Adding mid-air gestures to help editing on smartwatch text entry

Andreas Komninos
University of Strathclyde
Glasgow, Scotland, UK
Andreas.Komninos@strath.ac.uk

Mark Dunlop
University of Strathclyde
Glasgow, Scotland, UK
Mark.Dunlop@strath.ac.uk

This short paper was presented at the *ACM CHI 2017 Workshop on Ubiquitous Text Interaction*, May 2017, Denver, Colorado, USA.

Abstract

Text entry has been shown to have reasonable performance on smart watches but space is very tight and editing has been shown to be a major factor slowing down text entry on many devices. In this position paper we propose use of mid-air gestures to control editing functions so that screen estate and touch gestures can be focused on text entry.

Author Keywords

Text entry; editing text; mid-air gestures; interaction.

ACM Classification Keywords

H.5.2 User Interfaces: Interaction styles, Graphical user interfaces (GUI).

Introduction

While standard QWERTY text entry on smartwatches does appear to permit fast entry [8], editing is difficult as space is very limited. Mid-air input has been examined in the past but only as a complete replacement for touch interactions. The idea we would like to explore is the use mid-air gestures to assist (but not completely replace) touch input

Background

The fat-finger problem [18] is a concern for text entry on touch screen devices as the users' fingers are both blunt instruments and obscure large areas of the screen. This led to various attempts at novel watch-top text entry methods (e.g. [6,9,10,13]) that reduce the need for over-precise tapping on the tiny touch screens of smart watches. While all successful in permitting accurate text entry, their interaction style or learning curve restricted entry speed. An alternative is to use strong language models with a standard QWERTY layout to compensate for inaccurate tapping and also to support gesture writing [22]. Velocitap [19] and WatchWriter [8] both show text entry speeds that are in-line with those expected from touchscreen phones and considerably faster than custom layouts or approaches.

Correcting text can be a major limitation on overall text entry speed as it has been shown that input speeds decline after correcting an error [5]. Users often over rely on backspace for correction as cognitively and physically easier than more optimal edition techniques such as carat relocation and text selection – a problem that is exacerbated on touchscreens where precise location of the editing carat with blunt fingers is difficult. Arif et al. found that " 99% of the time participants corrected their errors with the backspace key" [3]. Adding arrow keys to a keyboard is a common solution on desktops to allow precise carat location but these take up space on a touchscreen further reducing the space available for the main alphabetic keys. Across keyboard gestures have successfully been used for carat movement [16], editing [7], and to replace space, shift, enter and backspace [2]. However, on-keyboard gestures are typically not compatible with gesture-writing approaches and there is very limited space off-keyboard on a smartwatch.

Mid-air input has been examined in the past but only as a complete replacement for touch interactions [11,12,20]. In [11] a mid-air gesture input technique for large displays is examined and found to produce reasonable input speeds, however the need for gesturing for various controls (e.g. start input, end input, delete, undo, select) slowed participants down. In [12] the authors examine head tracking as a way to implement a swipe-like interface, although this is more suited to impaired individuals without motor control of their hands. In [20], a technique similar to [11] is studied and found to produce comparable input speeds with a push-gesture QWERTY virtual keyboard, although in the experiment, the authors did not investigate the impact of errors (a forced error correction strategy was used)

In other research [14], it is highlighted that the lack of physical feedback on touchscreen keyboards further impacts writing speed, because the users have to frequently shift their attention between the keyboard and input area, in order to examine whether the touches have had the intended effect, which also applies to error correction via backspacing. In [17] a similar "slowing down" effect is observed for the first character, after a user switches keyboard layouts (e.g. from alphabetic to numeric and back).

To summarise, input in mobile or ubiquitous touch-based keyboards suffers from a range of factors, including key size, accuracy of gesture recognition algorithms, inadequate control of cursor and control keys and lack of support for seamless transition between keyboard modes. Although keyboard layout optimisations, gesture typing and mid-air typing have been explored in the past, we have not been able to find any literature that examines how these techniques could be used together to solve the input problem. Most importantly, we are interested in exploring how large and computationally robustly recognisable mid-air gestures might replace some of the non-input control elements of a virtual keyboard, in order to prevent input slow-down during error correction and keyboard control.

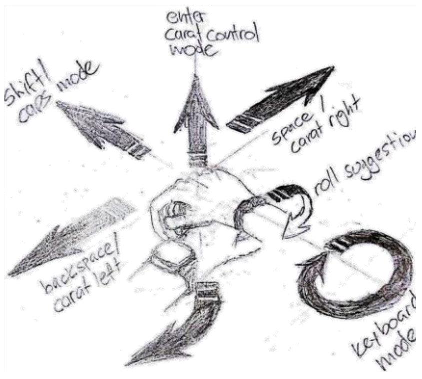

Figure 1: Suggested interactions

Edit command gestures

The keyboard layout for tiny screens should maximise the space for alphabetic text entry in normal use to improve tap accuracy as far as possible given layout constraints. As such the alphabetic QWERTY layout should dominate. The non-alphabetic controls that are needed for full text entry are:

- **Space** input: While traditionally needed after most words or punctuation, some language models do not need space explicit space input (e.g. [19]). It still, however, helps considerably in disambiguating taps. Touch-screen keyboards typically automatically insert taps after a gestured word or on picking from a suggestion bar, however reliance on this prevents out-of-dictionary typing.
- **Shift** and caps-lock: Keyboard now typically auto-capitalise but occasional use of shift is still needed, for example for words that can be proper nouns or common nouns (e.g. *Mark* & *mark*).
- **Punctuation** input: Typically limited on the primary keyboard to periods and commas.
- **Enter**, accept, OK: Now often overloaded the traditional carriage-return key is used on mobiles as a carriage-return in paragraph based typing and as a short cut to OK/accept in other situations. Even in prose based text entry it's use is very much limited from the early electric keyboards where manual line breaks were required.
- **Number, symbol and Emoji** input: a wide range of non-alphabetic characters and extended character set symbols are needed in addition to support for the increasing use of Emojis [15]
- **Backspace**: The principle editing tool of most users – people typically engage in "psychotic backspacing" often in preference to more efficient editing processes.
- **Suggestion picking**: Alternative suggestions are needed for any intelligent text entry method to allow users to pick alternative interpretations of their input.
- **Dynamic carat positioning**: Moving the carat dynamically to a new position in the text. Difficult on a small touch screen as tapping is inaccurate and a limited amount of text is usually visible on tiny screens during typing. The accuracy problem is often solved with pop-up zoomed in displays and reduced movement sensitivity during carat movements.
- **Precise carat positioning**: on desktops using ← ↑ → and ↓ keys to move the carat this is typically omitted on touch screens.

Potential above screen gestures

Above device gestures could be enabled by the user wearing mini sensor on their, say, right hand while tapping on a watch on their left wrist or by using near surface interaction affects that can identify dual point interaction (e.g. [4,21]). Figure 1 summarises our suggestions.

For the non-editing commands identified we suggest minimising use through automation and suggestion picking or gesture typing, reducing their use to occasional explicit input and then approximately following the on-touchscreen gestures suggested by Arif et al. [2]:

- Space input: short above surface gesture – e.g. right to left swipe.
- Shift: short above surface gesture – e.g. a vertical up swipe
- Enter: a gesture synonymous with hiding the keyboard or moving down – e.g. a diagonal down-left stroke or maybe fast lift of wrist from screen
- Punctuation input: comma and period should be on alphabetic keyboard
- Number, symbol and Emoji input: non-short above surface gesture to change keyboard mode (not covered in [2]) – e.g. a circle gesture

For editing it is harder to automate as users are typically correcting either their or the language model's mistakes. We suggest more reliance on word level interaction, or smart editing (e.g. [1]) with character level interaction limited

- Changing suggested word at carat position from word list: Dynamic gesture with on-screen feedback – e.g. wrist rolling movement
- Backspace: probably most used gesture so fast easy – e.g. right-to-left swipe.
- Dynamic carat positioning: Dynamic gesture with on-screen feedback – e.g. two finger 2D direct manipulation (mouse style) movement
- Precise carat positioning: rarely supported so omit or modally overload space/backspace.

Cross Device Standardisation

There is very little cross device standardisation for gestural interaction. We would also like to explore the same configuration on a range of standard (e.g. mechanical QWERTY keyboard) and limited input surface scenarios, or where entry is hindered by the physical properties of the input device, for example metal weather and vandal resistant keyboards on ATMs keyboards or pervasive displays. We hope that at least some functions (e.g. delete) are ecologically valid across a range of ubiquitous computing devices and might benefit from a set of universal gestures that can be easily learned and intuitively used, leveraging from the proliferation of sensor-enabled wearable devices (e.g. smartwatches, smart clothing, smart jewellery etc).

References

1. Ahmed Sabbir Arif, Sunjun Kim, Wolfgang Stuerzlinger, Geehyuk Lee, and Ali Mazalek. 2016. Evaluation of a Smart-Restorable Backspace Technique to Facilitate Text Entry Error Correction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (CHI ’16), 5151–5162. https://doi.org/10.1145/2858036.2858407
2. Ahmed Sabbir Arif, Michel Pahud, Ken Hinckley, and Bill Buxton. 2014. Experimental Study of Stroke Shortcuts for a Touchscreen Keyboard with Gesture-redundant Keys Removed. In *Proceedings of*

Graphics Interface 2014 (GI ’14), 43–50. Retrieved from http://dl.acm.org/citation.cfm?id=2619648.2619657

3. Ahmed Sabbir Arif and Wolfgang Stuerzlinger. 2010. Predicting the Cost of Error Correction in Character-based Text Entry Technologies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI ’10), 5–14. https://doi.org/10.1145/1753326.1753329
4. Sangwon Choi, Jaehyun Han, Sunjun Kim, Seongkook Heo, and Geehyuk Lee. 2011. ThickPad: A Hover-tracking Touchpad for a Laptop. In *Proceedings of the 24th Annual ACM Symposium Adjunct on User Interface Software and Technology* (UIST ’11 Adjunct), 15–16. https://doi.org/10.1145/2046396.2046405
5. Matthew JC Crump and Gordon D Logan. 2013. Prevention and correction in post-error performance: An ounce of prevention, a pound of cure. *Journal of Experimental Psychology: General* 142, 3: 692.
6. Mark D Dunlop. 2004. Watch-top text-entry: Can phone-style predictive text-entry work with only 5 buttons? In *Mobile Human-Computer Interaction-MobileHCI 2004*, 342–346.
7. Vittorio Fuccella, Poika Isokoski, and Benoit Martin. 2013. Gestures and Widgets: Performance in Text Editing on Multi-touch Capable Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI ’13), 2785–2794. https://doi.org/10.1145/2470654.2481385
8. Mitchell Gordon, Tom Ouyang, and Shumin Zhai. 2016. WatchWriter: tap and gesture typing on a smartwatch miniature keyboard with statistical decoding. 3817–3821.
9. Andreas Komninos and Mark Dunlop. 2014. Text input on a smart watch. *Pervasive Computing, IEEE* 13, 4: 50–58.
10. Luis A Leiva, Alireza Sahami, Alejandro Catalá, Niels Henze, and Albrecht Schmidt. 2015. Text Entry on Tiny QWERTY Soft Keyboards. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 669–678.
11. Anders Markussen, Mikkel Rønne Jakobsen, and Kasper Hornb\a ek. 2014. Vulture: A Mid-air Word-gesture Keyboard. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems* (CHI ’14), 1073–1082. https://doi.org/10.1145/2556288.2556964
12. Adam Nowosielski. 2017. Swipe-Like Text Entry by Head Movements and a Single Row Keyboard. In *Image Processing and Communications Challenges 8: 8th International Conference, IP&C 2016 Bydgoszcz, Poland, September 2016 Proceedings*, Ryszard S. Choraś (ed.). Springer International Publishing, Cham, 136–143. Retrieved from http://dx.doi.org/10.1007/978-3-319-47274-4_16
13. Stephen Oney, Chris Harrison, Amy Ogan, and Jason Wiese. 2013. ZoomBoard: A Diminutive Qwerty Soft Keyboard Using Iterative Zooming for Ultra-small Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI ’13), 2799–2802. https://doi.org/10.1145/2470654.2481387
14. Tim Paek, Kenghao Chang, Itai Almog, Eric Badger, and Tirthankar Sengupta. 2010. A practical examination of multimodal feedback and guidance signals for mobile touchscreen keyboards. 365–368.
15. Henning Pohl, Christian Domin, and Michael Rohs. 2017. Beyond Just Text: Semantic Emoji Similarity Modeling to Support Expressive Communication

👫➔📱😄. *ACM Trans. Comput.-Hum. Interact.* 24, 1: 6:1–6:42. https://doi.org/10.1145/3039685

16. Jean-Baptiste Scheibel, Cyril Pierson, Benoît Martin, Nathan Godard, Vittorio Fuccella, and Poika Isokoski. 2013. Virtual Stick in Caret Positioning on Touch Screens. In *Proceedings of the 25th Conference on L'Interaction Homme-Machine* (IHM '13), 107:107–107:114. https://doi.org/10.1145/2534903.2534918
17. Andrew Sears and Ying Zha. 2003. Data entry for mobile devices using soft keyboards: Understanding the effects of keyboard size and user tasks. *International Journal of Human-Computer Interaction* 16, 2: 163–184.
18. Katie A. Siek, Yvonne Rogers, and Kay H. Connelly. 2005. Fat Finger Worries: How Older and Younger Users Physically Interact with PDAs. In *Proceedings of the 2005 IFIP TC13 International Conference on Human-Computer Interaction* (INTERACT'05), 267–280. https://doi.org/10.1007/11555261_24
19. Keith Vertanen, Haythem Memmi, Justin Emge, Shyam Reyal, and Per Ola Kristensson. 2015. VelociTap: Investigating Fast Mobile Text Entry Using Sentence-Based Decoding of Touchscreen Keyboard Input. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (CHI '15), 659–668. https://doi.org/10.1145/2702123.2702135
20. Mateusz Wierzchowski and Adam Nowosielski. 2016. Swipe Text Input for Touchless Interfaces. In *Proceedings of the 9th International Conference on Computer Recognition Systems CORES 2015*, Robert Burduk, Konrad Jackowski, Marek Kurzyński, Michał Woźniak and Andrzej Żołnierek (eds.). Springer International Publishing, Cham, 619–629. Retrieved from http://dx.doi.org/10.1007/978-3-319-26227-7_58
21. John Williamson. 2016. Fingers of a Hand Oscillate Together: Phase Syncronisation of Tremor in Hover Touch Sensing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (CHI '16), 3433–3437. https://doi.org/10.1145/2858036.2858235
22. Shumin Zhai and Per Ola Kristensson. 2012. The word-gesture keyboard: reimagining keyboard interaction. *Communications of the ACM* 55, 9: 91–101.