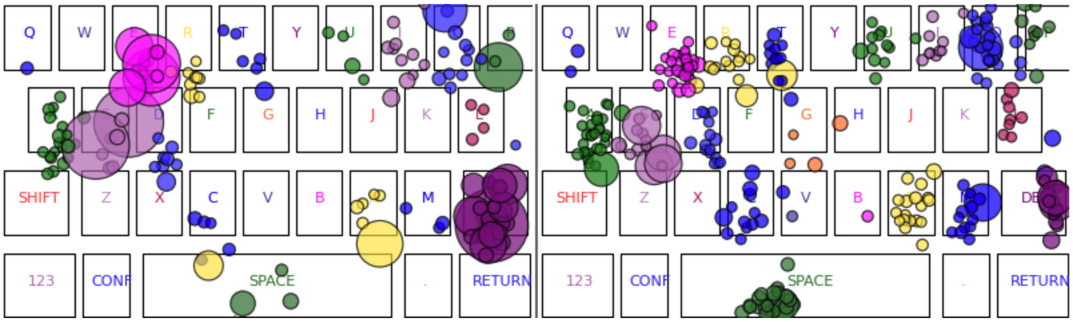


Figure 7. A circle indicates a pause; size represents its duration. Left - week 1 for P1, Right - week 8 for P1.

Keys near physical edges require less time to press but do not result in lower error rates. We found significant differences between keys located near the device's edge, such as Q, A, P, and L, and all other keys regarding movement time [week 8, Z=2.032, p<.05]. Nevertheless, this difference does not result in accuracy improvements. In fact, border keys have a slightly higher substitution rate (week 8, 7% vs. 5.4%, n.s.).

Insertion errors have smaller movement times and distances. Insertion errors are related to unintentionally and accidentally entered characters. Knowing how to filter these keystrokes can result in performance improvements. When analyzing movement times and distances, we found significant differences between correct entries and insertion errors [$F_{1,4}$=23.287, p<.01; $F_{1,4}$=24.119, p<.01] throughout the eight-week period. These results suggest that touch data can be used to classify insertions.

While hit point and movement analysis examined where users land on the screen and how they explore the keyboard, respectively, an examination of lift point allows us to understand the final step of selecting a key. It is particularly relevant to understand in what conditions substitution errors occur.

Lift points are spread-out over keys' boundaries. Figure 8 illustrates all lift points for week 8. Data shows that points are spread over intended keys and particularly close to their edges. Unlike sighted users [Findlater et al. 2011, Nicolau and Jorge 2012], there is not a clear touch offset direction, which can have significant implications when building touch models for this user group. Moreover, hit point deviations (standard deviations) remain unchanged across time with 25.6px in week 1 and 24.3px in week 8, which is approximately half the size of a key. This suggests that users may be prone to slip errors; that is, slipping to a nearby key just before selecting it.

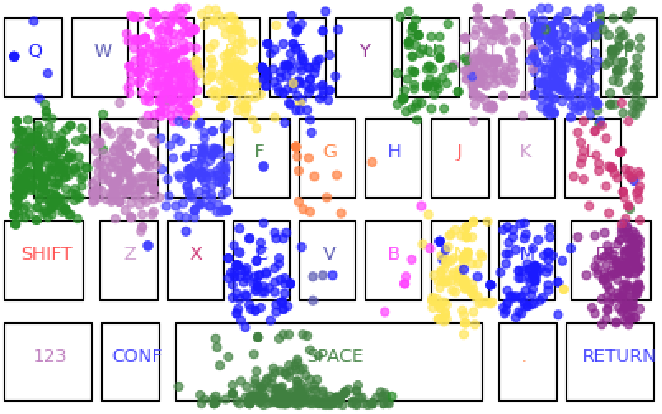


Figure 8. Lift points for all participants in week 8.

There is more to substitution errors than slips. We classified as finger slips all entries where the last visited key was the intended target. Although we are not applying a time threshold, this measure gives us all entries that need to be considered as slip errors. Overall, in week 1, 37.5% (SD=17%) of substitution errors were slips. In week 8 we obtained a similar value of 38.4% (SD=12%) [$F_{1,7}$=2.095, p>.05]. Notice that slip errors account for fewer than 50% of substitution errors by week 8. Taking into account that users should receive speech feedback before selecting the intended key, we analyzed whether participants' finger paths crossed it at some point during movement. In week 8, for 64% (SD=9.8%) of substitution errors, participants were inside the boundaries of the target at some point in their touch paths; however, failed to select it in a timely manner. After identifying some of the instances where these errors occurred, we conducted a manual examination of the recorded videos. We noticed that most of the cases were related to a significant delay between speech feedback, which resulted in a mismatch between the key being heard and touched at that moment. Participants tried to compensate for this delay by performing corrective movements, but often resulted in entering the incorrect key. Further research should explore this issue by investigating the effect of auditory delay on input accuracy.

For some substitutions, intended keys are not even visited. According to the results described above, in week 8 there are still 36% of substitutions where participants did not even visit the key they were aiming to press. This means that they performed a selection without hearing the intended key. From visual inspection of individual keystrokes' movements, we derived several reasons for this behavior: *1) Accidental touches* – similarly to insertion errors, participants unintentionally touch the keyboard close to the intended character. These keystrokes are short in distance and time. *2) Phonetically similar keys* – this happens when users cross a key that sounds similar to the intended character (e.g. while aiming for M, the user lands on B, moves to the right, enters N, and lifts the finger), resulting in a substitution error. *3)*

Overconfidence on spatial model – in some substitution instances it seems that participants overly rely on their spatial understanding of the keyboard by performing a gesture and selecting a key without waiting for feedback. Lastly, *4) Feeling lost and giving up* – some exploration paths show fine-grain movements near the intended key, going back and forth; however, participants never hit the intended character.

6. EVERYDAY TYPING RESULTS

This section presents the input performance gathered from the field study. We start by validating the proposed algorithm to compute intent; then, we report on our dataset and results from everyday typing data.

6.1 Validating Intent Algorithm

In order to validate our approach to compute intent from a series of keystrokes (see Section 3), we compared the computed intended sentence with the existing ground truth, i.e. required sentences from the laboratory evaluation. In summary, we ran the algorithm for all transcribed sentences in the laboratory dataset and compared the computed intent with the original required sentence. Although participants' writing style may be different in the real-world, such results can shed light about our algorithm's effectiveness.

Table V. Minimum word distance, minimum string distance, and uncorrected error rate.

	Algorithm Performance Measures		**User Performance Measures**	
	MSD (Intended, Required)	**Accuracy Computed Intent**	**Uncorrected (Transcribed, Intended)**	**Uncorrected (Transcribed, Required)**
W1	13.30%	67.96%	7.6%	8.6%
W2	3.96%	82.02%	2.8%	4%
W3	1.98%	92.42%	2.2%	3%
W4	3.90%	86.32%	2.9%	3.4%
W5	4.90%	88.44%	2.8%	3.2%
W6	1.96%	94.06%	1.7%	1.7%
W7	2.00%	92.76%	2.0%	2.9%
W8	2.20%	93.22%	1.3%	1.6%

The first column of Table V shows the character difference between the required sentence and the computed intent sentence. Overall, differences are small and decreased with time from 13% (SD=9.7%) to 2.2% (SD=1.7%). It is noteworthy that in week 1, differences between the required sentence and computed intent are substantially larger than in remain weeks. Such result can be explained by the significantly higher uncorrected error rate in week 1 (see Table III, M=8.72%), as this was users' first contact with a smartphone. Participants were still learning how to input text on their devices. In fact, one of the participants achieved an uncorrected error rate of 20%, resulting in nearly unreadable sentences. Obviously, transcribed sentences with more errors are generally hard to understand user's intent. Thus, our approach might not be as effective when transcribed sentences have low quality. However, low quality sentences are uncommon after week 1; when given the chance, blind users tend to correct most errors. This is shown in the following weeks performance (see Section 5.2). After the first week, participants average uncorrected error rate stabilized between 1.6% and 4%, resulting in higher accuracy in the computation of intent. Required sentences and computed intent differed by only 2-5 characters in every 100 characters (Table V, first column).

The second column of table V shows the percentage of computed intended words that match the original required words. Overall, results follow the same trend of character-level data. The algorithm's accuracy is 68% (SD=11.4%) in week 1 and significantly improves in the following weeks, yielding a correct intended word 93% of times in week 8.

The third and fourth columns of Table V are related with the effect of using the required sentence or computed intent on users' performance results. These columns show uncorrected error rates when comparing users' transcribed sentences with required and computed intent sentences, respectively. Overall, both measures are similar, demonstrating the effectiveness of the proposed method. Differences are always within a 1.2% error (M=0.67%, SD=0.4%); however, uncorrected error rates using the computed intent sentences are consistently lower than using the original required sentence, giving an optimistic view of error performance. The result is related with our approach to compute intent, which aims to find the most similar word to the transcribed text, minimizing the differences between transcribed and intended sentences. This knowledge should be taken into account when analyzing field results. Nevertheless, despite the slight difference in user performance, we found no significant differences between the two measures [Z=-1.214, *p*=.225, *r*=.38], whether we use the required sentence or computed intent; that is, there is no significant differences between the third and fourth column of Table V.

It is noteworthy that these results may be related to the general high quality of transcribed sentences in the laboratory study (M=1.6% SD=1.4% *uncorrected error rate* by week 8). Overall, 87% of words were considered correct by the spellchecker. It is still an open question, whether blind users maintain this level of accuracy in everyday mobile typing tasks.

6.2 Segmented Trials

Table VI illustrates the number of segmented text-entry trials per participant and week. In week 12, we discarded 10 trials of P4 because he was using a Bluetooth keyboard. In week 5, the same participant had 5 trials where his typing speed significantly improved from 4 to 25 *words per minute*. Manual inspection showed that typing was done without exploration movements, suggesting that a sighted person was using the device. These trials were removed from the dataset.

None of the participants used text-editing operations, such as caret movement, copy, or cut. Similarly, participants did not use auto-correct or prediction. It is unclear why participants did not use those features, whether it was due to lack of knowledge, desire to use or difficulty. Previous work suggested some of these operations are hard to accomplish non-visually [Azenkot et al. 2012].

Segmented trials contained a total of 3,030 words of which 86% were considered correct. Interestingly, this is a similar proportion of correct words as in the laboratory, which give us confidence about our data analysis approach. We obtained an average of 1.5 words per trial; we believe there was a small number of words per trial mostly due to two reasons: 1) writing style – mobile typing tasks are usually short text messages, search queries, and contact management; and 2) pauses -

participants usually paused after writing 1-2 words. Pause thresholds correspond to 3 standard deviations to the average time between keystrokes (see Section 3.2).

Table VI. Number of trials per participant and week.

	P1	P2	P3	P4	P5
W1	10	101	0	19	1
W2	0	47	3	25	2
W3	18	73	19	2	0
W4	15	311	42	2	0
W5	18	80	11	23	0
W6	10	35	26	18	0
W7	20	82	0	136	2
W8	48	21	18	53	41
W9	20	89	36	55	42
W10	23	80	23	17	43
W11	31	55	52	32	0
W12	0	164	27	1	0
Total	213	1,138	257	383	131

6.3 Everyday Typing Performance

In this section we report on participants' input speed and errors during everyday typing tasks, using the previously described segmented trials. We also compare participants' real-world performance with laboratory performance, highlighting their main differences and similarities.

Average six words per minute after 12 weeks. Figure 9 shows participants' input speed over 12 weeks. Overall, the average *input speed* in the real-world improved from week 1 (M=3.2 SD=.8 WPM) to week 12 (M=5.9 SD=.2 WPM). As in laboratory performance, we found a significant effect of *Week* on *WPM* [$F_{7,741}$=16.334, *p*<.001] with all participants improving typing speed over time. Still, learning rates were lower in real-world data with an improvement of 0.2 WPM per week.

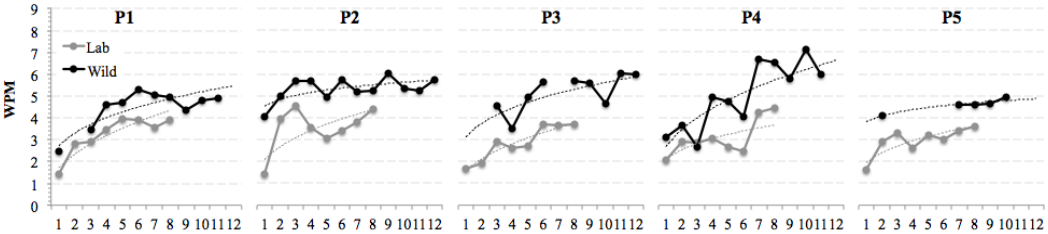


Figure 9. Words per minute for each participant over 12 weeks.

Everyday typing is faster than laboratory. In Figure 9, notice that everyday typing speed is consistently higher than laboratory results. The difference in performance between real-world and laboratory is 1.6 WPM and 1.4 WPM in week 1 and week 8, respectively. We found this difference to be statistically significant [$F_{1,1175}$=243.917, *p*<.001].

Time between keystrokes is smaller in the real-world. In order to further understand why the difference in input speed occurred, we performed an analysis of touch behaviors. Results show that average *distance covered* [$M_{Everyday}$=78px, M_{Lab}=65px, $F_{1,1301}$=.368, *p*=.544], and *average exploration time* [$M_{Everyday}$=2.3s, M_{Lab}=2s, $F_{1,1301}$=2.611, *p*=0.106] are similar between laboratory and real-world data. On the other hand, *inter-key interval* (i.e. time between keystrokes) is significantly smaller in everyday typing tasks [$M_{Everyday}$=592ms, M_{Lab}=1060ms, $F_{1,1175}$=205.686, *p*<.001]. These results suggest that participants were faster to initiate the action to acquire a key, which may be relate to the nature of a composition task in real-world typing.

Uncorrected error rates are higher in the real-world. Figure 10 illustrates participants' *uncorrected error rate* over 12 weeks. The average error rate in the real-world was 10% (SD=10) and 6% (SD=2) for week 1 and 12, respectively. Participants performed significantly more errors during everyday typing tasks [$F_{1,1301}$=34.633, *p*<.001], within 9% of laboratory performance. These results suggest that laboratory results give a skewed view towards more accurate, although slower, typing performance. Participants tend to correct the majority of typing errors in laboratory settings, achieving *uncorrected error rates* between 0% and 3.3% by week 8.

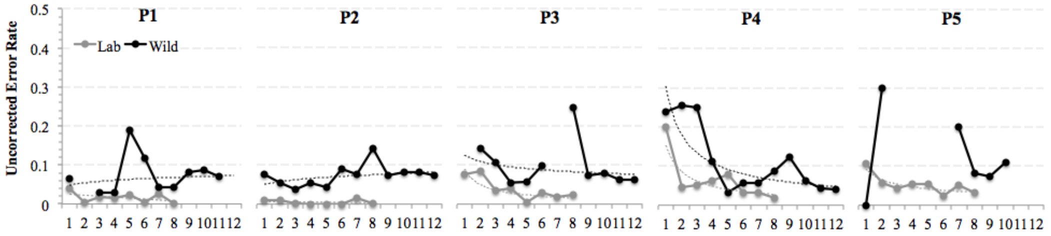


Figure 10. Uncorrected error rate for each participant over 12 weeks.

Corrections are less effective in everyday typing. *Corrected error rates* illustrate the percentage of erased characters that were erroneous. High rate means that most characters were erroneous. Overall, *corrected error rates* were significantly higher in laboratory settings [$F_{1,1301}$=28.105, *p*<.001]. In week 1, the average *corrected error rate* was 38% (SD=46%) in the real-world and 75% (SD=11%) in the laboratory. In week 8 it was 41% (SD=45%) and 73% (SD=16%) in the real-world and in the laboratory, respectively. In addition to being less effective, participants spent relatively (to entered characters) less time correcting sentences during everyday typing than in the laboratory ($F_{1,1175}$=409.400, *p*<.001).

6.4 Character-Level Errors

In addition to overall input performance, we performed a fine-grained analysis on everyday typing data by categorizing types of errors: substitutions, omissions, and insertions.

Substitutions continue to be the most common type of error. As in laboratory performance, substitutions (incorrect characters) are consistently higher than insertions and omissions. In week 1, participants achieve an average *substitution error rate* of 17% (SD=4.6%) and finished by week 12 with an *error rate* of 9% (SD=2%). *Insertion error rates* vary between 9% and 3%, while *omissions error rates* were between 2.5% and 1%.

Differences in magnitude of errors between laboratory and real-world are mostly due to substitutions. Substitution error rates revealed to be the most different between real-world and laboratory data [$F_{1,1301}$=4.111, *p*<.001]. *Omission error rates* (omitted characters) were similar between everyday and laboratory typing data [$F_{1,1301}$=.076, *p*=.783]. For instance, in week 8 average omission rate was 3% (SD=2.6%) in the wild and 1.3% (SD=.7%) in the laboratory. The average *insertion error rate* (added characters) in week 1 was 9% (SD=10%) in the real-world and 4% (SD=1.7) in the laboratory. It was 6% (SD=3) and 1% (SD=.6) in week 8 for real-world and laboratory data, respectively. *Insertion error rates* remained consistently higher in the wild than in the lab [$F_{1,1301}$=28.810, *p*<.001].

7. DISCUSSION

In this section we describe major results, implications for future design of virtual keyboards, and limitations of our work.

7.1 Summary of Major Results

According to laboratory results, participants achieve an *average typing speed* of 4 WPM and 4.7% total error rate after eight weeks of usage. Although performance keeps improving after eight weeks, *learning rate is slow* (0.3 WPM per week). Previous research has shown similar results [Azenkot et al. 2012]. An open question until now was: why and how did users improved typing performance? Overall participants seem to gain a better spatial model of the keyboard by *landing closer to targets*, performing more *time- and movement-efficient paths* towards intended targets, and *less target re-entries*, which resulted in lower number of pauses to hear auditory feedback.

Regarding real-world performance, *input speed* is on average 1.5 times faster. One reason for this increase could be differences in the input task itself. In laboratory studies participants are required to memorize and transcribe a sentence, while in the real world they are performing a composition task. Indeed, *average pause* between keystrokes was smaller in everyday typing tasks, suggesting participants needed less time to think about the next action.

Regarding uncorrected error rates, there is also a difference between laboratory and everyday results. While *uncorrected error rate* is ~1% in the laboratory, it remains above 7% in the real-world. This goes in line with previous field studies with motor-impaired users [Evans and Wobbrock 2012]. Real-world writing is usually more informal, especially in messaging applications. Some examples in our dataset include words where participants were trying to express emotions, such as "yeaaaah" or "noooo". This specific "error pattern" accounted for 0.36% of words. Other examples include abbreviations or slang expressions.

Nevertheless, blind users are usually careful with the text quality, shown by a similar proportion of correct words in the lab and in the wild (86-87%). Overall, this means that incorrect words have more errors in everyday typing tasks.

Character-level analysis revealed that most erroneous characters are *substitutions*. This result occurred in both laboratory and field evaluations. However, in contrast with sighted typing patterns, results do not show a clear offset pattern. Instead, touch *points are scattered over intended keys* and particularly near edges. Finally, participants naturally correct the overwhelming majority of errors (98.4%), which corresponds to about *13% of their typing time*. Moreover, one third of corrections are *counterproductive* as users delete correct characters.

7.2 Implications for Design

Easier, effective, and efficient correction. Corrections are still time consuming and inefficient. None of our participants used cursor-positioning operations throughout the study. It seems that these actions are only expected to be used by expert typists, preventing novice users to do fine-grain corrections. Also, participants did not use auto-correct or auto-complete solutions, although these have great potential to be used in non-visual text-entry to correct missed errors (such as omissions) and improve typing speeds.

Synchronize speech output with touch input. Results suggest that 64% of substitution errors can be due to a mismatch between speech output and touch information. Future non-visual keyboards should prioritize synchronization between input and output modalities.

Filter unintentionally added characters. Accidental touches originate substitution and insertion errors, which in turn take time to correct. However, most of these errors can be filtered out by monitoring movement's time and distance, since they are significantly shorter than correct entries.

Use language-based solutions. The majority of omission errors (68%) go by undetected and therefore uncorrected. Language-based solutions such as spellcheckers seem to be the only plausible solution. Nevertheless, mainstream auto-correct approaches should also be able to deal with some substitution errors, since current algorithms usually weight word corrections by key distance. Although blind users do not show a predominant touch offset direction, most substitution errors were adjacent keys.

Leverage land-on and movement information. Non-visual typing comprises much more than just lift positions. Movement data can provide evidence of what particular key users are trying to select. Future key recognizers should leverage this information and try to predict the most probable targets (see [Pasqual and Wobbrock 2014, Wobbrock et al. 2009] for pointing prediction). This information could be used with language models to narrow the search space of word-corrections or provide character suggestions when users delete a letter.

Touch models need to adapt to expertise. Leveraging movement data is particularly relevant on early stages of learning when users perform longer exploration paths. While expert users may land on the intended target most of the times, novice users still need to search for the intended key and wait for auditory feedback. Therefore, touch models need to be able to adapt to different typing behaviors (i.e. abilities) and learning rates.

Evaluation settings and the speed-accuracy trade-off. Results show that laboratory studies underestimate the typing speed of blind users. On the other hand, real-world performance is more error-prone. The number of errors blind users commit during everyday typing tasks is higher than in laboratory settings. While this speed-accuracy tradeoff is well known in the literature, correction solutions have the potential for a greater impact in users' everyday tasks.

7.3 Limitations

Our participants only included five novice blind users. Despite being a small number of participants they represent a crucial user group when the goal is to designing easy-to-use solutions and identify challenges with current virtual keyboards. Although typing performance and touch behaviors can be significantly different for expert users, the derived implications may still apply. For instance, using more efficient correction strategies or language-based solutions can further improve experts' typing performance. Future research should replicate the analysis reported in the paper with more experienced blind typists in order to examine character-level errors and touch movement behaviors.

In this paper, we contribute with a method to compute intent and performance from everyday non-visual typing tasks. An alternative and common method to collected typing performance from field data is to prompt users with target sentences throughout the day [Trewin 2004]. The method has a clear advantage of knowing what users intent to type; however, participants could treat prompted typing tasks formally, giving a biased view of real-world input performance. Nevertheless, we believe this to be an interesting research topic for future work.

8. CONCLUSION AND FUTURE WORK

We have investigated text-entry performance of 5 blind users over the course of twelve weeks in both laboratory and real-world settings. Results show that users improve both entry speed and accuracy, although at slow rate. Improvements are mostly due to a combination of factors, such as landing closer to intended keys, performing more efficient keyboard explorations, lower number of target re-entries, and lower movement times.

Regarding correction strategies, users correct most of typing errors, which consumes on average 13% of input time. Substitutions errors were the most common error type in both laboratory and field settings. Nevertheless, results show performance differences between laboratory and field data. In summary, users type faster but less accurately in everyday tasks, which suggests that future error correction solutions will have a higher impact in the real-world.

Overall, we provide a thorough examination on how novice blind users learn how to type on a virtual keyboard. Future research can leverage our approach to analyze field data and apply the design implications that emerged from our results to improve non-visual typing performance. As future work, we intend to extend and integrate our analysis tools into a widespread accessibility service and conduct a large-scale study on non-visual input performance.

ACKNOWLEDGMENTS

We thank all participants from Fundação Raquel e Martin Sain in Lisbon.

REFERENCES

Lisa Anthony, YooJin Kim, and Leah Findlater. Analyzing usergenerated youtube videos to understand touchscreen use by people with motor impairments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI'13). ACM, NY, USA, 1223-1232. http://dx.doi.org/10.1145/2470654.2466158

Shiri Azenkot, Jacob O. Wobbrock, Sanjana Prasain, and Richard E. Ladner. 2012. Input finger detection for nonvisual touch screen text entry in Perkinput. In *Proceedings of Graphics Interface* (GI '12). Canadian Information Processing Society, Toronto, Ont., Canada, Canada, 121-129.

Matthew N. Bonner, Jeremy T. Brudvik, Gregory D. Abowd, W. Keith Edwards. 2010. No-Look Notes: Accessible eyes-free multi-touch text entry. *Pervasive Computing*, 409–426.

Abigail Evans and Jacob Wobbrock. 2012. Taming wild behavior: the input observer for obtaining text entry and mouse pointing measures from everyday computer use. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '12). ACM, NY, USA, 1947-1956. http://doi.acm.org/10.1145/2207676.2208338.

Leah Findlater, Jacob Wobbrock, and Daniel Wigdor. 2011. Typing on flat glass: examining ten-finger expert typing patterns on touch surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI'11). ACM, New York, NY, USA, 2453–2462. http://dx.doi.org/10.1145/1978942.1979301.

Leah Findlater and Jacob Wobbrock. 2012. Personalized Input: Improving Ten-Finger Touchscreen Typing through Automatic Adaptation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI'12). ACM, New York, NY, USA, 815-824. http://dx.doi.org/10.1145/2207676.2208520.

Jon Froehlich, Mike Chen, Sunny Consolvo, Beverly Harrison, and James Landay. 2007. MyExperience: a system for in situ tracing and capturing of user feedback on mobile phones. In *Proceedings of MobileHCI '07*. 57-70. http://dx.doi.org/10.1145/1247660.1247670

Krzysztof Gajos, Katharina Reinecke, and Charles Herrmann. 2012. Accurate measurements of pointing performance from in situ observations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI'12). ACM, New York, NY, USA, 3157-3166. http://doi.acm.org/10.1145/2207676.2208733.

João Guerreiro, André Rodrigues, Kyle Montague, Tiago Guerreiro, Hugo Nicolau, and Daniel Gonçalves. 2015. TabLETS Get Physical: Non-Visual Text Entry on Tablet Devices. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI'15), 39-42. http://dx.doi.org/10.1145/2702123.2702373.

Tiago Guerreiro, Paulo Lagoá, Hugo Nicolau, Daniel Gonçalves, and Joaquim A. Jorge. 2008. From tapping to touching: Making touch screens accessible to blind users. *IEEE MultiMedia*, 48–50.

Faustina Hwang, Simeon Keates, Patrick Langdon, John Clarkson. 2004. Mouse movements of motion-impaired users: a submovement analysis. *Proceedings of the 6th international ACM SIGACCESS conference on Computers and accessibility* (New York, NY, USA), 102–109. http://dx.doi.org/10.1145/1028630.1028649.

Amy Hurst, Jennifer Mankoff, and Scott E. Hudson. 2008. Understanding pointing problems in real world computing environments. In *Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility* (Assets '08). ACM, New York, NY, USA, 43-50. http://doi.acm.org/10.1145/1414471.1414481.

Simeon Keates and Shari Trewin. 2005. Effect of age and Parkinson's disease on cursor positioning using a mouse. In *Proceedings of the 7th international ACM SIGACCESS conference on Computers and accessibility* (2005), 68–75. http://dx.doi.org/10.1145/1090785.1090800.

Heidi Koester, Edmund LoPresti and Richard Simpson. 2005. Toward Goldilocks' pointing device: determining a "just right" gain setting for users with physical impairments. In *Proceedings of the 7th international ACM SIGACCESS conference on Computers and accessibility (Assets '05)*. ACM, New York, NY, USA, 84-89. http://dx.doi.org/10.1145/1090785.1090802.

Per Ola Kristensson. 2009. Five challenges for intelligent text entry methods. *AI Magazine*. 30, 4 (2009), 85. http://dx.doi.org/10.1609/aimag.v30i4.2269.

I. Scott MacKenzie, Tatu Kauppinen, and Miika Silfverberg. 2001. Accuracy measures for evaluating computer pointing devices. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2001), 9–16. http://dx.doi.org/10.1145/365024.365028.

I. Scott MacKenzie and William Soukoreff. 2002. A character-level error analysis technique for evaluating text entry methods. *Proceedings of the second Nordic conference on Human-computer interaction* (2002), 243–246. http://dx.doi.org/10.1145/572020.572056.

I. Scott MacKenzie and William Soukoreff. 2002. Text entry for mobile computing: Models and methods, theory and practice. *Human-Computer Interaction*. 17, 2 (2002), 147–198.

I. Scott MacKenzie and William Soujoreff. 2003. Phrase sets for evaluating text entry techniques. CHI'03 e*xtended abstracts on Human factors in computing* (2003). 754-755. http://dx.doi.org/ 10.1145/765891.765971

Sergio Mascetti, Cristian Bernareggi, and Matteo Belotti. 2012. TypeInBraille: quick eyes-free typing on smartphones. *Proceedings of International Conference on Computers for Handicapped Persons*. 615-622. http://dx.doi.org/10.1007/978-3-642-31534-3_90

Charles E. McCulloch and John M. Neuhaus. 2001. *Generalized linear mixed models*. Wiley Online Library.

Kyle Montague, Hugo Nicolau, and Vicki L. Hanson. 2014. Motor-impaired touchscreen interactions in the wild. In *Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility* (ASSETS '14). ACM, New York, NY, USA, 123-130. http://doi.acm.org/10.1145/2661334.2661362.

Kyle Montague, André Rodrigues, Hugo Nicolau, and Tiago Guerreiro. 2015. TinyBlackBox: Supporting Mobile In-the-Wild Studies. In *Proceedings of the international ACM SIGACCESS conference on Computers and accessibility (Assets '15)*, 379-380. http://dx.doi.org/10.1145/2700648.2811379.

Maia Naftali. and Leah Findlater. Accessibility in context: understanding the truly mobile experience of smartphone users with motor impairments. In *Proceedings of the international ACM SIGACCESS conference on Computers and accessibility*. 209-216. http://dx.doi.org/10.1145/2661334.2661372

Hugo Nicolau and Joaquim Jorge. 2012. Elderly Text-Entry Performance on Touchscreens. In *Proceedings of the 14th international ACM SIGACCESS conference on Computers and accessibility*. 127-134. http://dx.doi.org/10.1145/2384916.2384939.

Hugo Nicolau and Joaquim Jorge. 2012. Touch typing using thumbs: understanding the effect of mobility and hand posture. *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems (CHI'12)*, 2683–2686. http://dx.doi.org/10.1145/2207676.2208661.

Hugo Nicolau, Kyle Montague, Tiago Guerreiro, João Guerreiro, and Vicki L. Hanson. 2014. B#: chord-based correction for multitouch braille input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '14). ACM, NY, USA, 1705-1708. http://doi.acm.org/10.1145/2556288.2557269.

Hugo Nicolau, Kyle Montague, Tiago Guerreiro, André Rodrigues, and Vicki L. Hanson. 2015. Typing Performance of Blind Users: An Analysis of Touch Behaviors, Learning Effect, and In-situ Usage. In *Proceedings of the international ACM SIGACCESS conference on Computers and accessibility* (ASSETS '15). ACM, NY, USA, 273-280. http://dx.doi.org/10.1145/2700648.2809861

João Oliveira, Tiago Guerreiro, Hugo Nicolau, Joaquim Jorge, and Daniel Gonçalves. 2011. Blind people and mobile touch-based text-entry: acknowledging the need for different flavors. In *Proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility* (ASSETS '11). ACM, NY, USA, 179-186. http://doi.acm.org/10.1145/2049536.2049569.

Phillip T. Pasqual. and Jacob Wobbrock. 2014. Mouse pointing endpoint prediction using kinematic template matching. *Proceedings of the 32nd annual ACM conference on Human factors in computing systems* (2014), 743–752. http://dx.doi.org/10.1145/2556288.2557406.

André Rodrigues, Kyle Montague, Hugo Nicolau, Tiago Guerreiro. 2015. Getting Smartphones to TalkBack: Understanding the Smartphone Adoption Process of Blind Users. In *Proceedings of the international ACM SIGACCESS conference on Computers and accessibility* (ASSETS '15). ACM, NY, USA, 23-32. http://doi.acm.org/10.1145/2700648.2809842

André Rodrigues, Hugo Nicolau, Kyle Montague, Luís Carriço, and Tiago Guerreiro. 2016. Effect of target size on non-visual text-entry. *Proceedings of the 18th International conference on human-computer interaction with mobile devices and services* (2016). 47-52.

William Soukoreff and I. Scott MacKenzie. 2003. Metrics for text entry research: an evaluation of MSD and KSPC, and a new unified error metric. *Proceedings of the SIGCHI conference on Human factors in computing systems* (2003), 113–120. http://dx.doi.org/10.1145/642611.642632.

Caleb Southern, James Clawson, Brian Frey, Gregory Abowd, and Mario Romero. 2012. An evaluation of BrailleTouch: mobile touchscreen text entry for the visually impaired. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services* (MobileHCI '12). ACM, NY, USA, 317-326. http://doi.acm.org/10.1145/2371574.2371623.

Hussain Tinwala and I. Scott MacKenzie. 2010. Eyes-free text entry with error correction on touchscreen mobile devices. In *Proceedings of the 6th Nordi CHI* (2010), 511–520. http://dx.doi.org/10.1145/1868914.1868972.

Shari Trewin. 2004. Automating accessibility: the dynamic keyboard. In *Proceedings of the 6th international ACM SIGACCESS conference on Computers and accessibility (Assets'04)*. ACM, New York, NY, USA, 71-78. http://dx.doi.org/10.1145/1028630.1028644.

Jacob O. Wobbrock and Brad A. Myers. 2006. Analyzing the input stream for character- level errors in unconstrained text entry evaluations. *ACM Trans. Comput.-Hum. Interact.* 13, 4 (December 2006), 458-489. http://doi.acm.org/10.1145/1188816.1188819

Jacob O. Wobbrock. 2007. Measures of text entry performance. In Text Entry Systems, MacKenzie and Tanaka-Ishii (eds.). San Francisco: Morgan Kaufmann, 47-74.

Jacob Wobbrock, James Fogarty, Shih-Yen Liu, Shunichi Kimuro, and Susumo Harada. 2009. The angle mouse: target-agnostic dynamic gain adjustment based on angular deviation. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2009), 1401–1410. http://dx.doi.org/10.1145/1518701.1518912.

Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. 2011. The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '11). ACM, New York, NY, USA, 143-146. http://doi.acm.org/10.1145/1978942.1978963

Georgios Yfantidis and Grigori Evreinov. 2006. Adaptive blind interaction technique for touchscreens. *Universal Access in the Information Society*. 4, 4 (2006), 328–337. http://dx.doi.org/10.1007/s10209-004-0109-7.

View publication stats