An Evaluation of BrailleTouch: Mobile Touchscreen Text Entry for the Visually Impaired

Caleb Southern[1], **James Clawson**[1], **Brian Frey**[2], **Gregory D. Abowd**[1], **Mario Romero**[1]

[1]Georgia Institute of Technology
85 Fifth Street NW, Atlanta GA 30332, USA
caleb.southern@gatech.edu, jamer@cc.gatech.edu,
abowd@gatech.edu, mario@gatech.edu

[2]University of Maryland, Baltimore County
1000 Hilltop Circle, ITE Building, Suite 415
Baltimore MD 21250, USA
frey1@umbc.edu

ABSTRACT

We present the evaluation of BrailleTouch, an accessible keyboard for blind users on touchscreen smartphones. Based on the standard Perkins Brailler, BrailleTouch implements a six-key chorded braille soft keyboard. Eleven blind participants typed for 165 twenty-minute sessions on three mobile devices: 1) BrailleTouch on a smartphone; 2) a soft braille keyboard on a touchscreen tablet; and 3) a commercial braille keyboard with physical keys. Expert blind users averaged 23.2 words per minute (wpm) on the BrailleTouch smartphone. The fastest participant, a touchscreen novice, achieved 32.1 wpm during his first session. Overall, participants were able to transfer their existing braille typing skills to a touchscreen device within an hour of practice. We report the speed for braille text entry on three mobile devices, an in depth error analysis, and the lessons learned for the design and evaluation of accessible and eyes-free soft keyboards.

Author Keywords

Accessibility; text entry; blindness; mobile devices; touchscreens; multi-touch interaction; chording; gestures.

ACM Classification Keywords

H.5.2. [Information Interfaces and Presentation]: User Interfaces – Input devices and strategies, Voice I/0.

K.4.2. [Computers and Society]: Social issues – assistive technologies for persons with disabilities.

INTRODUCTION

While volunteering at a local vision rehabilitation center, we observed a teenager using a mysterious portable device with six buttons. Although bulky and awkward, it was entirely effective at making her giggle at what we realized were text messages from a friend. She was like any other teenager with a smartphone, an expert at texting. The only difference was that she was doing it on the braille keyboard of a Braille Sense Plus. Upon researching this device, we were shocked to discover that it cost $6000. We recognized a potential cause for a profound digital divide, and a significant design challenge: *How can we support affordable text entry for the visually impaired on inexpensive commodity mobile devices?*

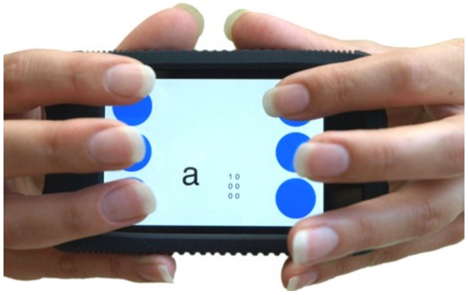

Figure 1. BrailleTouch on a smartphone.

As evidenced by today's young adults, texting is a dominant form of mobile communication and connectivity. American cell owners between the ages of 18 and 24 send or receive, on average, 109.5 text messages compared to 17.1 calls per day [23]. Today's technologies support mobile text entry poorly due to the size constraints of mobile devices. Phones have too many buttons, physical or virtual, that are too small. Most users have no option but to look at their device while typing. This problem is particularly limiting for the visually impaired. With 284 million visually impaired people worldwide [26], it is imperative to create affordable forms of accessible text entry.

Previously, we presented the design of BrailleTouch, an eyes-free text entry system for mobile devices (see Figure 1) [7, 22]. The contributions of the evaluation reported here are: 1) the first formal measure of English text entry speed and accuracy on a standard braille keyboard; and 2) an analysis of typing speed and accuracy across three experimental conditions: a traditional electronic braille keyboard; a soft braille keyboard on a touchscreen tablet; and the BrailleTouch soft keyboard on a smartphone.

RELATED WORK

In this section, we tell the history of manual, mechanical, and electronic braille writing and the most relevant methods for eyes-free mobile text entry.

MobileHCI'12, September 21–24, 2012, San Francisco, CA, USA.
Copyright 2012 ACM 978-1-4503-1105-2/12/09...$10.00.

Device Name	Braille Display	Price (USD 2012)
GW Micro Braille Sense Plus	Yes	6000
GW Micro Voice Sense	No	2000
Humanware VoiceNote BT	No	2000
Freedom Scientific PACmate BX 400	No	1500
AFB Braille+ Mobile Manager	No	1400
Humanware Maestro	No	1300
EmpTech Nano Braille PDA	No	1000

Table 1. Common commercial braille notetakers.

A Brief History of Braille Writing

In 1821, twelve-year-old Louis Braille witnessed Barbier's tactile writing system demonstration [16]. Deeply inspired, he optimized the raised-dot system to a cell that fits under a single fingertip and encodes 63 (2^6–1) characters on a 3×2 binary matrix, with dots numbered one through six in column-major order (see Figure 2) [3].

In 1829, Braille and Barbier introduced the slate and stylus, the analog to pen and paper for sighted people [25]. Until the advent of electronic braille "notetakers", the slate and stylus was the only portable device for blind people to write and read notes. Still in use, it consists of a hinged guide for dot-by-dot manual input with an awl. A person writes braille characters on the back of the page from right to left in mirror image, and turns the page over to read from the front.

Frank Hall invented the first mechanical braillewriter in 1895, which introduced the standard six-key chorded keyboard [25]. The keyboard unfolds the braille cell using dots 1 and 4 as hinges, resulting in the key arrangement of (3, 2, 1) and (4, 5, 6). The Perkins Brailler adopted the Hall keyboard and remains the standard today. In addition to the six main keys, the Perkins has a spacebar, a backspace key on the far right, and a paper advance key on the far left (see Figure 3).

In 1986, predating the PDA by a decade, Dean Blazie introduced the Braille 'n Speak, the world's first portable electronic braille note taking device [10]. Featuring the Perkins-style keyboard and speech output, it became the first alternative to the slate and stylus for taking braille notes while on-the-go. Today there are many braille notetakers, unfortunately all expensive (see Table 1).

While there are many studies of braille reading performance, there exist only two studies measuring braille writing performance on the six-key keyboard, none in English. Lee et al. measured the speed of a commercial braille keyboard for Korean text entry at 27.9 words per minute (wpm) with 2.8% error [15]. Oliveira et al. measured the speed of Portuguese text entry on a Perkins Brailler at 16.7 wpm [20].

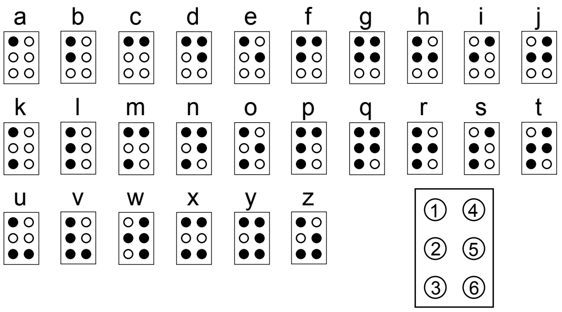

Figure 2. The braille cell and code for English.

Eyes-Free Mobile Text Entry

Although portable, most braille notetakers are too large to be considered mobile devices. We discuss a range of eyes-free mobile keyboards designed for the visually impaired.

Touchscreen Braille Keyboards

Kane et al. presented *Slide Rule*, an interface to control mobile devices with multi-touch input and audio feedback [12]. While Slide Rule employs a soft QWERTY keyboard for text entry, the authors suggested a soft braille chorded keyboard as future work. Fard et al. presented a design for such a device, but did not implement it [6]. Although not mobile, Keles developed a braille keyboard on the IMPAD touch input surface [14].

Oliveira et al. presented *BrailleType*, where the user sequentially enters one dot at a time and signals the end of the braille character with a double tap [21]. Its speed was 1.45 wpm with 9.7% error. Mascetti et al. presented *TypeInBraille*, in which the user types the braille cell one row (2 dots) at a time and enters `space` with a flick gesture [18]. TypeInBraille's speed was 6.3 wpm with 3% error. Azenkot et al. presented *Perkinput*, a chorded input for touchscreens where the braille cell is typed one column (3 dots) at a time [1]. Perkinput's speed was 6.1 wpm with an uncorrected error rate of 3.5%. After the date of our evaluation, Radomir Bruder, a visually impaired hobbyist from Slovakia, released but did not evaluate the *Type Brailler Learn Braille* app for the iPhone [4]. The user can chord all six braille dots simultaneously with the screen facing away, in a similar manner to BrailleTouch.

Other Mobile Keyboards for the Blind

Oliveira et al. and Azenkot et al. both evaluated accessible soft QWERTY keyboards, similar to Apple's *VoiceOver* split-tap keyboard for the iPhone [1, 21]. In the split-tap interaction, the first tap produces a voiced output of the character under the finger and a second tap

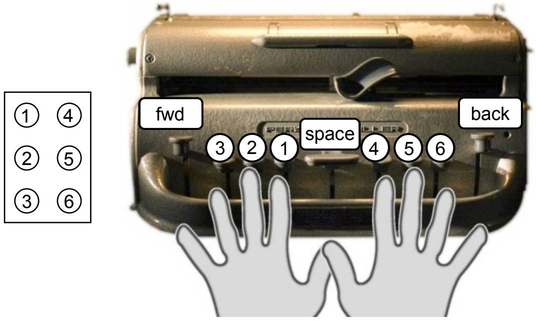

Figure 3. Mapping braille on to the Perkins Brailler.

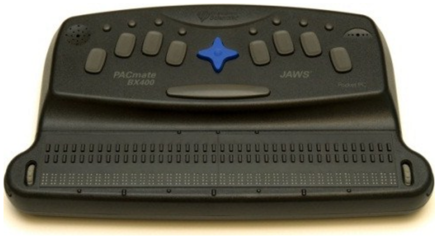

Figure 4. Freedom Scientific PACmate BX400.

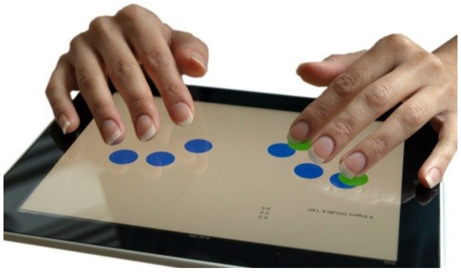

Figure 5. Perkins-style keyboard on a tablet.

selects the character. The speed of split-tap text entry was 2.11 to 3.99 wpm with 5.2% to 6.4% error.

Bonner et al. presented *No-Look-Notes*, a multi-touch eyes-free text entry system that divides the touchscreen into eight pie segments for a first-level selection of character sets and provides a second-level menu for individual character selection [2]. No-Look-Notes' speed was 1.33 wpm with 11% error.

Guerreiro et al. measured three eyes-free text-entry techniques for the standard physical 3×4 numeric keypad on a mobile phone: *Multi-tap*, *NavTap*, and *BrailleTap* [11]. Multi-tap assigns groups of characters to keys 2 through 9. For instance, users press key 2 once for A, twice for B, and three times for C. Guerreiro et al. measured multi-tap's speed at 0.88 wpm with 15.28% error. NavTap provides a navigation interface for a 5×6 virtual table of the alphabet where vowels occupy the first column. Its speed was 1.37 wpm with 9.87% error. BrailleTap maps the 3×2 braille cell to the six keys on the upper right corner of the 4×3 keypad. Users input a character one dot a time, and signal the end of the dot sequence by pressing key 4. BrailleTap's speed was 3.6 wpm with 6.55% error.

Lee et al. presented a braille text entry glove [15]. The typing speed for their experimental braille glove was 24.3 wpm with 5.2% error. Similarly Miele et al. developed but have not evaluated an accelerometer glove that allows the user to chord braille on any flat surface, such as a desk [19].

BrailleTouch Design

The design goal for BrailleTouch was to provide affordable and accessible text entry for the visually impaired on commodity smartphones [7, 22]. Our initial challenge was to accommodate six fingers simultaneously on the limited screen area of a multi-touch smartphone. To solve this challenge, we designed BrailleTouch to be held with two hands, with the screen facing *away* from the user (see Figure 1), similar to the Lucid Touch see-through interaction [27]. The rationale was that blind users need not see the screen. The index, middle, and ring fingers of each hand wrap around the edges of the phone and land on a 3×2 keyboard with the same finger-to-key convention used on standard braille keyboards. We implemented the `space`, `backspace`, and `enter` control keys through gestures: right-flick, left-flick, and two-finger right-flick, respectively.

STUDY DESIGN

We investigated the typing performance of blind users on three experimental braille keyboards: 1) BrailleTouch on a touchscreen smartphone; 2) a soft braille keyboard on a touchscreen tablet; and 3) the PACmate, a commercial braille keyboard with physical buttons. We describe the participants, apparatus, methodology, and procedure.

Participants

We recruited eleven visually impaired participants (4 male) through a newsletter circulated by the Center for the Visually Impaired in Atlanta, GA, USA, and through the Georgia Radio Reading Service (GaRRS) statewide radio broadcast. Additional volunteers contacted us through referrals from other participants. The average age was 47.4 (SD=14.1, range: 28 – 65). Nine participants were blind from birth. All participants were self-described as "proficient with the six-key braille typing system." We did not require or assume prior experience with touchscreen devices; only three owned an iPhone (see Table 2).

P	Age	Sex	CVL	Handed	Br	Job	Ed	iPhone
A1	57	M	0	R	5	N	16	
A2	57	F	0	R	7	N	16	
A3	65	F	0	L	7	N	18	
A4	30	F	0	R	4	N	16	
A5	58	F	0	R	7	Y	13	Y
A6	49	F	0	L	6	Y	19	Y
B1	46	M	32	R	29	Y	17	Y
B2	38	M	0	R	5	N	12	
B3	29	F	27	R	11	Y	12	
C1	28	F	0	L	5	Y	12	
C2	64	M	35	A	7	N	12	

Table 2: Participants. CVL = current vision level since age; Br = age learned braille; Ed = education level (grades).

Apparatus

Our study apparatus consisted of the three experimental braille input devices networked with a PC which ran the typing test software, presented uniform audio feedback for all devices, and recorded performance logs.

For the control condition, we used a Freedom Scientific PACmate BX400 braille notetaker (see Figure 4). For the tablet experimental condition, we implemented a soft braille keyboard on an Apple iPad, based on the PACmate keyboard layout (see Figure 5). The PACmate is a commercial braille notetaker with six physical keys in the standard Perkins arrangement, plus a space bar, `backspace` on the left, and `enter` on the right. The PACmate's keyboard design is comparable to that of other braille notetakers (see Table 1). On the tablet, the user initially defines a comfortable keyboard location on the screen through an eight-finger press-and-hold gesture, an then types. The smartphone experimental condition was BrailleTouch on an Apple iPod Touch (see Figure 1).

Each of the three braille input devices wirelessly transmitted typed characters to a central PC, which provided uniform audio feedback for all three typing conditions and ran the typing test software. Therefore, the input interaction on each of the three braille keyboards was the only independent variable that differed between the experimental conditions during the typing task.

Methodology

We modified the standard text entry study methodology to accommodate visually impaired participants.

From Transcription to Dictation

In traditional text entry studies, sighted participants read short phrases from a screen and transcribe them on the experimental keyboard. Because visually impaired participants cannot use visual phrase stimuli, we presented phrases to the participants through audio for a dictation task. We employed a speech synthesizer, the American English voice in Google's online language translation utility, to ensure uniform delivery [9]. Dictation, in the form of taking notes on a braille keyboard, is a common real-world task for blind users, hence the name "braille notetaker" for the class of devices including the PACmate.

Phrase Set

MacKenzie and Soukoreff developed a standard set of 500 English phrases commonly used in text entry studies [17]. These phrases approximate the letter frequency of the English language. As is typical in text entry studies, we asked our participants to only input lower-case letters and spaces, with no numbers or punctuation.

A dictation-based text entry study introduces three auditory confounds not present in visual transcription tasks: 1) phrases can be aurally ambiguous; 2) users may not know the spelling of a word; and 3) users may not comprehend the delivered speech. We mitigated aural ambiguity by removing problematic phrases from the standard MacKenzie-Soukoreff phrase set. Criteria for removing phrases included homophones, compound words, and mondegreens (phrases aurally similar to others). We removed 65 of the 500 phrases from the standard phrase set. MacKenzie and Soukoreff provide software for analyzing the correlation of a phrase set to English [17]. The correlation for their original phrase set is .954, and for our modified set is .952.

Typing Test Software

We modified the Twidor typing test software, freely available at [8], by adding speech synthesis to present the phrases and audio feedback for the keyboard input. For each character typed, Twidor played a typewriter click sound effect, similar to the mechanical Perkins brailler. When the user backspaced, the system played a *swoosh* sound effect, in addition to the speech of the letter erased. Finally, we played a distinct sound effect when the user pressed `enter` to submit a phrase.

Unlike with sighted users who can see presented phrases on the screen, the visually impaired might not know how to spell a word, might not remember the entire phrase while typing, or may not comprehend the delivered speech. In an effort to address these challenges we implemented two hints in our typing test software to mitigate these factors. The first hint (chord 4-5-6) spelled out the entire phrase, letter by letter, before the timer started. The second hint (chord 1-4-5-6) repeated the entire phrase.

Experimental Procedure

We structured our study as a 3×5 within-subjects design with two independent variables: *Device* (PACmate, tablet, smartphone) and *Session* (1 to 5). During each session, participants performed a 20-minute typing test on each of the three devices, separated by a short break. Sessions lasted approximately 90 minutes. Each session was separated by at least 90 minutes and no more than three days. We counterbalanced the three conditions across participants and sessions to control for ordering effects.

During each typing test, participants typed as many blocks of ten phrases from dictation as they could start within a 20-minute period (see Video Figure 1). We instructed them to type the phrases "as quickly and accurately as possible." After each block of ten phrases, we provided verbal feedback on their most recent speed (wpm) and accuracy, and encouraged participants to briefly rest their hands.

At the beginning of the study, we trained participants and gave them the opportunity to practice on the three devices. To avoid contaminating participants with actual test phrases, we presented them with phrases from an alternative corpus, the Children's Phrase Set [13]. Before each typing test, we allowed participants time to briefly warm up with two practice phrases: (1) the alphabet; and (2) "this is a practice sentence".

We compensated participants at the rate of \$0.125 × wpm × accuracy for each 20-minute typing test, with a floor of \$2.00 per test. In addition, we offered a \$5 bonus for completing the entire study. This is a similar scheme to other text entry evaluations [5]. Therefore, all participants whose results are presented here earned at least \$35 (5 sessions × 3 conditions × \$2 + \$5 bonus). Our rationale for rewarding performance was to motivate participants to

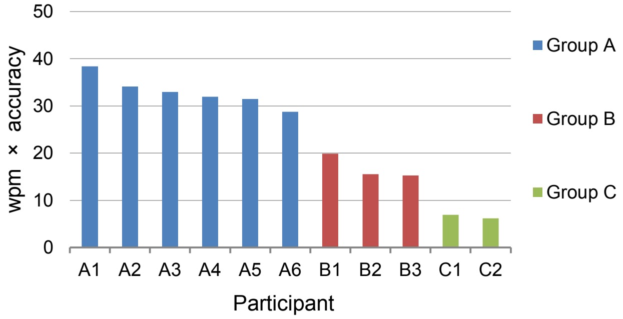

Figure 6: Participant groups clustered on typing Speed×Accuracy on the first session on the PACmate.

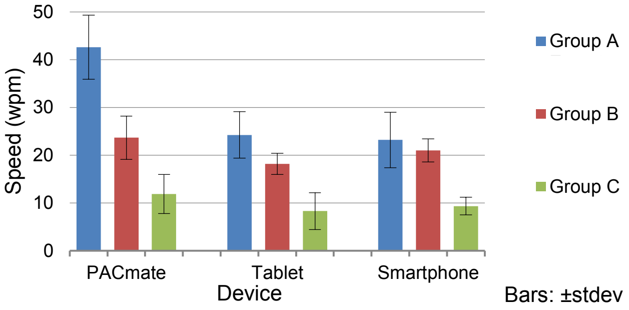

Figure 7: Speed per group per device on last session.

continue to type as quickly and accurately as possible throughout the study.

RESULTS

The eleven participants typed a total of 9092 phrases across 165 twenty-minute typing tests on three braille keyboards. We shall discuss: 1) braille typing performance on a standard keyboard; 2) effects of device and session; 3) performance-based groups of participants; 4) typing speed; and 5) typing accuracy.

Standard Braille Typing Performance

We present the first formal evaluation of English text entry on a standard physical braille keyboard, the PACmate. In the final session, participants averaged 32.9 wpm (SD=14.1), which is consistent with Lee's study in Korea which reported typing speeds of 27.9 wpm on a Perkins-style keyboard [15]. The Total Error Rate (described below) for the PACmate on the final session was 9.41% (SD=6.12%).

Device and Session

There was no significant effect on the touchscreen devices between *Session* 3 and either 4 or 5, indicating that people who know the braille keyboard can transfer their typing skills to a touchscreen within an hour of practice. Surprisingly, we found no significant effect on speed between the tablet and the smartphone, despite their different interaction techniques.

We analyzed the data using a mixed model analysis of variance with fixed effects for *Device* and *Session* and a random effect for *Participant*. We also included *Order* in the model to test for ordering effects. Our counterbalanced order of testing layouts achieved the desired outcome, as the main effect and interactions for *Order* were not statistically significant.

We found a main effect of *Device* on speed ($F_{2,20}$ = 19.968, $p < .001$) as well as a main effect of *Session* on speed ($F_{4,40}$ = 22.369, $p < .001$). To analyze these effects further, we conducted post-hoc pairwise comparisons using Bonferroni correction. The post-hoc test for *Device* shows a significant effect on speed between the PACmate and the tablet (t_{10} = 4.739, $p < .001$) and a significant effect on speed between the PACmate and the smartphone (t_{10} = 4.767, $p < .001$).

The post-hoc test for *Session* shows a significant effect on speed between Sessions 1 and 3 (t_{10} = -5.709, $p < .001$), Sessions 1 and 4 (t_{10} = -4.834, $p < .001$), Sessions 1 and 5 (t_{10} = -6.326, $p < .001$), Sessions 2 and 3 (t_{10} = -4.237, $p < .002$), Sessions 2 and 4 (t_{10} = -5.772, $p < .001$), and Sessions 2 and 5 (t_{10} = -6.780, $p < .001$). No other statistically significant effects on speed were found for *Session*, indicating that from session 3 on, significant learning did not occur. However, due to a significant effect between the first and last session, we use Session 5 for the speed and error analyses below.

Participant Groups

Similar to [20], we observed a wide variance between participants' individual braille typing abilities on the PACmate control device. These skill differences appeared to be related to participants' success in adapting to the touchscreen braille keyboards. As such, using k-means clustering on their performance, we classified participants into three groups (see Figure 6). We determined three groups to be the best clustering set based on the minimum inconsistent clustering coefficients for a hierarchical cluster tree. The classification was based on speed × accuracy (the same metric used to compute compensation) from the first typing session on the PACmate control device. Group A consists of six expert participants (A1-A6); Group B is three participants with moderate performance (B1-B3); and Group C is two participants with poor performance (C1-C2).

Speed

On the PACmate in the fifth session (see Figure 7), Group A averaged 42.6 wpm (SD = 6.7); Group B averaged 23.7 wpm (SD = 4.5); and Group C averaged 11.9 wpm (SD = 4.1). Participants in Groups A and B both typed quickly on the two touchscreen devices. Group A averaged 24.2 and 23.2 wpm on the tablet and smartphone, respectively. Group B averaged 18.2 and 21.0 wpm on the tablet and smartphone, respectively. Group C did not achieve comparable speeds, averaging 8.3 and 9.4 wpm on the

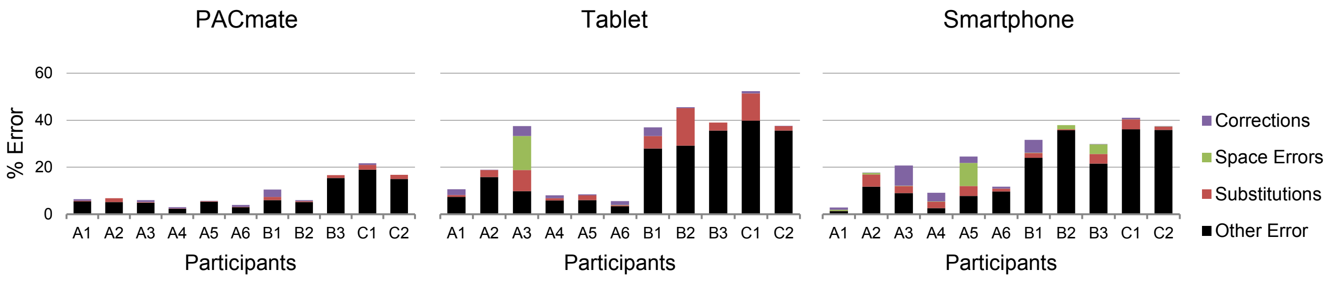

Figure 8: Four error types per participant per device: corrections, space errors, substitutions, other errors.

tablet and smartphone, though this is not much slower than their 11.9 wpm on the PACmate control device.

Accuracy

We first calculated accuracy using MacKenzie and Soukoreff's Total Error Rate, which reflects insertion, deletion, substitution errors, as well as the keystrokes needed to make corrections (i.e. backspaces) [24]. Total error combines the number of errors made and corrected (IF), the number of errors made but not corrected (INF), the number of keystrokes invested in error correction (F), and the number of correct keystrokes (C), such that,

$$Total\ Error\ Rate = \frac{INF+IF}{C+INF+IF} \times 100\%.$$

We report the total error rates for the final session. The expert braille typists in Group A averaged 14.8% and 14.5% error on the tablet and smartphone, respectively. These figures are roughly three times the 5.3% total error on PACmate Session 5. The five participants in Groups B and C completed the study continuing to exhibit error rates on the touchscreen devices prohibitive for real-world usage. Group B averaged 40.5% and 33.1% total error on the tablet and touchscreen, respectively, as compared with 11.0% error on the PACmate. Group C averaged 45.0% and 39.3% total error on the tablet and touchscreen, respectively, compared to19.3% error on the PACmate.

Deconstructing the Error

To better understand errors, we examined the logs of the typing tests and identified three common patterns of errors: *Corrections*; *Substitutions*; and *Space Errors*. Figures 8 and 9 show the contributions of these three classes of error, along with the remaining *Other Error*. The full height of the stacked bars represents the Total Error.

Minimum String Distance (MSD)

Correction errors occur when the user backspaces over a character and then types a replacement character. These are captured in the Total Error Rate. We observed that, for some participants, the final corrected input strings they submitted were much more accurate than what was reflected in their Total Error scores. To quantify this observation, we employed another error metric, the Minimum String Distance (MSD) [24].

MSD captures three types of errors that represent the difference between the presented string and the transcribed string: insertions, deletions, and substitutions. The Levenshtein distance, or MSD, between two character strings is the minimum number of insertion, deletion, and substitution operations necessary to transform one string into another. Oliveira et al. report MSD error in their study of blind text entry [20]. By subtracting the MSD Error from the Total Error for each of the 165 typing sessions, we were able to measure the portion of the Total Error that was due to participants backspacing and correcting their text input.

In order to identify specific instances of substitutions, or input characters being substituted for presented characters, we implemented software to walk through the pairs of strings in the typing logs and line up the pairs of substitution characters, by accounting for insertion and deletion errors.

Below, the bottom line is a string transcribed by participant A5. The underscore represents a deletion error, or omitted character. The top line is the presented string. The underscore represents an input insertion error. By aligning the two strings, accounting for insertions and deletions, we identify character pairs in substitution errors, indicated by the carets on the third row.

```
just what the doctor_ ordered
dofg _f`t the doctor` o"de"ed
^^^^  ^^               ^  ^
```

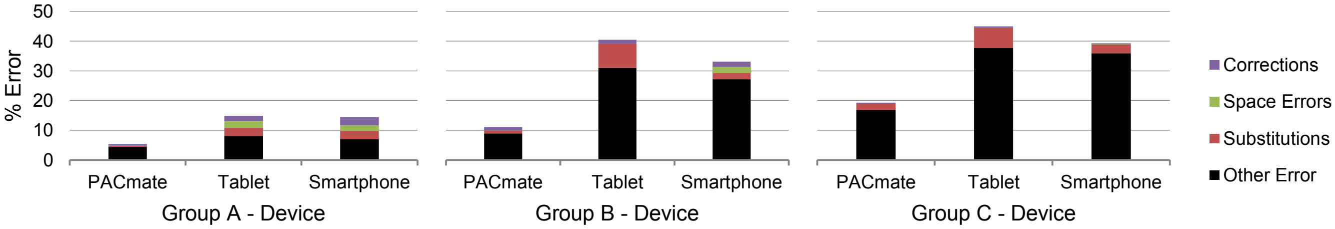

Figure 9: Four average error types per participant group per device.

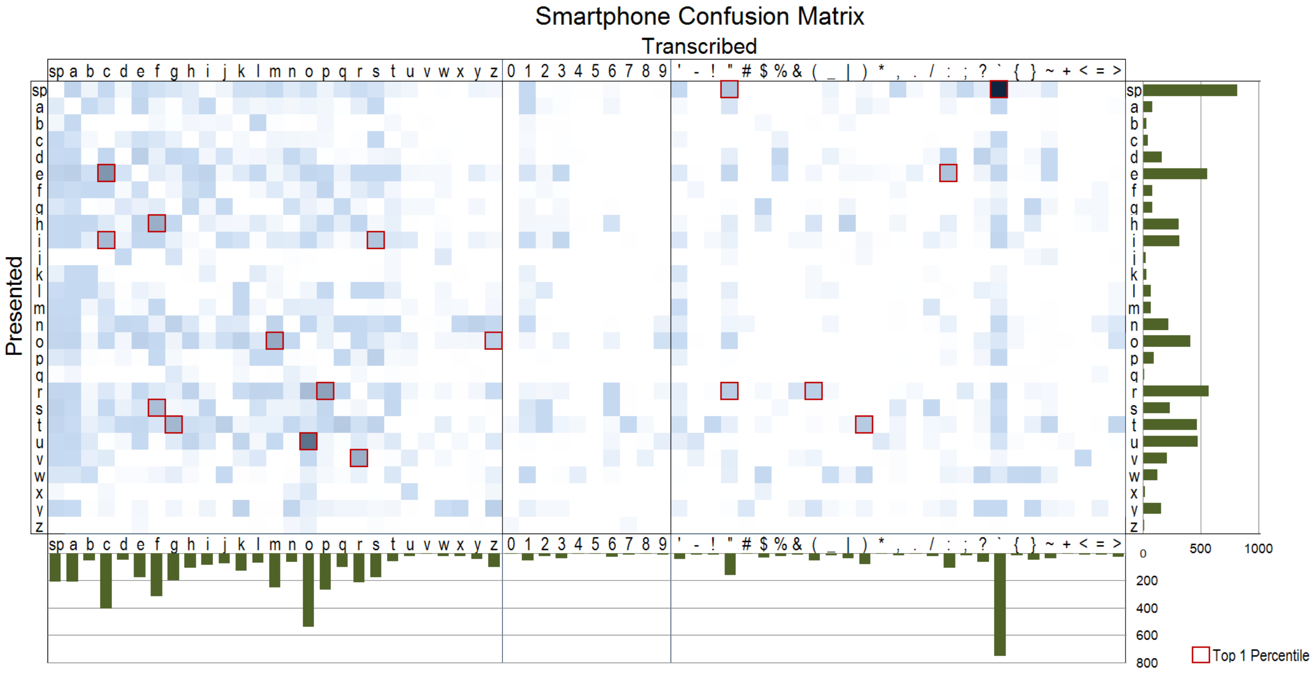

Figure 10: Confusion matrix for substitution errors for Brailletouch on smartphone.

The MSD algorithm calculates an $m \times n$ matrix, where m and n are the lengths of the two strings being compared.The operations to transpose one string into the other can be determined by walking backward through the MSD matrix from the bottom-right to the top left.

One of the most frequent substitutions we observed was space ← grave accent (`). We broke out *Space Errors* as a special case of substitution errors. On the smartphone, the space character is typed with a flick gesture to the right. Most participants performed this gesture with their right index fingers. On the braille keyboard, the right index finger falls over dot-4, which represents the grave accent character. Thus, our software failed to accurately recognize the flick gesture for many participants, instead interpreting the touch event as a tap on dot-4. We also observed this same substitution error pair on the tablet soft braille keyboard. While some participants hit the soft space bar with their thumb, as our design anticipated, others moved their right index finger down from dot-4 to tap the spacebar. Thus the same substitution error occurred frequently on both the smartphone and the tablet, although with different causes.

Confusion Matrices

In order to quantify the effect on the error rate from substitution errors, we identified each substitution character pair from the presented and transcribed strings and populated a confusion matrix with the total number of errors for that character pair (see Figure 10). The rows in the matrix represent the 27 presented characters (*a* to *z* plus *space*). The columns represent the 64 possible characters in the transcribed input string typed by the user (63 binary combinations of the six braille dots, plus *space*).

We identified the top 1 percentile of substitution errors for each of the three devices from the confusion matrices (see Table 3). This first percentile accounts for 47% of the total error. Most of these substitution errors could be explained by a 1-dot difference in the corresponding braille chords. For example, in the substitution pair v ← r, our software interpreted a dot-5 where the adjacent dot-6 should be. We shall examine this in detail in the Discussion section.

DISCUSSION

We first compare the speed and accuracy of our BrailleTouch system in the context of comparable technologies. We then discuss the design implications of this study for our system, especially in regard to automatic error correction to improve accuracy. Finally, we discuss lessons learned for conducting text entry research with the visually impaired.

Speed and Accuracy

Our touchscreen braille keyboards offer a significant speed advantage over other touchscreen braille keyboards for the visually impaired, all of which report participants' typing speeds of less than 7 wpm. Users with expert and moderate braille typing ability were able to type at approximately 18-24 wpm on both the tablet and smartphone.

One surprising result is that there is no statistically significant effect on speed or accuracy between the two touchscreen devices. We expected to see an effect between each of the three devices. The PACmate and tablet differ in that one has physical buttons, while the other has virtual buttons on flat glass. The tablet and the smartphone differ ergonomically and in the keyboard orientation. Our analysis shows that moving from physical buttons to a touchscreen matters, but rotating one's hand position (tablet vs. smartphone) does not.

Our expert participants' average touchscreen error rate of ~15% was in line with results from other assistive technology text entry evaluations, such as MultiTap (15%) and BrailleType (8%) for the visually impaired [20] and Edgewrite for people with motor disabilities (29.56%) [28]. Concerned with the prohibitive error rates we observed from Groups B and C, we further examined what types of errors the participants were committing, and how we could change our touchscreen designs to improve accuracy.

	Presented Character	Transcribed character	Presented braille	Transcribed braille	Dot-level analysis	Gesture-level interpretation
1.	sp	`	swipe	⠈	4-insertion	Missed swipe
2.	u	o	⠥	⠕	5-del + 6-ins	R. F2 drift down
3.	e	c	⠑	⠉	5-de l+ 4-ins	R. F2 drift up
4.	r	p	⠗	⠏	5-del + 4-ins	R. F2 drift up
5.	o	m	⠕	⠍	5-del + 4-ins	R. F2 drift up
6.	v	r	⠧	⠗	6-del + 5-ins	R. F2 drift up
7.	h	f	⠓	⠋	5-del + 4-ins	R. F2 drift up
8.	t	g	⠞	⠛	3-del + 1-ins	L. F1 F2 drift up
9.	i	c	⠊	⠉	2-del + 1-ins	L. F2 drift up
10.	s	f	⠎	⠋	3-del + 1-ins	L. F1 F2 drift up
11.	e	:	⠑	⠱	6-ins	
12.	r	o	⠗	⠕	2-del	
13.	sp	“	swipe	⠐	5-ins	Missed swipe
14.	i	s	⠊	⠎	3-ins	
15.	t	)	⠞	⠾	6-ins	
16.	o	z	⠕	⠵	6-ins	
17.	r	“	⠗	⠐	(1+2+3)-del	
18.	r	(	⠗	⠾	1-d + (4+6)i	

Table 3: Top 1% substitution errors on BrailleTouch. Ins=insertions; Del=deletions; R=right. F# = finger number.

Automatic Error Correction

In our analysis, we classified four components of the Total Error rate*: Corrections*; *Substitutions*; *Space Errors*; and *Other Errors*. The corrections represent user edits by backspacing, and thus do not need automatic solutions; rather they show an attention to accuracy by the participants. We shall discuss strategies for automatic correction of the *Space* and *Substitution Errors*, in order to improve accuracy in future design iterations.

Space Errors

The misinterpretation of `space` gestures can be addressed by adjusting constants in our software. On the smartphone, the space character is a flick gesture to the right. It was frequently misinterpreted as a dot-4, or a tap by the right index finger. Our gesture recognizer was based on two constants: the minimum distance and the maximum time between the touch down and touch up events on the screen. Through empirical analysis of raw touch position and timestamp data from blind users, we can determine better values for these two constants, which we believe will greatly reduce these space substitution errors. On the tablet, the spacebar was defined as any touch greater than a constant distance below the main soft braille buttons. Similarly, we can adjust this constant through empirical analysis of the actual touch positions of users.

Substitutions

We identified the most common substitution errors for each device (see Table 3). These data can inform two types of real-time automatic error correction in future design iterations. First, we can compare the typed characters to real words in a dictionary. If the input string does not match a real word, we can iterate through permutations of the input string that reflect the most frequent substitution errors.

Second, most of the common substitution errors reflect a single dot error within the braille cell. In other words, the user either deleted a dot (left out finger), inserted an extra dot (by pressing an extra finger), or substituted a finger (e.g. pressed dot 3 instead of adjacent dot 2). Insertions and deletions of dots can inform the dictionary approach described above. The finger substitutions suggest we should improve our spatial algorithms that assign an (x, y) position on the screen to one of the six braille soft buttons. This is effectively a classification problem. Future analysis of raw touch positions may provide simple heuristics that will improve soft braille key recognition, and therefore accuracy.

We observed that blind users' hands tend to drift across the touchscreen over time, given the lack of tactile landmarks. The constrained size of the smartphone mitigated the drift issue, while drift was more pronounced on the tablet due to its larger screen area. We believe that this drift across the screen is related to some of the substitution errors that involve adjacent braille dots. In future work, we will investigate the use of a Kalman filter to correct for users' hand drift over time.

Methodology

We designed our study to produce quantitative results that could be compared with other text entry research in the literature. However, we had to modify several practices that have become standards in the text entry research community because our subjects were visually impaired.

Audio Feedback

In our study, we provided participants with only keyclick audio feedback for each character entered. We chose this over having the system speak each character entered because the Google speech synthesizer we employed could not keep up with the high text entry speeds we observed in our pilot studies. However, many users complained of the lack of voice feedback, as they had difficulty determining if they had entered the correct character. Screen reader accessibility software can speak individual characters fast enough to match high text entry speeds, and in a manner that is familiar to many visually impaired users. In future studies we will employ character- and word-based auditory feedback.

Dictation Phrase Set

Although we removed 65 aurally ambiguous phrases from the standard Mackenzie-Soukoreff text entry phrase set, some remaining phrases were still problematic for our participants. The best solution is to create a new phrase set optimized for dictation typing tests.

Braille and the Blind Community

The significance of braille literacy in the blind community is a hotly discussed topic. Indeed fewer people may be reading tactile braille than were a generation ago, due in part to text-to-speech technology. However braille writing, which is the application for BrailleTouch, is a different matter than braille reading. Since we published the initial design of BrailleTouch, the authors have been personally contacted by hundreds of enthusiastic people from all over the world who want this technology. One study participant was so excited about the potential of BrailleTouch that she donated her $4000 PACmate to the research team. Furthermore, the six-key Perkins-style braille keyboard is much easier to learn than tactile braille reading. With no prior knowledge of braille, the authors were able to learn the alphabet within several hours and soon after achieve 12-16 wpm typing speeds.

Most blind people who know braille are accustomed to both reading and writing in Grade 2, or contracted, braille, a shorthand which requires fewer characters than the printed form of the same text. Similar to other braille text entry studies [1, 18, 21], we employed Grade 1 braille, which uses one braille chord for each character. While we initially thought that contracted braille was just convenient shorthand, our participants informed us that contracted braille has the significance of a language within the blind community. As a result, many participants reported translating dictated phrases into Grade 1 braille (which they never use) in their heads before typing the characters. One described this as “like hearing English and having to type French.” It is essential that future braille text entry research employs contracted braille in order to be relevant to the blind community.

CONCLUSIONS AND FUTURE WORK

We presented the evaluation of BrailleTouch, an eyes-free text entry system for mobile touchscreen smartphones, with eleven blind participants. The six expert braille typists averaged speeds of 23.2 wpm with 14.5% error on a smartphone. In response to the high Total Error rates we observed, especially in Groups B and C, we classified three common types of errors and presented strategies to improve accuracy in future design iterations. BrailleTouch has the potential to revolutionize mobile text entry for the visually impaired by replacing assistive technology that costs $1000s with an application on a commodity smartphone. Sighted users may also benefit from the design lessons learned for eyes-free typing on a touchscreen. In future work, we will evaluate BrailleTouch in real-world scenarios by deploying it on the smartphones of blind users.

ACKNOWLEDGMENTS

We thank the Atlanta Center for the Visually Impaired, the Georgia Radio Reading Service, the Atlanta Vision Loss Center, and especially the study volunteers. We also gratefully acknowledge support from a Nokia University Research Award.

REFERENCES

1. Azenkot, S., Wobbrock, J. O., Prasain, S., and Ladner, R. E., Input Finger Detection for Nonvisual Touch Screen Text Entry in Perkinput, in *Proc. Graphics Interface*. Canadian Information Processing Society, Toronto, (2012).
2. Bonner, M., Brudvik, J., Abowd, G., and Edwards, K., No-Look Notes: Accessible Eyes-Free Multi-Touch Text Entry, in *Proc. Eighth International Conference on Pervasive Computing* Springer: Heidelberg, (2010), 409-426.
3. Braille, L., *Procedure for Writing Words, Music and Plain-song Using Dots for the Use of the Blind and Made Available to Them*, Institut National des Jeunes Aveugles (Royal Institution Of Blind Youth): Paris, (1829).
4. Bruder, R. *Type Brailler Learn Braille Lite*. (2012) May 7, 2012 [cited 2012 June 14]; Available from: http://itunes.apple.com/app/type-brailler-learn-braille/id509166743?mt=8.
5. Clawson, J., Lyons, K., Starner, T., and Clarkson, E., The Impacts of Limited Visual Feedback on Mobile Text Entry for the Twiddler and Mini-QWERTY Keyboards, in *Proc. Ninth IEEE International Symposium on Wearable Computers*, IEEE Computer Society, (2005), 170-177.
6. Fard, H. G. and Chuangjun, B., Braille-based Text Input for Multi-touch Screen Mobile Phones, in *School of Computing*, Blekinge Institute of Technology, (2011), 68.

7. Frey, B., Southern, C. and Romero, M. BrailleTouch: Mobile Texting for the Visually Impaired., in *Proc. HCII*. Springer Berlin / Heidelberg, (2011).

8. Fusia, J. *Twidor - The Twiddler Tutor*. (2005) [September 20, 2011]; Available from: http://wearables.cc.gatech.edu/projects/twidor/ download.html.

9. Google. *Google Translate*. [September 19, 2011]; Available from: http://translate.google.com/translate_tts?tl=en&q= google+voice.

10. Gordon, G., The Past, Present, and Future of Portable Note-takers, in *Technology And Persons With Disabilities Conference*, Center On Disabilities - California State University Northridge: Los Angeles, (2002).

11. Guerreiro, T., Lagoá, P., Santana, P., Gonçalves, D., and Jorge, J., NavTap and BrailleTap: Non-Visual Texting Interfaces, in *Proc. Rehabilitation Engineering and Assistive Technology Society of North America Conference (Resna)*: Washington, D.C., (2008).

12. Kane, S. K., Bigham, J. P., and Wobbrock, J. O., Slide rule: Making Mobile Touch Screens Accessible to Blind People Using Multi-touch Interaction Techniques, in *Proc. 10th International ACM SIGACCESS Conference on Computers and Accessibility*, ACM: Halifax, Nova Scotia, Canada, (2008), 73-80.

13. Kano, A., Read, J. C., and Dix, A., Children's Phrase Set for Text Input Method Evaluations, in *Proc. 4th Nordic Conference on Human-Computer Interaction: Changing Roles*, ACM: Oslo, Norway, (2006).

14. Keles, M., INEXPENSIVE BRAILLER: Writing Braille Using IMPAD, in *Department of Computer Science*, New York University: New York, (2009), 40.

15. Lee, S., Hong, S. H., Jeon, J. W., Choi, H. G., and Choi, H., Design of Chording Gloves as a Text Input Device, in *Computer Human Interaction - Lecture Notes in Computer Science*, 3101/2004 (2004), 201-210.

16. Lorimer, P., Origins of Braille, in *Braille: Into the Next Millenium*, J. M. Dixon, Editor. National Library Service for the Blind and Physically Handicapped of the Library of Congress: Washington D.C.,18-39, 2000.

17. MacKenzie, I. S. and Soukoreff, R. W., Phrase Sets for Evaluating Text Entry Techniques, in *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, ACM: Ft. Lauderdale, FL, (2003), 754-755.

18. Mascetti, S., Bernareggi, C., and Belotti, M., TypeInBraille: Quick Typing on Smart Phones by Blind Users, in *Universitá Degli Studi Di Milano*: Milan, Italy, (2011).

19. Miele, J. A. and Edwards, O. R., WearaBraille: Development of a Wireless, Virtual, Braille Keyboard, The Smith-Kettlewell Eye Research Institute: San Francisco, CA, (2011).

20. 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: Dundee, Scotland, UK, (2011), 179-186.

21. Oliveira, J., Guerreiro, T., Nicolau, H., Jorge, J., and Gonçalves, D., BrailleType: Unleashing Braille over Touch Screen Mobile Phones, in *Human-Computer Interaction – INTERACT 2011*, P. Campos, et al., Editors, Springer Berlin / Heidelberg: Lisbon, Portugal, (2011), 100-107.

22. Romero, M., Frey, B., Southern, C., and Abowd, G., BrailleTouch: Designing a Mobile Eyes-Free Soft Keyboard, in *MobileHCI 2011*: Stockholm, Sweden, (2011), 707-709.

23. Smith, A., Americans and Text Messaging, in *Pew Internet & American Life Project*, Pew Research Center: Washington, D.C., (2011).

24. Soukoreff, R. W. and MacKenzie, I. S., Metrics for Text Entry Research: An Evaluation of MSD and KSPC, and a New Unified Error Metric, in *Proc. SIGCHI Conference on Human Factors in Computing Systems*, ACM: Ft. Lauderdale, FL, (2003), 113-120.

25. Tobe, C. B., Callahan, E., and Callahan, M., Embossed Printing in the United States, in *Braille: Into the Next Millennium*, J. M. Dixon, Editor. National Library Service for the Blind and Physically Handicapped of the Library of Congress: Washington, D.C., (2000), 41-71.

26. W.H.O. *World Health Organization Fact Sheet N°282 on Visual impairment and blindness*. (2011); Available from: http://www.who.int/mediacentre/factsheets/fs282/en/.

27. Wigdor, D., Forlines, C., Baudisch, P., Barnwell, J., and Shen, C., Lucid Touch: A See-through Mobile Device, in *Proc. 20th Annual ACM Symposium on User Interface Software and Technology*, ACM: Newport, Rhode Island, USA, (2007).

28. Wobbrock, J. O., Myers, B. A., Aung, H. H., and LoPresti, E. F., Text Entry from Power Wheelchairs: Edgewrite for Joysticks and Touchpads, in *Proc. 6th international ACM SIGACCESS Conference on Computers and Accessibility*, ACM: Atlanta, GA, USA, (2004), 110-117.