
Text Entry for Ultra-Small Touchscreens Using a Fixed Cursor and Movable Keyboard

Tomoki Shibata

Tufts University
Medford, MA, USA
tshibata@cs.tufts.edu

Daniel Afergan

Tufts University
Google Inc.
Mountain View, CA, USA
afergan@google.com

Danielle Kong

Tufts University
Medford, MA, USA
danielle.kong@tufts.edu

Beste F. Yuksel

Tufts University
Medford, MA, USA
byukse01@cs.tufts.edu

I. Scott MacKenzie

York University
Toronto, ON, Canada
mack@cse.yorku.ca

Robert J.K. Jacob

Tufts University
Medford, MA, USA
jacob@cs.tufts.edu

Abstract

Emerging ultra-small wearables like smartwatches pose a design challenge for touch-based text entry, due to the “fat finger” problem, wherein users struggle to select elements much smaller than their fingers. In order to address the challenge, we introduce *DriftBoard*, a panning based text entry technique in which the user types by navigating the movable QWERTY keyboard with respect to the fixed cursor point. In this paper, we describe interactions on the proposed typing system, and report a preliminary result of our user study on a watch-size touchscreen with a text copy task comparing DriftBoard to two published ultra-small keyboards, ZoomBoard (tapping-based) and Swipeboard (swiping-based). Within the user study, DriftBoard performed comparably (no significant difference) to ZoomBoard in the major metrics of text entry speed and error rate, and outperformed Swipeboard, which suggests that the proposed panning-based typing is another promising input form for text entry on ultra-small touchscreens.

Author Keywords

Text entry; ultra-small touchscreens; fat finger problem; fixed cursor and movable keyboard

ACM Classification Keywords

H.5.2 [User Interfaces]: Input devices and strategies, Interaction styles

Permission to make digital or hard copies 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 bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s). Copyright is held by the author/owner(s).

CHI'16 Extended Abstracts, May 7–12, 2016, San Jose, CA, USA.

ACM 978-1-4503-4082-3/16/05.

<http://dx.doi.org/10.1145/2851581.2890230>

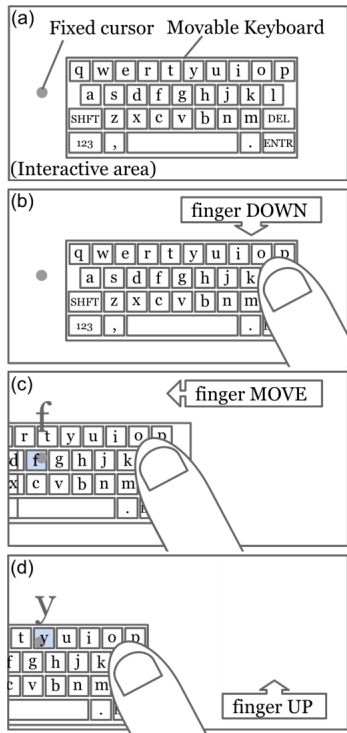


Figure 1: (a) DriftBoard basic. To type, the user (b) places a finger down on the interactive area, (c) moves the finger to navigate the movable keyboard towards the fixed cursor, and then (d) releases the finger up from the interactive area, where “y” is typed in this case. Note that the bound of the movable keyboard can go outside of the interactive area, and that key highlighting and enlargement are provided for whatever character is underneath the fixed cursor.

Introduction

User input on ultra-small wearables is far more limited than output. For example, smartwatches offer instant access to notification messages, but responsive interactions to them still pose a challenge. Notably, text entry with a conventional software keyboard on a smartwatch will suffer due to the fat finger problem [7], wherein a finger of the low spatial resolution will not be able to precisely point the tiny character keys.

Related Work

Not limited to text entry, but input forms for wearables have been actively studied. For example, exploring input spaces on skin, Skinput [4], or on the back side of a device, Back-of-device [1], carrying a sharper stylus on a finger tip, NanoStylus [8], and so on.

For text entry, several leading approaches that manipulate the keyboard display or introduce new gesture sequences have shown promise against the hurdle. For instance, ZoomBoard [6] employs a tapping-based approach where tap actions at certain times zoom in the keyboard display; therefore, the desired key becomes large enough to touch. Swipeboard [3] employs a swiping-based approach where each character is associated with a unique sequence of swiping actions; therefore, a user does not need to hit an exact location on the screen.

Yet, not all available touch-based interactions have been examined as input forms of text entry.

DriftBoard Interaction

Our goal is 1) to design a touch-based input technique which overcomes the fat finger problem, and 2) to validate its potential for practical use with a hope of facilitating text entry on ultra-small touchscreens.

To accomplish this, we begin with proposing *DriftBoard*, a panning-based text entry technique using a fixed cursor point and movable QWERTY keyboard. To type with DriftBoard, the user performs a panning action: dragging the movable QWERTY keyboard towards the fixed cursor point, and then releasing the finger from the touchscreen, which types the character that is currently underneath the fixed cursor. (If no character key is acquired by the fixed cursor at the time of the finger release, no character is typed.)

From a technical view, a panning action can be described with the three primitive finger events: finger DOWN, MOVE and UP. Within one panning action, the finger DOWN event initiates a panning; series of finger MOVE events continuously translate the location of the movable keyboard, and then the finger UP event completes the panning action and finally types the character acquired by the fixed cursor.

Figure 1. illustrates the interaction on DriftBoard with the three key components (the fixed cursor, movable keyboard and interactive area) and user’s finger events forming a panning action. Figure 2. shows its physical size for a user study, and Figure 4. demonstrates a user typing with DriftBoard on an Android smartwatch of a round display.

This pan technique differs in concept from conventional tap input techniques, in which the keyboard (target) remains stationary and the user navigates a cursor/finger (acquisition point) - here, the user navigates the keyboard (target) and the cursor (acquisition point) remains stationary.

Because a panning action can be initiated and completed at any part of the interactive touch area, even regardless of on the keyboard or not, and the middle part of a panning allows the user to refine the keyboard location respect to the fixed cursor, we believe our panning-based typing is an efficient means to address the fat finger problem.

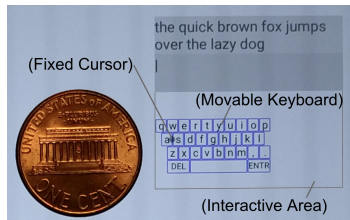


Figure 2: DriftBoard prototype on an Android touchscreen device and a U.S. one cent coin next to it as a reference scale. In the user study, participants were asked to type the same sentence appearing on top.

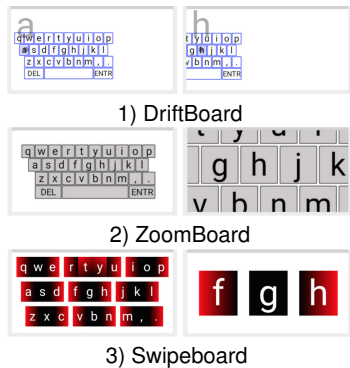


Figure 3: Appearances on the touchscreen for the user study. 1-Left) The fixed cursor of DriftBoard is acquiring “a”, and 1-Right) is acquiring “h”. 2-Left) The initial state of ZoomBoard, and 2-Right) a zoomed state by a tap near “h”. 3-Left) The initial state of Swipeboard showing all of the character blocks, and 3-Right) the state showing the “fgh” block.

Method

In order to investigate the potential of the panning-based typing mechanism, we conducted a controlled user study with a text copy task utilizing DriftBoard (panning-based) on a smartwatch size touchscreen (the interactive area: 28 x 14 mm, roughly half of 1.6 inches (diagonal) screen), as well as the two published ultra-small keyboards, ZoomBoard [6] (tapping-based) and Swipeboard [3] (swiping-based), replicated on the screen of the same size.

Participants: 10 participants (5 females) of age between 19 and 33 (Mean:22.8, SD:4.2) who are right handed, native English speakers or bilingual from birth and have normal or corrected-normal vision. All were paid 10 U.S. dollar per hour for their participations, in addition the performance bonus of 5 U.S. dollar.

Apparatus: All three input techniques were implemented as Android Applications. We used a tablet, Nexus 7 (2013), and covered up the unused surface area to offer feeling of a watch size. The device was placed on a table while participants performed the task. Figure 3. presents appearances of each input technique on the touchscreen.

Procedure: The experiment was composed of 2 sessions (one per a day) for each participant with 3 input techniques per session, 5 blocks per input technique and 5 phrases per block. All phrases (lowercase only including space and no punctuation) were randomly picked from MacKenzie & Soukoreff’s phrase set [5]. The participants were instructed 1) to use the index finger of the right hand 2) to memorize phrase (although the presented phrase does not disappear), 3) to correct a typo only when they realized immediately after they made one but ignore it if they already made any progress on transcribing, and finally 4) to type as quickly and accurately as possible. In the first session, before experiencing each input technique, the participants watched the experimenter enter a few characters with ver-

bal instructions. They were then asked to begin their trial blocks without any practice. In the second session, before starting each input technique, the participants were asked to type “hello world” to refresh their past experiences.

Design: A within-subjects design with two independent variables: 3 input techniques and 10 trial blocks (to capture participants’ improvements on typing). For the preliminary analysis, the dependent variables were text entry speed measured as words per minute [wpm] and error rate [%]. To offset learning effects across the input techniques, each of all 6 possible orders by the 3 input techniques was assigned to one or two participants. Text entry speed and error rate were calculated by the methods described by Castellucci and MacKenzie [2]. Entry time was measured from the first to the last finger event on the interactive area, excepts for typing the ENTER key, which was used to indicate the completion of transcribing a phrase.



Figure 4: DriftBoard prototype in action on a smartwatch of a round display (Motorola Moto 360). Left) The fixed cursor is on top of “r”. Right) The location of the fixed cursor remains the same, but it acquires “z” as the result of the keyboard move. Note that in the DriftBoard concept, the size of the movable keyboard does not necessary to be the same as the device’ frame size. Note also that DriftBoard is applicable to more shapes than just a rectangle.

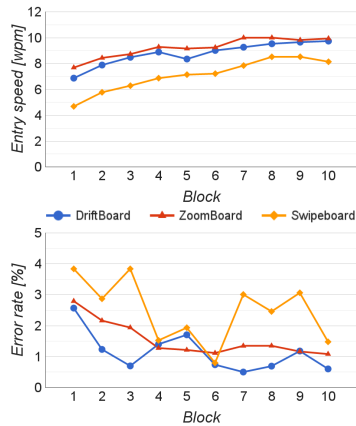


Figure 5: Text entry performances, speed and error rate.

Replication notes:

(ZoomBoard): Swipe to type a space and swipe to delete functionals [6] were not implemented. The scaling factor for zoom-in was set to about x3 [6].

(Swipeboard): The ENTER key was assigned to a double swiping down action.

(ZoomBoard and Swipeboard): "2-second timeout" [3] was implemented.

(DriftBoard): The location of the movable keyboard was reset to one shown in Figure 2. every time when a new phrase appears.

Results

We here report a preliminary analysis on 1,455 phrases out of the collected 1,500 total by excluding 7 erroneous data due to hardware issues, and 38 outliers, which appeared as error rate of more than 50% or efficiency (ratio of actual and optimal keystrokes per character) of less than 50%.

Figure 5. shows averaged performances of the text copy task. With about the total of one hour experience (at the 10th block), DriftBoard achieved Text entry speed: 9.74 wpm and Error rate: 0.6%. On entry speed, the ANOVA indicates significant effects of input method ($F_{2,18} = 30.127, p < .0001$), of block ($F_{9,81} = 36.617, p < .0001$), and of interaction ($F_{18,162} = 1.746, p < .05$). Post-hoc comparisons with Fisher LSD test on input method reveal no significant difference involving ZoomBoard, and significant differences between Swipeboard and DriftBoard. On error rate, the ANOVA indicates a significant effect of input method ($F_{2,18} = 5.553, p < .05$), and no significant effects of block ($F_{9,81} = 1.799, p > .05$) and of interaction ($F_{18,162} = .760, ns$). Post-hoc comparisons with Fisher LSD test on input method reveal no significant difference involving ZoomBoard, and significant differences between Swipeboard and DriftBoard.

Conclusion

In this paper, we introduced a panning-based text entry mechanism, using a fixed cursor and movable QWERTY keyboard, to overcome the fat finger problem for text entry on ultra-small touchscreens. We demonstrated the potential of the panning-based approach compared with a tapping-based and swiping-based approach by replicating two published systems, ZoomBoard and Swipeboard. With the preliminary but promising result of the analysis, we believe the panning-based typing mechanism is a highly viable alternative for touch-based text entry on ultra-small touchscreens.

We will perform a full analysis on the collected data report it in the near future.

Acknowledgments

We thank Ronna ten Brink, Remco Chang from Tufts University and NSF (IIS-1218170 and IIS-1065154) for support.

References

- [1] Patrick Baudisch and Gerry Chu. 2009. Back-of-device Interaction Allows Creating Very Small Touch Devices. In *Proc. CHI 2009*. 1923–1932.
- [2] Steven J. Castellucci and I. Scott MacKenzie. 2011. Gathering Text Entry Metrics on Android Devices. In *Proc. CHI EA 2011*. 1507–1512.
- [3] Xiang 'Anthony' Chen, Tovi Grossman, and George Fitzmaurice. 2014. Swipeboard: A Text Entry Technique for Ultra-small Interfaces That Supports Novice to Expert Transitions. In *Proc. UIST 2014*. 615–620.
- [4] Chris Harrison, Desney Tan, and Dan Morris. 2011. Skinput: Appropriating the Skin As an Interactive Canvas. *Commun. ACM* 54, 8 (Aug. 2011), 111–118.
- [5] I. Scott MacKenzie and R. William Soukoreff. 2003. Phrase Sets for Evaluating Text Entry Techniques. In *Proc. CHI EA 2003*. 754–755.
- [6] Stephen Oney, Chris Harrison, Amy Ogan, and Jason Wiese. 2013. ZoomBoard: A Diminutive Qwerty Soft Keyboard Using Iterative Zooming for Ultra-small Devices. In *Proc. CHI 2013*. 2799–2802.
- [7] Katie A. Siek, Yvonne Rogers, and Kay H. Connelly. 2005. Fat Finger Worries: How Older and Younger Users Physically Interact with PDAs. In *Proc. INTERACT 2005*. 267–280.
- [8] Haijun Xia, Tovi Grossman, and George Fitzmaurice. 2015. NanoStylus: Enhancing Input on Ultra-Small Displays with a Finger-Mounted Stylus. In *Proc. UIST 2015*. 447–456.