Interacting with Computers Advance Access published May 4, 2016

doi:10.1093/iwc/iww003

Comparison of Three QWERTY Keyboards for a Smartwatch

Jonggi Hong[1], Seongkook Heo[1], Poika Isokoski[2] and Geehyuk Lee[1]*

[1]*KAIST, Computer Science Department, 291 Daehakro, Yuseong, Daejeon 305-701, South Korea*
[2]*University of Tampere, School of Information Sciences, Tampere, Finland*
**Corresponding author: geehyuk@gmail.com*

The QWERTY keyboard has been a *de facto* standard for computer text entry and continues to be one for mobile text entry such as for smartphones. It is not clear, however, that it will continue to be an option for text entry for much smaller devices such as smartwatches. In a series of user experiments, we examined the performance of the QWERTY keyboard when it is reduced to fit a small smartwatch screen. At the same time, we examined whether the ZoomBoard and the SplitBoard, which are QWERTY keyboards augmented by zooming and panning strategies, respectively, would be effective in comparison with a plain QWERTY keyboard. In Experiment 1, we evaluated the text entry performance of new users on the three QWERTY keyboards. In Experiment 2, we evaluated the relative performance of the three keyboards for three different screen sizes. In Experiment 3, we further observed how the keyboard performance changed when used in a mobile situation. Main results are: (i) users could adapt to a plain QWERTY keyboard even in the smallest screen cases. (ii) The SplitBoard consistently showed a better performance than other keyboards in all tested sizes. (iii) The SplitBoard showed a better performance than other keyboards in a mobile condition (treadmill) and was preferred most by participants.

RESEARCH HIGHLIGHTS

- We examined the performances of the ZoomBoard and the SplitBoard in comparison with a plain QWERTY keyboard for three different screen size conditions and for static and mobile usage conditions.
- The SplitBoard showed a better performance than the other keyboards consistently in all conditions and was preferred most by the participants.
- In addition, we could observe that the participants could adapt to a plain QWERTY keyboard surprisingly well even in the smallest screen case.

Keywords: QWERTY keyboards; ZoomBoard; SplitBoard; smartwatch text entry

Editorial Board Member: Javier Bargas-Avila

Received 27 May 2015; revised 18 February 2016; accepted 22 February 2016

1. INTRODUCTION

Diverse smartwatches are being introduced to the market these days.[1] Whereas the name smartwatch appeared only recently, its concept has a longer history. Notable examples of early 'smart' watches are GNU/Linux wristwatch (Mann, 2000) and IBM wristwatch computer (Raghunath and Narayanaswami, 2002). Unlike, these early smartwatches, modern smartwatches have computing power and battery life comparable with that of early mobile phones, and are evolving steadily into a computing platform for mobile applications migrating from smartphones.[2] One of the hurdles to this migration seems to be the small form factor of a smartwatch that makes it difficult to accommodate the GUI for the smartphone applications. Many researchers are now proposing new user interfaces and interaction techniques to overcome the user interface problems of smartwatches.

[1]Smartwatch Database, http://www.smartwatchgroup.com/smartwatch-database/.

[2]Smartwatch Applications, http://www.smartwatchgroup.com/smartwatch-applications/.

Among the many user interface challenges of a smartwatch, our focus in this paper is on text entry. Simple smartwatches, such as sport smartwatches and health-care smartwatches, may not require text entry, but general-purpose smartwatches are likely to require text entry as they are expected to become a platform for mobile applications migrating from smartphones, such as sending a short message. There are many different options for text entry on a smartwatch, including soft keyboards, handwriting input, speech input and using special hardware. We are yet to see which will become the main stream for smartwatches, but our focus in this paper is on soft keyboards. To design a soft keyboard appropriate for a smartwatch, we may consider the following two approaches: (i) inventing a new specialized design for the small form factor of a smartwatch (Partridge *et al.*, 2002; Raghunath and Narayanaswami, 2002; Dunlop *et al.*, 2014)[3] and (ii) carrying over the QWERTY keyboard which is the *de facto* standard text-entry device for a computer (Oney *et al.*, 2013; Chen *et al.*, 2014; Hong *et al.*, 2015).

The first approach may be preferred because a designer will then be able to consider more possibilities without being constrained by existing designs or practices. The problem of the first approach, however, is that users have to learn a new skill. The history of mobile text entry methods and their adoption in fact testifies the importance of the initial 'pick-up-and-use usability' of mobile text entry methods (Dunlop and Masters, 2009). In particular, smartwatch users are not expected to do massive text entry on a smartwatch, and therefore it is difficult to expect that they will invest time to learn a new text entry skill for a smartwatch. In this respect, we considered the second approach to be more practical; it carries over the familiar QWERTY keyboard and will not require users to learn a new skill.

An obvious question at this point is if users would be able to use a QWERTY keyboard on the small screen of a smartwatch. The size of a letter key when the QWERTY layout is fitted into a $1'' \times 1''$ smartwatch screen will be about $2\,\text{mm} \times 2\,\text{mm}$, which seems too small for a finger. This was the first question that motivated us to start the current study. A natural next question was what would be the best strategy to help users use a QWERTY keyboard. We could see the following two strategies: zooming and scrolling. In the first approach, a user touches a region on a QWERTY keyboard to magnify it and then touch a target key to enter a letter. This strategy was investigated by Oney *et al.* (2013). They verified that this strategy was effective for a very small device. The second strategy is to use an already magnified view of a QWERTY keyboard and let the user to scroll it to find and hit a target key. This approach was shown to be effective by Hong *et al.* (2015). Scrolling in this case may be continuous (Cha *et al.*, 2015) or discrete (Hong *et al.*, 2015), but the latter choice showed a better performance. A discrete scrolling operation in this case was actually a simple switching operation between the left and the right halves of a QWERTY keyboard.

[3]Minuum Keyboard, http://minuum.com.

The goal of the current study is to answer the following three questions. (i) Will people be able to use a small QWERTY keyboard on a smartwatch? (ii) When and how much will the zooming strategy benefit users? (iii) When and how much will the scrolling strategy benefit users? In order to answer these questions, we conducted a series of user experiments with the following three kinds of QWERTY keyboards: an ordinary QWERTY keyboard (the QWERTY), a zooming QWERTY keyboard (the ZoomBoard), and a scrolling QWERTY keyboard (the SplitBoard). As users are not expected to do massive text entry on a smartwatch, we expected that initial learnability would be the most important requirements for a smartphone text entry method. Therefore, we compared the initial learnability of the three keyboards in the first experiment (also reported by Hong *et al.*, 2015).

The results of the first experiment prompted us to think that the effectiveness of the different strategies may depend on the screen size. Therefore, we designed and conducted the second experiment where we compared the performances of the three keyboards for different screen sizes. In the second experiment, we had a chance to observe the relatively long-term learnability of the three methods because participants had to spend more time with the same keyboard design for different screen sizes. An unresolved question after the two experiments was how the efficiency of the text entry methods would change in a mobile situation, and we designed and conducted the third experiment, where we studied how their performances depend on the usage environment: a static setting vs. a mobile setting.

2. RELATED WORK

2.1. General considerations in wrist watch text entry

One could potentially implement text entry on a wrist device using speech recognition, handwriting recognition, using a separate device such as a Bluetooth keyboard, using the skin near the watch as input surface (Weigel *et al.*, 2014), employing the wristband as a keyboard (Perrault *et al.*, 2013; Ahn *et al.*, 2015; Funk *et al.*, 2014) or perhaps through a few buttons in the watch frame and an encoding scheme such as MDITIM (Isokoski and Raisamo, 2000) or EdgeWrite (Wobbrock *et al.*, 2003). Physical miniature keypads have also been seen on many electronic wristwatches and watch phones. We anticipate that many technologies may be in use in the same device as we have seen on smartphones. Some users utilize the speech-based services while others rely on touchscreen interaction exclusively.

In the introduction we followed from Dunlop and Masters (2009) and argued that minimizing learning effort has led to the QWERTY soft keyboard being the dominant text entry method on smartphones with touchscreens. The same reasons that have driven phone manufacturers to touchscreens on smartphones are working on the watch platform. The capacitive touchscreen has no moving parts or seams that can leak water or gather dirt. No external pointing devices are needed. It can sustain a reasonable amount of pressure and creates no input unless touched by an object with finger-like capacitance. Minimalist uncluttered design of the device body is easier to do with a touchscreen than with physical keyboards.

Taking the touchscreen as a starting point, we will focus on soft keyboards because they are the dominant text entry technique on bigger touchscreen devices and a likely starting point for the design of a successful text entry technique on a wrist watch also.

2.2. Soft keyboards in wrist device text entry

Many soft keyboards have been developed for text entry in mobile devices. We will discuss the trade-off between key size and key ambiguity, and describe work on zooming keyboards.

2.2.1. The keysize-ambiguity trade-off

A central theme in miniature soft keyboards is the balance between the number of keys and the ambiguity of a key press (MacKenzie, 2002). With fewer keys an individual key can be made bigger. However, then each key is mapped to more than one character requiring disambiguation.

The multi-tap text entry technique for 12-key telephone keypads was a widespread early example of explicit user-initiated key disambiguation. In multi-tap each key is associated with multiple characters. The first press on the key shows the first character. It will be entered if the user presses a different key or does not press any key during a time-out period of 0.5–2 s (varies between implementations). If the user presses the same key again during the timeout period, the next character on that key will be shown etc.

The English language multi-tap requires on average over two key presses for each character (MacKenzie, 2002). Multi-tap can be implemented on any number of keys. With one key the ambiguity is at its maximum and many key presses are needed to enter a character on average. In addition to the number of keys, the Keystrokes Per Character (KSPC) value depends on the order of the characters within each key. For optimal KSPC more frequent characters should be associated with fewer key presses.

The difficulty in optimizing the character placement is maintaining a layout that is easy for new users to adopt. The Less-Tap by Pavlovych and Stuerzlinger (2003) and the QWERTY-like keypad by Hwang and Lee (2005) offered layouts that were familiar to new users. In the QWERTY-like keypad the layout mimicked the QWERTY layout for placement of characters on keys and character frequency guided the ordering within keys. An experimental evaluation showed that the QWERTY-like keypad was superior to the standard telephone character layout (Hwang and Lee, 2005). In the Less-Tap system the assignment of characters to keys was alphabetical (i.e. the same as in multi-tap) and the order of characters within a key was again decided based on character frequency. In an experiment Less-Tap too outperformed multi-tap (Pavlovych and Stuerzlinger, 2003).

In dictionary-based disambiguation the computer matches the key sequence to known words. Often there is only one match or the most probable match is the desired word. In such cases no further input is needed. In other cases the user needs to select from a list of matching words. The 12-key telephone keypad dictionary-based disambiguation brings KSPC down to about 1 (MacKenzie, 2002). The T9 system (James and Reischel, 2001) was the most widely used implementation on keypad-based mobile phones. Similar systems have been seen on touchscreen devices. The 1Line keyboard is an example of a disambiguation systems with touchscreen specific features such as bezel tap and finger gestures. It combines the three rows of the QWERTY layout into one and then utilizes a disambiguation to figure out which word the user intended to write (Li *et al.*, 2011). The Minuum[4] keyboard is a product that utilizes many of the ideas in the 1Line keyboard design.

2.2.2. Miniature QWERTY keyboards, zooming and scrolling keyboards

A two-step selection technique where the first pointer location is used as the center of the region to zoom in and the second location makes the actual selection is a well-known technique in eye-pointing (Bates and Istance, 2002) where it is used to overcome the noise in the tracker signal. A similar technique, called ZoomBoard, for finger-pointing on small soft keyboards was proposed by Oney *et al.* (2013). In an evaluation the ZoomBoard outperformed a QWERTY keyboard. However, these results were measured on a very small display (16 mm × 6 mm). Typical wristwatch displays are larger than this.

Inspired by the ZoomBoard, Leiva *et al.* (2015) studied designs for small QWERTY keyboards utilizing call-out and shift techniques. In call-out a small indicator pops up next to the finger tip showing the touched key. In shift, the indicator shows the region under the finger. Leiva *et al.* added zooming to shift to make the tiny keys easier to see. The ZoomBoard was included in the evaluation for comparison. All keyboards were tested in three sizes ranging from 16 mm × 6.5 mm to 28.4 mm × 11.4 mm. Key sizes for the alphabetic keys ranged from 1.5 to 2.6 mm. In terms of text entry rate the ZoomBoard outperformed the others in the smallest keyboard size reaching 6.0 wpm. At larger sizes the differences in text entry rates were not statistically significant. The study did not include an unaltered QWERTY keyboard for comparison.

As an alternative to zooming Cha *et al.* (2015) utilized panning a keyboard that was too big to fit on the wristwatch screen. The panning was done with the usual dragging gestures

[4] Minuum Keyboard, http://minuum.com.

utilizing different control-display gains. When the area with the desired key was visible on screen, the character was entered by tapping as usual. Cha *et al.* evaluated their design in a simulation on a tablet computer. They found that the average text entry rate across their conditions was 10.9 wpm.[5] They suspect that the improvement in text entry rate over the ZoomBoard was due to the fact that sometimes several characters could be entered before the keyboard needed to be moved. In the ZoomBoard the zooming tap was needed for each character.

2.3. Small screen tapping and pointing precision

Pointing has been studied extensively, but most of the work did not involve direct finger pointing. Instead mouse or stylus has often been used. In the wristwatch touchscreen text entry situation we are interested in one-finger pointing at small targets (<4 mm diameter) on modern capacitive touchscreens.

Often cited early work on touchscreen pointing performance was done by Sears *et al.* in the early 1990s. Touchscreen technology has changed but these results still serve as lower-bound measurements of the performance of touchscreen pointing. Reviewing even earlier work, Sears (1991) found that the side of a square target needed to be at least 20 mm. Sears' own measurements led to recommending 22.7 mm key size.

While finger tapping accuracy in Sear's study was low, precision of the sensing technology was good. With appropriate feedback (lift-off strategy) targets as small as 1.7 mm × 2.2 mm (Sears and Shneiderman, 1991) could be selected. The actual positioning accuracy of a finger has been measured to be an order of magnitude better. Using close-up video-based measurement 0.17 mm was found as the limit in precision of the finger using rolling gestures (Bérard and Rochet-Capellan, 2012). The notable difference between the 20 mm key size recommendation and the 0.17 mm finger positioning accuracy highlights the difference that continuous feedback has in improving finger pointing accuracy. In a touch-typing situation, however, both lift-off and finger rolling with continuous visual feedback are slow in comparison to tapping.

Later in an experiment with an information kiosk with a touchscreen, 10 mm keys were found sufficient for entering single characters (Colle and Hiszem, 2004). When entering multiple characters error rates rose so that 20 mm keys were needed. Relatively small keys were also found usable in measurements that involved mixed forefinger and thumb tapping. The conclusion was that targets larger than 8 mm can be reliably selected on a smart phone-like handheld touchscreen (Sheik-Nainar, 2010). A similar key size of 9.2–9.6 mm was recommended by Parhi *et al.* (2006) for one-handed thumb tapping on a mobile device.

Even with modern smart phone touchscreens the results seem to depend on the experimental task. Henze *et al.* (2011) collected data on 100 000 000 taps in a circle-tapping game distributed through Google's application market. In their data, the 8 mm targets that Sheik–Nainar found acceptable were missed in over 20% of cases. Such discrepancy makes us suspect that people are good at adapting to strict accuracy requirements, but only when motivated to do so. Presumably, the circle tapping game where most targets were larger did not motivate similar accuracy as Sheik–Nanair's experiment where 8 mm was the largest target size.

Azenkot and Zhai (2012) reported text entry rates on a QWERTY soft keyboard with 6 mm key width. They measured 36 wpm for one finger, 34 wpm for one thumb, and 50 wpm for two thumbs. Error rates were 8, 7 and 11%, respectively. The high text entry rates and the high error rates are a result of not allowing error correction during writing. However, these results suggest that smartphone users today perform much better than the old touchscreen tapping results predicted.

The evaluation of the ZoomBoard (Oney *et al.*, 2013) offers interesting results on finger tapping accuracy. The display area used for the ZoomBoard was 16.5 mm × 6.1 mm. The unzoomed keys were 1.5 mm × 1.5 mm in size. When zoomed in, the ZoomBoard keys were 4.4 mm × 4.4 mm. In a study with the ZoomBoard and the ordinary one-step QWERTY text entry on the same 16.5 mm × 6.1 mm area the 1.5 mm keys on the QWERTY keyboard were found too small. KSPC was 2.88 (vs. optimal at 1) and 14% of phrases were erroneous after corrections. The ZoomBoard was found usable suggesting that 4.4 mm keys were big enough for text entry. However, the KSPC metric at 2.15 (vs. optimal performance at 1.84) suggest that the ZoomBoard use was not completely error free. The number of uncorrected errors for the ZoomBoard was insignificant at 1 error per 1430 characters. Text entry rate for the ZoomBoard was 9.3 wpm and the text entry rate for the QWERTY keyboard was 4.5 wpm.

A number of studies (Henze *et al.*, 2011, 2012; Sears, 1991) have shown that the touch points on touchscreens tend to be systematically shifted away from the center of the targets. Depending on the task and user's grip on the device the orientation and magnitude of the shift vary (Henze *et al.*, 2011). On extremely small keyboards the shift is of particular interest because the error margin is so small.

Given the previous result that targets smaller than 20 mm cause difficulties for users and also negative reactions by the users, it is rather surprising how well small (<10 mm) QWERTY soft keyboards have done on smart phones. It seems that the discomfort caused by the small keys was tolerable given the other good properties of touchscreen phones. There may also be a skill component involved. When people are better aware of how touchscreens work, they may be able to compensate for the typical problems.

We now face a similar threshold when considering wristwatch text entry. It seems impossible that the QWERTY

[5]Cha *et al.* report 11.9 wpm. However, they computed their words per minute metric using 4.58 as the divisor instead of the usual 5.

Figure 1. Casio Databank wrist computer (original image by cinsky: https://flic.kr/p/5M9LAw).

keyboard would be usable if further shrunk to the wristwatch sized displays. Is this really so? The work by Leiva *et al.* (2015) suggests that the motor component of hitting very small keys may not be too difficult when aiming is aided by a callout. Also, thanks to the intelligent touch event decoder, the QWERTY keyboard-based VelociTap system was operable even in a tiny 2.5 cm wide configuration (Vertanen *et al.*, 2015). Later in this paper we will present similar results for the QWERTY keyboard without intelligent decoding.

2.4. Earlier wristwatch text entry designs

Studies on wristwatch text entry with touchscreen input remain rare. Most studies (including those cited above) that aim for relevance on wristwatch sized devices were completed using other platforms. For example the ZoomBoard study was done on an iPad (Oney *et al.*, 2013) and Leiva *et al.* (2015) and Vertanen *et al.* (2015) used smartphones.

Wrist-top computers have a history that reaches to the 1970s when electronic wristwatches with calculator functions were introduced. The Casio Databank model shown in Fig. 1 also included text entry capabilities. However, the user interfaces were based on buttons.

Partridge *et al.* (2002) built their own wristwatch-sized prototype for investigating a text entry concept that they named TiltType. Only anecdotal usage data were reported based on informal demonstrations of the device.

Dunlop *et al.* (2014) designed a 7-key text entry technique for the Sony Smartwatch 2. It had an alphabetic character arrangement with three ambiguous keys at the top and three at the bottom. A big space key in the middle also showed the entered text. A later paper (Komninos and Dunlop, 2014) also gives performance metrics. The average text entry rate was 8.1 wpm.

In an earlier work, we compared the QWERTY and the ZoomBoard with a system of our own design called SplitBoard (Hong *et al.*, 2015) and also in a second experiment to a tap-slide keyboard that we called SlideBoard and the QWERTY-like keypad (Hwang and Lee, 2005) described above. The performance of the SlideBoard and the QWERTY-like keypad paralleled the QWERTY and the ZoomBoard. Thus they are no longer discussed in the further work in this paper.

3. THREE QWERTY KEYBOARDS

In this section, we describe the three QWERTY keyboards that we compared in the current study: an ordinary QWERTY keyboard (the QWERTY), a zooming QWERTY keyboard (the ZoomBoard) and a scrolling QWERTY keyboard (the SplitBoard). The ZoomBoard and the SplitBoard are defined and described in earlier publications (Oney *et al.*, 2013; Hong *et al.*, 2015), but we will describe them again in this section for the sake of completeness. In addition, we will provide the design details specific to our prototypes of the three keyboards that we implemented for our experiments. The designs of the three keyboards vary slightly in the three experiments; we will explain the experiment-specific differences in the corresponding sections later.

Most touchscreen keyboards use a language model for input disambiguation and error correction. The use of a language model is expected to be effective for all of the three QWERTY keyboards. However, our goal in this study was not to design the best keyboard product but studying the effect of different key selection strategies on text entry efficiency. Therefore, we did not use language models. Instead we focused on the *raw* key entry efficiency. The effectiveness of a language model is likely to be different for different keyboards. Finding the most effective way of using a language model for each keyboard will be an important research problem when the interaction techniques investigated in this paper are shown to be effective.

3.1. The QWERTY

In this paper, we use the name ‘the QWERTY’ to specifically refer to an ordinary QWERTY keyboard fitted in the small touchscreen of a smartwatch. Figure 2 explains the design of the QWERTY and a task display. The upper screen area marked with an orange rectangle is a display area, which is used for the display of a sentence to enter and the visual feedback of an entered sentence in our experiments. All remaining screen space below the orange box is used for the QWERTY. As the picture illustrates, the alphabetic key part of the QWERTY layout is anisotropically scaled to fit the area leading to non-square keys. The enter and the backspace keys are moved to

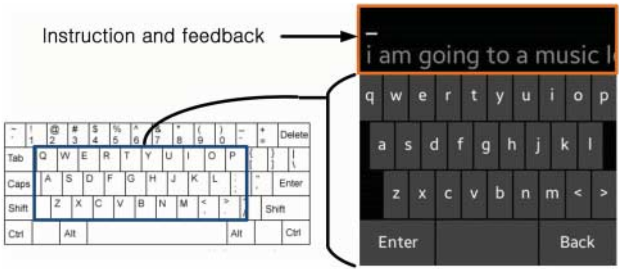

Figure 2. The QWERTY.

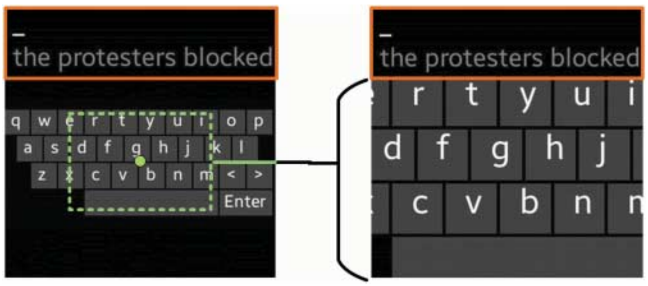

Figure 3. The ZoomBoard: (left) before zooming and (right) after zooming.

the bottom left and bottom right corners, respectively. A take-off selection strategy is used, i.e. a key entry is triggered when the finger is lifted.

3.2. ZoomBoard

The ZoomBoard uses a zooming strategy to give access to a large QWERTY layout on a small touchscreen. It enables users to easily type a key by using two-stage tap input; the first tap enlarges the tapped area and the second tap triggers input. Using this strategy, users can type more comfortably with enlarged keys while utilizing their knowledge of the QWERTY layout. However, users always have to perform two taps to input one character. Both taps need to be visually guided.

Figure 3 shows the ZoomBoard prototype that we implemented for our experiments. As in the case of the QWERTY, the upper screen area marked with an orange rectangle is used for instruction and feedback and the remaining area under it is used for the ZoomBoard. Initially, the keyboard area shows a QWERTY keyboard. The size of the QWERTY keyboard is limited by the width of the screen. We do not stretch the QWERTY keyboard anisotropically to fill the available area because we believe it to be more important to maintain the original aspect ratio than to maximize the key size in the unzoomed state. This is in accordance with the original design of the ZoomBoard by Oney *et al.* (2013). When a user touches a region in the QWERTY keyboard, it is magnified and centered at the touch point.

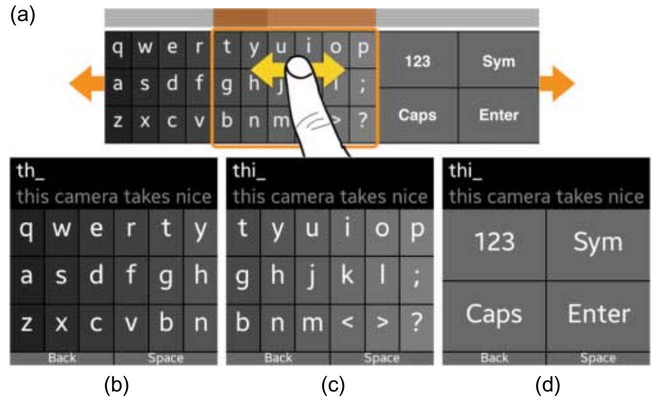

Figure 4. The SplitBoard: (a) the concept of the SplitBoard and (b–d) the three parts of the SplitBoard. Flicking left or right on the screen, users can switch among the three parts.

The ZoomBoard uses left and right swipe gestures as a backspace and a space instead of tapping keys. Therefore, the ZoomBoard does not have a backspace key. However, the space key is retained like in the original ZoomBoard paper (Oney *et al.*, 2013). As in the case of the QWERTY, a take-off selection strategy is used, i.e. zooming or a key entry is triggered when the finger is lifted.

3.3. SplitBoard

The SplitBoard uses a scrolling strategy to give access to a large QWERTY layout on a small touchscreen. As shown in Fig. 4, the SplitBoard consisted of three parts: the first two are the two halves of the QWERTY layout and the third part is for function keys. Users switch among the three parts by a flick gesture as if scrolling the three parts through the screen.

Text is entered by tapping on the keys. The space bar and the backspace key are located at the bottom of the screen. These two keys are present in all parts so that a user can access them easily at any time. Though the space bar and the backspace key are very thin, they are easy to select because they can be selected by touching the bezel at the bottom of the screen.

Each of the first two parts has 3×6 alphabet keys. As shown in Fig. 4, the two center columns of the QWERTY layout appear in both parts. This overlap reduces the number of switches between the two parts. The optimal number of overlapping columns is a trade-off between a key size and the number of switches between the two parts. On a smaller screen, a key size may be more important factor and less overlap may be favored. On a larger screen, a key size may be less important and more overlap may be favored. According to Hong *et al.* (2015), the number of gestures per character (GPC) changes as a function of the number of the overlapping columns; the GPC in the case of two overlapping columns is 1.28. As in the case of the QWERTY, a take-off selection strategy is used.

Figure 5. The smartwatch used in Experiment 1.

4. EXPERIMENT 1: NOVICE TYPING PERFORMANCE

The goal of Experiment 1 was to compare the novice performance of the three keyboards.[6] We chose to measure novice performance because it is critical for early user experience and because smartwatch users are not expected to perform massive text entry. In addition, we expected that novice performance would not be very different from expert performance because all of the three keyboards use the familiar QWERTY layout and, therefore, have good immediate usability.

4.1. Participants

We recruited 12 participants from our university (3 females, average age of 22.4 (SD $=$ 4.2)). The participants were not native English speakers, but were using English in class every day. They were familiar with touchscreen interfaces because they all were smartphone users. However, most of them did not have a prior experience of using a smartwatch (11 negative, 1 unknown).

4.2. Apparatus

The three soft keyboard designs described in the previous section were used on a Samsung Galaxy Gear smartwatch shown in Fig. 5. The size of the display is 29.3 mm × 29.3 mm, and the size of the upper area for instruction and feedback is 29.3 mm × 8 mm.

The sizes of the alphabetic keys in the QWERTY, the ZoomBoard and the SplitBoard are shown in Table 1. When the ZoomBoard is zoomed in, the width and height of the keys are two times the initial width and height. The size of space and backspace keys in the SplitBoard is 14.5 mm × 1.8 mm. Despite the thin size, participants could use the space and backspace keys easily because the keys are at the edge of the screen, thereby having larger effective touch areas.

[6]This experiment is one of the two experiments reported by Hong *et al.* (2015).

Table 1. The sizes of alphabetic keys in the three keyboards in Experiment 1.

	ZoomBoard (mm)		
QWERTY (mm)	Unzoomed	Zoomed	SplitBorad (mm)
2.9 × 5.3	2.9 × 2.9	5.8 × 5.8	4.8 × 6.5

4.3. Procedure

Participants sat on a chair in a quiet room and rested their arms on the desk. They were allowed to move their arms close to their face if they wanted to see the small keyboards better. The smartwatch was worn on the non-dominant wrist. The index finger of the dominant hand was used to operate the keyboards. Participants were asked to transcribe a given phrase as fast and accurate as possible. Phrases to transcribe were selected randomly from the 500 phrase set by MacKenzie and Soukoreff (2003). All non-alphabet characters except the space were removed from the phrases.

In each block, participants used the three keyboards in a counterbalanced order. In the first block, the experimenter gave instructions on how to use the keyboards at the beginning of using each keyboard. Participants transcribed one phrase as a practice before starting real trials. Working with a small display for an hour or so is tiring, and therefore the participants were given 3 min breaks between blocks. Participants were allowed to skip the breaks if they wished so.

After finishing all blocks, participants were asked to answer a questionnaire. The questionnaire asked them to rank the three keyboards in the order of preference (without any further instruction), and provided spaces for their thoughts on the pros and cons of the three keyboards.

4.4. Design

The independent variables were *Keyboard* and *Block*. The experiment was a 3 (*Keyboard*) × 5 (*Block*) within-subject factorial design. Participants repeated five blocks, and in each block they transcribed five phrases with each of the three keyboards. The order of the three keyboards in a block was fully counterbalanced. We used common metrics for text entry research suggested by Soukoreff and MacKenzie (2003): words per minute (WPM), total error rate (TER) and uncorrected error rate (UER). We measured them for each block.

4.5. Results

In this experiment, as well as in the next two experiments, we used the non-parametric test procedure with Aligned Rank Transform (ART) detailed in Wobbrock *et al.* (2011) for comparing means for both WPM and TER. A non-parametric method was chosen because some parts of WPM and TER data

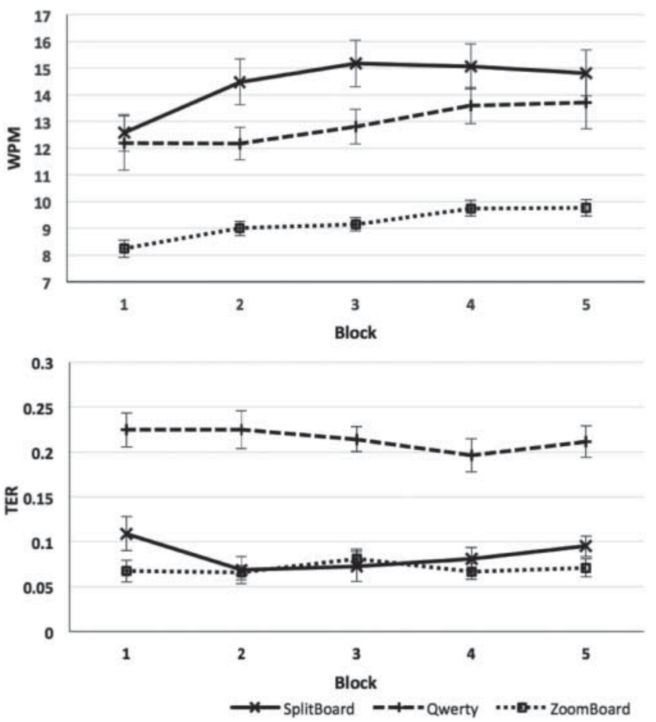

Figure 6. The WPMs and TERs of the QWERTY, the ZoomBoard and the SplitBoard in Experiment 1.

did not satisfy the normality condition and also because of the relatively small numbers of participants.

The first graph in Fig. 6 shows the WPMs of the three keyboards over the five blocks. The SplitBoard was the fastest. The WPM of the SplitBoard was 12.6 wpm in Block 1 and increased to 14.8 wpm in Block 5. The WPM of the QWERTY was 12.1 wpm in Block 1 and 13.7 wpm in Block 5. The ZoomBoard started with 8.2 wpm and ended with 9.8 wpm. A repeated measures ANOVA after ART (3 keyboards × 5 blocks) showed a significant main effect of *Keyboard* on WPM ($F(2, 22) = 212.12$, $P < 0.001$, $\eta_p^2 = 0.63$). Pairwise comparisons with Šidák correction showed that the SplitBoard was faster than the ZoomBoard ($P < 0.001$) and the QWERTY ($P < 0.001$), and the QWERTY was faster than the ZoomBoard ($P < 0.001$).

The second graph in Fig. 6 shows the TERs of the three keyboards over the five blocks. The TERs of three keyboards did not change significantly between blocks. The ZoomBoard was the most accurate keyboard, having the TER of 0.07. The TER of the QWERTY was the highest at 0.21 on average over the five blocks. The average TER of the SplitBoard over the five blocks was 0.09. A repeated measures ANOVA after ART (3 keyboards × 5 blocks) showed that *Keyboard* had a significant main effect on TER ($F(2, 22) = 155.70$, $P < 0.001$, $\eta_p^2 = 0.70$). Pairwise comparisons with Šidák showed that the QWERTY had a higher TER than the ZoomBoard ($P < 0.001$) and the SplitBoard ($P < 0.001$). The TERs of the ZoomBoard and the SplitBoard were not significantly different ($P = 0.135$).

The average UERs of the SplitBoard, the QWERTY and the ZoomBoard were 0.0058, 0.0058 and 0.003, respectively. All UERs were small enough not to have practical significance.

All participants said that they liked the SplitBoard most. Some participants mentioned that the SplitBoard had a familiar key layout. Some mentioned that the SplitBoard was inconvenient when they need to enter words that require frequent switching between the keyboard sections. Eleven participants disliked the QWERTY most. They said that the keys of the QWERTY were too small to enter text comfortably. Using the small keys might have led to increased user fatigue. There was no significant difference in TER between the SplitBoard and the ZoomBoard, but some participants replied that the ZoomBoard was the most accurate keyboard. On the other hand, some participants replied that the drastic and frequent change of the keyboard scale by zooming in and zooming out was visually tiring. No participant selected the SplitBoard as the least preferred keyboard.

5. EXPERIMENT 2: TYPING PERFORMANCE AND SCREEN SIZE

One of the unexpected results of Experiment 1 was the relatively poor performance of the ZoomBoard contrary to the results of Oney *et al.* (2013). The main difference between the two experiments was the screen size; the screen size in Experiment 1 was almost twice the screen size used by Oney *et al.* (2013). It seemed that the advantage of the ZoomBoard is dependent on the screen size. In fact, it seemed that the relative advantages of the three QWERTY keyboards may all depend on the screen size. This led us to design Experiment 2, where our goal was to measure the relative performances of the three keyboards for three different screen sizes. In addition to this main goal, we expected to have a chance to observe the performances of the three keyboards after more training; participants in this experiment carried out 9 blocks with each keyboard, whereas participants in Experiment 1 carried out only 5 blocks with each keyboard.

5.1. Participants

We recruited 18 participants from our university (7 females, average age of 20 (SD = 2.7)). They were not native English speakers, but were using English in class every day. They were familiar with touchscreen interfaces because they all were smartphone users. However, most of them did not have a prior experience of using a smartwatch (16 negative, 1 positive, 1 unknown).

5.2. Apparatus

The Galaxy Gear smartwatch used in Experiment 1 was used again. Also, the same three soft keyboard designs used in

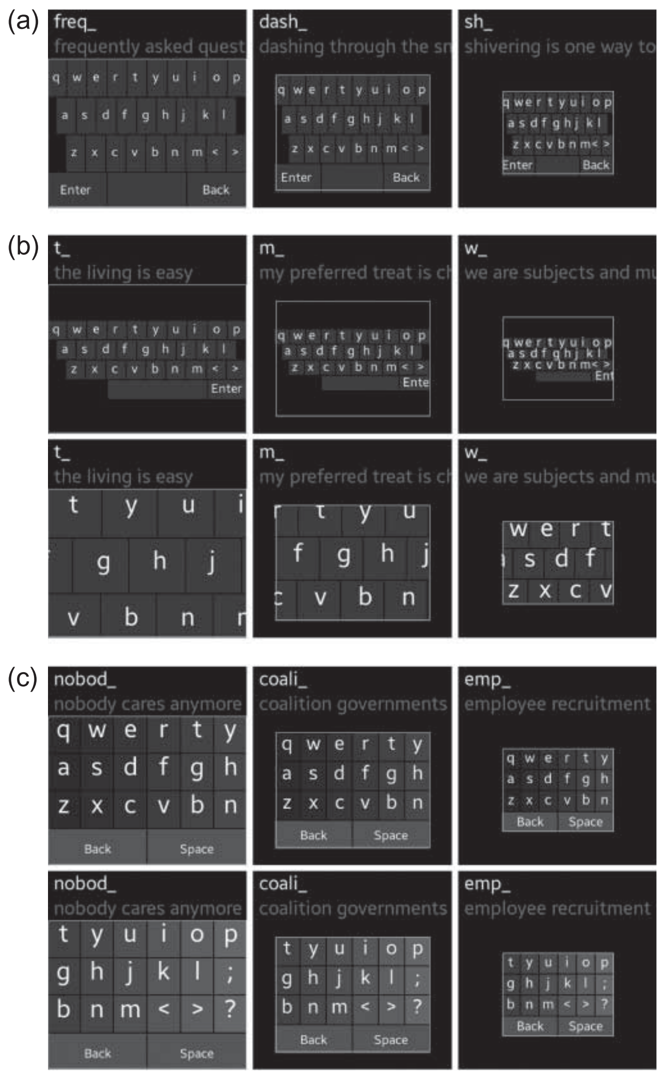

Figure 7. The keyboard designs in the three screen size cases of Experiment 2: (a) the QWERTY designs, (b) the ZoomBoard designs and (c) the SplitBoard designs.

Experiment 1 were used except that the four rows of keys in the SplitBoard were of the same height now. The thin backspace and space were usable in Experiment 1 because they were located along the bezel of the touchscreen. In this experiment, however, as the size of the SplitBoard varied as shown in Fig. 7, the backspace and space keys were not always located at the bezel any more. Therefore, we chose to make the four rows of keys in the SplitBoard of the same height in this experiment.

We selected three different keyboard sizes with the same aspect ratio: 29.3 mm × 21.3 mm (large), 22.9 mm × 16.6 mm (medium) and 16.5 mm × 11.2 mm (small). The biggest size was used by Hong *et al.* (2015) and the smallest size was used by Oney *et al.* (2013). Figure 7 shows the three keyboards in the three different sizes. The sizes of alphabetic keys are shown in Table 2. The function keys (backspace, space and enter keys)

Table 2. The sizes of alphabetic keys in the three keyboards in Experiment 2.

Keyboard size	QWERTY (mm)	ZoomBoard (mm)		SplitBoard (mm)
		Unzoomed	Zoomed	
Small	1.6 × 3.0	1.6 × 1.6	4.6 × 4.6	2.7 × 3.7
Medium	2.3 × 4.1	2.3 × 2.3	6.7 × 6.7	3.8 × 5.1
Large	2.9 × 5.3	2.9 × 2.9	8.5 × 8.5	4.8 × 6.5

located at the bottom of the keyboards have the same height as the alphabetic keys. Their widths are three to five times larger than the alphabetic keys as shown in Fig. 7. When the ZoomBoard is zoomed in, the width and the height of the keys are 2.93 times the initial width and height. This zoom ratio was also used by Oney *et al.* (2013).

5.3. Procedure

The device was worn on the non-dominant wrist. Participants used the index finger of the dominant hand to tap on the keys. Participants were asked to transcribe a given phrase as fast and as accurately as possible. They sat on a chair in a quiet room and rested their arms on the desk. They were allowed to move their arms close to their face if they wanted to see the small keyboards better.

Each participant completed the transcription task for each of the 3 (3 sizes) × 3 (3 keyboards) conditions in a counterbalanced order as described in the next subsection. In each block, participants transcribed 4 phrases from the 500 phrase set by MacKenzie and Soukoreff (2003) using each of the three keyboards (12 phrases in total). Before using each keyboard first time, participants were given an instruction on how to use the keyboard. Participants transcribed one sentence as a practice before they start real trials. The number of phrases for a block was determined so that each participant could finish the whole experiment within one and half hours. They were given 3 min breaks between blocks, but the break was optional as it was in Experiment 1.

After finishing all blocks, participants answered a questionnaire. The questionnaire asked them to rank the three keyboards in the order of preference in each of the three size cases (without any further instruction), and provided spaces for their thoughts on the pros and cons of the three keyboards.

5.4. Design

The independent variables were *Session*, *Size* (keyboard size) and *Keyboard*. The experiment was a 3 (*Session*) × 3 (*Size*) × 3 (*Keyboard*) within-subject factorial design. In other words, the whole experiment consisted of three sessions, and each session consisted of three blocks for the three sizes, and in

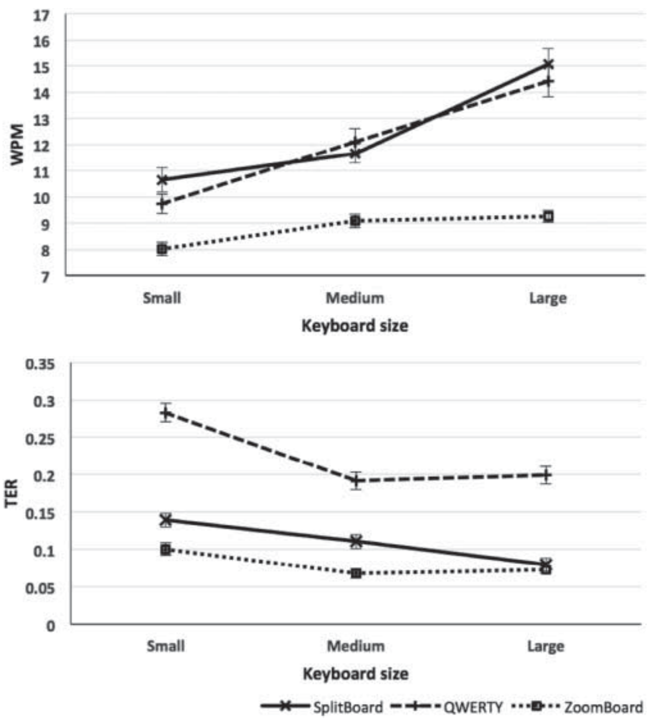

Figure 8. The WPMs and TERs of the three keyboards in the three different size cases of Experiment 2.

Table 3. The average UER of the SplitBoard, the QWERTY, and the ZoomBoard for the three size cases in Experiment 2.

	Small	Medium	Large
QWERTY	0.0107	0.0096	0.0063
ZoomBoard	0.0063	0.0016	0.0029
SplitBoard	0.0028	0.0043	0.0024

Table 4. Participants' preferences over the three keyboards in the three size cases of Experiment 2.

	Small	Medium	Large
Most preferred			
QWERTY	0 (0%)	6 (33%)	7 (39%)
ZoomBoard	13 (72%)	2 (11%)	1 (6%)
SplitBoard	5 (28%)	10 (56%)	10 (56%)
Total	18 (100%)	18 (100%)	18 (100%)
Least preferred			
QWERTY	16 (89%)	9 (50%)	3 (17%)
ZoomBoard	1 (6%)	7 (39%)	14 (78%)
SplitBoard	1 (6%)	2 (11%)	1 (6%)
Total	18 (100%)	18 (100%)	18 (100%)

each block participants used the three keyboards. The order of the keyboard sizes were counterbalanced by the Latin square method, and the order of the three keyboards were fully counterbalanced. As in Experiment 1, we measured WPMs, TERs and UERs for each of the factorial conditions.

5.5. Results and discussion

The first graph in Fig. 8 shows the WPMs of the three keyboards for the three different sizes. As expected, the WPM decreased as the sizes of the keyboards decreased. A repeated measures ANOVA after ART (3 sizes × 3 keyboards) to compare the means of WPMs over the three sessions showed a significant main effect of *Size* on WPM ($F(2, 34) = 103.54$, $P < 0.001$, $\eta_p^2 = 0.30$). Pairwise comparisons with Šidák correction showed that the performances between all pairs of the sizes were significantly different ($P < 0.001$). There was also a significant main effect of *Keyboard* on WPM ($F(2, 34) = 183.63$, $P < 0.001$, effect size = 0.52). Pairwise comparisons with Šidák correction showed that the SplitBoard and the QWERTY were similar in WPM ($P = 0.262$), and the ZoomBoard was significantly slower than the SplitBoard ($P < 0.001$) and the QWERTY ($P < 0.001$). There was a significant *Size* × *Keyboard* interaction effect on WPM ($F(4, 68) = 14.81$, $P < 0.001$, $\eta_p^2 = 0.10$).

The second graph in Fig. 8 shows the TERs of the three keyboards for the three different sizes. A repeated measures ANOVA after ART (3 sizes × 3 keyboards) to compare the means of TERs over the three sessions showed a significant main effect of *Size* on TER ($F(2, 34) = 50.46$, $P < 0.001$, $\eta_p^2 = 0.12$). Pairwise comparisons with Šidák correction showed that all pairs of sizes had a statistically significant difference in TER ($P < 0.001$) except for *Large* and *Medium* pair ($P = 0.868$). There was also a significant main effect of *Keyboard* on TER ($F(2, 34) = 241.18$, $P < 0.001$, $\eta_p^2 = 0.82$). Pairwise comparison with Šidák correction showed that all pairs of keyboards had a significant difference in TER ($P < 0.001$). There was a significant *Size* × *Keyboard* interaction effect on TER ($F(4, 68) = 7.97$, $P < 0.001$, $\eta_p^2 = 0.04$).

The average UERs of the SplitBoard, the QWERTY, and the ZoomBoard for the three size cases are summarized in Table 3. All UER values were small enough not to have practical significance.

Participants' preference over the three keyboards varied depending on the keyboard sizes as shown in Table 4. Participants liked the ZoomBoard most in the smallest size case. The SplitBoard was preferred most in the medium and the largest size cases. The least number of participants disliked the SplitBoard in all size cases. In the largest size case, the ZoomBoard was selected as the most inconvenient keyboard. From the questionnaire answers, we could find similar feedback as in Experiment 1. In addition, there was some feedback related to the size of the keyboards. Participants said that the SplitBoard was easy and fast in all size cases. They mentioned that they could learn the SplitBoard quickly. Some participants had difficulty in doing flicking gesture in the smallest size case. Participants said that the QWERTY was fast in the largest size case, but caused frequent errors in the smallest size. As mentioned in Experiment 1, the ZoomBoard was thought to be the most accurate method in the smallest size case. Several participants mentioned that zooming-in was unnecessary in the largest size case.

6. EXPERIMENT 3: TYPING PERFORMANCE AND MOBILITY

The goal of the experiment was to observe how the performance of the three QWERTY keyboards changes due to user motion. For this goal, we measured the performance of the three keyboards in two different conditions: standing and walking conditions.

6.1. Participants

We recruited 12 students from our university (4 females, average age of 22.42 (SD = 3.5)). One of them were left-handed and all others were right-handed. No participant had a prior experience of using a smartwatch.

6.2. Apparatus

The Galaxy Gear smartwatch used in Experiments 1 and 2 was used again, and the same three soft keyboard designs that were used in the largest size case of Experiment 2 were used. Right-handed participants wore the device on the left wrist, and a left-handed participant wore the device on the right wrist.

6.3. Procedure

First, we measured participants' preferred walking speed (PWS). We asked them to adjust the treadmill speed by the step of 0.1 km/h until they felt that it was similar to their usual walking speed. The average PWS of all participants was ∼3.8 km/h (SD = 0.24).[7] Participants then participated in a training session, where we demonstrated how to use the three keyboards and asked them to enter phrases displayed on the screen using each keyboard.

After the training session, participants completed two evaluation sessions: one while standing and the other while walking on the treadmill. In the standing condition, they entered text while standing as shown in Fig. 9. In the walking condition, they entered text while walking on the treadmill at their PWS. In both conditions, they entered 5 phrases given on the wristwatch screen using each keyboard (15 sentences in total). To avoid over-concentration and to simulate a more natural mobile situation, we asked the participants to relax for 30 s between sentences. During the 30 s relaxation pause in the walking condition, they swung their arms as they normally would while walking.

After finishing all sessions, participants answered a questionnaire. The questionnaire asked about their preferences over the three keyboards in the standing and walking conditions ('Which keyboard do you prefer most (least) for typing while standing (walking)?'), and provided spaces for describing reasons for their answers.

[7] These numbers are based on the treadmill's own speed indication, which we did not calibrate. We expect the indication to be approximately correct.

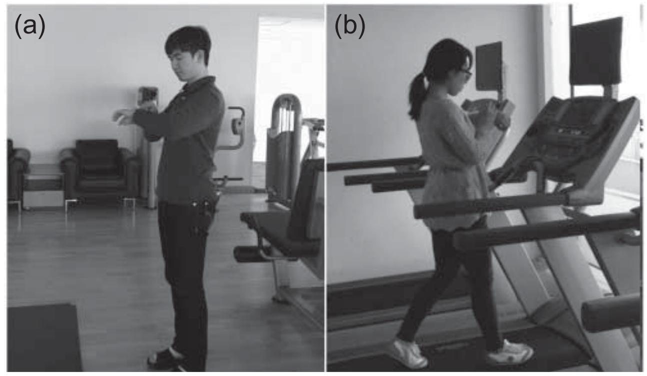

Figure 9. Postures in Experiment 3: (a) in the standing condition and (b) in the walking condition.

6.4. Design

The independent variables were *Context* (Standing or Walking) and *Keyboard*. The experiment was a 2 (*Context*) × 3 (*Keyboard*) within-subject factorial design. The order of the contexts was fully counter-balanced; half of the participants completed Walking condition first, and others completed Standing condition first. In each condition, the order of the three keyboards was counterbalanced using a Latin Square. We measured WPMs and TERs for each of the factorial conditions.

6.5. Results

The first graph in Fig. 10 shows the WPMs of the three keyboards in the two contexts. A repeated measures ANOVA after ART (2 contexts × 3 keyboards) showed a significant main effect of *Keyboard* on WPM ($F(2, 55) = 58.80$, $P < 0.001$, $\eta_p^2 = 0.846$). Pairwise comparisons with Šidák correction showed that the SplitBoard and the QWERTY were significantly faster than ZoomBoard ($P < 0.0001$), but there was no significant difference in WPM between the SplitBoard and the QWERTY ($P = 0.19$). There was also a significant main effect of *Context* on WPM ($F(1, 55) = 11.35$, $P < 0.01$, $\eta_p^2 = 0.495$). Walking condition was slower than Standing condition. There was no significant *Context* × *Keyboard*

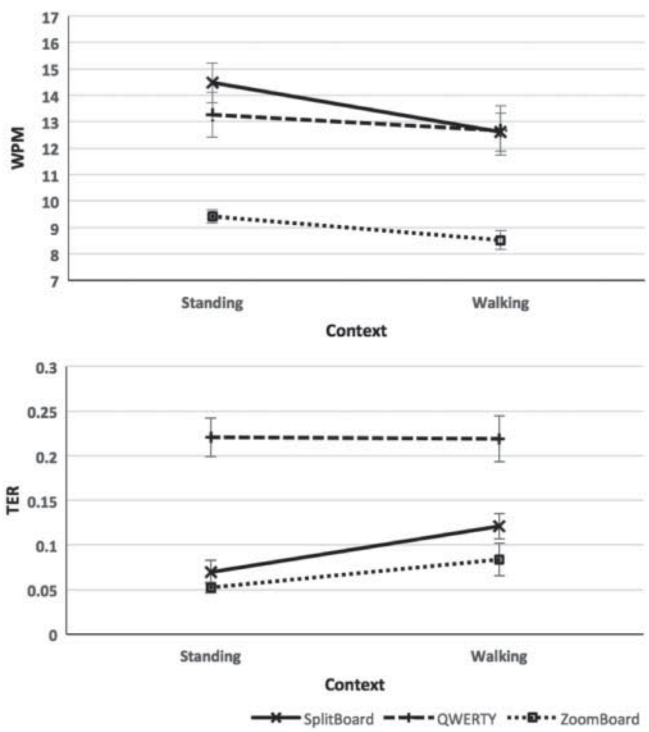

Figure 10. The WPMs and TERs of the QWERTY, the ZoomBoard and the SplitBoard in Experiment 3.

interaction effect on WPM ($F(2,55) = 0.62$, $P = 0.54$, $\eta_p^2 = 0.295$).

The second graph in Fig. 10 shows the TERs of the three keyboards in the two contexts. A repeated measures ANOVA after ART (2 contexts × 3 keyboards) showed a significant main effect of *Keyboard* on TER ($F(2,55) = 82.27$, $P < 0.001$, $\eta_p^2 = 0.835$). Pairwise comparisons with Šidák correction showed that the SplitBoard and the ZoomBoard had significantly lower TERs than the QWERTY ($P < 0.0001$), and the ZoomBoard had a significantly lower TER than SplitBoard ($P < 0.05$). There was also a significant main effect of *Context* on TER ($F(1,55) = 4.14$, $P < 0.05$, $\eta_p^2 = 0.393$). Walking condition had higher TERs than Standing condition. There was a significant *Context* × *Keyboard* interaction effect on TER ($F(2,55) = 3.83$, $P < 0.05$, $\eta_p^2 = 0.221$).

Table 5 shows the results of the post-test questionnaire on the preference. The SplitBoard received most of the votes in the walking (75%) and standing (67%) contexts. Preference for the ZoomBoard increased from 0 to 25% when participants were mobile. The QWERTY and the ZoomBoard shared the majority of the least preferred votes almost equally in both contexts. In their comments participants explained their preference for the SplitBoard by saying that it was both fast and error free. The dislike for ZoomBoard was explained by saying that entering text with it was tiring and took too much time.

Table 5. Participants' preferences over the three keyboards in the two contexts of Experiment 3.

	Standing	Walking
Most preferred		
QWERTY	3 (25%)	1 (8%)
ZoomBoard	0 (0%)	3 (25%)
SplitBoard	9 (75%)	8 (67%)
Total	12 (100%)	12 (100%)
Least preferred		
QWERTY	4 (33%)	7 (58%)
ZoomBoard	7 (58%)	5 (42%)
SplitBoard	1 (8%)	0 (0%)
Total	12 (100%)	12 (100%)

7. DISCUSSION

7.1. Comparison with earlier work

Leiva *et al.* (2015) conducted a study similar to ours where they compared three QWERTY keyboard designs (ZoomBoard, Callout and ZShift) in three different size conditions. Their ZoomBoard design was very similar to the ZoomBoard in our study. The Callout uses a callout showing the currently touched key, and the ZShift uses a callout showing a magnified view of the neighborhood of the currently touched key. Leiva *et al.* found no statistically significant differences in text entry rate among the three keyboards in any of the three size cases except that the ZoomBoard turned out to be superior to other keyboards in the smallest size case. The performance of their ZoomBoard was comparable to our ZoomBoard, and the performances of all of the three keyboards were far below that of the SplitBoard or the QWERTY in our study.

In Oney *et al.* (2013), the performance of the basic QWERTY keyboard was worse than that of the ZoomBoard. They reported that some participants even gave up the task due to the excessively frequent errors of the basic QWERTY keyboard. In contrast, in our Experiment 2, the text entry speed of the QWERTY in the smallest size case, which is equivalent to the basic QWERTY keyboard in Oney *et al.* (2013), was higher than that of the ZoomBoard. This contradicting result may be due to the difference between the two basic QWERTY keyboard designs. Even though the two keyboards used the same screen area, the QWERTY keyboard in Oney *et al.* (2013) had square keys while the QWERTY in our study had vertically stretched keys as shown in Fig. 7. The bigger keys may have lowered the number of errors and made the QWERTY faster. The contradicting result may be also due to the difference in the amount of training time. In Oney *et al.* (2013), participants transcribed 12 phrases for each keyboard, whereas in our experiment the participants transcribed 36 phrases for each keyboard. The longer training time in our case may have helped participants develop a precise pointing skill. It is also possible that the quality of the touchscreen may have been different between the two studies, which also may be one of the possible reasons for the contradicting results between the two studies.

7.2. The unexpectedly good performance of the QWERTY

The most unexpected finding from the current research was that users could adapt so well to the tiny QWERTY keyboard even in the smallest size (key size $1.6\,\text{mm} \times 3.0\,\text{mm}$). In Experiment 2, the QWERTY was comparable to the SplitBoard and was much better than the ZoomBoard for all sizes. We expected that the QWERTY would not be good any more if we move to a mobile situation. Contrary to our expectation, its relative performance remained almost unchanged as we have seen in Experiment 3. Throughout the three experiments, the QWERTY showed the highest TERs and therefore was disliked most by all participants, but the WPM of the QWERTY was still comparable to the SplitBoard and was much better than the ZoomBoard. If we had considered WPM only, we would have concluded that the QWERTY was the best text entry methods for smartwatches even without a dictionary-based technique to supplement its high error rate.

7.3. The potential benefit of using a language model

As we explained in Section 3, we decided not to use language models in the current study, and focused on the *raw* key entry efficiency instead. The effectiveness of a language model will be different for different keyboards, and finding the most effective way of using a language model for each keyboard will be an important next step as we mentioned earlier. A language model may be used to improve a text entry method in many different ways. One of them is to reduce typing errors by adjusting the receptive fields of the keys based on their conditional probabilities. Another is to substitute a more likely word for a misspelled word based on a dictionary. Yet another use of a language model is to speed up text entry by suggesting an auto-completion target based on a dictionary. The first two examples are for preventing or recovering from an error and will benefit a text entry method with a high error rate more. The third example use of a language model will be equally effective for most text entry methods. In this respect, an anticipation that we can have at the moment from the experimental results of the current research is that the QWERTY, which exhibited a good WPM but the highest TER, may benefit most from the use of a language model.

Recently, Vertanen *et al.* (2015) presented VelociTap, a soft keyboard decoder supporting a sentence-based text entry approach. They could show 35 wpm on a $25\,\text{mm} \times 16\,\text{mm}$ QWERTY keyboard. It seems that with such a language model support the QWERTY may indeed be a viable option for smartwatch text entry.

7.4. The good characteristics of the SplitBoard

The SplitBoard showed a better performance than other keyboards in all experiments. We enumerate here a few reasons that we think may explain the good performance of the SplitBoard in comparison with other keyboards. First, zooming brings about changes in both the position and scale of the keyboard, while panning brings about a change only in the position of the keyboard. Therefore, a panning strategy may have enabled a more stable and predictable typing experience.

Second, the SplitBoard requires less number of gestures for entering a character than the ZoomBoard. The GPC of the SplitBoard is 1.28; a user needs 1 tap and 0.28 flick gestures for a character. On the other hand, the GPC of the ZoomBoard is 1.84; a user needs 1.69 taps for a non-space character and 0.15 swipes for a space character.

Third, the flick gestures of the SplitBoard for switching the keyboard is relatively light weighted because they do not require aiming. In contrast, both of the two tap gestures for entering a character in the case of the ZoomBoard may be heavier because they require aiming. The tap gestures in the case of the QWERTY must be of course heaviest because they require very precise aiming.

Lastly, the QWERTY layout happened to be a good match for the SplitBoard. It is easy to calculate the GPC of the SplitBoard if we assume that the probability of each key is independent and identical, and it is 1.4 when the SplitBoard has two overlapping key columns between the left and right halves. In contrast, the GPC of the SplitBoard that we calculated taking the QWERTY layout and the English language model into account was 1.28. This means that the QWERTY layout was helpful in reducing the number of switching between the left and right halves of the SplitBoard. In fact, we attempted to improve the performance of the SplitBoard by adjusting the key layout in the early phase of the current study, but the improvement was quite marginal.

7.5. Limitations and future work

One of the limitations of the current study may be the relatively small numbers of participants in the experiments. The numbers of participants in Experiments 1, 2 and 3 were 12, 18 and 12, respectively, which are small compared with that of traditional text entry studies. Considering the relatively small sample sizes, we report an eta squared value for each of the ANOVA analyses of the experimental results.

Another limitation may be the narrow age range of the participants. All participants in the experiments were in their 20s and late 10s. A prior study has shown that the performance of using a touchscreen is influenced by age (Findlater *et al.*, 2013). This may imply that people in different age groups may show different performances on a soft keyboard. We have to admit that the conclusions that we derive from the current study may not be generalized to other age groups.

Yet another limitation may be the relatively short duration of the experiments. The experiments in this study have relatively short training and evaluation periods. Though initial usability is an important issue in adopting a keyboard for smartwatches, a longitudinal study would have shown the ultimate performance that users can achieve through the everyday, regular use of the three keyboards.

8. CONCLUSION

We started the current study with the following questions. (i) Will people be able to use a small QWERTY keyboard on a smartwatch? (ii) When and how much will the zooming strategy benefit users? (iii) When and how much will the scrolling strategy benefit users? The answer to the first question seems to be both positive and negative. Participants in the experiments could develop the skill to use the QWERTY unexpectedly well, but all disliked it due to its high error rates. We may conclude that users can use a small QWERTY keyboard well but would not like to use it.

The answers to the other questions were the main results of the three experiments. Among these results the most important were. (i) The ZoomBoard showed a worse performance than other keyboard in all conditions, and may only be considered for very small screen cases. (ii) The SplitBoard consistently showed a better performance than other keyboards in all conditions. (iii) The performances of all keyboards were slightly (1–2 wpm) worse in the walking condition than in the standing condition.

The SplitBoard was successful in improving the QWERTY keyboard in a smartwatch environment. The SplitBoard resulted in a performance advantage and a large user preference improvement over the ordinary QWERTY keyboard. Immediate future work is to improve the SplitBoard by incorporating language model-based features. We will then be able to predict better the future of the SplitBoard as a practical smartwatch text entry solution.

FUNDING

This work was supported by the IT R&D program of MKE/KEIT [KI10041244, SmartTV 2.0 Software Platform] and by the Academy of Finland [MIPI, 266285].

REFERENCES

Ahn, Y., Hwang, S., Yoon, H., Gim, J. and Jung-hee, R. (2015) Bandsense: Pressure-sensitive Multi-touch Interaction on a Wristband. In *Proc. 33rd Annual ACM Conf. Extended Abstracts on Human Factors in Computing Systems, CHI EA'15*, pp. 251–254. ACM.

Azenkot, S. and Zhai, S. (2012) Touch Behavior with Different Postures on Soft Smartphone Keyboards. In *MobileHCI 2012*.

Bates, R. and Istance, H. (2002) Zooming Interfaces!: Enhancing the Performance of Eye Controlled Pointing Devices. In *Proc. 5th Int. ACM Conf. Assistive Technologies*, pp. 119–126.

Bérard, F. and Rochet-Capellan, A. (2012) Measuring the Linear and Rotational user Precision in Touch Pointing. In *Proc. 2012 ACM Int. Conf. Interactive Tabletops and Surfaces, ITS '12*, New York, NY, USA, pp. 183–192. ACM.

Cha, J.-M., Choi, E. and Lim, J. (2015) Virtual sliding qwerty: a new text entry method for smartwatches using tap-n-drag. Appl. Ergon., 51, 263–272.

Chen, X., Grossman, T. and Fitzmaurice, G. (2014) Swipeboard: A Text Entry Technique for Ultra-Small Interfaces that Supports Novice to Expert Transitions. In *UIST 2014*, pp. 615–620.

Colle, H. and Hiszem, K. (2004) Standing at a kiosk: effects of key size and spacing on touch screen numeric keypad performance and user preference. Ergonomics, 47, 1406–23.

Dunlop, M.D., Komninos, A. and Durga, N. (2014) Towards High Quality Text Entry on Smartwatches. In *Proc. Extended Abstracts of the 32nd Annual ACM Conf. Human Factors in Computing Systems – CHI EA'14*, pp. 2365–2370. ACM Press.

Dunlop, M.D. and Masters, M.M. (2009) Pickup usability dominates: a brief history of mobile text entry research and adoption. Int. J. Mob. Hum. Comput. Interact., 1, 42–59.

Findlater, L., Froehlich, J.E., Fattal, K., Wobbrock, J.O. and Dastyar, T. (2013) Age-related Differences in Performance with Touchscreens Compared to Traditional Mouse Input. In *Proc. SIGCHI Conf. Human Factors in Computing Systems*, pp. 343–346. ACM.

Funk, M., Sharami, A., Henze, N. and Schmidt, A. (2014) Using a Touch-sensitive Wristband for Text Entry on Smart Watches. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems, CHI EA'14*, pp. 2305–2310. ACM.

Henze, N., Rukzio, E. and Boll, S. (2011) 100,000,000Taps: Analysis and Improvement of Touch Performance in the Large. In *MobileHCI 2011*, pp. 133–142.

Henze, N., Rukzio, E. and Boll, S. (2012) Observational and Experimental Investigation of Typing behaviour using Virtual Keyboards for Mobile Devices. In *Proc. 2012 ACM Annual Conf. Human Factors in Computing Systems – CHI'12*, pp. 2659. ACM Press.

Hong, J., Heo, S., Isokoski, P. and Lee, G. (2015) Splitboard: A Simple Split Soft Keyboard for Wristwatch-sized Touch

Screens. In *Proc. 33rd Annual ACM Conf. Human Factors in Computing Systems*, pp. 1233–1236. ACM.

Hwang, S. and Lee, G. (2005) Qwerty-like 3 × 4 Keypad Layouts for Mobile Phone. In *CHI'05 Extended Abstracts on Human Factors in Computing Systems*, pp. 1479–1482.

Isokoski, P. and Raisamo, R. (2000) Device Independent Text Input: A Rationale and an Example. In *Proc. Working Conf. Advanced Visual Interfaces*, pp. 76–83.

James, C. and Reischel, K. (2001) Text Input for Mobile Devices: Comparing Model Prediction to Actual Performance. In *CHI'01*, pp. 365–371.

Komninos, A. and Dunlop, M. (2014) Text input on a smart watch. IEEE Pervasive Comput., 13, 50–58.

Leiva, L.A., Sahami, A., Catala, A., Henze, N. and Schmidt, A. (2015) Text Entry on Tiny QWERTY Soft Keyboards. In *Proc. 2015 Conf. Human Factors in Computing Systems – CHI'15*.

Li, F.C.Y., Guy, R.T., Yatani, K. and Truong, K.N. (2011) The 1line Keyboard: A Qwerty Layout in a Single Line. In *Proc. 24th Annual ACM Symp. User Interface Software and Technology, UIST'11*, New York, NY, USA, pp. 461–470. ACM.

MacKenzie, I. (2002) KSPC (keystrokes per character) as a Characteristic of Text Entry Techniques. *Hum. Comput. Interact. Mob. Devices*, pp. 405–416.

MacKenzie, I.S. and Soukoreff, R.W. (2003) Phrase Sets for Evaluating Text Entry Techniques. In *CHI'03 Extended Abstracts on Human Factors in Computing Systems*, pp. 754–755.

Mann, S. (2000) A GNU/Linux wristwatch videophone. Linux Journal.

Oney, S., Harrison, C., Ogan, A. and Wiese, J. (2013) ZoomBoard: A Diminutive QWERTY Soft Keyboard using Iterative Zooming for Ultra-small Devices. In *Proc. SIGCHI Conf. Human Factors in Computing Systems*, pp. 2799–2802.

Parhi, P., Karlson, A.K. and Bederson, B.B. (2006) Target Size Study for One-handed Thumb use on Small Touchscreen Devices. In *Proc. 8th Conf. Human–Computer Interaction with Mobile Devices and Services, MobileHCI'06*, New York, NY, USA, pp. 203–210. ACM.

Partridge, K., Chatterjee, S., Sazawal, V., Borriello, G. and Want, R. (2002) TiltType: Accelerometer-supported Text Entry for Very Small Devices. In *Proc. 15th Annual ACM Symp. User Interface Software and Technology*, pp. 201–204. ACM.

Pavlovych, A. and Stuerzlinger, W. (2003) Less-Tap: A Fast and Easy-to-learn Text Input Technique for Phones. In Graphics Interface.

Perrault, S., Lecolinet, E., Eagan, J. and Guiard, Y. (2013) Watchit: Simple Gestures and Eyes-free Interaction for Wristwatches and Bracelets. In *Proc. SIGCHI Conf. Human Factors in Computing Systems*, pp. 1451–1460. ACM.

Raghunath, M.T. and Narayanaswami, C. (2002) User interfaces for applications on a wrist watch. Pers. Ubiquitous Comput., 6, 17–30.

Sears, A. (1991) Improving touchscreen keyboards: design issues and a comparison with other devices. Interact. Comput., 3, 253–269.

Sears, A. and Shneiderman, B. (1991) High precision touchscreens: design strategies and comparisons with a mouse. Int. J. Man–Mach. Stud..

Sheik-Nainar, M. (2010) Contact Location Offset to Improve Small Target Selection on Touchscreens. In *Proc. Human Factors and Ergonomics Society 54th Annual Meeting*, pp. 610–614.

Soukoreff, R.W. and MacKenzie, I.S. (2003) Metrics for Text Entry Research: An Evaluation of MSD and KSPC, and a New Unified Error Metric. In *Proc. SIGCHI Conf. Human Factors in Computing Systems*, pp. 113–120. ACM.

Vertanen, K., Memmi, H., Emge, J., Reyal, S. and Kristensson, P.O. (2015) Velocitap: Investigating Fast Mobile Text Entry using Sentence-based Decoding of Touchscreen Keyboard Input. In *Proc. 33rd Annual ACM Conf. Human Factors in Computing Systems, CHI'15*, New York, NY, USA, pp. 659–668. ACM.

Weigel, M., Mehta, V. and Steimle, J. (2014) More than Touch: Understanding How People use Skin as an Input Surface for Mobile Computing. In *Proc. SIGCHI Conf. Human Factors in Computing Systems*, pp. 179–188. ACM.

Wobbrock, J.O., Findlater, L., Gergle, D. and Higgins, J.J. (2011) The Aligned Rank Transform for Nonparametric Factorial Analyses using Only ANOVA Procedures. *CHI'11 Conf. Human Factors in Computing Systems*, pp. 143–146.

Wobbrock, J.O., Myers, B.A. and Kembel, J.A. (2003) EdgeWrite: A Stylus-based Text Entry Method Designed for High Accuracy and Stability of Motion. In *Proc. 16th Annual ACM Symp. User Interface Software and Technology*, pp. 61–70.