Text Entry Interface Design And Evaluation: Literature Review

Tiankai Jiang

20834939

t57jiang@uwaterloo.ca

August 2, 2020

Abstract

This thesis analyzed 40 papers about text entry interface design and performance evaluation. And it aims to serve as a guideline and reference for researchers who want to conduct experiment in this field. The paper mainly focuses on methods to improve text entry user experience for physical keyboards and touch screen keyboards on mobile devices, and evaluation design details.

Keywords— text entry, usability evaluation, design principle, touch screen keyboard, physical keyboard

1 Introduction

The keyboard is one of the most important text entry methods. The origin of the keyboard can be traced back to the beginning of the 17th century in Europe. In 1868, an American inventor Christopher Latham Sholes got the patent and franchise of typewriter. Later, the prototype of the modern keyboard layout(QWERTY) was developed. Beginning from the 1950s, keyboards were massively used in computers. And since smartphones came out in the early 21st century, keyboards were migrated to smartphones in both physical form and a virtual touch screen form. Currently, there are more than 10 billion mobile devices(smartphones and tablets) in use worldwide and the keyboard still serves as a primary input method. With the widespread of keyboard and the

extension of keyboard usage scenario, e.g. on the smart TV, on the smartwatch, for the physically challenged, while driving, etc. the variation and improvement of the conventional keyboard on mobile devices are required to improve the performance and applicability.

40 papers related to text entry (mostly keyboard) design and evaluation were analyzed in this paper and the paper aims to answer the following questions:

- What aspects could be considered to improve the user experience of the keyboard?
- What aspects could be considered to evaluate a keyboard?

2 Methods to Improve Text Entry User Experience

User experience is related to the potential typing speed, the error rate on the keyboard, the mental workload while typing, and the occupied area of the keyboard on the screen, etc. Several aspects were mentioned in the reviewed papers to improve text entry user experience:

- Visual improvement, e.g. making more space for content display.
- Feedback improvement, e.g. adding physical or voice feedback.
- Performance improvement, e.g. increasing typing speed and lowering error rate.
- Workload improvement, e.g. decreasing users' mental workload.

2.1 Physical Keyboard Modification

Although the number of mobile phones with a physical keyboard has been decreasing sharply in recent years, qwerty phones remain the preference for a minority of people, and companies like BlackBerry still produce qwerty smartphones. Also, mobile devices with a physical keyboard is an essential for some people with disabilities.

In the early days, mobile devices are not as portable as it is today. So their keyboard is relatively larger. Green et al. (2004) proposed a specialized keyboard, which compressed the 4-line characters and numbers in conventional keyboards into one line and used modifier keys and multitap to achieve character selection.

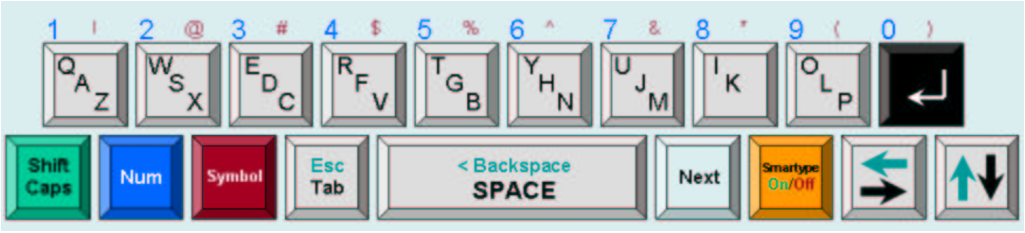


Figure 1: The reduced qwerty keyboard from (Green et al., 2004)

This design was motivated in two aspects, that the reduced keyboard only takes up 15% to 25% of the size of a normal keyboard and as we can see in Figure 1, each key is basically the compression of a normal keyboard in the vertical direction so that the users could use their knowledge of the conventional keyboard. The experiment showed that after an average of 4.75 hours of practice, participants could achieve half (10.4wpm) of their normal typing speed (22.5wpm). Although this was not an improvement in speed, considering the reduction in size and comparisons with its previous similar experiments, this experiment achieved a reasonable keyboard performance and learnability. It may seem naive currently since we have a touch screen and more accurate sensors so we can reduce the key size. This design did not modify the size of the key, therefore, it is useful under some special conditions, e.g. text input on a small device in factories where the operators have to wear thick gloves so accurate operation cannot be performed.

One of the biggest drawbacks of the physical keyboard on mobile devices is that it takes up a lot of space that potentially can be used to present more content. Scott et al. (2010) presented RearType, a text input system that uses normal keyboard keys but locates at the back of a device, aiming to maximize the use of the display for visual output.

(a) Front (b) Back

Figure 2: The RearType Keyboard from (Scott et al., 2010)

The RearType split the QWERTY keyboard from the middle and rotated them by 90 degrees. So the keys have the same relative position as before, despite at the back, and the modifier keys located at the front of the devices, within the approach of two thumbs (Figure 2). So users can grab the devices using both hands while typing using all fingers. It is unreasonable to assume that users could touch type with this keyboard, so a semi-transparent floating widget on the screen was used to show which key was pressed.

The result showed that after 1-hour training, participants could achieve an average typing speed of 15 wpm, which was not statistically different from a touch screen keyboard.

Similarly, Kim et al. (2012) presented a physical QWERTY layout keyboard at the back of a mobile phone. Their design was more delicate than the previous one as it was for mobile phones only. The keyboard was placed at the top back of the mobile phone, split in half as a spatial cue for easy recognition of the keys' location. Users used their index fingers and/or middle fingers to type. The experiment found that after 40 minutes of exercise, participants could achieve a text entry rate of 15.3 wpm, with an error rate of 12.2%.

Both experiments proved that our knowledge about the QWERTY layout can be transferred smoothly to the back of the device. This kind of design also seems crazy and unreasonable with little practical appeal. After all, the input speed did not increase with these designs. In the next section, a design similar to these keyboards was shown, but with a more modern approach.

2.2 Touch screen at the back

Generally, a touch screen is more versatile than physical keyboards, for it can not only be used for typing, but also gesture recognition and display. Since the existence of, no matter touch screen keyboard or physical keyboard, at the front of the device will reduce the area for displaying content, instead of adding a physical keyboard at the back, why not replace it with a touch screen at the back?

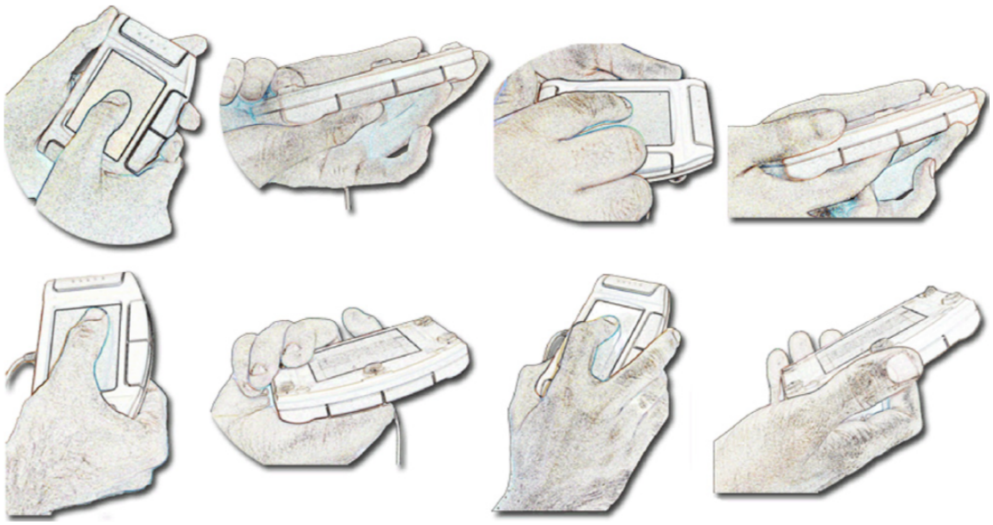


Figure 3: Touch screens on both sides of the device in (Wobbrock et al., 2008)

Wobbrock et al. (2008) conducted experiments to check the feasibility of making gestures on the screen at the back of a mobile device. Their findings indicate that the index finger works well on the front and back of a device. And they showed that users could make letter-like gestures on the front and back of a device and that the index finger on the front of the device performed best. Taken together, these studies indicate that device designs can go further to leverage a richer set of finger and thumb interactions.

The experiment was taken in 2008 when the touch screen technology was still in its initial stage. The accuracy and the texture of the touchscreen were far from perfect.

Combining the result of this experiment and the previous two, we know that we are capable of both typing and making gestures at the back of a device. With the development of OLED display, the screen of a mobile phone can be extended to the back of the phone, such as Xiaomi Mix Alpha, and many high-end

smartphones nowadays have two displays, at each side of the phone, with much higher accuracy. The back type keyboard/gesture prototypes may not be used directly, but they serve as a good reference if a touch screen keyboard at the back of the device is needed.

2.3 Adding Physical Feedback to Touchscreen

Generally, one of the drawbacks of a touch screen is that there is no physical feedback when users make a press. So they do not know if they have truly touched the button, which may lead to extra keypress.

Hoggan et al. (2008) conducted an experiment to compare devices with a physical keyboard, a standard touchscreen, and a touchscreen with tactile feedback. The result indicated that adding tacticle feedback to the touch screen could improve the performance of typing significantly. Later, they did the same experiment with a high specification actuator, which could roughly indicate the touch location. The result showed that accuracy and speed improved even further.

Lee and Zhai (2009) surveyed the impact of tacticle feedback and audio feedback on the performance of text entry. The result indicated there was a significant improvement in typing speed and error rate when using tacticle feedback and audio feedback. But there was no significant difference between using tacticle feedback and using audio feedback, and combining them together would not improve the result further.

Both experiments indicated it was essential to add feedback to the touch screen. The application of physical feedback can be seen clearly through the evolution of the iPhone. Since the first generation of the iPhone, there was audio feedback when typing. From iPhone 6s, 3D touch was added, giving users a sense of "soft" and "heavy" when performing certain tasks. And later on iPhone XS, Apple replaced 3D touch with haptic touch, which not only showed different feedback strength but also showing the feedback at the specific point where users pressed the screen using a two-dimensional linear motor, making iPhone one of the phones with the best typing experience.

2.4 Touch Screen Keyboard Layout Modification

One big advantage of a software keyboard is that it can be modified to whatever layout we desire with little cost. The layout for the QWERTY keyboard was initially designed to avoid sticking in the typewriters. So, it does not necessarily ensure the performance on the computer keyboard, let alone the performance on mobile devices. Therefore, a series of new keyboard layout were purposed in order to speed up typing on mobile devices.

MacKenzie and Zhang (1999) proposed a new keyboard layout for mobile devices (OPTI). Previous research summarized a 27×27 table, showing the probability of the next character given the current character. The one more character, in addition to 26 letters, was space. The new layout was designed based on minimizing the path between every two characters according to that table.

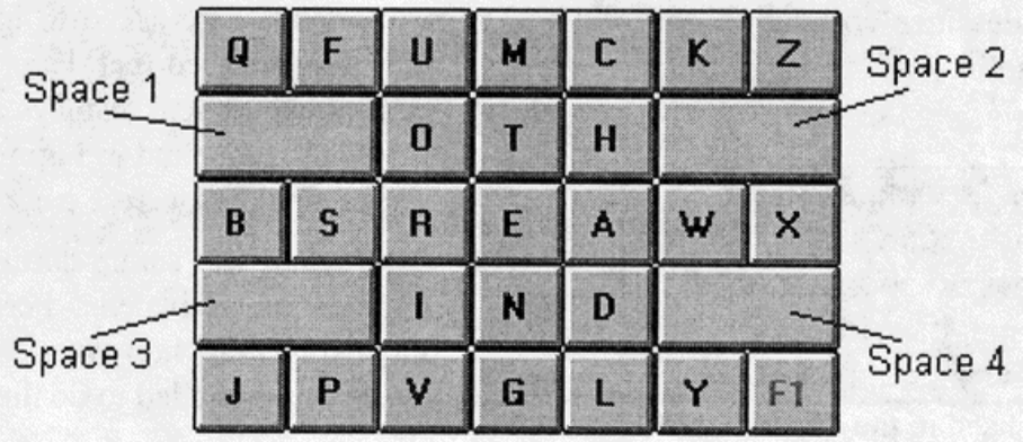


Figure 4: New keyboard layout from (MacKenzie and Zhang, 1999)

The layout was shown in Figure 4. The research assumed that users used one finger or a stylus pen for text entry. The result indicated that after 20 45-minutes training session, the text input speed increased from 17.0 wpm to 44.3 wpm. And some expert users achieved 58.2 wpm, which is 35% faster than for the QWERTY layout.

Bi et al. (2010) mentioned that some kind of layouts might be more efficient than QWERTY layout, however, if the difference between QWERTY and the new layout were too large, it would be difficult and inconvenient to migrate from the conventional layout to the new one. Hence, they proposed Quasi-Qwerty

optimization. The overall performance would be increased and at the same time, the new layout was closed to the QWERTY layout, avoiding the problem mentioned above.

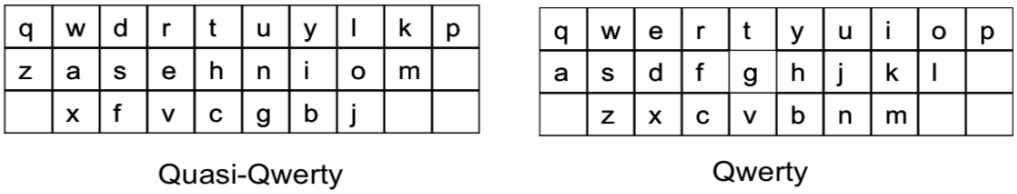


Figure 5: New keyboard layout (left) from (Bi et al., 2010)

Using the sum of Fitts' law movement time between all pairs of letters along with the constraint on the QWERTY layout, the new layout was designed and shown in Figure 5. The predicted entry speed for Quasi-Qwerty keyboard was not as high as the freely optimized one, but higher than the conventional QWERTY keyboard. However, in the experiment, the initial entry time was shorter for the Quasi-Qwerty keyboard than the freely optimized one, which means novice users found it easier to transfer to this design. This paper reminded us that the performance was not the only criteria when designing a keyboard. Human factors such as the motivation to transfer to the new design should also be considered.

The reduced keyboard layout mentioned in Green et al. (2004) could not only be applied to the physical keyboard. With the development of tablets, a similar one-line keyboard layout was proposed by Li et al. (2011), aiming for touch screen mobile devices with a large screen, to maximize the displaying area for contents so that users did not have to swipe up and down constantly while typing. The new keyboard was only 39.8% height of the native keyboard.

Figure 6: New keyboard layout from (Li et al., 2011)

This design (Figure 6) also compressed the three-line QWERTY keyboard into one line. And based on the

usage rate for each key, the keyboard further reduced (combined) some keys. The input process was close to the input on a T9-like keyboard. We typed the keys and selected the desired words predicted by the algorithm. The difference was that the "Enter", "Backspace" and word selection were done by gestures, which further reduced the number of keys. The experiment result showed participants could type at a rate of over 30 WPM after just five 20-minute typing sessions. And the predicted text entry speed was 68 wpm.

Above are three papers trying to improve the typing experience on mobile devices. Each of them had their consideration: typing speed and error rate, users' motivation of switching, and the size of the keyboard. When designing a new layout, these criteria can also be considered for future researchers.

2.5 Touch Screen Keyboard Layout Modification For Smaller Devices

The smartwatch industry has been flourishing in recent years. And with the development of mobile CPU, the functions of smartwatch went from simple calculation, digital time telling in the 2000s, to text messaging, sleep tracking, workout tracking in the 2010s. Although the speech recognition for text input on a smartwatch is rather convenient now, sometimes out of privacy and other issues, we still want to choose to type for text entry. It would be awkward to put the QWERTY keyboard directly on the smartwatch since the screen size is too small to type. Therefore, new layouts for small touch screen devices were required.

One of the natural thoughts that came up when tackling this problem was enlarge the keys. Oney et al. (2013) proposed ZoomBoard, in which the keys were iteratively zoomed to a comfortable size to be seen.

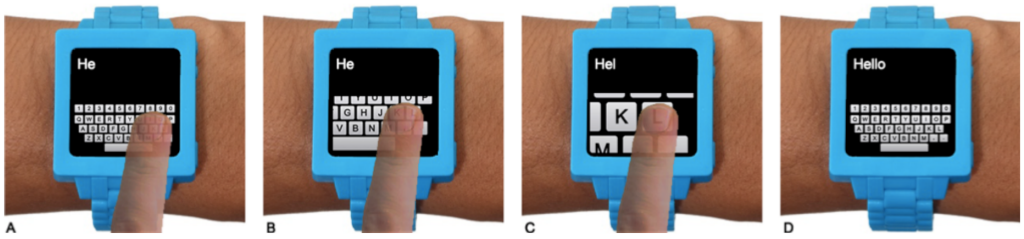


Figure 7: New keyboard layout from (Oney et al., 2013)

Figure 7 above shows the working mechanism of the Zoomboard. In order to correctly detect which part of the keyboard and which specific area on the keyboard was selected, the authors experimented with three

zooming strategies. And they found that upon pressing the screen, the keyboard zoomed, and leaving the target directly under the finger was the best approach. The ZoomBoard could be as tiny as the size of a US penny. The experiment found after the final trial, participants could achieve an average typing speed of 9.3 wpm, which was more than 100% faster than directly using the QWERTY keyboard on the screen. And the error rate was significantly lower as well.

Another approach to solving this problem was much the same as the one-line keyboard (Li et al., 2011) mentioned above. If the size of the individual key was too small, just combine multiple keys together and enlarged the key size. Komninos and Dunlop (2014) brought out a keyboard for the smartwatch with six keys.

Figure 8: New keyboard layout from (Komninos and Dunlop, 2014)

Figure 8 above shows the layout of the keyboard. When users pressed a key, the word with the most probability based on the current character sequence was shown in the middle area of the screen. Then users could tap in the center screen to rotate through all the possible words. The backspace, numbers, and capitalization were done by different swiping gestures on the screen. And the keyboard even supported adding accent marks by long pressing. The result indicated that the average typing speed was 8.1 wpm. It was not better than that of the ZoomBoard but this keyboard supported word suggestion, numbers, and accent marks, which reduced the typing actions required and was more functional.

Chen et al. (2014) outlined the Swipeboard for text entry on smartwatch. It leveraged users' spatial memory of a QWERTY keyboard, making it easy to learn, and it could even be operated under eyes-free condition.

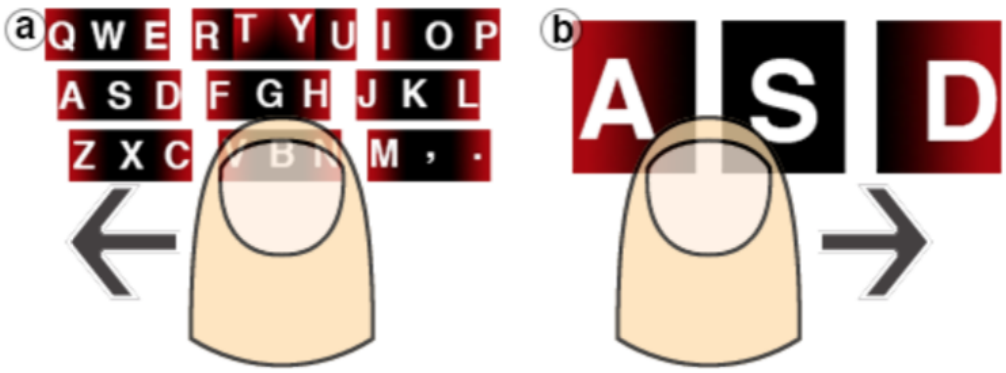


Figure 9: New keyboard layout from (Chen et al., 2014)

Figure 9 shows the mechanism of SwipeBoard. The keyboard could be as small as 12×12mm. Users used two swipes to select a character. In the first swipe, one of the nine regions where the character located was selected and in the second swipe, the character in the region was selected. The backspace, numbers, and symbols were also toggled using gestures. The experiment showed that after 2 hours of training, participants could achieve a typing speed of 19.58 wpm, which was reported 15% faster than their baseline. The authors also mentioned the potential migration of this keyboard to eyewear, such as to Google Glasses.

Hong et al. (2015) proposed SplitBoard, another text entry method on smartwatches. The QWERTY layout was divided into two sections in the SplitBoard, and the section could be switched using horizontal flick gestures (Figure 10). The keys were selected by a simple press. This approach was more straight-forward than previous solutions. The backspace and space and functional keys(nums, caps, symbols) were shown directly instead of toggled using gestures.

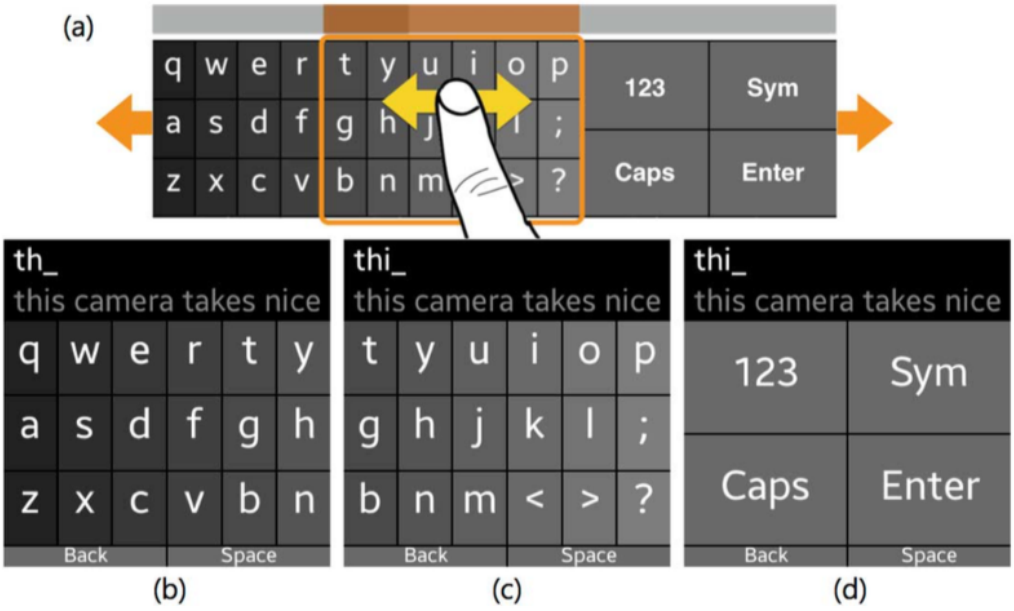
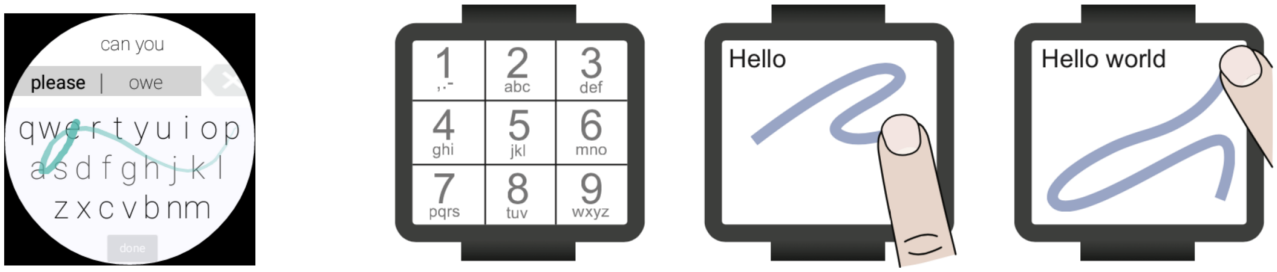


Figure 10: New keyboard layout from (Hong et al., 2015)

The experiment result revealed that SplitBoard was faster than ZoomBoard and the native QWERTY keyboard on Samsung Galaxy Gear (29.3mm×29.3mm touch screen) and the total error rate (TER) was the lowest among three keyboards and the uncorrected error rate (UER) was the same as QWERTY keyboard and lower than ZoomBoard.

Gordon et al. (2016) put forward WatchWriter (Figure 11a). Since it was difficult to modify the hardware, they moved their focus on the decoding algorithm. Instead of typing character by character, the strategy of WatchWriter was swipe typing. Users continuously swiped onto the approximate area of the keys in the word and relied on the algorithm to suggest the word. The space between words was automatically inserted and it also contained an algorithm to avoid autocorrect special words such as website URL. WatchWriter achieved a typing speed of 22-24 wpm with near-zero error rate.



(a) WatchWriter from (Gordon et al., 2016)

(b) Invisiboard from (Mottelson et al., 2016)

Figure 11: Comparison between WatchWriter(left) and Invisiboard(right)

Finally, Mottelson et al. (2016) proposed Invisiboard. It was a combination of the keyboard in Komninos and Dunlop (2014) and the WatchWriter (Gordon et al., 2016). Text input was done by swiping gestures, not on a QWERTY-like keyboard but a T9-like keyboard. Figure 11 shows the difference between them. Subjects achieved an average of 10.6 wpm typing speed after 30 minutes practice and some participants demonstrated a speed of over 30 wpm.

To summarize, future researches on keyboard layout modification for small screen devices could refer to ideas mentioned in the above five papers. Each of the methods had its own advantages and drawbacks. ZoomBoard could allow users to select a character in only one touch, but it did not have auto suggestions, and as mentioned by Hong et al. (2015), it would be slower than native QWERTY keyboard if the screen size was slightly larger. SwipeBoard required at least two presses to select a character. SplitBoard cannot be applied on devices as small as the one used in ZoomBoard. And WatchWriter required the computing power of the smartwatch to perform real-time prediction. The combination of these ideas could potentially lead to a better design.

2.6 Hiding The Touch Screen Keyboard

A lot of effort has been devoted to maximizing the displaying area on the touch screen. For instance, Scott et al. (2010) and Kim et al. (2012) moved the physical keyboard to the back of the device, and Li et al. (2011) compressed the keyboard into one line. All those methods assume that a visible keyboard was necessary for

typing. However, the standard keyboard layout has not been changed for decades and most of the users have been using the QWERTY keyboard for at least several years. Some researchers have been thinking about hiding the keyboard completely, letting users type only with their own memory about the keys' position. Completely hiding the keyboard could free up a large portion of the screen for content display.

Findlater et al. (2011) examined and compared the typing pattern under three conditions, when there was no visible keyboard and with/without error feedback and when there were visible keyboard and error feedback. The experiment assumed users type with ten fingers on a touch screen tablet, thus they could use their memory of touch typing experience on the physical keyboard. The error feedback in the experiment was asterisked – users could not tell what they type but they could know if they missed or made an extra press. The result indicated that after 5 practice sessions, and when there was no keyboard and no feedback, users could achieve 59 wpm typing speed if they assumed their input was correct, which was about 100%faster than that on touch screen devices (30 wpm) and about the same as that on the physical keyboard (60 wpm) (Dhakal et al., 2018). With the feedback under the invisible keyboard condition, the typing pattern experienced a larger spread of hits per key comparing to the condition when the keyboard was visible, but still, the classification accuracy was about 90%. The result revealed that typing on a flat surface with no visible keyboard was possible. And some advice for invisible keyboard and algorithm design was given by the authors, such as the space bar should be shorter and taller; the key size should be larger and there should be a gap in the middle of the keyboard.

Based on this finding, Zhu et al. (2018) explored the possibility of typing with an invisible keyboard on the mobile phone. The difference between typing on an invisible keyboard on a flat surface and the mobile phone was that the latter used only two thumbs or an index finger, under which condition the knowledge about ten fingers muscle memory may not be used. The research found that users could utilize their memory of the layout of the QWERTY keyboard even with only one or two fingers typing. And with the modification on the decoding algorithm, an average speed of 37.9 wpm could be achieved after a few sessions of practice, making it indistinguishable from the typing speed on a normal keyboard.

Those researches revealed that an invisible user interface was possible for mobile keyboards. It was an effective way to maximize the screen display area. To implement this design, several things should be modified

compared to the native keyboard, such as the key size, the curvature of the keyboard, and the decoding algorithm. In addition, since the keyboard was invisible, we should wisely deal with the components originally covered by the keyboard, avoiding conflicts between those components and the keyboard. And the workload should be further investigated by future research. After all, no one wants to switch to a keyboard that may make him/her distracted or exhausted.

2.7 Utilizing Gestures

The use of gestures has shown its own advantages on ultra-small screens, like WatchWriter (Gordon et al., 2016) and Invisiboard (Mottelson et al., 2016). Comparing with keyboards for desktop computers, keyboards on tablets and mobiles are relatively small as well. Researches have been conducted to explore whether the use of gestures on these devices would improve the typing experience.

Zhai and Kristensson (2003) designed *SHARK* shorthand gesturing stylus keyboard. The design was based on two observations: (1) some characters in a word were connected in stylus keyboard, therefore swiping on them using gestures was faster than hitting the keys one by one; (2) users tended to remember the pattern of a word, that is, the trajectory of stylus movement from key to key, rather than individual key taps. The experiment showed after 45 to 50 minutes of the practice session, subjects could reach a speed of 15 wpm. And most participants found it fun and even considered replacing physical typing from time to time. 15 wpm was not fast at all, as some of the optimized text entry methods on smartwatches like WatchWriter (Gordon et al., 2016) could achieve 24 wpm. But considering the experiment was performed in 2003, at which time the devices were less optimized, it was a big step towards the application of gesture typing.

Zhai et al. (2009) proposed ShapeWriter, a gesture keyboard on iPhone. ShapeWriter was a stroke gesture-based text and command input method. Instead of tapping each letter, the user traces over all letters in a word sequentially on the touch screen keyboard. The difference between this text entry method and the ones in all other papers was that ShapeWriter was actually put on AppStore. So the feedback could be directly reflected by the user comments. 454 (81.6%) reviews were completely positive, and 12.5% reviews were somewhat positive. Most of the positive reviews were about the performance of this new keyboard and wanted the keyboard to be

applied system-wide. Only 5.9% reviews were completely negative, and most of them complained about their inadaptation. Overall, gesture typing could be accepted by the public. And that is why Apple officially added this feature in iOS 13.

Figure 12: Bimanual keyboard from (Bi et al., 2012)

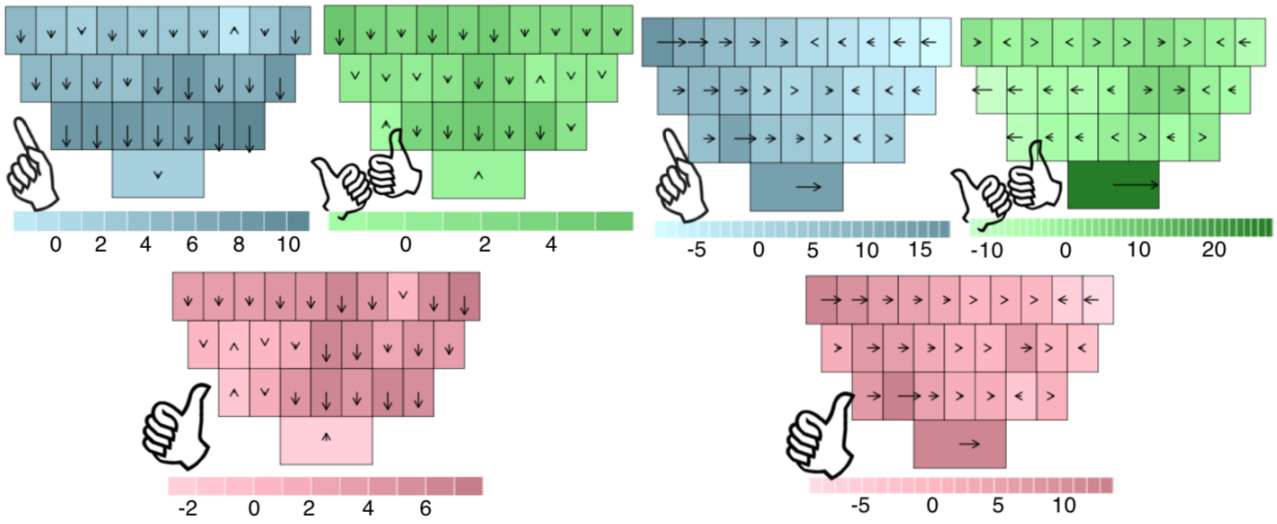
Bi et al. (2012) proposed a bimanual gesture keyboard. For most gesture keyboards, the input was done by only one hand. However, most touch screen devices currently have the ability to multitouch and the size of the tablet is suitable for two hands performing gestures simultaneously. Researchers designed the algorithm to recognize multiple keystrokes for the same word and suggested two approaches to detect the end of a word: finger-release, for which the release of both fingers indicated the end of a word, and space-required, for which the press of space button indicated the end of a word. The experiment showed that the finger-released version of the bimanual keyboard achieved about 85% input speed of the conventional unimanual gesture keyboard. And the error rate was higher for finger-released bimanual (3.3%) than unimanual (2.8%). The space-released bimanual keyboard was not as good as the finger-released one. Although the bimanual keyboard was not as efficient as the unimanual keyboard, it takes only 25% (landscape) and 47% (portrait) space of the unimanual keyboard, and users expected about 50% less keystroke length when typing. Figure 12 shows the comparison between unimanual and bimanual gesture keyboards.

In conclusion, using a gesture keyboard is a proper way to balance the input performance and the keyboard size. Thus, it should be considered when trying to improve the overall user typing experience.

2.8 Considering Postures

Some users preferred to type with two thumbs on touch screen mobile devices and others preferred using index fingers. The typing pattern may be influenced by different postures and thus influence typing performance and user experience. Researches have been performed to explore the relationship between typing postures and typing patterns.

Mackenzie and Soukoreff (2002) surveyed the model of two-thumb text entry on the physical keyboard on PDA, which is the earliest research about the typing posture on mobile devices. They predicted an upper bound of text entry rate of 60.74 wpm based on the linguistic and motor component of the interaction.



(a) Vertical Offset

(b) Horizontal Offset

Figure 13: Postures VS Touch Points Offset from (Azenkot and Zhai, 2012)

Azenkot and Zhai (2012) compared the touch behavior of different typing postures, namely, one finger, one thumb, and two thumbs. They found that text entry with two thumbs was the fastest (50.03 wpm), one finger was far lower (36.34 wpm) and one thumb was the lowest (33.78 wpm). The error rate exhibited the same pattern as the entry rate: the higher the text entry rate, the higher the error rate. They also explored the touchpoints of all three postures. Vertically, touchpoints of all postures tend to be higher than the center of the key. And horizontally, touchpoints of one finger tend to be close to the center of the keyboard, that is,

positive offset for keys on the left and negative offset for keys on the right; touchpoints of two thumbs tend to be close to the edge of the keyboard, that is, negative offset for keys on the left and positive offset for keys on the right; touchpoints of one thumb were similar to that of one finger but the pattern was not symmetric. Figure 13 shows the typing pattern. Overall, all postures minimized their finger movement when typing.

This research provided guidance for keyboard design when considering postures. The distribution and the decoding algorithm of the keyboard could be customized based on users' different typing postures.

ContextType (Goel et al., 2013) was a method using typing postures to improve the mobile touch screen text entry. Researchers first collected data for all four typing postures: using two thumbs, the left thumb, the right thumb, or the index finger. Then a touch pattern model was built based on that data and applied to the software keyboard. The keyboard would automatically detect one of the four typing postures based on information collected before while the user was typing and automatically switch to another algorithm if the change was detected. The experiment showed the typing speed was not improved but the error rate was lowered by 20.6% for all four postures by the application of ContextType. This experiment confirmed the theory proposed in Azenkot and Zhai (2012) and proved the adaptation of posture context could indeed make some improvement to the typing experience.

2.9 Algorithm Enhancement

The improvement in the algorithm could significantly improve the typing experience. And usually, a new algorithm was combined with some other technologies or research findings, such as Invisiboard (Mottelson et al., 2016) combined a new reduced keyboard with new decoding algorithm; ContextType (Goel et al., 2013) combined the keyboard with a posture detection algorithm, etc. In this subsection, several approaches concentrating on the algorithm design to enhance typing were introduced.

Clawson et al. (2008) designed a system – Automatic Whiteout++ to detect and correct errors made by typists on physical mini-QWERTY keyboard. In their previous system Automatic Whiteout, the algorithm could already detect typos like row substitution error (e.g. cat to cay), key repeat error (e.g. cat to caat), roll-on insertion error (e.g. cat to cart) and roll-off insertion error (e.g. cat to catr). In this new system,

bi-letter and tri-letter frequency were introduced to help with error detection. And a decision tree was used to decide whether or not a word was a typo. The experiment showed the new system could detect and correct up to 32.37% of errors. One drawback of this system was that it did not use the dictionary information, maybe it was constrained by the computing power for mobile devices at that time, combining of which may lead to a better result.

Vertanen et al. (2015) proposed VelociTap, a pure keyboard decoder that supported a sentence-based text entry approach. The decoder used a probabilistic keyboard model, a character language model, and a word language model. The algorithm was designed by adding the keyboard model log probability to the language model log probability multiplied by a scaling factor. And the result was used to search for the top 50 probable hypotheses. The system was trained with billions of words from blogs and social media. The experiment result indicated the system could achieve a similar text entry speed as Google keyboard but a much lower error rate.

In addition to the decoder algorithm improvement, key resizing and adaptation algorithms were popular and effective as well. Findlater and Wobbrock (2012) proposed an algorithm based on touch location, touch area, finger movement, previous keypress, and the position of hand and arm. The keys with the top probability for the next candidate were enlarged or moved slightly to a preferred position. The result showed that while remaining a low error rate, the adapted keyboard outperformed the conventional one.

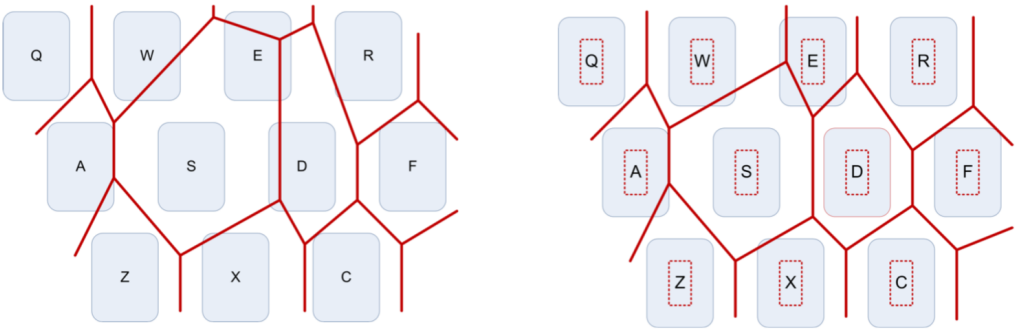


Figure 14: Keyboard adaptation (Gunawardana et al., 2010), left: unconstrained, right: constrained

Furthermore, Gunawardana et al. (2010) did usability research on auto-adaptive keyboards, the result showed those keyboards could lower the error rate. However, the authors also advised the designers to add constraints to the adaptation algorithm (Figure 14(right)), since a too extreme algorithm would make it impossible

to tap some keys, as it is shown in Figure 14(left).

Finally, Yin et al. (2013) combined the posture adaptation and keyboard layout adaptation together. Using a hierarchical spatial backoff model with submodels of different levels of complexity, they reduced the language-model-independent error rate by 13.2%.

2.10 Text Entry User Experience Improvement: Conclusion

In this section, potential methods to improve the user experience when typing were summarized. Shrinking the keyboard, moving the keyboard to the back, or making keyboard invisible could be used to maximizing the display area. Modifying the keyboard layout, adding physical feedback, and other methods related to postures or gestures could be used to improve the typing speed or lower the error rate or both. And designing a new decoding/adaptive algorithm could be beneficial in all aspects. When designing a highly efficient and user attracting keyboard on mobile devices, the factors in this section should be considered.

3 Keyboard Evaluation

This section briefly discussed tools and criteria used for keyboard evaluation. Most papers discussed in this section have been mentioned in the previous section. In total 36 papers were examined.

3.1 Subjects

The subject selection is essential for experiment design. 25 papers mentioned their participant selection. The number of participants ranged from 5 to 45, with an average of 16 and a mode of 12. Seven papers mentioned their participants were familiar with the QWERTY keyboard on mobile devices before. Two papers (Zhu et al., 2018; Azenkot and Zhai, 2012) mentioned that their participants all had experience on the touch screen. Another two (Bao et al., 2011; Findlater et al., 2011) hired both normal people and expert typists in their experiment. For one study related to posture (Wobbrock et al., 2008), participants were required to be right-handed and for one study related to keyboard evaluation (Hong et al., 2015), participants were required to be native in English.

3.2 Phrase Sets

When evaluating a topic related to text entry, participants enter texts of phrases provided by researchers and the corresponding data are collected. The selected phrase sets should be moderate in length, easy to remember, and representative of the target language. 20 papers mentioned their phrase sets selection. 14 of them used a phrase set from (MacKenzie and Soukoreff, 2003). It was a collection of 500 phrases that met the criteria mentioned above and was widely used for text entry experiments. Other experiments used phrases set that met their own demand such as a modification of UK dictionary (Komninos and Dunlop, 2014), email sentences with 1–12 words from the W3C email corpus, and the non-spam messages in the TREC 2005 corpus (Vertanen et al., 2015) and 12 four-letter words (Chen et al., 2014).

3.3 Speed

Typing speed is one of the most important criteria to evaluate the performance of a keyboard. The most common method to measure the typing speed is using words per minute (WPM). 23 reviewed papers used WPM to measure the speed. It is defined as:

$$WPM = \frac{|T| - 1}{S} \times \frac{1}{5} \times 60$$

where S is time measure in seconds, from the first keypress to the last. And $|T|$ is the length of the final text entered by the participant.

In addition to WPM, character per minute (CPM) can also be used. CPM was used in (Bi et al., 2010). And for some special keyboards, the time to enter each character are not the same, e.g. in SwipeBoard (Chen et al., 2014), entering letter A was much faster than entering letter N. Therefore, character entry time, which is the time between two valid key hits, was used to record entry time for each character. Three papers (Hoggan et al., 2008; Komninos and Dunlop, 2014; Bi et al., 2010) also used phrase completion time, which is the average time to complete each phrase, to measure the typing speed.

3.4 Error Rate

The error rate is another essential aspect to measure the text entry performance. It is more complicated to measure error than speed. Here are some notations defined by (Soukoreff and MacKenzie, 2003):

- $|\cdot|$: length of something.
- *Presented Text(P)*: text a participant should enter.
- *Transcribed Text(T)*: text a participant actually entered.
- *Input Stream(IS)*: all keystrokes sequence while entering P.
- *Correct(C)*: correct characters in T.
- *Incorrect Not Fixed(INF)*: incorrect characters in T.
- *Fixes(F)*: modifier keys and backspace, delete and navigation keys.
- *Incorrect Fixed(IF)*: character in IS but not in F and not in T.
- *Minimum String Distance(MSD)*: minimum number of insertion, deletion or substitution of a single character to transform T into P.

Several commonly used error measurement method are:

- *Error Rate(ER)*:= $\frac{INF}{|T|} \times 100\%$
- *MSD Error Rate(MSD ER)*:= $\frac{MSD(P,T)}{Max(P,T)} \times 100\%$
- *KeyStroke Per Character(KSPC)*:= $\frac{|IS|}{|T|}$
- *Erroneous Keystroke Error Rate(EKS ER)*:= $\frac{INF+IF}{|P|} \times 100\%$
- *Total Error Rate(Total ER)*:= $\frac{INF+IF}{C+INF+IF} \times 100\%$
- *Not Corrected Error Rate(Not Corrected ER)*:= $\frac{INF}{C+INF+IF} \times 100\%$
- *Corrected Error Rate(Corrected ER)*:= $\frac{IF}{C+INF+IF} \times 100\%$

In the reviewed papers, five used ER, three used KSPC, three used corrected and uncorrected ER, three used MSD, and three used total ER. Soukoreff and MacKenzie (2003); Soukoreff (2010); Arif and Stuerzlinger (2009) thoroughly discussed the usage and the advantages/drawbacks for each error rate.

3.5 Workload

A keyboard cannot be considered as good if it takes too much mental workload of the user. In the reviewed papers, six used NASA TLX (Hart, 1986) to measure the mental workload of the subjects.

3.6 Subjective Rating

The users' preferences and comments are also important for keyboard evaluation. Three in reviewed papers contained designed questions to ask for the subjective rating and preference of the keyboards. And one (Zhai et al., 2009) collected comment data from App Store to evaluate the keyboard.

3.7 Learnability

When measuring the performance, enough time should be given to the participants to get familiar with the new keyboard, in order to get a reasonable and objective result. Ten in reviewed papers provided training sessions (from 20 minutes to 2 days) for participants to get accustomed to the keyboard or reporting the result for the first session and the last session separately, to give the readers a sense of the learnability of the design.

3.8 Touch Points Analysis

Analysis of the user touchpoints is a good way to explore the typing pattern of participants and a good way to get the reference to adjust the layout, design, and materials of the keys/keyboards. Two papers plotted the heatmap of the user touchpoints and the other three analyzed the vertical and horizontal offset between the user hit points and the center of each key.

4 Conclusion

This paper first summarized methods to improve the text entry user experience from nine aspects. Then it presented things to consider when evaluating a keyboard. It could serve as a reference for starter researchers in this field. One missing point in this paper is about the language models, which can be critical when suggesting words as user typing. Different language model could significantly influence the performance and users' experience of typing. However, the language model is basically pure mathematics, which is beyond the understanding of the author. A survey about different language models should be added in the future. Also, great design and performance do not necessarily mean the success of a keyboard. Since the public has been using the QWERTY keyboards for so long, it is not realistic to replace them with a new one in a short time. An ingenious advertising and a slow transition should be considered.

References

Arif, A. S. and Stuerzlinger, W. (2009). Analysis of text entry performance metrics. In *2009 IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH)*, pages 100–105.

Azenkot, S. and Zhai, S. (2012). Touch behavior with different postures on soft smartphone keyboards. In *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '12, page 251–260, New York, NY, USA. Association for Computing Machinery.

Bao, P., Pierce, J., Whittaker, S., and Zhai, S. (2011). Smart phone use by non-mobile business users. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, page 445–454, New York, NY, USA. Association for Computing Machinery.

Bi, X., Chelba, C., Ouyang, T., Partridge, K., and Zhai, S. (2012). Bimanual gesture keyboard. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, UIST '12, page 137–146, New York, NY, USA. Association for Computing Machinery.

Bi, X., Smith, B. A., and Zhai, S. (2010). Quasi-qwerty soft keyboard optimization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, page 283–286, New York, NY, USA. Association for Computing Machinery.

Chen, X. A., Grossman, T., and Fitzmaurice, G. (2014). Swipeboard: A text entry technique for ultra-small interfaces that supports novice to expert transitions. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, UIST 14, page 615–620, New York, NY, USA. Association for Computing Machinery.

Clawson, J., Lyons, K., Rudnick, A., Iannucci, R. A., and Starner, T. (2008). Automatic whiteout++: Correcting mini-qwerty typing errors using keypress timing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, page 573–582, New York, NY, USA. Association for Computing Machinery.

Dhakal, V., Feit, A. M., Kristensson, P. O., and Oulasvirta, A. (2018). Observations on typing from 136 million keystrokes. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–12, New York, NY, USA. Association for Computing Machinery.

Findlater, L. and Wobbrock, J. (2012). Personalized input: Improving ten-finger touchscreen typing through automatic adaptation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, page 815–824, New York, NY, USA. Association for Computing Machinery.

Findlater, L., Wobbrock, J. O., and Wigdor, D. (2011). Typing on flat glass: Examining ten-finger expert typing patterns on touch surfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, page 2453–2462, New York, NY, USA. Association for Computing Machinery.

Goel, M., Jansen, A., Mandel, T., Patel, S. N., and Wobbrock, J. O. (2013). Contexttype: Using hand posture information to improve mobile touch screen text entry. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, page 2795–2798, New York, NY, USA. Association for Computing Machinery.

Gordon, M., Ouyang, T., and Zhai, S. (2016). Watchwriter: Tap and gesture typing on a smartwatch miniature keyboard with statistical decoding. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, page 3817–3821, New York, NY, USA. Association for Computing Machinery.

Green, N., Kruger, J., Faldu, C., and St. Amant, R. (2004). A reduced qwerty keyboard for mobile text entry. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '04, page 1429–1432, New York, NY, USA. Association for Computing Machinery.

Gunawardana, A., Paek, T., and Meek, C. (2010). Usability guided key-target resizing for soft keyboards. In *Proceedings of the 15th International Conference on Intelligent User Interfaces*, IUI '10, page 111–118, New York, NY, USA. Association for Computing Machinery.

Hart, S. G. (1986). Nasa task load index (tlx). volume 1.0; paper and pencil package.

Hoggan, E., Brewster, S. A., and Johnston, J. (2008). Investigating the effectiveness of tactile feedback for mobile touchscreens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, page 1573–1582, New York, NY, USA. Association for Computing Machinery.

Hong, J., Heo, S., Isokoski, P., and Lee, G. (2015). Splitboard: A simple split soft keyboard for wristwatch-sized touch screens. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 1233–1236, New York, NY, USA. Association for Computing Machinery.

Kim, H., Row, Y.-k., and Lee, G. (2012). Back keyboard: A physical keyboard on backside of mobile phone using qwerty. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '12, page 1583–1588, New York, NY, USA. Association for Computing Machinery.

Komninos, A. and Dunlop, M. (2014). Text input on a smart watch. *IEEE Pervasive Computing*, 13(04):50–58.

Lee, S. and Zhai, S. (2009). The performance of touch screen soft buttons. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, page 309–318, New York, NY, USA. Association for Computing Machinery.

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 *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, page 461–470, New York, NY, USA. Association for Computing Machinery.

Mackenzie, I. and Soukoreff, R. (2002). A model of two-thumb text entry. *Proceedings - Graphics Interface*.

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*, CHI EA '03, page 754–755, New York, NY, USA. Association for Computing Machinery.

MacKenzie, I. S. and Zhang, S. X. (1999). The design and evaluation of a high-performance soft keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '99, page 25–31, New York, NY, USA. Association for Computing Machinery.

Mottelson, A., Larsen, C., Lyderik, M., Strohmeier, P., and Knibbe, J. (2016). Invisiboard: Maximizing display and input space with a full screen text entry method for smartwatches. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '16, page 53–59, New York, NY, USA. Association for Computing Machinery.

Oney, S., Harrison, C., Ogan, A., and Wiese, J. (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, page 2799–2802, New York, NY, USA. Association for Computing Machinery.

Scott, J., Izadi, S., Rezai, L. S., Ruszkowski, D., Bi, X., and Balakrishnan, R. (2010). Reartype: Text entry using keys on the back of a device. In *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '10, page 171–180, New York, NY, USA. Association for Computing Machinery.

Soukoreff, R. W. (2010). *Quantifying Text Entry Performance*. PhD thesis, CAN. AAINR64939.

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 *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, page 113–120, New York, NY, USA. Association for Computing Machinery.

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 *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, page 659–668, New York, NY, USA. Association for Computing Machinery.

Wobbrock, J. O., Myers, B. A., and Aung, H. H. (2008). The performance of hand postures in front- and back-of-device interaction for mobile computing. *Int. J. Hum.-Comput. Stud.*, 66(12):857–875.

Yin, Y., Ouyang, T. Y., Partridge, K., and Zhai, S. (2013). Making touchscreen keyboards adaptive to keys, hand postures, and individuals: A hierarchical spatial backoff model approach. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, page 2775–2784, New York, NY, USA. Association for Computing Machinery.

Zhai, S. and Kristensson, P.-O. (2003). Shorthand writing on stylus keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, page 97–104, New York, NY, USA. Association for Computing Machinery.

Zhai, S., Kristensson, P. O., Gong, P., Greiner, M., Peng, S. A., Liu, L. M., and Dunnigan, A. (2009). Shapewriter on the iphone: From the laboratory to the real world. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '09, page 2667–2670, New York, NY, USA. Association for Computing Machinery.

Zhu, S., Luo, T., Bi, X., and Zhai, S. (2018). Typing on an invisible keyboard. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–13, New York, NY, USA. Association for Computing Machinery.