Flat is the New Pitch-Black: Discussing Blind use of Touchscreens

Mario Romero
KTH
Lindstedtsvägen 5
114 28 Stockholm
marior@kth.se

Abstract

The increasingly ubiquitous touchscreen, from the smart phone to the treadmill, is a significant hurdle for blind individuals who cannot rely on their sense of touch for decoding its interface. Advances in smart phone screen readers, such as the iPhone's Voice Over, have enabled blind users to effectively navigate touchscreens. While Voice Over efficiently outputs information, text input remained a challenge. To address this, we previously introduced BrailleTouch, a soft braille keyboard for efficient blind text entry on touchscreens. In this position paper, we present the tailored touch-based user experience design and evaluation techniques we developed for BrailleTouch which we have not previously discussed.

CHI 2014, April 26 to May 1, 2014, Toronto, Canada.
ACM xxx-x-xxxx-xxxx-x/xx/xx.

Author Keywords

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

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 touching a mysterious portable device with eight buttons and hundreds of pins. She furiously chorded the keys of what appeared to be a miniature piano with only eight black keys. Intermittently, she ran her fingers over the tiny pins that eagerly popped up and down at lightning speed. She smiled. Then, slowly, as if in recognition, she started giggling and soon broke into a *silent* LOL. She was lightning quick yet dead silent. Her trouble was she was in the middle of a class presentation and did not want to alert her teacher, another blind woman, of this extracurricular activity.

Although bulky and awkward, the device was entirely effective at making her giggle at what we soon realized were text messages from a friend. She was like any

Figure 1: Sense Plus provides touch feedback on the shape, position, and pressing of the entry keys, with a perceptible-by-touch click. It provides character-level feedback through a refreshable braille display at the bottom.

Figure 2: BrailleTouch is a soft braille keyboard that provides audio feedback at character or word-level through the iOS Voice Over infrastructure.

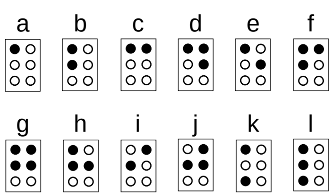

Figure 3: Braille alphabet.

other teenager with a smartphone, an expert at texting, except she was not using her eyes or ears to communicate. Everything happened through touching and pressing. She was typing on the braille keyboard of a Braille Sense Plus and reading on its refreshable braille display. We researched the Sense Plus and were shocked to discover that it cost $6000 (see Figure 1). We recognized a potential cause for a profound digital divide, and a significant design challenge: How can we support mobile texting for the visually impaired on commodity mobile devices?

We designed BrailleTouch (see Figure 2), a soft braille keyboard for smart phones [6]. Next, we reported the results of a laboratory study of BrailleTouch focusing on typing speed and accuracy [7]. In this position paper, we report the touch-centric user experience design and evaluation techniques we developed for our user study. We have not previously discussed our tailored UX touch-centric d esign and evaluation features that we present in the following sections.

Related Work

Kane et al. presented Slide Rule, an interface to control mobile devices with multi-touch input and audio feedback [5]. While Slide Rule employs a soft QWERTY keyboard for text entry, the authors suggested a soft braille chorded keyboard as future work and started the race that follows. Guerreiro et al. presented BrailleType, a braille system that sequentially enters a double tap [3]. Azenkot and Ladner presented Perkinput, a chorded input for touchscreens where the braille cell is typed one column at a time [2]. These papers include evaluations of their systems, measuring speed, accuracy, and user experience.

Discussion on the Design of BrailleTouch

Previously, we reported the high-level operation of BrailleTouch. The keyboard works with the flat touchscreen facing away from the user. It is a column-major numbered 3x2 binary matrix. Characters are input through chording combinations of the six buttons. In figure 2, the user's left and right indices chord the letter ”c”, for example. The white numbered circles are *not* the buttons. They are meant as place markers for sighted and low-vision users of BrailleTouch. In fact, the interface is not button based.

The design of BrailleTouch exploits the structure of braille. Louis Braille designed the alphabet to be read by touch. He did not optimize it for encoding length, but for tactile decoding. For example, "a" is the only single-dot character, "b" is the only two-adjacent-vertical-dot character, "c" is the only two-horizontal-dot character, and so on (see figure 3). Braille realized that if there were other single-dot characters in his encoding, readers by touch would have no way to disambiguate between these encodings and "a".

We exploit this structure. For example, if the touch is on the left side of the screen, and there are three points of contact, it can only be an "l". If there are two points of contact and they are close together, it is a "b", and if they are far apart, it is a "k". This design detail produced significant improvements. Before we designed fixed-sized buttons which generated abundant error (see Figure 4).

The second impactful change we introduced is replacing a central space bar with gestures. We introduced right-swipe for space, left-swipe for back space, double-right-swipe for enter, and double-left-swipe for escape.

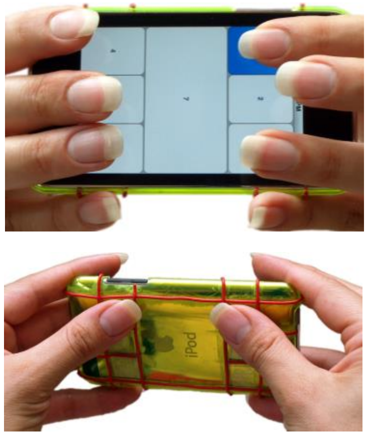

Figure 4: First prototype of BrailleTouch. Fixed buttons and a space bar and rear tactile map. User rejected the tactile map as they don't read with their thumbs.

Figure 5: Experimental setup. We connected the test devices to an automated speech synthesizer delivering the standard phrase set and recorded video and sound of the users' hands.

(Readers can freely download BrailleTouch to an iPhone from the App Store and try its features [1]). These gestures created a significantly more enjoyable and effective user experience. The static space bar constantly generated mistakes and frustration among users.

The third design feature that subtly modifies the user experience is a temporal window for correction. When users mistakenly chord a character like "c" when they meant to type "a", they can correct their mistake without having to delete the "c". They simply lift the right index and wait for a second and the chord sequence is recognized as an "a". This feature significantly facilitates text correction while typing.

The fourth feature we discuss here is the lifting of the fingers. It is difficult for users to enter a chord with multiple fingers all at exactly the same time. It is simpler to record the entire chord sequence after the last finger has lifted from the screen, even if the fingers are not lifted all at the same time (but within a few hundred milliseconds). This has an enormous impact on people's unconscious response to BrailleTouch. Users "don't know why, but it just works for [them]."

Users reported feeling truly blind on touchscreens. From an analytical perspective, we can understand this. Touchscreens lack tangible symbolic landmarks. Thus, flexibility and robustness were key factors for BrailleTouch's blind touch interface, where users could exploit proprioceptive cognition more than symbolic tactile or spatial features. This is a point of discussion for the workshop, where we will explore high-level symbolic cognition playing a secondary role to sub-symbolic stimuli in user experience evaluation [4].

The fifth feature and the most troublesome for users, is the placement of the hands on the screen. Traditional braille keyboards have 1x6 or 1x8 keyboards. People reported having to translate their mental images of how to type from this linear configuration to the 3x2 matrix of BrailleTouch. It surprised us to discover that most people combined finger-memory with spatial patterns for touch typing. We disrupted these patterns. It was surprising to us because we assumed that folding the 1x6 keyboard back into the 3x2 braille cell would actually help users. We discovered that typing and reading are very different cognitive skills, from our perspective, much more so than for sighted individuals. Furthermore, some people reported feeling that the keys were upside down, that is the 1 and 4 should be at the bottom for them. After deliberation, we determined that the users where experiencing the mental image of flipping a page on its vertical axis, so that the top-front ends at the bottom-back. We and most users understood the flip to be a horizontal flip, where the top remains the same. Nevertheless, the final challenge for users was the fact that we maintained finger to key mapping, thus, we flipped the keyboard left and right. To explain to users how to understand the change, we focused on finger-to-key mapping. Yet, many users reported having to get used to the changes. The commercial version of BrailleTouch allows the users to flip the keyboard layouts.

Finally, an interesting discussion on a failed part of the original design is the tactile features we placed on the back of the screen (see figure 4, bottom). We expected participants to read the buttons from the back of the screen with their thumbs. Yet, we learned that people don't have enough sensitivity and training to read with their thumbs. Users rejected the marked case.

Figure 6: Three conditions: physical braille keyboard, tablet with virtual braille keyboard, smartphone braille keyboard. Notice the masking tape and the rubber case for the phone.

Presented Character	Transcribed character	Presented braille	Transcribed braille
u	o	⠥	⠕
e	c	⠑	⠉
r	p	⠗	⠏
o	m	⠕	⠍

Table 1: Top BrailleTouch typing errors. Notice right hand fingers are out of place by one shift up.

Discussion on the Evaluation of BrailleTouch

We conducted a quantitative study of speed and accuracy based on detailed touch data logs and we analyzed the results in more detail than we have previously had the space to report (see Figures 5 & 6). We take this opportunity to report on the relevant features of the evaluation that relate to sub-symbolic touch as the central mode of simultaneous interaction and evaluation.

Table 1 presents the symbolic (characters) and sub-symbolic actions (finger placements) resulting in the top errors committed while typing with BrailleTouch. The columns "presented character" and "transcribed character" are symbolic-level actions. Many users mistakenly type an "o" when they mean to enter a "u", a "c" for an "e", and so on. While there is no discernable pattern for these errors at a symbolic layer analysis, a simple sub-symbolic glance at the columns "presented" and "transcribed braille" reveals very clearly that the common mistake is a finger on the right hand misplaced one cell up (⠕ for ⠥, ⠉ for ⠑, etc.).

At the sub-symbolic layer, the fingers are in control. The source of the problem is fine motor control, proprioceptive memory and the lack of tactile landmarks demarking the boundaries of interaction. Notice in figure 6 the physical braille keyboard is made up of physical buttons that people can touch without activating then press and feel and hear that they have pressed it. Tactile and haptic feedback abound. The flat screens of the mobile devices lack this feedback and, thus, represent the pitch-black referenced in the title.

It is interesting to note that for blind users, braille chording operates at a sub-symbolic level. This is the second point of discussion for the workshop. Braille users understand braille chord sequences at a sub-symbolic level. Their basic literacy education trains them to integrate braille chords into their sub-symbolic, felt experience of the world. This provides a rich research opportunity to rigorously study the connections between what is touched and felt and what is touched and decoded.

References

[1] http://brailletouchapp.com/

[2] 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).

[3] Guerreiro, T., Lagoá, P., et al., NavTap and BrailleTap: Non-Visual Texting Interfaces, in *Proc. Rehabilitation Engineering and Assistive Technology Society of North America Conference (Resna)*: Washington D.C., (2008).

[4] Isbister, K., Höök, K., Sharp, M., & Laaksolahti, J. (2006). The sensual evaluation instrument. *CHI* (p. 1163). New York, New York, USA: ACM.

[5] 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 *Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility*, ACM: Halifax, Nova Scotia, Canada, (2008), 73-80.

[6] Romero, M., Frey, B., Southern, C., & Abowd, G. D. (2011). BrailleTouch: designing a mobile eyes-free soft keyboard*. MobileHCI* (pp. 707-709). ACM.

[7] Southern, C., Clawson, J., Frey, B., Abowd, G., & Romero, M. (2012). An evaluation of BrailleTouch: mobile touchscreen text entry for the visually impaired. *MobileHCI* (pp. 317-326). ACM