Swipeboard: A Text Entry Technique for Ultra-Small Interfaces That Supports Novice to Expert Transitions

Xiang 'Anthony' Chen[1][2], Tovi Grossman[1], George Fitzmaurice[1]

[1]User Interface Group
Autodesk Research
{firstname.lastname}@autodesk.com

[2]HCI Institute
Carnegie Mellon University
xiangchen@acm.org

ABSTRACT

Ultra-small smart devices, such as smart watches, have become increasingly popular in recent years. Most of these devices rely on touch as the primary input modality, which makes tasks such as text entry increasingly difficult as the devices continue to shrink. In the sole pursuit of entry speed, the ultimate solution is a shorthand technique (e.g., Morse code) that sequences tokens of input (e.g., key, tap, swipe) into unique representations of each character. However, learning such techniques is hard, as it often resorts to rote memory. Our technique, Swipeboard, leverages our spatial memory of a QWERTY keyboard to learn, and eventually master a shorthand, eyes-free text entry method designed for ultra-small interfaces. Characters are entered with two swipes; the first swipe specifies the region where the character is located, and the second swipe specifies the character within that region. Our study showed that with less than two hours' training, Tested on a reduced word set, Swipeboard users achieved 19.58 words per minute (WPM), 15% faster than an existing baseline technique.

Author Keywords

Text entry, input technique, mobile device, Swipeboard.

ACM Classification Keywords

H.5.2 [Information interfaces and presentation]: User Interfaces - Graphical user interfaces; Input devices and strategies.

INTRODUCTION

Typing is an important part of our digital activities, either with a desktop computer or other smart devices. For smart devices, soft keyboards have become the dominant text entry modality. However, as smart devices become miniaturized, it is increasingly difficult to type on a soft keyboard, due to the 'fat finger' problem [13].

To mitigate this problem, prior work has proposed various text entry solutions, that may involve new gestures [12,18,19] or additional sensors [11,15]. However, not all of these techniques can be adapted to ultra-small interfaces, and those that can usually have a limited entry speed well below 20 WPM [10].

In contrast, shorthand techniques (e.g., Morse code) promises a much faster entry speed (reportedly, the world's fastest Morse code entry is 140 WPM [4]). While such techniques can be easily adapted to an ultra-small interface, learning them is hard, as it often resorts to rote memory. This creates a dilemma: techniques palatable to users only give incremental improvements, while shorthand techniques can go significantly faster but create a high barrier for user learning and acceptance.

In this paper, we present Swipeboard, a technique that supports fast text entry on ultra-small interfaces. Swipeboard divides the keyboard into nine regions (Figure 1a). Entering any character requires two swipes: the first swipe specifies the region where the character is located, and the second swipe specifies the character within that region (Figure 1b). This technique leverages our spatial memory of a QWERTY keyboard to ease novice learning, while an expert can perform efficient, eyes-free, shorthand input. Further, the technique is *target agnostic*, requiring no knowledge of target locations or dimensions [17].

We present a performance model and a user study to demonstrate how Swipeboard users' behavior will transition from novice to experts, and how their performance can increase through learning and practice: in less than two hours, participants transcribing from a reduced word set achieved 19.58 WPM on a 12mm x 12mm interface, 15% faster than an existing baseline technique.

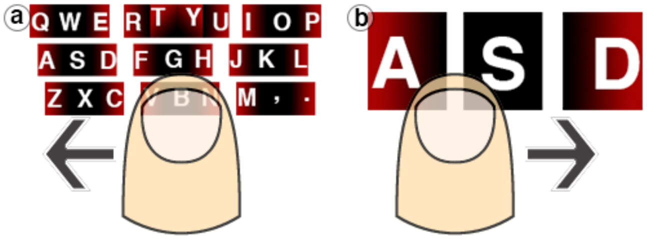

Figure 1. Swipeboard: the first swipe specifies one of the nine regions subdivided from a QWERTY keyboard; the second swipe specifies the character. For example, swiping left goes from (a) to (b), then swiping right types 'D'.

UIST '14, October 05 - 08 2014, Honolulu, HI, USA
Copyright 2014 ACM 978-1-4503-3069-5/14/10···$15.00.
http://dx.doi.org/10.1145/2642918.2647354

RELATED WORK

The miniaturization of ultra-small interfaces limits the use of conventional numeric keypads, keyboards, and touch input. Wigdor and Balakrishnan designed a chord input to improve the traditional *Multitap* on the few and small phone keys [16]. The use of a touch screen inspires other possibilities. *MultiWidget* uses a dialing gesture along the watch's edges to specify a numeric value [1]. This suggests a watch's tangibility could ease the input. *Zoomboard* uses iterative zooming for enlarging and acquiring keys on a soft keyboard [10]. This mitigates the 'fat finger' problem, but still requires a fair amount of visual attention and motor skills.

In contrast, shorthand text entry techniques are potentially eyes-free with high entry speed, especially for expert users. Two approaches for designing such techniques have been explored. A *continuous* approach, such as *EdgeWrite*, maps a word to a continuous stroke [18]. A *discrete* approach, such as *H4-Writer*, uses a base-4 Huffman coding to generate unique key sequences for each letter [8]. *Quikwriting* uses grouping to map characters to a base-9 encoding [12]. Unfortunately, such techniques usually require extensive learning and reference character maps, making them impractical for broad adoption.

What would be more desirable is a system that guides novices "*to perform like experts*" [2]. For example, Kurtenbach's marking menus provide *self-revelation* to help novices navigate and select an item, while *guidance* and *rehearsal* further train novices towards using the menu like experts [7]. Kristensson and Zhai also articulate this idea in building the text entry system *Shark*^{2} [5]. With *Shark*^{2}, novices simply trace a word by searching and inking through its constituent letters, experts eventually can memorize and execute this process as one single stroke, and the system accommodates users' behavior between these two 'extremes'. Unfortunately the technique may not scale well to ultra-small interfaces.

In summary, existing approaches that are based on the QWERTY layout either plateau at limited WPM, or are impractical on ultra-small interfaces; in contrast, shorthand techniques that rely on non-conventional layouts or gestures can achieve faster entry but are usually difficult to learn. In this paper, our goal is to design a text entry technique that supports novice to expert behaviors, and gradually scales users' performance to achieve a high entry speed.

SWIPEBOARD

To achieve this goal, we design Swipeboard – a text entry technique that leverages our spatial memory of a QWERTY keyboard layout to encode each alphabetic character into two touch actions that navigate to the key of that character. From a motor perspective, this makes the technique similar to the two-level simple marking menu technique [6]. Learning from Zhao and Balakrishnan's work [20], we also used simple, discrete actions, which are more suitable for ultra-small interfaces, rather than requiring compound, continuous strokes.

Swipeboard divides a QWERTY keyboard into nine regions (Figure 1a); the first swipe specifies the region where the character is located, e.g., swiping left goes to "ASD" (Figure 1b), or swiping up goes to "RTYU", and a tap, anywhere on the interface, brings up "FGH". A second swipe (or tap) specifies the character within the first region: in the cases of three keys (e.g., Figure 1b), swiping left selects the key on the left, tapping anywhere selects the key on the center, and swiping right selects the key on the right; in the one case of four keys ("RTYU"), swiping left selects 'R', up-left selects 'T', up-right selects 'Y' and right selects 'U'. If the first swipe (or tap) is incorrect, a swipe down, or a 2-second timeout, restores the keyboard to the top level (Figure 1a). Other combinations of swipes are reserved for additional functions. A double-swipe down-left deletes a character, double-swipe down-right enters a space, and a double-swipe up switches to symbols and numbers.

Swipeboard has several beneficial features. First the technique is *target agnostic*: the actions can occur anywhere on the display, and no spatial target selection is required. This supports use on ultra-small interfaces even when visual cues might be illegible to human eyes. Second, the gestures are based on the familiar QWERTY layout, supporting novice use. Finally, the technique should allow an expert to chunk a single character entry into an efficient and eyes-free double swipe (or tap), dispensing with reliance on the visuals, thus supporting fast text entry.

We have found the technique can be used at very small sizes, with the limiting factor being the legibility of the keyboard (for novices' learning), not the input area (our study participants were able to use Swipeboard as small as 12 mm by 12 mm). In general, this technique can be used on smart watches (Figure 2a), and potentially smart eyewear, with an input area on the side (Figure 2b).

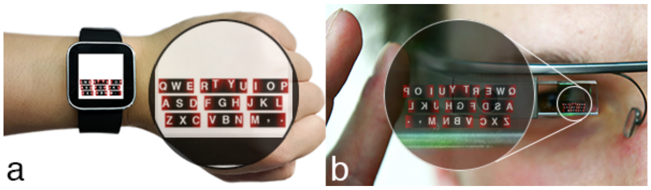

Figure 2. Swipeboard is usable as small as 12 mm by 12 mm, supporting potential usage on (a) smart watches or (b) smart eyewear.

PERFORMANCE MODEL

Swipeboard encodes each character with a first swipe (or tap) to locate the key on the keyboard, and a second swipe (or tap) to specify the key. We call a swipe (or tap) an *action*. Users will begin in a novice phase, where they must identify the gesture for a target before each action, while the finger is up. We call this process *up*. Text entry in this novice phase will be perceived and carried out as two separate *units of interaction* that consists of a total of four steps (1st up, 1st action, 2nd up, and 2nd action).

Essentially, novices will rely on the visuals of the keyboard, as well as their spatial memory of a QWERTY keyboard, to specify a key. This mechanism allows them to learn and rehearse each character entry. Gradually, they will transition towards an expert phase by trying to associate two actions for a given character. The time for the *up* actions will gradually decrease as less reliance on visual search cuts the decision making time. When reaching the expert phase, each character entry will be chunked into a single unit of interaction that consists of two consecutive actions, dispensing with deliberate searching steps. In particular, we predict Swipeboard performance to be:

$$T = T_{u1} + T_{a1+} + T_{u2} + T_{a2},$$

where the total time T is the sum of two *up* steps (T_{u1}, T_{u2}) and two *action* steps (T_{a1}, T_{a2}). Based on research on gesture performance modeling, we expect a stroke or tap time to be approximately 100 ms [3]. Thus, for a novice, we have a character entry time, T_{nov}:

$$T_{nov} = T_{u1} + 100 + T_{u2} + 100$$

To model optimal expert use, we make two assumptions. First, their initial reaction time is optimal, so we assign a value of 200ms to T_{u1} as an approximate human limit [14]. Second, we assume the expert will chunk the two swipes into a single action, reducing T_{u2} to the time taken to lift the finger between the two swipes. Mackenzie has found that it takes approximately 127ms to lift up *and* tap in place [9]. When repeating such action the finger would be up approximately half of the time, therefore we estimate our *up* time T_{u2} as half of that, 64ms. As such, we predict optimal upper limit character entry time, T_{opt} as:

$$T_{opt} = 200 + 100 + 64 + 100$$

This equates to 464ms per character, or 25.87 WPM. The model shows that the difference between Swipeboard novices and experts lie in the time while the finger is up. The novice must determine the correct swipe at each level mostly through visual search, while an expert immediately performs the actions upon recalling what actions are associated to the desired key. Below, we present a study that provides data to support our model.

USER STUDY

The goal of our study is to measure the effectiveness of Swipeboard for entering text on an ultra-small interface. Specifically, we compared our technique with Zoomboard – a state-of-the-art text entry method for miniaturized devices [10]. Our study observes how Swipeboard users' behavior transitions from novice to expert, and how the performance increases through learning and practice.

Design

The study consisted of a series of text entry trials. To accelerate the learning elicited within the timeframe of our lab study, each trial consisted of a single four letter word, with each word consisting of four characters from the set: E, T, A, N and S, such as 'EATS', 'SEAT', 'NEST'. Our goal of choosing this reduced word set is *not* to replace a regular text entry study, but rather to develop a way to quickly test the novice-expert performance growth by artificially accelerating the learning of the techniques focusing on a small word set. This would help provide insights into Swipeboard's optimal entry time from a motor skill perspective. The character set was chosen so that a broad range of the possible swipe directions would be required.

To prevent the transfer effects between techniques inherent in within-subjects designs, we employed a repeated measures between-subjects factorial design. The independent variable was *Technique* (ZoomBoard vs. Swipeboard). Participants were randomly assigned to one of the two groups. They were asked to complete two sessions of trials on separate days with one day in between. Each session contained 4 blocks of trials. For each block, participants were asked to enter 108 words, (nine consecutive sets, each containing 12 distinct words). In summary, the experimental design was:

2 Techniques x
8 participants per Technique x
2 sessions per participant x
4 blocks per session x
108 words per block
= 13824 data points.

Apparatus

We implemented both SwipeBoard and ZoomBoard with iOS, deployed on an Apple iPad tablet device – a platform that was also used in Oney et al.'s study [10]. At development time, this served as a simulation platform for rapidly prototyping and experimenting with different input and output resolutions, and more importantly, with full touch capability, which was unavailable on most commercial ultra-small interfaces at the time we conducted the study. Note that the design of Swipeboard, however, is not restricted to this platform; it can be generalized and transferred to other miniaturized interfaces, e.g., ring-sized, wrist-worn, or head-mounted devices.

We used a keyboard dimension of 12 mm by 12 mm (about ¼ the size of a regular smart watch) on the iPad display wherein we enabled and used native iOS touch events to implement the two text entry techniques. All input events outside of this zone were ignored. We measured the completion time for each word (from tapping the 'start' button to the complete and correct entry of a word), counted the errors, and, for each character, logged the time for each *up* and *action* events. Two sizes were used to render each key on each text entry step, based on Zoomboard's original paper [10] – small size: 1.5mm x 1.5mm and large size: 4.4mm x 4.4mm. For Zoomboard, small size was used before zooming and large size afterwards. For Swipeboard, small and large sizes were used before and after the first swipe (or tap), respectively.

Participants

We recruited 16 participants from a local university. They were between 21 to 26 years old, 9 female and 7 male. One of them was mixed-handed and two left-handed. All were smart phone users. All but one typed with a soft keyboard on a daily basis.

Task and Stimuli

Each participant was asked to finish a series of text transcribing tasks using our selected word set.

At the beginning of each task, a word prompted the participants. They were asked to read and memorize the presented word, then to press the 'start' button after which the word was removed and a technique was activated for transcribing that word. Timing started when the 'start' button was pressed, and stopped when the entire word is transcribed correctly. When an erroneous character was entered, the display highlighted it with red, and the word will re-appear. The time for correcting errors was counted into the total completion time. Upon finishing a word, the system automatically advanced to the next one.

The two groups of participants used ZoomBoard and SwipeBoard, respectively, as described below:

ZoomBoard. To enter a character, a user will tap a corresponding key on the keyboard. This first tap will not select the key, but will only zoom the keyboard into the tapped region. Then the user will perform a second tap on the desired key in order to select it. The user can also swipe down to zoom out from the current region. The keyboard will also zoom out to the original size after the second tap or a two-second timeout of no user actions. At any time, the user can swipe left to delete an entered character, and swipe right to enter a 'space'.

SwipeBoard. To enter a character, a user will firstly swipe to select one of the nine regions where the corresponding key is located (Figure 1): upper-left, up, upper-right, right, lower-right, down, lower-left, left; except for the central region, where a tap anywhere within the interface is used for selection. After the first swipe or tap, the interface will show the selected region of three or four keys. In the cases of three keys (Figure 1), the user will swipe left to select the left key, right for the right key, and tap anywhere for the key at the center. In the only case of four keys ("RTYU"), the user will swipe left for 'R', up-left for 'T', up-right for 'Y' and right for 'U'. At this step, the user can also swipe down to return to the full keyboard view. At any time, swiping lower-left twice deletes an entered character, and swiping low-right twice enters a 'space'.

Results and Analysis

In this section we discuss the results based on text entry speed and error rates. We then relate the data to our performance model.

Text Entry Speed

Figure 3 shows the aggregated WPM results (including error correction time). Although Swipeboard did not start faster (9.09 WPM) than Zoomboard (10.76 WPM), participants were able to quickly learn the technique and, as shown in Figure 3, achieve a higher entry speed than Zoomboard's from the third block of the study onwards (after about 35 to 45 minutes' of usage time). As the study went on, Swipeboard continued to accelerate. By the end of the study (the 8th block performed in the 2nd session), Swipeboard's WPM was 19.58, 15% faster than Zoomboard (17.08 WPM).

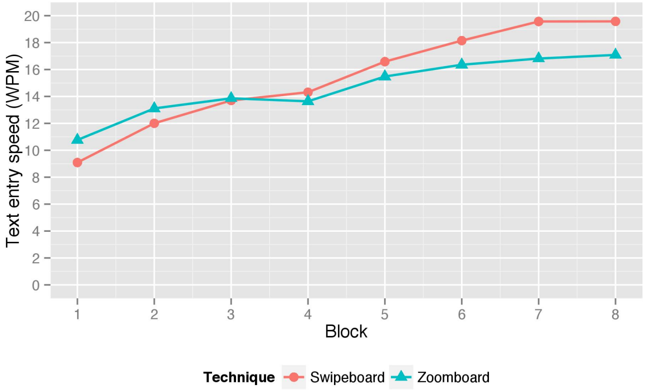

Figure 3. With less than two hours' training, Swipeboard users achieved 19.58 words per minute (WPM) entering text on a 12mm x 12mm interface, 15% faster than Zoomboard [10].

The overall effect of technique was significant ($F_{1, 14}$ = 6.503, p = .0112). The effect of block ($F_{7, 105}$ = 52.99, p < .0001), and the interaction between technique and block ($F_{7, 98}$ = 3.952, p = .0004) were also significant. This indicates that there was a learning effect in both technique and the Swipeboard produced a faster learning curve, as well as a higher text entry speed compared to Zoomboard.

Error Rate

As both techniques require two steps for entering a character, we recorded two types of errors inherent to this design:

- *Soft error*: error made in the first step of entering a character, such as, for Zoomboard, zooming into the wrong area of the keyboard, and for Swipeboard, swiping or tapping to the wrong regions of keys.
- *Hard error*: error made in the second step of entering a character, such as, for Zoomboard tapping the wrong key on the zoomed in area, and for Swipeboard, swiping or tapping for the wrong character within a region of key.

Essentially, soft errors reflected the difficulty of making the first step. Overall, the soft error rates were 4.183% for Swipeboard and 4.739% for Zoomboard. The difference was not significant ($F_{1, 14}$ = .9764, p = .3237). Hard errors are errors that resulted in a mistyped character. The hard error rates were 13.30% for Swipeboard and 14.90% for Zoomboard and the difference was significant ($F_{1, 14}$ = 3.933, p = .0481). Figure 4 shows the two error rates of both techniques.

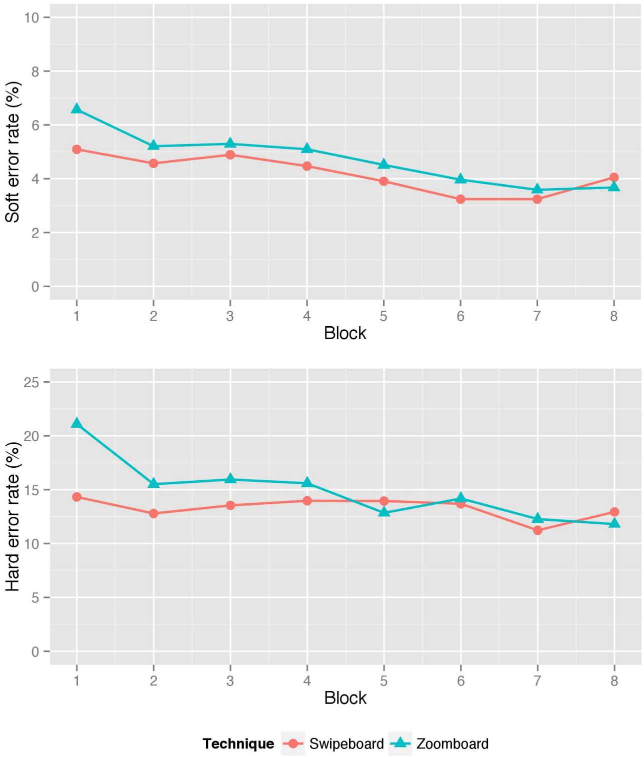

Figure 4. Soft and hard error rates of Swipeboard and Zoomboard: soft errors are made in the first step; hard errors in the second step and result in mistyped characters.

Performance Model

Figure 5 shows a breakdown associated with our model of the two techniques’ completion time, plotted across all blocks in the study. As the time for both *actions* was almost always constant, we focused on the *up* time – the portion that actually contributed to the difference in the two techniques’ entry speed.

For the 1st *up*, Swipeboard participants started much slower than Zoomboard. However, through learning and practice, the time rapidly dropped and the gap between the two techniques greatly decreased. In the second half of the study, the difference had been reduced to about 65 ms. As Swipeboard users transitioned from novices to experts, they relied less on visually searching the keyboard but instead gradually memorized the association between characters and the swipes/taps, resulting in a shorter reaction time in this initial interaction. For the 2nd up, Swipeboard participants also demonstrated a similar learning curve: from novices to experts, their reaction time continuously became shorter, whereas Zoomboard’s time only had a minor decrease. Zoomboard’s second tap suffers from the ‘fat finger’ problem, where there is an upper bound as to how fast a user can select a small target. Swipeboard, on the other hand, is target agnostic, providing users with a larger room for improvement.

LIMITATIONS, TRADE-OFFS AND FUTURE WORK

We discuss existing limitations and trade-offs of our work and point to possible future research directions.

Expanding the word set. Our study used a reduced word set. The five letters were selected to strike a balance between high usage frequency and a variety of swipes/taps when using Swipeboard as the entry technique. Although this five-letter set already covers 43% of all letter frequency occurrences in English, a full letter set (and perhaps including some symbols) is still needed if we are to reflect the learning curve of a user typing in day-to-day scenarios.

Extending to more platforms. Swipeboard can benefit a variety of interactive platforms, such as smart rings, smart watches, and smart eyewear. Our study in this paper is very basic and cannot address each of these devices individually. More testing is required to study ergonomic issues, e.g., devices’ orientations and placements. Modalities other than touch can also be included into comparison.

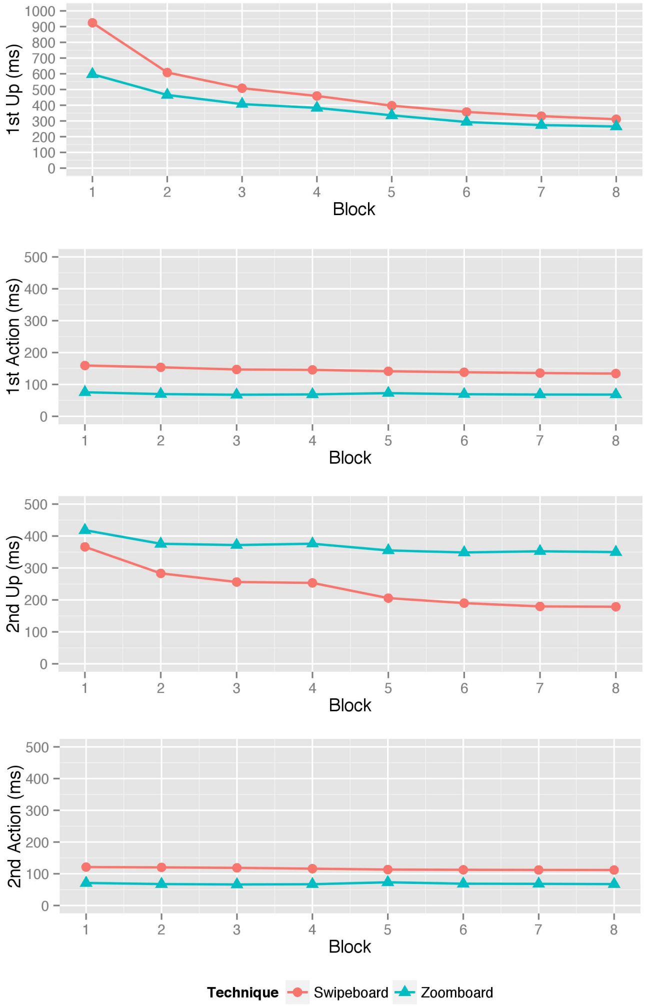

Figure 5. In error-free trials, a break-down of the techniques’ completion time shows that as Swipeboard users transitioned from novices to experts, they spent less time deciding on each swipe actions.

Observing more training. Based on our performance model, there is still room for improvement in participants' Swipeboard performance. Anecdotally, the first author trained about a total of four hours of Swipeboard practice and can achieve an entry speed of 32.9 WPM.

Considering individual character entry. Swipeboard users transition from novices to experts by trying to memorize the association between the swipes/taps and the characters. Our data shows that some character entries, however, seemed easier to memorize and execute than the others. Figure 6 shows the aggregated character entry time in our study. There was a difference in per character difficulties, especially for novices (see how first block participants entered 'A' much faster than 'N' – the former character requires two identical and consecutive swipe-left); however, as participants got better in using the technique, the gaps became smaller. This suggests that with Swipeboard, not all the character entries are created equal. Future work should address this by, for example, providing more training on harder character entries, or adjusting the recognizer to learn how users would enter the more difficult characters.

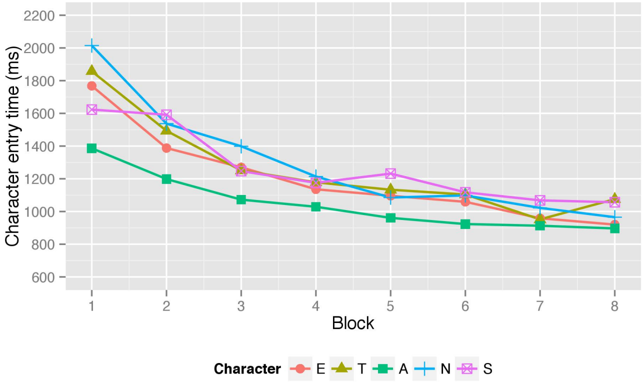

Figure 6. Individual character entry time shows that with Swipeboard some characters entries are easier to memorize and execute; such difference is most notable at the novice level.

Incorporating other models. Future work can also incorporate a probabilistic model when recognizing entries of characters with different usage frequencies. Language models can also be used where only the first swipes or taps are needed to specify a word – this interestingly makes Swipeboard similar to the traditional T9 on numeric keypads, except that the target agnostic swipes/taps require no actual keys to be devised or rendered on the interface.

CONCLUSION

With the miniaturization of smart devices, typing on an ultra-small interface is becoming a pressing design challenge. Our technique leverages people's spatial memory of a QWERTY keyboard to learn, use, and master a shorthand text entry method that is otherwise hard to learn and be accepted by general users. Swipeboard's design accommodates a wide range of user behavior, from novices to experts, and allowed users, with less than two hours' training, to achieve an entry speed of 19.58 WPM when typing a reduced word set on a 12mm x 12mm interface. Swipeboard's target agnostic nature also makes it a useful technique for small wearable devices where visual cues might be illegible or unavailable to the users.

REFERENCES

1. Blasko, G. and Feiner, S. Evaluation of an Eyes-Free Cursorless Numeric Entry System for Wearable Computers. *IEEE Wearable Computers*, (2006), 21–28.
2. Buxton, W.A.S. Chunking and phrasing and the design of human-computer dialogues. *IFIP Congress* (1995), 494–499.
3. Cao, X. and Zhai, S. Modeling human performance of pen stroke gestures. *CHI '07*, 1495-1504.
4. Cornwell, J.M. Morse Code at 140 WPM. *The National Association for Amateur Radio*, 2008.
5. Kristensson, P.-O. and Zhai, S. SHARK 2. *ACM UIST '04*, 43-52.
6. Kurtenbach, G., Fitzmaurice, G., Baudel, T., and Buxton, B. The design of a GUI paradigm based on tablets, two-hands, and transparency. *ACM CHI '97*, 35–42.
7. Kurtenbach, G.P. The design and evaluation of marking menus. Diss., University of Toronto, 1993.
8. MacKenzie, I.S., Soukoreff, R.W., and Helga, J. 1 thumb, 4 buttons, 20 words per minute. *ACM UIST '11*, 471-480.
9. MacKenzie, I.S. and Zhang, S.X. The design and evaluation of a high-performance soft keyboard. *ACM CHI '99*, 25–31.
10. Oney, S., Harrison, C., Ogan, A., and Wiese, J. ZoomBoard. *ACM CHI '13*, 2799-2802.
11. Partridge, K., Chatterjee, S., Sazawal, V., Borriello, G., and Want, R. TiltType. *ACM UIST* (2002), 201-204.
12. Perlin, K. Quikwriting. *ACM UIST '98*, 215–216.
13. Siek, K.A., Rogers, Y., and Connelly, K.H. Fat finger worries: how older and younger users physically interact with PDAs. *INTERACT 2005*. 267–280.
14. Welford, A.T. Reaction time, speed of performance, and age. *Annals of the New York Academy of Sciences 515*, 1 (1988), 1–17.
15. Wigdor, D. and Balakrishnan, R. TiltText. *ACM UIST* , 81–90.
16. Wigdor, D. and Balakrishnan, R. A comparison of consecutive and concurrent input text entry techniques for mobile phones. *ACM CHI '04*, 81–88.
17. Wobbrock, J.O., Fogarty, J., Liu, S.-Y.S., Kimuro, S., and Harada, S. The angle mouse. *ACM CHI '09*, 1401–1410.
18. Wobbrock, J.O., Myers, B.A., and Kembel, J.A. EdgeWrite. *ACM UIST '03,* 61–70.
19. Zhai, S. and Kristensson, P.-O. Shorthand writing on stylus keyboard. *ACM CHI '03*, 97-104.
20. Zhao, S., and Balakrishnan, R. Simple vs. compound mark hierarchical marking menus. *ACM UIST '04*. 33-42).