MAUU(D)5900

MASTER THESIS

in

Universal Design of ICT

October 2018

An Accessible Directions-based Text Entry Method Using Two-thumb Touch Typing

Linghui Ye

Department of Computer Science

Faculty of Technology, Art and Design

OSLO AND AKERSHUS
UNIVERSITY COLLEGE
OF APPLIED SCIENCES

Contents

Abstract

Text entry on smartphone is a common activity in people's daily lives, and this activity heavily relies on users' visual feedback and advanced motor function. Generally, the average touchscreen size of current smartphones is between 5 and 5.6 inches and can be difficult for users who are visually impaired or have reduced motor function to input text. Thus, this study proposed an eyes-free text entry strategy for smartphones based on bimanual gestures and QWERTY layout. The QWERTY keyboard layout was split into two symmetric sections and each part contains multiple characters. The users entre text by moving their thumbs in the direction of the desired characters. Furthermore, a longitudinal user study with 20 participants was performed to evaluate the proposed text entry method. During the four training sessions, the participants achieved text entry speeds of 11.1 WPM in eyes-free mode and 14.1 WPM in eyes-on mode. The experiment results show that the proposed method holds potential for supporting users with low vision and certain types of reduced motor function for entry text on smartphones.

Keywords

Text entry; Smartphone text entry; Accessibility; Eyes-free text input; Bimanual input.

Acknowledgement

Foremost, I would like to express my gratitude to my supervisor professor Frode Eika Sandnes for his continuous support of my master study and research. During this period, I have experienced pregnancy and birth, without his on-going support and encouraging, I could not have completed it. He has offered a lot of helpful feedback and valuable comments during the whole master project. I have learned so much from him. Many thanks also go to my coordinate professor Norun Christine Sanderson who was always very helpful and friendly.

I would also like to offer my great thanks to my friends and my class fellows who participated the evaluation experiment. Since the confidential agreement I cannot list their names. They have always been incredibly positive and patient during the four sessions' test. Moreover, they offered their help without requesting for any return.

Finally, I am particularly grateful for my family, especially my husband, for his encouragement and support. Special thanks should be also given to my parents who came to Norway to help me take care of my children, so that I could spare more effort to my master work.

1. Introduction

Touch-based smartphones are handheld personal computers that performs many computer functions such as Internet access and running applications on the mobile operating system (Android and IOS). One of the main tasks performed on smartphones is the input of text, including sending SMSs, writing e-mails, typing notes, filling in forms, posting social media updates, completing documents, etc. The text entry on smartphones is heavily reliant on visual feedback and users with low vision and/or reduced motor function may find it challenging to input text given the small touchscreen size. Hence, this study proposed a text entry strategy for the touchscreen and implemented the prototype on a smartphone. A longitudinal mixed experiment design with four sessions was conducted to evaluate the porotype's effectiveness (speed and accuracy) and learning effects. According to a report by Zhai et al. (2004), a good text entry method should have certain attributes. First, it should be efficient. After practice users should be input text at a sufficiently fast speed without making many errors. Next, it should be easy to learn and use with a low learning threshold. Then, it should impose a low cognitive, perceptual, and motor demand on users. Last, it should be fun to use. Fun can be a source for people to use a technique frequently and constantly (Zhai, Kristensson, & Smith, 2004).

Currently, most smartphones employ the QWERTY keyboard layout which most users are familiar with. Although there are many optimized alternative software keyboard layouts, such as the Dvorak keyboard (Cassingham, 1986), Colemak keyboard layout (COLEMAN, 2013), and Fitaly keyboard (MacKenzie & Zhang, 1999), while, these keyboard layouts usually require users to invest effort to learn. Most users are unlikely to accept totally new keyboard layouts. Thus, several researches optimized QWERTY layout by combining some algorithms such as Metroplis energy minimization algorithms with QWERTY keyboard layout to reduce finger move distance (Bi, Smith, & Zhai, 2010; Zhai & Smith, 2001). Furthermore, to reduce the visual search time, some attempts have therefore employed prediction algorithm-based language model which is able to dynamically add characters based on the entered text context (Raynal, 2014). Regarding to the proposed text entry strategy, to lower the learning threshold, the proposed text entry method is based on the QWERTY layout. Most smartphone users are familiar with the QWERTY layout, and some of them are able to input text without even having to look at the individual keys. Bi et al. (2010) reported that users usually tend to refrain from learning reduced or rearranged keyboards but wish to use QWERTY layout which they are already familiar with (Bi et al., 2010). Furthermore, QWERTY keyboards have performance merits compared to other soft keyboards. MacKenzie et al. (1999) conducted an empirical test to compare the text entry rates among six keyboards (QWERTY, ABC, Dvorak, Fitaly, JustType, and telephone) and found the text entry rate of the QWERTY soft keyboard to be the fastest among all the six soft keyboards (MacKenzie, Zhang, & Soukoreff, 1999). Moreover, a new technique requires users to invest time in learning. Thus, the proposed text entry strategy is based on the QWERTY character arrangement. This text entry method is less suitable for some users who are born with visual and motor disabilities, because they may not be familiar with the QWERTY keyboard layout.

Currently, the average touchscreen size of a smartphone is between 5 and 5.6 inches. The virtual keyboards on smartphone consist of some rectangle soft keys. The key sizes were almost 15X20 mm with 2-mm between-key spacing. Due to the small form factor of the smartphones, it is difficult for users to locate the specific key precisely. Visual clues about virtual keys' location and the necessary motor control are required for users to successfully hit these small soft keys. Hence, using virtual keyboard to enter text on smartphones can be extremely difficult for individuals with low vision and/or reduced motor function. Especially for the aging population, usually these users experience reduced eyesight and motor function. Smartphone text entry can be particularly challenging for these individuals. Since the limited touchscreen size of the smartphone, ten finger touch typing is not an efficient way for smartphone text entry. Although some attempts have employed a Markov-Bayesian pattern matching method to realize ten-finger touch typing on touch interface and achieved text entry rates at 45 WPM (Shi, Yu, Yi, Li, & Shi, 2018), the accuracy of this text entry strategy needs to be improved. Generally, most users enter text on smartphone using their single index finger, two thumbs or a single thumb (Azenkot & Zhai, 2012). The user study has conducted to compare the three hand postures text entry speeds on smartphone, and the results showed that the two thumbs were associated with a higher text entry rate (50 WPM) compared to the one-finger (36 WPM) and the single Thumb (33 WPM) text entry speeds. With the two thumbs working together, the finger travel distance can be reduced, which could improve the text entry rate. Moreover, the text entry error rate of the two-thumb posture was highest, at 10.80%. The lowest error rate was obtained with one-thumb input (7.00%), and single index finger text entry error rate was 8.17%. To compromise the text entry speed with the error rate, the proposed text input method employs two-thumb touch-typing. Buxton and Myers (1986) have pointed out that some input tasks can be accelerated if both hands are used collaboratively (Buxton & Myers, 1986). Several researches (Bi, Chelba, Ouyang, Partridge, & Zhai, 2012; MacKenzie & Soukoreff, 2002b; Sandnes & Aubert, 2007) also have addressed bimanual text input.

This master thesis presents a touchscreen text entry method. First, related works on text entry are described in Section 2. The proposed text entry strategy does not rely on precise pointing at on-screen targets. Instead of pointing, the method uses dragging in different directions, lowering the demand for visual feedback. The principle behind the text entry prototype is described in Section 3. A longitudinal user study involving 20 participants was performed to evaluate the proposed method. The methodology of this experiment is discussed in Section 4. The results and discussion are presented in Section 5. At last, the conclusion is provided in Section 6.

2. Related work

Smartphone-based text entry has been embedded in our work and life environment, and numerous researchers have worked on new ideas to improve text entry techniques. Text entry research has a long history. There are mainly two categories of text entry inventions. One part of the studies focuses on how users input text, including speech input, gesture and gaze-based input. The other studies pay attention to the actual keyboards including physical and soft keyboards and keyboard layout.

Speech is the most natural form of communication between humans, while, speech is not a widely used text entry technique. This can be attributed in part to the variations in error rates when processing speech (Munteanu & Penn, 2015). Error correction is particularly difficult with speech commands. Furthermore, Karat et. (1999) have reported that the effective speed of text entry by continuous speech recognition was lower than keyboard based text entry (Karat, Halverson, Horn, & Karat, 1999). Another way to input text is via gestures (Goldberg & Richardson, 1993; Wobbrock, Myers, Aung, & LoPresti, 2004). Touchscreen gestures such as tapping, dragging, holding, pinching, etc. are commonly used in touch interface interaction. These touchscreen gestures are not associated with hitting a specific absolute target. Compared to hitting absolute targets, touchscreen gestures are more natural interactions and less rely on users' visual ability and advanced motor function. Hence, gestures can be used for some users with reduced eyesight and/or users with certain types of degenerated motor functions, such as the old people with presbyopia and trembling hands are not able to accurately hit the absolute targets. Handwriting recognition is one of representative text entry approach using gestures. Since the general users have experience with writing on paper, handwriting recognition is easy and acceptable approach to most people. Moreover, handwriting recognition technology has made tremendous progress in recent years (Plamondon & Srihari, 2000). Some devices have used alphabet character based handwriting recognition, such as Graffiti and Unistroke (Castellucci & MacKenzie, 2008). Since the alphabet used can be reliably recognized, it is widely used for different users. For example, a styles-based unistroke input technique – EdgeWrite defines the edge characters which are similar to the alphabets to help users with motor impairment (Wobbrock, Myers, & Kembel, 2003). People have high expectations toward handwriting recognition technique, while, it has limited speed (Zhai et al., 2004). Experiments have shown that simpler gestures such as UniStroke is faster (15.8 WPM) than the more complex Graffiti gestures (11.4 WPM) as there is less distance for the fingers to travel and hence may give faster text entry rates (Castellucci & MacKenzie, 2008). Attempts have also been made to input text using single stroke gestures, for example, navigating menu hierarchies to retrieve a specific letter using some simple gestures (Sandnes et al., 2012). Besides that some techniques such as Quikwriting (Perlin, 1998) and cirrin (Mankoff & Abowd, 1998) combined absolute hitting targets and gestures to enter characters with continuous stylus movement. The recently approach in this category is Swype (Cuaresma & MacKenzie, 2013) which is an application on Android devices. Swype is a soft keyboard for touchscreen smartphone which uses shape writing recognition for inputting words. Users employ Swype to enter text by drawing the word in one continuous motion without lifting the finger on the QWERTY keyboard. While, the fundamental weakness is that it relies heavily on users' visual feedback, since the users must continuously recognize the rearranged characters. Other text entry strategies target a wide range of text entry paradigms, including tilt (Wigdor & Balakrishnan, 2003; Yeo, Phang, Castellucci, Kristensson, & Quigley, 2017), multi-tap keystroke-based techniques and joystick -based techniques(Chau, Wobbrock, Myers, & Rothrock, 2006; Silfverberg, MacKenzie, & Kauppinen, 2001; Wobbrock, Aung, Myers, & Lopresti, 2005).

With respect to the keyboard, several years ago most mobile phones were assembled with physical keypads. The most commonly used physical keyboard is the 12-key keypad where each key corresponds to multiple characters. The ambiguity of multiple possible characters is commonly resolved by multitap or lexical models (MacKenzie & Soukoreff, 2002c). After that some mobile phones with small physical QWERTY keyboards, such as BlackBerry and Nokia Communicator, became popular. But, these keyboards were small, only suitable for one or two finger input text. Hence, the text entry performance needs improvement. Now smartphones employ touchscreen technology instead of physical keys. The devices use on-screen soft keyboards. The most common soft keyboard is QWERTY which has been a standard layout for more than 100 years, it is widely used (Noyes, 1983). Unfortunately the QWERTY keyboard yields limited performance since common consecutive character pairs appear on the polarizing positions of the keyboard (Bi et al., 2010). This leads to more frequent movements and greater move distances when people enter text. To improve the motor movement efficiency of the standard QWERTY layout, many researches focused on rearranging the characters of the QWERTY layout. For example, Dvorak layout which reduce finger movement distance by placing the most common characters along the home row (Lewis, Potosnak, & Magyar, 1997). Opti is another optimized soft keyboard layout, with frequent characters positioned in the center, infrequent characters in the perimeter (MacKenzie & Zhang, 1999; MacKenzie et al., 1999).

Although the current smartphone touchscreen size has become large, only less than half of the screen size is used for displaying the soft keyboard. It is not easy for users to locate characters due to the small key size, condensed layout and no tactile feedback from the soft keyboard (Sears & Zha, 2003). Furthermore, to enter text on touchscreens requires constant visual attention, thus some studies focus on eyes-free interaction text entry. For example, Jain and Balakrishnan (2012) proposed that using bezel, the physical touch-insensitive frame surrounding a touchscreen display, as a text entry method for eyes-free interaction (Jain & Balakrishnan, 2012). Kim et al. (2012) designed a physical QWERTY keyboard on the back of the mobile phone (Kim, Row, & Lee, 2012). Since users cannot refer to the keyboard when they are entering text, the performance of the backside keyboard depends on user's spatial memory of QWERTY layout.

Furthermore, there is no tradeoff between the size of the keyboard and text entry accuracy (Dunlop, Durga, Motaparti, Dona, & Medapuram, 2012). Given that, some studies worked on reducing the number of QWERTY keys. One of notable way to reduce the size of the QWERTY keyboard is the half- QWERTY keyboard where the QWERTY keyboard is split into two halves

(Matias, MacKenzie, & Buxton, 1994). The half- QWERTY keyboard is small, the layout is familiar to users, and text can be inputted by using one hand only on half the keyboard (Matias, MacKenzie, & Buxton, 1993). Dunlop et al. (2012) proposed a semi-ambiguous keyboard for English – QWERTH which kept “D”, “F”, “G”, and “V” keys unique and grouped two characters as a key in the remaining characters. This keyboard layout not only smaller than QWERTY, but also caused less prediction problems (Dunlop et al., 2012). Chording keyboards which yield a high input rates have fewer keys. It requires that users have sufficient dexterity in their fingers as the text is entered by pressing multiple keys simultaneously (MacKenzie & Soukoreff, 2002c), and chording has also been used for achieving eyes-free digit input on smartphones (Azenkot, Bennett, & Ladner, 2013). Encoding is another approach to reduce the number of keys. Each character corresponds to a unique sequence of key presses (Boissiere & Dours, 2003; Jones, 1998). MacKenzie et al. (2011) designed an efficient four-key text entry method by using Huffman coding to assign minimized key sequences to characters (MacKenzie, Soukoreff, & Helga, 2011).

Text prediction and suggestions are also an efficient way to improve the performance of a text entry. It is unavoidable to entry error characters due to user’s imprecision of finger touch and spelling errors. Typically, there are two ways to provide appropriate suggestions to the users. On one hand, predicting unfinished characters based on user’s partial input, which is to reduce the number of input text characters and enhance the text entry speed. On the other hand, correcting user’s erroneous input, which is to improve the accuracy of the text entry strategy. several studies reported many results on text prediction. For example, the Reactive Keyboard (Darragh, Witten, & James, 1990) predicted words by finding the longest matching substrings in the previously entered text by using an adaptive dictionary-based language model. Goodman et al. (2002) combined language models with key press model to find the most probable key sequence when user entering text on soft keyboard (Goodman, Venolia, Steury, & Parker, 2002). Furthermore, Kristensson and Zhai (2005) proposed a geometric pattern matching technique to provide corrected suggestions to the users (Kristensson & Zhai, 2005). These text prediction inventions enhance the performance of the text entry methods in some extent.

Although, many researchers have made the text entry more accessible for people by using various approaches, e.g. changing the layout, reducing the number of keys, and providing appropriate suggestions (Bi, Ouyang, & Zhai, 2014; Gunawardana, Paek, & Meek, 2010; Rodrigues, Carreira, & Gonçalves, 2016), these solutions do not totally fit people with certain disabilities (visually impaired and motor-impaired users), such as older adults. For example, the characters on the soft keyboard are difficult to read for a user who is visually impaired. External Braille keyboards are expensive and cumbersome to transport (Guerreiro, Lagoá, Nicolau, Santana, & Jorge, 2008). Therefore, some of them use voice interaction, such as Apple's VoiceOver or TalkBack on Android, to input text on smartphone. These voice text entry methods provide an alternative for blind and visually impaired users to entering text on smartphone. Generally, these voice interaction techniques provide some acoustic prompts to the users, and they tap on the touchscreen based on these speech feedback and locate the position of the required letters on the virtual keyboards. Kane et al. (2008) conducted an experiment to compare audio-based multi-touch interaction techniques with button-based interaction strategy, and the results showed the voice interaction mode was more acceptable by blind users, although it is error prone (Kane, Bigham, & Wobbrock, 2008). Besides that, the text entry speed of voice interaction is rather slow. Oliveira et al. (2011) performed a study with 13 blind users who used Apple VoiceOver with QWERTY keyboard layout to entry text on a smartphone and the mean of text rate was only 2.1 WPM (Oliveira, Guerreiro, Nicolau, Jorge, & Gonçalves, 2011). Some researchers therefore aim to improve the performance (including both speed and accuracy) of voice interaction technology. One of attempts was using pseudo-pressure detection with QWERTY keyboard to implement a text-to-speech interface by using finger touching to provide exploration voice feedback and finger pressing to confirm the desired characters (Goh & Kim, 2014). Furthermore, this voice interaction is not always accurate, and it is difficult to use in noisy environment (Leporini, Buzzi, & Buzzi, 2012). Thus, some researchers focused on providing suitable text entry strategies for visually impaired users. Pollmann et al. (2014) utilized a feature some modern smartphones offer: the user's finger can be detected while it is hovering about 2 cm above the screen. They used this feature to enlarge the area of the keyboard under the finger to make the on-screen keyboards accessible for visual impaired users (Pollmann, Wenig, & Malaka, 2014). Furthermore, an audio-tactile text entry prototype based on multitap was designed for blind users (Buzzi, Buzzi, Leporini, & Trujillo, 2014).

Studies for overcoming text entry barriers on smartphone for individuals with limited motor function are limited. Some researchers utilized eye-tracking device with language modelling based graphic interface enable hands-free text entry, which could be convenient for users with limited hand movement (Ward, Blackwell, & MacKay, 2000). Eye gazing based text entry interaction is another direction to solve the text entry issues for motor impaired users (Kurauchi, Feng, Joshi, Morimoto, & Betke, 2016). Users input text by gazing and dwelling on the desired character for an amount of time on the virtual keyboard. Therefore, the text entry speed is limited. While in 2015 an eye typing technique was designed for individuals with a motor disability, which does not require the user to dwell on the characters through eye-gaze interaction. It automatically filters out unwanted characters from the sequence of characters gazed when input text (Pedrosa, Pimentel, & Truong, 2015). Kurauchi et al. (2016) also proposed a dwell-time-free gaze-typing method – EyeSwipe for people with motor impairments (Kurauchi et al., 2016). Users swipe through a virtual keyboard with their eye gaze to enter text via EyeSwipe. Besides that, keyboard scanning technique is another text entry method for reduced motor function users. The principle of keyboard scanning is automatically traverse the virtual keyboards in regular steps and once the desired key or key-group is highlighted a selection is made (Polacek, Sporka, & Slavik, 2017). Hence, the user only needs to hit an absolute target. Clearly, keyboard scanning is slow compared to most other text entry techniques. MacKenzie and Ashtiani (2011) proposed a text entry system – BlinkWrite for severe motor impaired individuals who cannot use a traditional eye gazing-based text entry interaction (MacKenzie & Ashtiani, 2011). This system uses eye-blinks as a single-key combined with a scanning ambiguous keyboard to input text.

Throughout these related works, almost no text entry strategy can include disabled users without compromising the user experience for non-disabled users. Therefore, the goal of this study is to design an accessible text entry approach which can be used not only by non-disabled users but also some users with visual or motor impairment including older people. Generally, most older people have vison degeneration and many also have trembling or stiff fingers which are caused by age-related diseases such as arthritis, stroke, and multiple sclerosis (Harper & Yesilada, 2008). In fact, there is no clear boundary between users who are categorized as "disabled" and those who are not (Fuglerud, 2014). For example, some users are trying to input text on smartphones without their reading glasses at hand, which is similar to the situation faced by visually impaired users. Moreover, some users attempt sending SMS messages when while walking (Mizobuchi, Chignell, & Newton, 2005), which is similar to the situation faced by users with reduced motor function. There are various causes of motor and visual impairment, and these can impact on user's ability to enter text. Therefore, this study reports on an attempt to allow for text entry with single directional finger touch gestures. One benefit of these simple gestures is that they require less finger distance than traditional UniStroke-type gestures. Moreover, they are bimanual in nature. There is thus a potential for high text entry speeds. And the target users of the proposed text entry strategy in this study are non-disability, visually impaired but not totally blind, and reduced motor function people. The next section outlines the design and implement of the proposed text entry method.

3. The prototype

The principle behind the text entry method is to input text by using two-finger movements on a touchscreen interface. Characters are organized into a QWERTY layout which is split into two parts (see Figure 3.1). Based on Fitts' law (Fitts, 1954) the input performance should go up when the size of the UI (User Interface) elements grows. For that reason, this text entry method used semi-ambiguous keyboards based on QWERTY in which it grouped some keys together and kept some other keys unique. The left part of the QWERTY layout is assigned to the left virtual keyboard (see Figure 3.2) and remaining characters are assigned to the right virtual keyboard (see Figure 3.3). Besides that, the touchscreen is also divided into two parts, the left part of the screen is the control domain of the left virtual keyboard and the right screen part is used to input right QWERTY characters. However, it is not possible to split the touch area equally into two parts. The main reason is that the users may move their finger through the middle line of the touch area. Furthermore, the path of the finger movements is not always straight. Therefore, defining the touch down point as $DownPoint(\, x_{down}, y_{down})$, released point as $UpPoint(\, x_{up}, y_{up})$, and the width of touch area as W. The following pseudo code shows how to determine which side of the touchscreen the user touchdown.

$$If\ \left(w - Math.\max\left(x_{down}, x_{up}\right) > Math.\min\left(x_{down}, x_{up}\right)\right)\{$$

$$Then\ touchPoint\ is\ on\ left\ part\ of\ touch\ area\}\ else\{$$

$$The\ touchPoint\ is\ on\ right\ part\ of\ touch\ area\}$$

When users touch on the smartphone display they imagine that their left and right touch fingers are located between D-G and J-K, respectively. The visual hint will be shown when users' finger first touch on the screen. The green crosses in the Figure 3.2 and 3.3 indicate the two fingers' touch position. The users visualize the QWERTY keyboard layout in their head, if the desired key is on the left side of the keyboard, the users use their left thumb as with touch and vice versa. To retrieve a specific letter the user drags his/her respective thumb in the direction of the desired letter. Finally, the finger is released to output the character. For example, if the user wants to input a character ‘T’, he or she just puts left thumb on the left side of the touchscreen and drag his/her finger to northeast (up and right) direction and release.

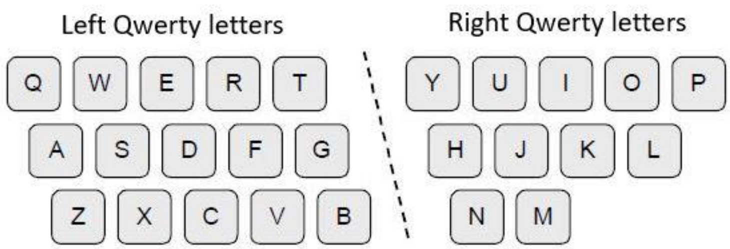


Figure 3.1 Left touch area inputting left QWERTY characters. Right touch area inputting right QWERTY characters (MacKenzie & Soukoreff, 2002b)

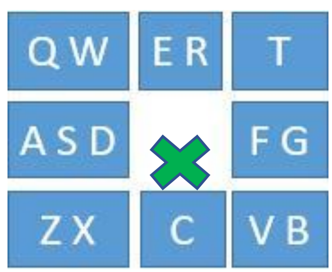


Figure 3.2 Left virtual keyboard. Green cross indicates the finger start touch point

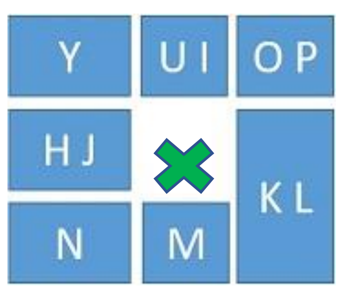


Figure 3.3 Right virtual keyboard. Green cross indicates the finger start touch point

Two text entry feedback modes, namely eyes-on mode application and eyes-free mode application were implemented for the android platform based on this prototype during phase I of the master project. These eyes-on and eyes-free terms refer only the gesture input region. In the eyes-free mode application, the left and right virtual keyboards (see Figure 3.2 and Figure 3.3) are hidden, unless help hints (see Figure 3.7) are requested. And in the eyes-on mode, the left and right virtual keyboards are shown. As for the user interface, there are two identical designs for the two applications. On one hand, the two textboxes on the top area of the smartphone touchscreen were used to present the original copy phrase and user's entered text. On the other hand, the bottom part of the smartphone touchscreen was used for inputting user gestures. Differences between these two applications are showed in Table 3.1. With respect to the eyes-on mode application, the bottom part of the display was also used for providing visual feedback on the QWERTY keyboard. Initially this area is blank. Once the users touch on this part of the screen, the display shows the groups for half a QWERTY keyboard layout. Specifically, if the finger is on the left side of the display the left half of the QWERTY keyboard (see Figure 3.2) is displayed, and if the first finger touch is on the right side of the display the right half of the QWERTY keyboard (see Figure 3.3) is displayed. Each of these halves are displayed as seven- or eight-character groups around the finger. And a specific character will be retrieved when user drag his/her finger to the direction of the desired character and releases the finger. For example, the user's finger touch down on the left part of touch area, and the left virtual keyboard appears. Next, when user's finger moves to northwest (*QW*), the selected characters group is "QW". And then the finger moves to northeast (*T*), the selected character is replaced by character group "T". Finally, the finger is released, and the character *T* is retrieved (see Figure 3.5). Apart from that, the eyes-on mode application providing visual feedback included a row below the text input field which provided some appropriate alternative suggestions for users to select when they input text. The eyes-on mode application just presented the suggestion word, rather than order the words in a list according to frequency. And the suggestion words were selected by tapping directly on the word. That is to say, the user could shortcut the input in the eyes-on mode by selecting a word based on a prefix, for instance, the user can select the desired word when it appears in the suggestion word list instead of entering all the characters. However, the users cannot refer to the left and right virtual keyboards, as well the suggestion word list in the eyes-free mode application. Therefore, users must only rely on their spatial memory of QWERTY layout to input text. Accordingly, simple audio feedback (a short beep each time a character was input) is used to confirm to users that they have successfully input a character. Additionally, when users are uncertain about how to retrieve a character, they can look up the help hints by moving their two fingers up on the touchscreen and the help hints will be shown for 1.5 seconds (see Figure 3.7).

Table 3.1 Difference between eye-mode and eyes-free mode application

Interaction Modality	Eyes-on mode	Eyes-free mode
Visible virtual keyboards	✓	
Suggestion word list	✓	
Help hints		✓
Audio feedback		✓

The layout of these two applications are similar. The user interface (UI) of eyes-on mode application comprises of four parts (see Figure 3.4). The presented text field shows text to be input, the entered text area displays the input text, the word suggestion area contains predictive words, and the remaining bottom part is the touch area on where users move their fingers to input characters. While, the eyes-free mode application's layout has no word suggestion area (see Figure 3.6).

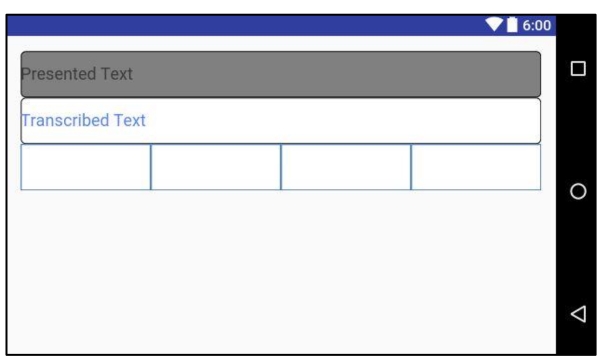
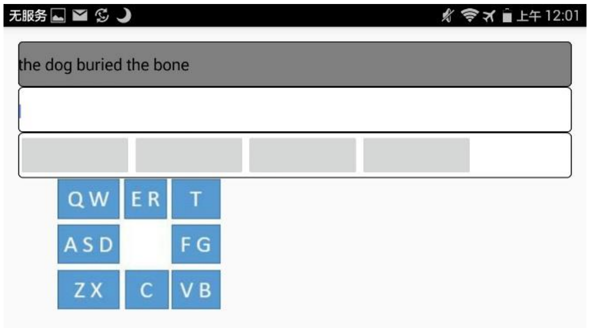
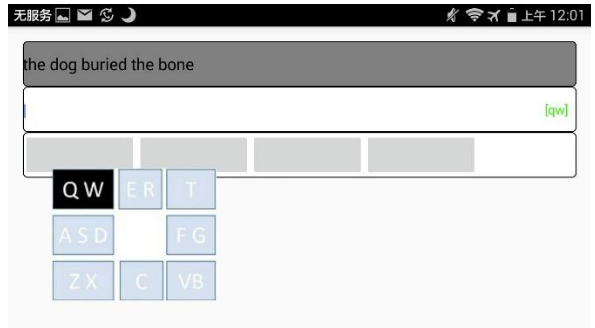


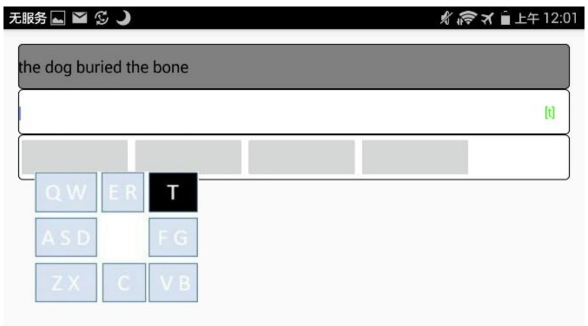
Figure 3.4 The layout of eyes-on mode application



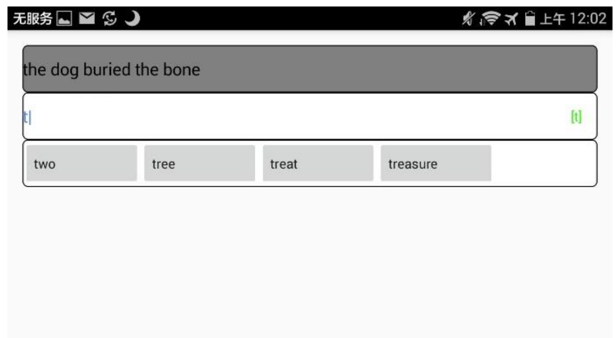
(a) The goal is to input the character *t*. Pressing the left side of the display.



(b) Moving the finger incorrectly in the north-west direction.



(c) Moving the finger to the north-east position.



(d) Releasing the finger, producing the character t.

Figure 3.5 Text input procedure with eyes-on mode application

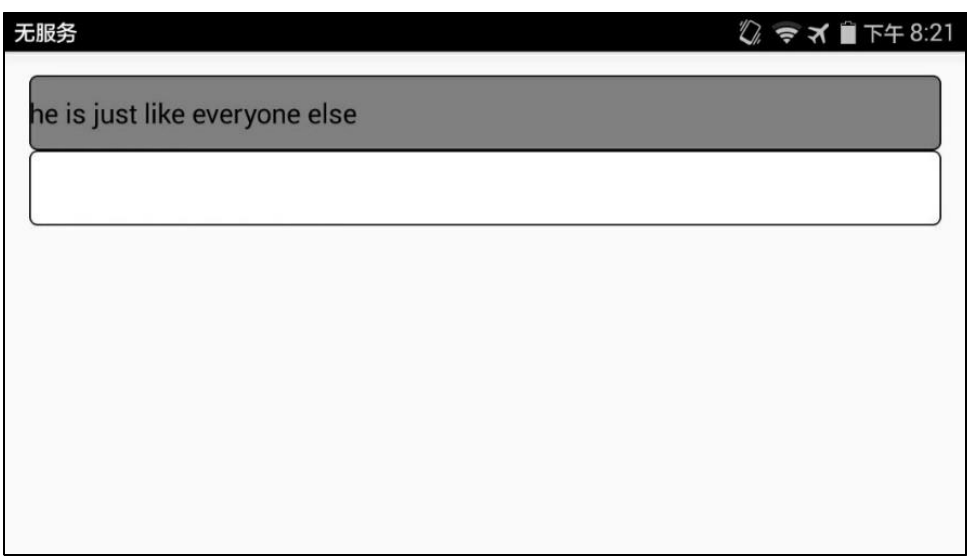


Figure 3.6 The layout of eyes-free mode application

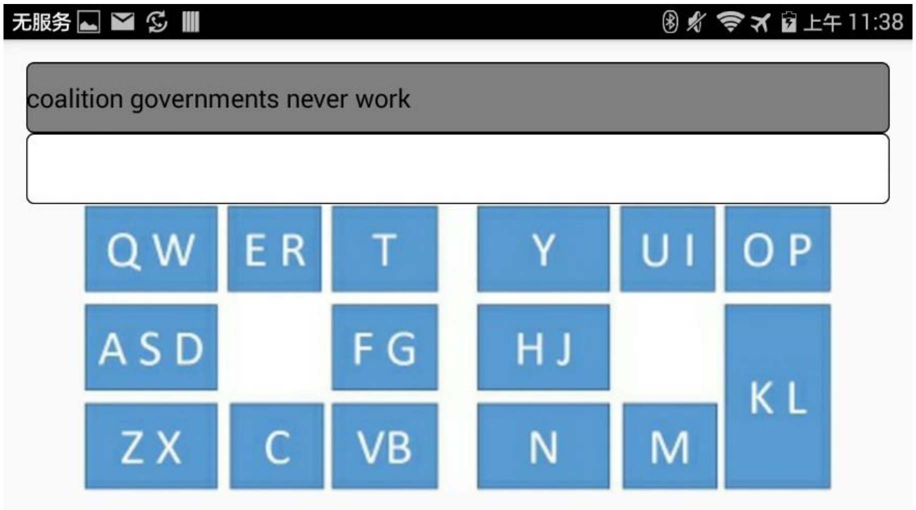


Figure 3.7 Visual help hints in the eyes-free mode application

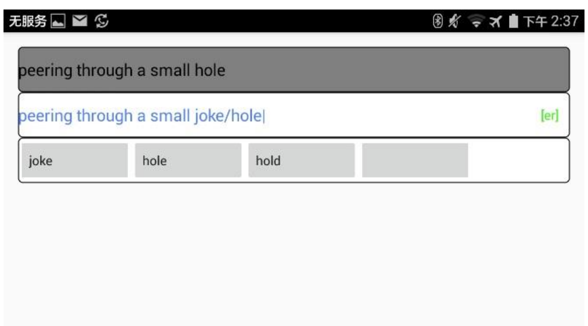
3.1 Physical direction

A character is selected by moving the thumb in the direction of this character. Thus, each character is represented by a direction. A map of directions and their associated characters are shown in Table 3.2. All words can be represented by a sequence of directions. For example, the word "MY", 'M' is in the north of right virtual keyboard, therefore, M' can be represented as "R_U" (R is right part of the touch area, and U is move upward). Similarly, 'Y' can be showed as "R_UL" (R is right part of the touch area, and UL is move upward and left). Then, the word "MY" can be showed as (R_U,R_UL). However, there are some deviations when users moving in the direction of desired character. Hence, the angle of a gesture can be showed in Equation 3.1. The touch area is regarded as a coordinate system and each touch down and touch up point can be represented by a horizontal and vertical value. The fingers movement form a specific angle, and the direction can be represented by a radial sector. The a is the angle of a gesture and x_{down}, y_{down} and x_{up}, y_{up} are the display coordinates of the finger press and finger release, respectively. The angle was converted to one of the eight directions by dividing the circle into eight equal sectors centered on the eight directions north, northeast, east, southeast, etc.

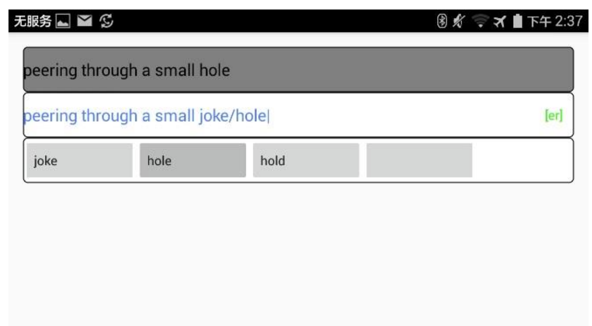
$$a = atan2(y_{up} - y_{down}, x_{up} - x_{down}), -\pi < a < \pi \qquad (3.1)$$

Table 3.2 Each direction associated with characters

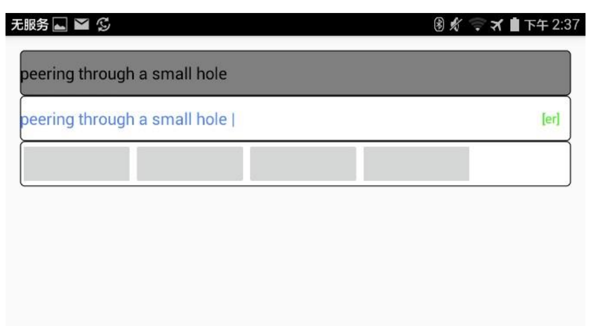
Movement direction	Left virtual keyboard	Right virtual keyboard
Up Left	QW	Y
Up	ER	UI
Up Right	T	OP
Left	ASD	HJ
Right	FG	KL
Down Left	ZX	N
Down	C	M
Down Right	VB	KL



(a) Word *joke* and word *hold* have same direction sequence.

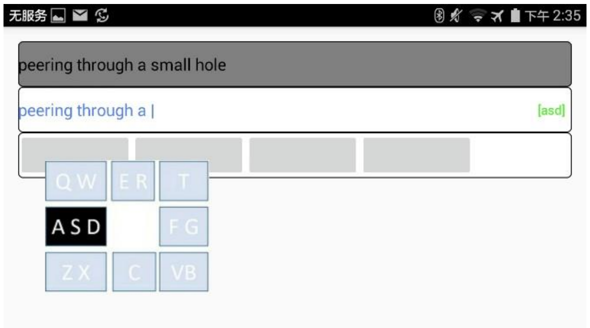


(b) pressing the goal word *hold* in the suggestion list.

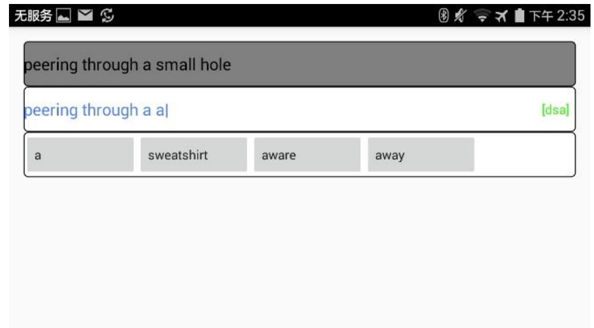


(c) Releasing the finger, producing the word *hold*.

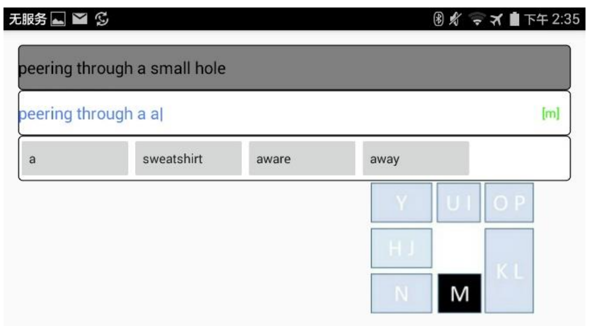
Figure 3.8 Resolving the same direction sequences word ambiguous with eyes-on mode



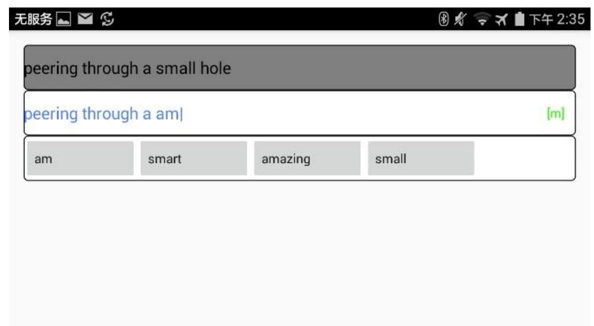
(a) The goal is to input the word *small*. First, pressing the left side of the display, moving finger in the west direction.



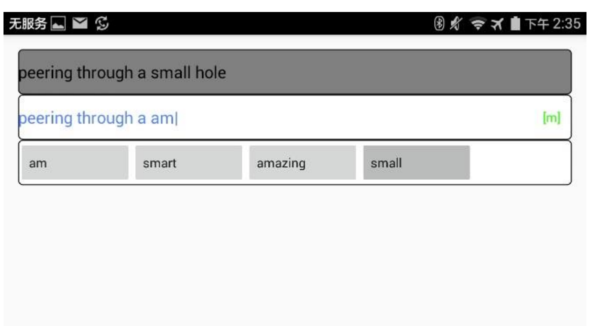
(b) Releasing the finger, producing the character *a*.



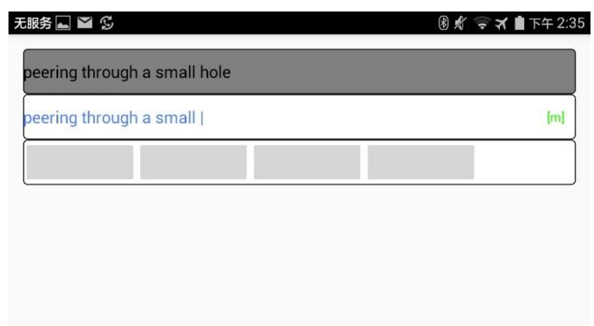
(c) Then, pressing the right side of the display, moving finger in the south direction.



(d) Releasing the finger, producing the character *m*.



(e) Last, pressing the goal word *small* in the suggestion list.



(f) Releasing the finger, producing the word *small*.

Figure 3.9 Ambiguous character input procedure with eyes-on mode

3.2 Resolving ambiguities

According to Table 3.2, most directions are assigned multiple characters. Only five directions have unique character assignments, namely up-left ('T'), down ('C') on left virtual keyboard and up-right ('Y'), down-right ('N') and down ('M') on right virtual keyboard. Except for direction left on the left virtual keyboard is assigned three characters ('A', 'S', and 'D'), the other directions are assigned two characters. To resolve these ambiguities, a word trie was employed in the prototype. A trie data structure (Fredkin, 1960) was used to map the sequence of directions to actual words based on a dictionary of the English language. To keep the experiment simple, a small dictionary with 1168 words was used in this application. The trie allows words to be suggested based on unique word prefixes. One node for each prefix and the words are stored in the leaf nodes (Fredkin, 1960). For each ambiguous character input with eyes-on mode, the displayed set of letters which were later replaced by the real letter when the entire word was resolved automatically (see Figure 3.9). Apart from that users can select the desired word when it appears in the suggestion word list instead of entering all the characters in the eyes-on mode. There are some ambiguities caused by some words sharing the same sequence of directions, for instance words too and *top* have same direction sequences (L_UR, R_UR, R_UR). But most sequences are unique. All the same direction sequence ambiguous words were output with eyes-on mode (see Figure 3.8). It is relatively easier for the user to extract the intended word when the ambiguities appear in sentence context. Regarding resolving same direction sequence ambiguities in the eyes-free mode, the users do not resolve ambiguous as this would require some sort of feedback (auditory, visual, etc.). Some robust word disambiguation works such as performing some elaborate language models to automatic word disambiguating could be resolved in the future work of this study.

3.3 Special characters

Some special character such as SPACE, BACKSPACE, and ENTER are essential functions for elementary text entry tasks. This study has tested both using single finger gestures and two fingers gestures to support these three special characters and found that the two fingers gestures were able to reduce the chance of character input to be mistaken for special characters, since the normal characters were inputted by single finger gestures. Furthermore, mirror movements of human hands are comparatively coordinated movement, and some experiments also found that symmetric gestures worked better than non-symmetric gesture(Matias et al., 1994). Therefore, these special characters were implemented by the two thumb symmetric gestures in this text entry method. Namely, SPACE was entered by moving two fingers outward to the sides. BACKSPACE was inputted by moving two fingers inward, which is used as the only means of error correction. And ENTER which used to moving the next phrase was realized by moving two fingers downward on the touchscreen. However, some advanced editing operations such as cursor navigation and entering various punctuation marks including question marks, quotations, period, etc., are not addressed in this study. The purpose of this study is to explore the proposed text entry strategy and not to develop a deployable general-purpose input mechanism. Thus, the text entry tasks should be as easy as possible for the users. While, these advanced symbols are used for complex text entry tasks.

4. Methodology

An experiment was conducted to assess the proposed text entry method usability and users' satisfaction, which is to evaluate whether users can use their memory of the QWERTY layout to operate the input techniques, how well they can access it, and how satisfied they are. This study used a quantitative research method which can provide reliable and accurate answer to the research question. More importantly, personal bias can be avoided by performing an objective experiment and using accepted computational techniques. An experiment with human participants was done for measuring and analyzing participants' performance basing on the proposed text entry strategy prototype.

4.1 Experimental design

Before performing the experiment, I carefully considered the way an experiment is designed and carried out. This involved deciding on the participants, the variables, the tasks, the procedure for briefing and preparing the participants, the hardware and software (materials or apparatus), the data collected and analyzed.

This experiment evaluated the effectiveness (speed and accuracy) of the proposed text entry strategy. A performance benefit might not appear immediately, it takes time for users learn the technique. And this master project also addressed the learning effect associated with the text entry strategy. Hence, this experiment was a longitudinal study which involves testing users over a prolonged period while their improvement in performance is measured. Given that, there was limited time and no payment for participants, they performed the task over four training sessions, each session lasted 20 minutes, while their improvement with practice is measured. As participants become familiar with the experimental procedure and task, the learning occurred.

It is very important to declare the independent variables (test conditions) and dependent variables (measured behaviors) of the evaluation experiment (MacKenzie, 2012). This experiment used a mixed design to test conditions (level and factors). There are two independent variables in this experiment. This is a longitudinal study, therefore, the "amount of practice" (training sessions) is a within-subjects independents variable and it had four levels reflecting the four practice sessions. The other independent variable is a between-subjects independent variable called feedback mode, and it had two levels, namely eyes-on mode and eyes-free mode. It was decided to treat feedback mode as a between-subjects factor as it is highly probable that the task based on eyes-on mode would greatly affect the task based on eyes-free mode. Furthermore, there are two dependent variables in this experiment. One dependent variable called text entry speed which was measured by how many words successfully entered per minute (WPM). Another dependent variable is accuracy which is also called error rate, and it is reported as the percentage of text entered incorrectly. The WPM and error rates will be discussion in data collection and analysis section (4.7).

Questionnaires were also used at the beginning and the end of the experiment to obtain participants' subjective opinions and feelings about the text entry technique. Items were formatted using a Likert scale to facilitate summarizing and analyzing the response. More importantly, since this experiment involved humans, ethical issues were considered before executing the experiment. The participants were informed about the hypotheses, goals and objective of the research. And they have the right not to participate, not to answer any questions, and to terminate participation at any time. They also have the right to anonymity and confidentiality. After that, a pilot test was performed to ensure that the software operated well, that the procedure was feasible, and the data collected were correct and available in an appropriate format for follow-up analyses.

4.2 Ethical considerations

It is very important to consider the ethical issues before performing the experiment. Since this user study involves humans, their autonomy, privacy and dignity should be considered. Specifically, when recruiting the participants, they should be informed with the goals, hypotheses, methodology of this research. Besides that, the experiment tasks, procedure, risks and benefits should be also announced to them. More importantly, they have right not to participate, not to answer any questions, and terminate the experiment at any time. And they also have right to anonymity and confidentiality. After that, people who agree to participate in this experiment should sign a consent form which includes the above ethic information.

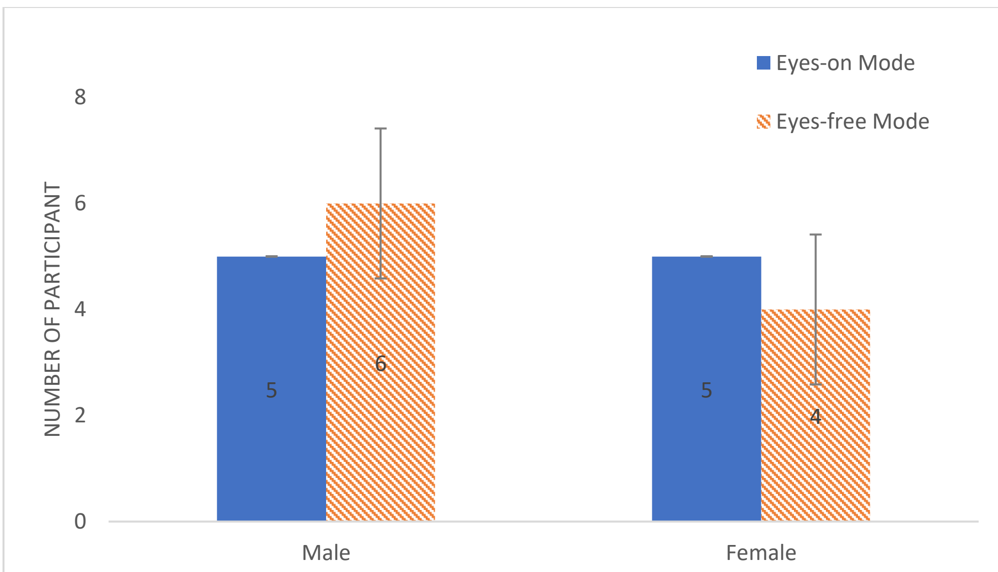


Figure 4.1 Gender information of the two groups. Error-bars show standard deviation.

4.3 Participants

A total of 20 participants were recruited for the experiment. Ten of them were recruited among my friends as the eyes-on group. They participated in the test of the eyes-on application. Next, ten students from the 2nd year class of master-level students of Universal Design of ICT classes as the eyes-free group, participated in the eyes-free experiment. The test panel comprised 11 males and 9 females. The gender information for the two groups is shown in Figure 4.1. There were five males and five females in the eyes-on group participated in the experiment involving eyes-on mode application, and there were six males and four females in the eyes-free group that that participated in the experiment involving eyes-free mode application. The participants were between 19 to 54 years of age. Figure 4.2 presents the age distribution for the two groups. The age for the two groups were balanced although the average age of the eyes-free group was slightly lower than the average age of the eyes group. None of the participants reported having reduced vision, reduced motor function or dyslexia diagnoses.

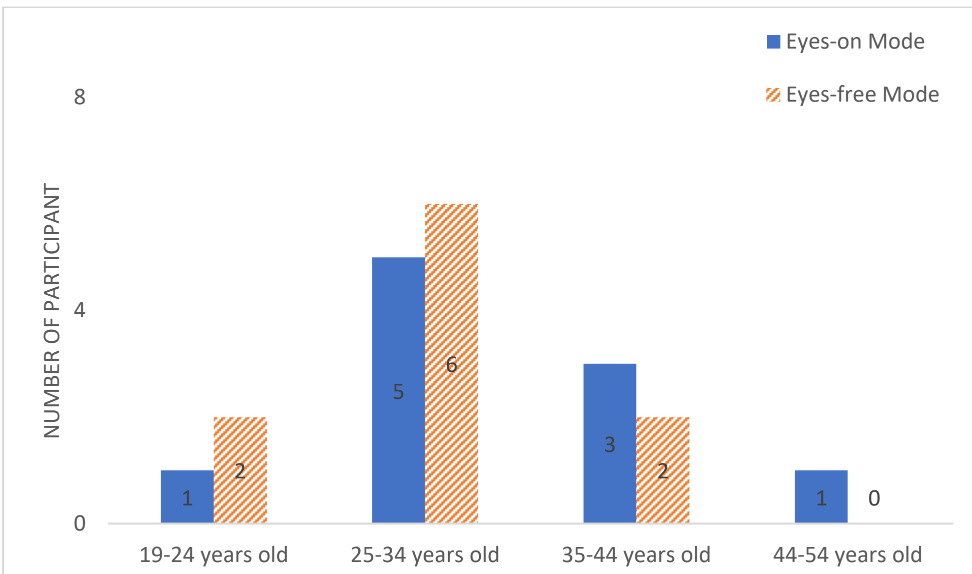


Figure 4.2 Age information of the two groups.

Quantitative research in descriptive studies may include a sample population of hundreds or thousands of participants to ensure that a valid estimate of a generalized relationship between variables (Hopkins, 2008). While, this master project is an experimental research and it is rational for the sample population be small and purposefully chosen, and it is intended to establish causality between variables. Martin (2007) recommend to use the same number of participants as used in similar research (Martin, 2007). After reviewing the literature, I found that the general text entry method evaluation studies (Bi et al., 2012; Cuaresma & MacKenzie, 2013; Goel, Jansen, Mandel, Patel, & Wobbrock, 2013; Nicolau & Jorge, 2012) recruit a dozen of participants. Apart from that, if there is an inherent difference in two conditions, it is always possible to achieve statistical significant results even with a very small group of participants (Hopkins, 2008). However, it is very difficult to recruit participants randomly from a population. Generally, most researchers obtain participants from convenient individuals (e.g., students from the university campus and friends from social public) (MacKenzie, 2012). To help identify the population, participants were given a brief questionnaire at the beginning of the experiment to gather demographic data, such as age and gender. Other information relevant to the research is gathered, such as familiarity with QWERTY layout, how often text is entered on a smartphone, and self-assessed text entry skills, by making a point on a line from 0 (lowest) to 6 (highest).

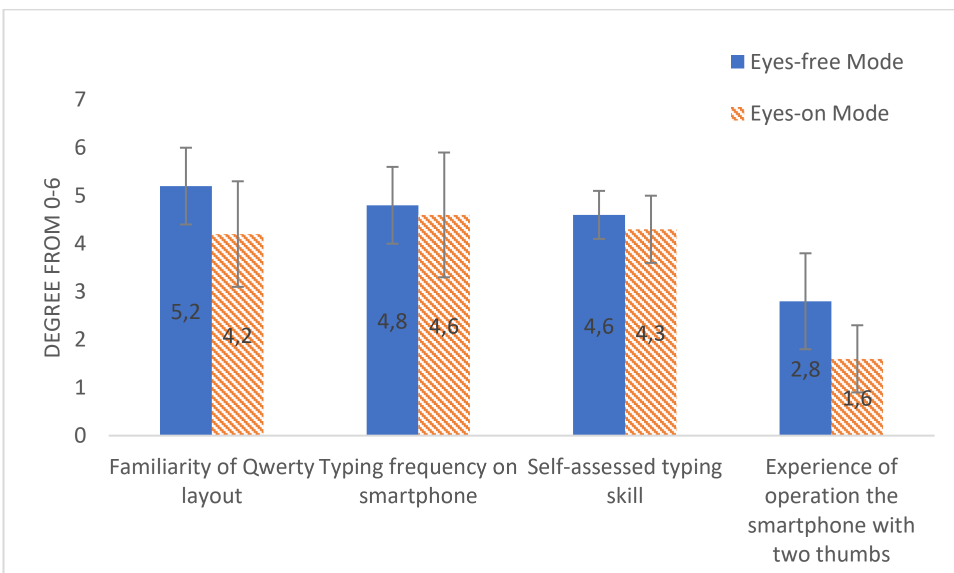


Figure 4.3 Characteristics of the two groups. Error-bars show standard deviation.

Reasonable criteria were taken to balance the two groups of participants. Figure 4.3 shows the detailed information of these balance factors between the two groups. The participants were asked to report their general familiarity with the QWERTY layout, eyes-on mode group (M = 4.2, SD = 1.1), eyes-free mode group (M = 5.2, SD = 0.8); their typing frequency on smartphone, eyes-on mode group (M = 4.6, SD = 1.3), eyes-free mode group (M = 4.8, SD = 0.8); and their self-assessed text entry skills in general, eyes-on mode group (M = 4.3, SD = 0.7), eyes-free mode group (M = 4.6, SD = 0.5), using a 6-point Likert scale. Furthermore, Mann Whitney U tests were conducted to compare these factors. There were no significant differences between the responses of the two groups in terms of their typing frequency on the smartphone layout (U = 53, p = .842), or their self-assessed text entry skills (U = -62, p = .327). And it is clearly that the participants in eyes-on mode group (M = 1.6, SD = 0.7) and eyes-free mode group (M = 2.8, SD = 1.0) did not have much experience of operation the smartphone with two thumbs. Besides that, there were significant differences in their familiarity with the QWERTY layout (U = 76, p = .044), and their experience of operation the smartphone with two thumbs between eyes-on mode and eyes-free mode groups (U = 82, p = .012).

4.4 Apparatus and materials

A HUAWEI C8817E smartphone with Android 4.4 KitKat operating system and 5.0-inch display was used in the experiment. Two android applications (eyes-on mode application and eyes-free mode application) running in the device were tested by participants. Clearly, these two applications were developed for the experiment, namely one for the tasks involving eyes-on mode and one for the tasks with eyes-free mode. Note that the two terms eyes-on and eyes-free, in the experiment software refer only the gesture input area. In the eyes-free mode, the QWERTY keys are hidden unless hints are requested. In the eyes-on mode, the QWERTY keys are shown. Besides that, the set of 500 English phrases proposed by MacKenzie (MacKenzie & Soukoreff, 2003) was used as copy text in the applications and it was used as it has been shown to be an effective benchmark (Kristensson & Vertanen, 2012). The phrases contain no punctuation symbols with all character in lowercase, thence it is easy to read. Apart from that, the phrases are moderate in length and the mean phrase length is 5.4 words per phrase (Sandnes & Aubert, 2007).And a simple English dictionary with 1168 words was used in this experiment for resolving the word ambiguities. Moreover, an integrated development environment (IDE) – Android studio was used to develop the applications. And another IDE – Eclipse was used to organize the collected data. Software JASP version 0.9.0.1 was used as the data analysis tool. A questionnaire was used before the experiment to gather information of demographics (age, gender, etc.), experience with text entry skills on smartphone, and familiarity with QWERTY keyboard. And at the end of the experiment another questionnaire was used to obtain participants' opinions and feelings about the text entry strategy. A consent form was signed by all participants before the experiment, which aimed to ensure that participants know that their participation was voluntary, that whey would incur no physical or psychological harm, that they could withdraw at any time, and that their privacy, anonymity, and confidentiality would be protected.

4.5 Task

The participants were asked to conduct an English language text copy tasks. Mackenzie and Soukoreff (2002) reported that text copy tasks are generally appropriate for an empirical evaluation (MacKenzie & Soukoreff, 2002c). Compared to text copy tasks, composing text tasks may involve substantial participants' thinking time which is difficult to measure (Wobbrock, 2007). While, in text copy tasks, the participants only focused on the text inputting activity and did not require to think what they should enter next. Therefore, copying text is usually preferred by most empirical text entry evaluation experiment (MacKenzie & Soukoreff, 2002c). In this text copying tasks, each phrase was randomly presented in a block, and each character entered appears directly below the intended character. The participants were asked to do the test as quickly and accurately as possible. They were asked to input the presented text by moving their fingers in relevant letters group directions and employ a next phrase two-finger gesture to move to the next phrase when completed. The participants assigned the task with eyes-free mode had the option of getting a visual hint where the full QWERTY keyboard was shown for 1.5 seconds by moving their two fingers up on the touchscreen. Moreover, they also allowed to correct or ignore mistakes and proceed whenever they were too difficult to correct.

4.6 Procedure

The participants performed the experiment separately and each training sessions scheduled two or three days apart. Each session lasted 20 minutes and the measurement part of the text entry session took 5 minutes. It is important to establish a good relationship with the participants from the very start, which include greeting to participants, introducing the experiment, and asking the participants to sign the consent forms. Besides that, a brief questionnaire was administered to gather demographic data and information on the participants' related experience such as their text entry self-assessment and their experience of operation the smartphone with two thumbs. This pre-experiment questionnaire was conducted via on-line questionnaire with Google forms. This opening took 5 min. After that, a brief of the principles of the text entry strategy was given and instructions of the applications explained and demonstrated to the participants. Appropriate practice trials were allowed for the participants. Then the participants performed the typing tasks for 5 min. At the last typing sessions, the participants filled in a face-to-face post experiment questionnaire to obtain the participants' opinions and self-assessed typing skills about the text entry method. The whole user study took over three months and all participants received no monetary rewards for their participation in the experiment.

4.7 Data collection and analysis

All the interactions performed on the smartphone during the experiment were logged in the external storage of the smartphone for the follow-up data analyses. When the participant performed the test, some information such as transcribed strings, presented string, input stream, the spatial and temporal details (inter- and intra-character time, angle and length) of the individual gestures, were logged in the smartphone log files. The presented string is shown in the presented text field. The transcribed string is entered by the participant by using the text entry method. The input stream is a sequence of character, including space, enter, and BACKSPACE, taken by participants during transcription. These log files are used for measuring the performance of the proposed text entry method. Speed and accuracy are the fundamental criteria to evaluate a text entry method. A repeated measures one-way ANOVA test with a significance level of 0.05 was used in the analysis of the performance of the text entry method.

4.7.1 Analysis speed

Text entry speed is considered as the most important measurement. Words per minute (WPM) is the most widely used way to measure text entry speed in empirical experiments. Since about 1905, the “standard word” is defined as a string of 5 characters, including spaces (Yamada, 1980). The formula for computing WPM shows in Equation 4.1, where T is the transcribed string entered by the participant, and $|T|$ is the length of the transcribed string. The s is the time in milliseconds measured from inputting the first character to the end of the last character, including BACKSPACE and without helping time for eyes-free mode application. And the unit for s term is millisecond, therefore the “60000” is milliseconds per minute. The “$\frac{1}{5}$” is words per character.

$$\mathrm{WPM} = \frac{|T|-1}{s} \times 6000 \times \frac{1}{5} \tag{4.1}$$

The WPM only consider the transcribed text, while there are many error corrections happens during the text entry test. For example, the participants may correct some characters several times before input the correct character, but the WPM does not take this into account. To do this, the (Gestures Per Character) GPC was used to compute the data rate. As noted, participants’ gestures (moving finger) represent all characters, including common characters and special characters (space and BACKSPACE) in this study. The GPC measure indicates how fast the participant moves his or her finger. Besides, an “empirical upper bound” of this text entry method can be estimated by assuming that all entered characters are correct. The formula for computing GPC displays in Equation 4.2, where IS is the input stream which contains all entered characters, including BACKSPACEs. The term s is the same meaning with the previous equation.

$$\text{GPC} = \frac{|IS| - 1}{s} \times 6000 \quad (4.2)$$

4.7.2 Analysis accuracy

There is a trade-off between speed and accuracy. For example, participants can enter text more quickly if they are willing to sacrifice accuracy. And participants perform with high accuracy if they slow down. To avoid skewing the experiment results, both speed and accuracy were measured. The text entry accuracy is about error rates when participants enter text. Typically, there are three approaches to measuring accuracy : the minimum string distance (MSD) error rate, keystrokes per character (KSPC), and a unified error metric (Soukoreff & MacKenzie, 2003; Wobbrock, 2007). KSPC measures the corrected errors which the participant committed during entry, while MSD measures the uncorrected errors which left in the transcribed string. Since users can shift errors back-and-forth between the MSD error rate and KSPC by investing more or less effort in error correction, none of these two error rate algorithms can measure the text entry accuracy precisely (Soukoreff & MacKenzie, 2003). Soukoreff and MacKenzie (2003) devised a unified error metric which combines the analysis of the presented string, transcribed string, and input stream to provide total error rate as the sum of the corrected error rate and uncorrected error rate (Soukoreff & MacKenzie, 2003). These three error rates (corrected, uncorrected, and total error rates) depend on classifying all entered characters into one of six categories (see Table 3.1) (Wobbrock, 2007). The formula for computing these three error rates present in Equation 4.3, 4.4, and 4.5, respectively, where C is the total number of characters that are not errors in the transcribed string, IF is the number of characters that are later corrected in the input stream, and INF is the number of incorrect character in the transcribed string (Soukoreff & MacKenzie, 2004; Wobbrock, 2007).

$$Not\ Corrected\ Error\ Rate = \frac{INF}{C+INF+IF} \times 100\% \quad (4.3)$$

$$Corrected\ Error\ Rate = \frac{IF}{C+INF+IF} \times 100\% \quad (4.4)$$

$$\text{Total Error Rate} = \frac{\text{IF+INF}}{C+INF+IF} \times 100\% \quad (4.5)$$

Table 4.1 Character classes used for analyzing accuracy (Wobbrock, 2007)

Correct (C)	Total number of correct characters in the transcribed text
Incorrect-not-fixed (INF)	Total number of incorrect characters in the transcribed text
Incorrect-fixed (IF)	All characters BACKSPACEd during entry
Incorrect-fixed that were correct (IFc)	All fixed characters that were correct
Incorrect-fixed that were in error (IFe)	All fixed characters that were wrong
Fix (F)	All BACKSPACEs

The above accuracy analysis is just capturing the number of erroneous characters (corrected and uncorrected error) without considering what those errors are. For example, the IF character class may contain the fixed right characters that have been either mistakenly or purposefully fixed (Soukoreff & MacKenzie, 2004). with respect to uncorrected errors in the transcribed string, Mackenzie and Soukoreff (2002) proposed a character-level error analysis technique which not only can calculate the minimum string distance (MSD) between the presented and transcribed string (how many errors committed), but also generate the detail of the errors (MacKenzie & Soukoreff, 2002a). These error generated characters can be used to reflect on whether the devise of text entry method is reasonable.

In many text entry studies, the number of corrected errors is much greater than the number of uncorrected errors (Wobbrock, 2007). Therefore, it is important to analysis the corrected errors in input streams. As noted, the participant committed some errors but did not notice them until entry several characters afterward. Then the participant deletes these errors and re-entered the correct characters. And some corrected characters are deleted by the participant during the fixing process. Therefore, the corrected errors comprise incorrect but fixed characters that were correct (IFc) and incorrect but fixed characters that were in error (IFe) (see Table 3.1). Finally, a detailed error direction (including uncorrected errors and corrected errors) information was generated by analyzing presented string, transcribed string, and input stream.

From the detailed error direction information, we can capture which directions are associated with the most errors and analyze the reason.

Since the text entry method is to input character by moving finger in the direction of the desired character, intra- and intercharacter time are also important information that can be used to evaluate the efficiency of this text entry method. The intra-character time (finger-holding-time) is the time from finger down to finger up (including finger moving time) when input a character. This measurement has been used to compare unistroke character speeds in Graffiti and EdgeWrite (Wobbrock et al., 2003). In this study, as noted, all characters represented by directions (see Table 2.2). And there are 16 directions and each direction have an average finger holding time. The formula for calculating the average finger holding time for one direction is in Equation 4.6, where n is the values of how many times the direction entered, d_j represents one of the direction. Since there are 16 directions, 0≤ $j \leq 15$. The term $HT(d_j)$ is the value of finger holding time for one direction d_j.

$$\overline{HT}(d_j) = \frac{\sum_{i=1}^{n} HT(d_j)}{n} \qquad (4.6)$$

Additionally, the inter-character time (finger-thinking-time) refers how long participants delay between the end of one character and the beginning of the next. The inter-character time can be useful for observing learning or recall which alphabets are difficult to input (Wobbrock, 2007; Wobbrock et al., 2005). The formula for the average finger thinking time for from one direction to another direction is provided in Equation 4.7. There are 16 directions, 0≤ $n \leq 15$ and 0≤ $m \leq 15$. The term of $TT(d_n)$ and $TT(dm)$ are time stamps from the sequence of the inputted direction. $TT(d_n) - TT(dm)$ is the value of thinking time that after the participant input direction d_m, he or she requires time to think and then input direction d_n. The term n is the values of how many times inputting from direction d_m to direction d_n.

$$\overline{TT}(d_m, d_n) = \frac{\sum_{i=1}^{n}\left(TT(d_n) - TT(d_m)\right)}{n} \qquad (4.7)$$

A thinking time matrix which shows the dynamic transition among the directions, was generated after all the training sessions.

4.7.3 Analysis learning

The ease of learning is an important factor for evaluating a text entry method (Lee & Zhai, 2004). One way to analyze learning of this text entry method is to graph entry rates in WPM over time and model the points according to the power law of learning (Card, Newell, & Moran, 1983). Analysis the helping information is another way to evaluate the text entry method. As noted, in eyes-free mode application, the participants can look up the help hints when they are uncertain about how to retrieve a character. The number of references to the help hints and the detailed helping directions logged in the helping information. The extent of refer the help hints were reported if there are learning effects.

5. Results and Discussion

5.1 Performance

5.1.1 WPM (Words Per Minute)

Text entry speed (word per minute) is a relatively conventional way to measure the performance of one text entry method. And the text entry speed outcomes of this proposed text entry strategy are shown in Figure 5.1. The results show that there is a significant improvement in text entry performance with practice ($F(3, 54) = 106.5$, $p < .001$). There is also a significant difference in performance between eyes-free mode and eyes-on mode group ($F(1, 18) = 4.917$, $p = .040$), where the text entry performance is higher with eyes-on mode. There is no interaction effects between the session and interaction mode ($F(3, 54) = 0.635$, $p = .596$).

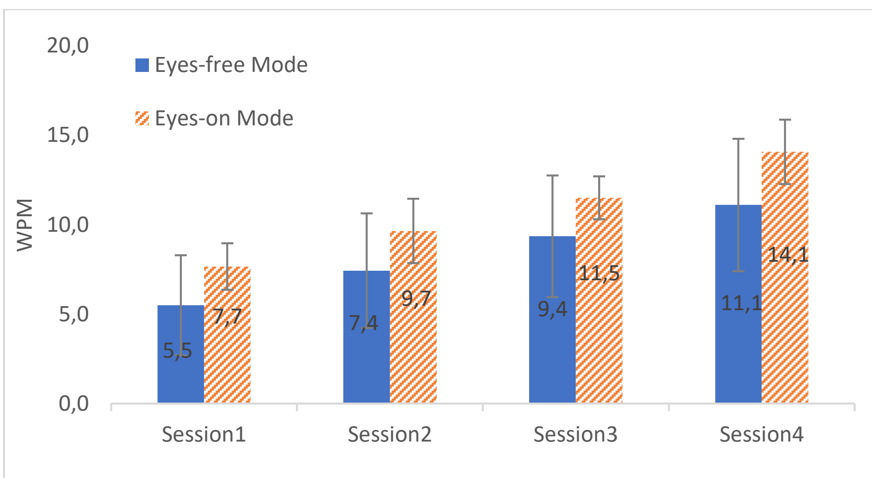


Figure 5.1 Text entry speed (words per minute). Error bars show standard deviation.

During the four training sessions, the text entry rate of eyes-on mode rises from 7.7 WPM in the first session to 14.1 WPM in the fourth session. Similarly, the text entry speed for the eyes-free mode application is 5.5 WPM in the first session and rises to 11.1 WPM in the last session. Clearly, the improvement of the text entry speed is near linear across the four-session training. And it is still believed that with further training, further improvement of the text entry speed would be possible. However, it would be expected that the improvement of the text entry speed would be logarithmic, and the progress would become smaller with prolonged training. Through observation of the Figure 5.1, interesting results can be found that the text entry speed of eyes-free mode seems to close to the text entry rate of the preceding session with eyes-on mode, for example, eyes-on session 2 text entry speed (7.4 WPM) is approximately to eyes-free session 1 text entry rate (7.7 WPM). Besides that, the spread with eyes-on mode is larger than the spread with eyes-free mode. That may be due to the text entry tasks for eyes-free mode is more difficult than the text entry tasks for eyes-on mode. Specifically, the eyes-on mode group participants can just select the word when it appears in the suggestion word list instead of entering all the characters and they also can refer to the left and right virtual keyboards when they enter text. And this spread difference may be because of bias between the two populations.

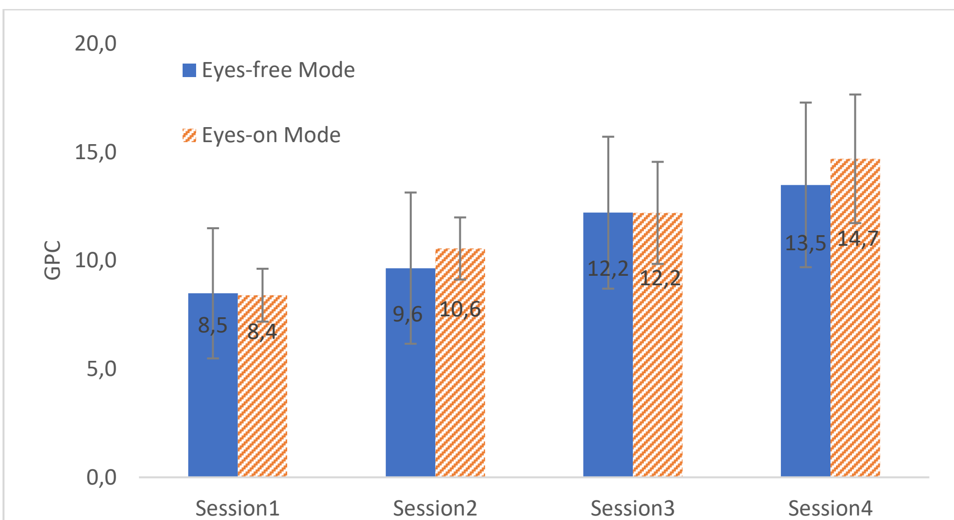


Figure 5.2 Moving gestures speed (gestures per character). Error bars show standard deviation.

5.1.2 GPC (Gestures Per Character)

According to the GPC formula in Equation 4.2, the results of GPC are shown in Figure 5.2. The results show that there is a significant improvement in text entry performance with practice ($F(3, 54) = 121.072$, $p < .001$). Even though the eyes-on mode application provides suggestion word list and visual keyboards, there is not a significant difference in GPC between eyes-free mode and eyes-on mode group ($F(1, 18) = 0.186$, $p = .0671$). That may be due to some users concentrated on dragging their thumbs and ignored the word suggestions. With the comparison to WPM results (see Figure 5.1), GPC performance is higher than WPM. Therefore, we can estimate the GPC results are the text entry method's "empirical upper bound" performance by assuming that all entered characters are correct.

Generally, the peak text entry speed can be yield when the text entry speed curve become logarithmic flattening. However, there is no sign of the text entry rates' logarithmic flattening out in this longitudinal experiment. Hence, the observed text entry speed might not represent the actual performance of this text entry method. Although, the observed mean text entry speeds for eyes-on mode (14.1 WPM) and eyes-free mode (11.1 WPM) are much lower than the two thumb text entry rates (50.0 WPM) (Azenkot & Zhai, 2012) obtained with the study of Azenko and Zhai (2012), they are much higher than the text entry performance (2.1 WPM) with ViceOver (Oliveira, Guerreiro, Nicolau, Jorge, & Gonçalves, 2011) reported by Oliveira et al. (2011). It is interesting that the obtained text entry speed for eyes-free mode is similar to the results of 11.1 WPM obtained with Graffiti with visual feedback (Castellucci & MacKenzie, 2008) and slightly higher than the text entry speed of 8.34 WPM obtained in the eyes-free Graffitii experiment (Tinwala & MacKenzie, 2010) conducted by Tinwala and Mackenzie (2010). In addition, expert evaluation was also conducted to show the peak performance of this text entry method. I as an expert of the method, managed to input the phrase "the quick brown fox jumps over the lazy dog" 10 times with eyes-free mode application, reaching a text entry rate of 24.85 WPM.

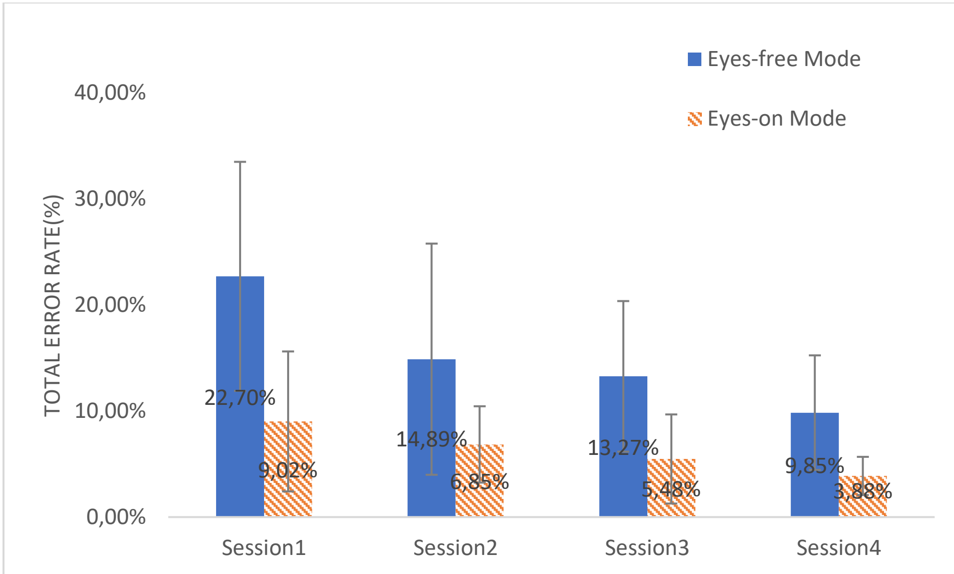


Figure 5.3 Total error rates (in percent). Error bars show standard deviation.

5.2 Errors

5.2.1 Total Error rates

Accuracy is another approach to evaluate a new text entry strategy. Generally, the types of text entry errors are various, such as corrected errors and not corrected errors. Figure 5.3 illustrates the total error rates of eyes-on and eyes-free mode applications. There is a significant difference in the total error rates between the two modes ($F(1, 18) = 12.40$, $p = .002$). Clearly, the total error rates in eyes-free mode are close to three times as high as the text entry total error rates in eyes-on mode. As expected, the high error rates with eyes-free mode are consistent with other eyes-free text entry studies. Besides that, the total error rates decreased during the four sessions. For the first session the total error rate with the eyes-free is 22.70% and falls to 9.85% for the forth session. The total error rate of eyes-on mode is 9.02% during the first session which falls to 3.88% during the last session. Thus, the practice affects the total error rates, while the experiment data did not satisfy the assumption of sphericity and the Friedman tests were used to show that there is a significant effect on training with eyes- on mode ($\chi^2(3) = 8.04$, $p = .045$) and eyes-free mode ($\chi^2(3) = 10.92$, $p = .012$).

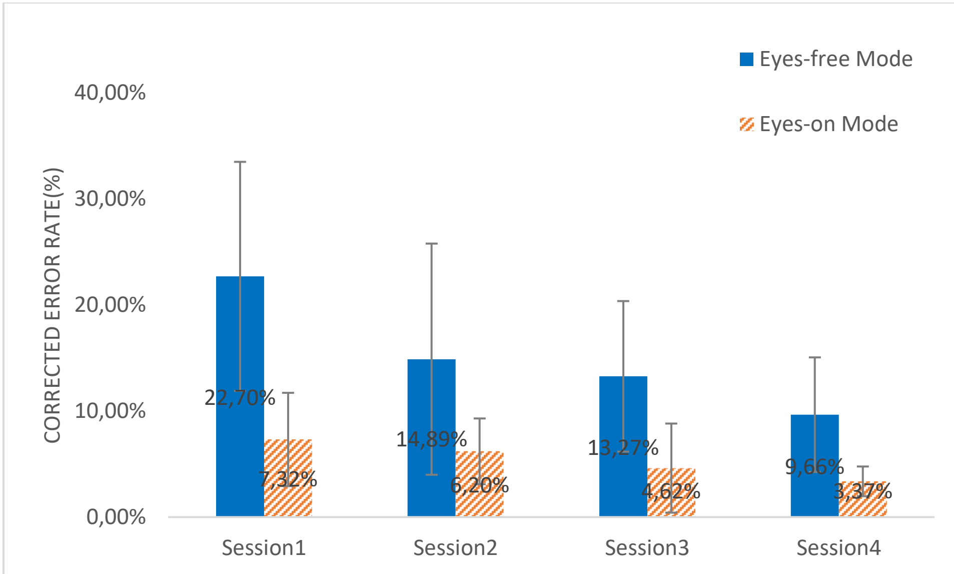


Figure 5.4 Corrected Error Rate (in percent). Error bars show standard deviation.

5.2.2 Corrected and uncorrected error rates

The total error rates consist of corrected error rates and uncorrected error rates. The number of corrected errors is much greater than the number of uncorrected errors. Figure 5.4 shows the corrected error rates for the two task types for the four sessions. Clearly, the trend of corrected error rates (see Figure 5.4) is similar with the total error rates (see Figure 5.3) and the error rates associated with eyes-free mode is nearly triple that of the eyes-on mode and this difference is significant ($F(1, 18) = 15.72$, $p < .001$). Training also affect the corrected error rates as there is a reduction from the first session to the last session with the eyes-free mode and a reduction from the first session to the last session with the eyes-on mode. The data did not satisfy the assumption of sphericity and Friedman tests were therefore used to show that there is both a significant effect on training for Eye Mode ($\chi^2(3) = 10.33$, $p = .015$) and Eye Free Mode ($\chi^2(3) = 11$, $p = .012$).

As noted, the corrected error rate consists of corrected and wrong error rates and correct but right error rates. Figure 5.5 and figure 5.6 illustrate the eyes-on mode and eyes-free mode detailed corrected error rates, respectively. Normally, the corrected but right errors were accidentally made by participants, for instance, the participant committed some errors but did not notice them until after entering several other characters. The participant then deletes these errors and re-entered the correct characters. And some corrected characters are deleted by the participant during the fixing process. Therefore, the corrected and wrong error rates are higher than corrected but right error rates in both eyes-on mode and eyes-free mode. Since the eyes-on mode provided alternative suggestions based on prefixes, the alternative suggestions and practice may affect the corrected and wrong error rates with eyes-on mode. There are no alternative suggestions in eyes-free mode and the practice has a significant effect on reducing the corrected and wrong error rate. Since the data did not satisfy the assumption of sphericity, Friedman tests were used to show that there is a significant effect on training for Eye Mode ($\chi^2(3) = 10.33$, $p = .015$).

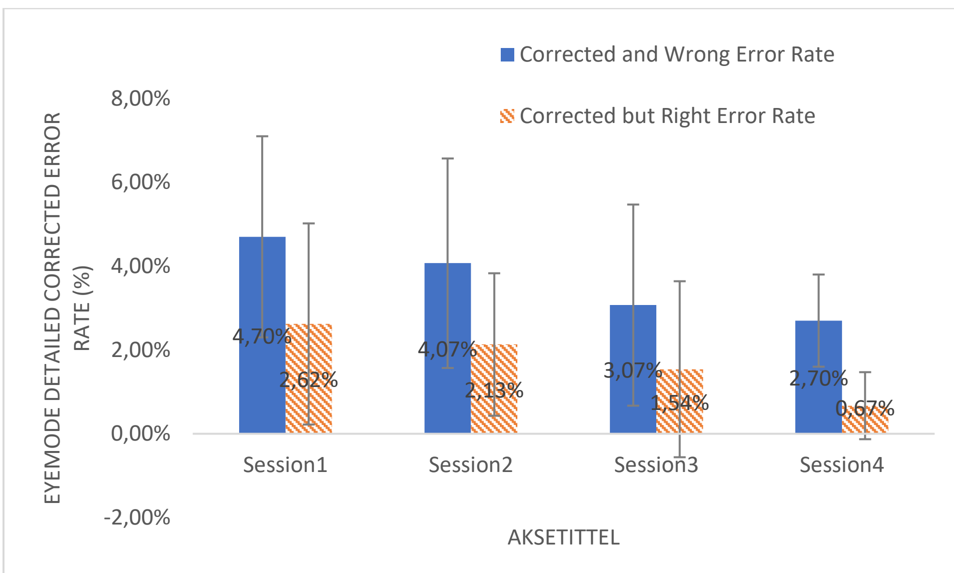


Figure 5.5 Detailed Corrected Error Rate (in percent) of Eyes-on Mode. Error bars show standard deviation.

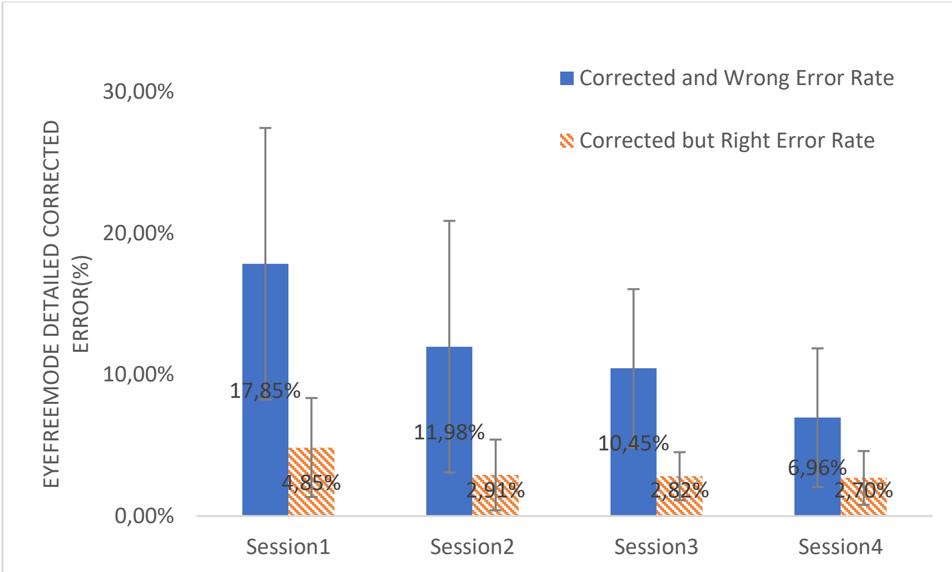


Figure 5.6 Detailed Corrected Error Rate (in percent) of Eyes-free Mode. Error bars show standard deviation.

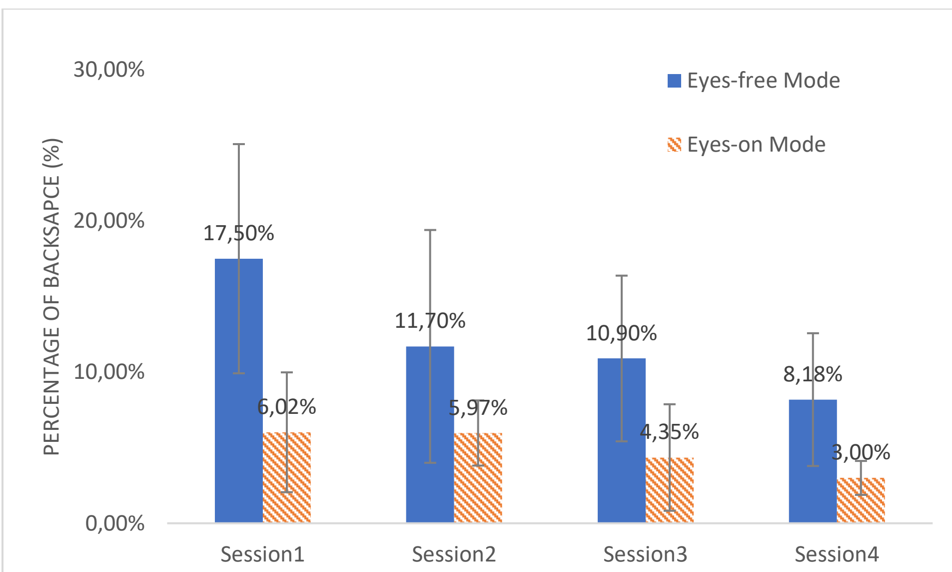


Figure 5.7 BACKSPACE Percentages. Error bars show standard deviation.

5.2.3 BACKSPACE percentages

The percentages of the BACKSPACE can be used to evaluate the performance of a text entry strategy. Figure 5.7 shows the ratio of the entered BACKSPACE and total inputted character. The results show that practice had a significant effect on entering BACKSAPCE between eyes-free and eyes-on applications ($F(3, 54) = 12.901$, $p < .001$). During the four training sessions, the percentages of inputted BACKSPACE in eyes-on mode falls from 17.5% in the first session to 8.18% in the fourth session. Similarly, the percentages of entering BACKSPACE in the eyes-free mode application is 6.02% in the first session and falls to 3.00% in the last session. Furthermore, there is also a significant difference in using BACKSPACE between eyes-free mode and eyes-on mode group ($F(1, 18) = 15.18$, $p < .001$). Clearly, the BACKSPACE ratio in eyes-free mode are close to three times as high as the BACKSPACE ratio in eyes-on mode.

5.2.4 Detailed errors associated with directions

Table 5.1 shows the total number of occurrence and errors of each direction in Eye mode and Eye Free Mode, respectively. As expected the north (*ER*) and west (*ASD*) direction for the left hand, and the north (*UI*) and the northeast (*OP*) direction for the right hand were inputted more frequently, the error occurrences were therefore high in these directions. However, the east (*FG*) direction for the left hand and the west (*HJ*) direction for the right hand were inputted less frequently with more errors. Gestures moving to these two directions (*FG* and *HJ*) are symmetric gesture and they are mirrored with each other, which may have high chance to be mistaken for each other.

Figure 5.8 and figure 5.9 show the detailed normalization error rates of each direction with eyes-on mode and eyes-free mode. The darker the background color, the higher the error rate in that direction. For left hand, more errors occurred in northeast and southwest direction and less errors occurred in northwest and southeast direction. While, for right hand, more errors occurred in northwest and southeast direction and less errors occurred in southwest and northeast direction. The results are accordance with the proposed theory, namely the lateral movement of the thumb is more flexible than the vertical movement.

Table 5.1 Total number of occurrence and error for each direction in Eyes-on Mode and Eyes-free Mode

	Eyes-on Mode		Eyes-free Mode	
Direction	Total Number of occurrence	Total Number of Error	Total Number of occurrence	Total Number of Error
ER	1792	36	1320	181
ASD	1573	43	1159	78
UI	933	16	741	107
OP	922	24	754	75
T	721	23	588	153
N	516	14	447	48
KL	507	7	363	41
HJ	441	17	360	75
FG	398	25	299	138
VB	261	2	176	27
C	260	11	201	32
Y	255	8	163	48
M	219	8	154	12
QW	162	7	146	14
ZX	36	4	30	7

QW 4.32%	ER 2.01%	T 3.19%
ASD 2.73%		FG 6.28%
ZX 11.11%	C 4.23%	VB 0.77%

Y 3.14%	UI 1.71%	OP 2.60%
HJ 3.85%		
N 2.71%	M 3.65%	KL 1.38%

Figure 5.8 Distribution of Normalized Eyes-on Mode Direction errors in percentages.

QW 9.59%	ER 13.71%	T 26.02%
ASD 6.73%		FG 46.15%
ZX 23.33%	C 15.92%	VB 15.34%

Y 29.45%	UI 14.44%	OP 9.95%
HJ 20.83%		
N 10.74%	M 7.79%	KL 11.29%

Figure 5.9 Distribution of Normalized Eyes-free Mode Direction errors in percentages.

Table 5.2 and Table 5.3 show the error metrices for eyes-on mode and eyes-free mode, respectively. The vertical title (in blue background) represents correct directions, and the horizontal title (in green background) represents wrong directions. The numbers in the matrix show the total number of times of that error, for instance, the total number of errors which moved from the correct direction north (*ER*) for the left hand to wrong direction northwest (*QW*) for the left hand, occurred 108 times in Table 5.2.

Table 5.2 The Error Matrix of Eyes-free Mode

	QW	ER	T	ASD	FG	ZX	C	VB	Y	UI	OP	HJ	KL	N	M
QW		1		3					10						
ER	108		22	26	1		1		1	13	4		2	2	
T	6	45		1	63			3	14	1	17	3			
ASD	24	11	2		10	3	4	5		2	1	8	2		
FG			4	13		1		64				47	1	6	1
ZX				2				2					1	2	
C				1		6		8		3		1	1	4	8
VB		1		2	1	5	10			1			6	1	
Y	7	2	7	2	1					7	5	15	2		
UI	8	48	3	1	2				2		36	1	4	1	1
OP	1	5	13	1			1		4	20			21	4	1
HJ		2	1	9	30		1	3		1	2		9	17	
KL				3	13		1	3		2	9	6			
N		2		1	1		1	3		1			1		37
M							1			1			2	8	

Table 5.3 The Error Matrix of Eyes-on Mode

	QW	ER	T	ASD	FG	ZX	C	VB	Y	UI	OP	HJ	KL	N	M
QW		1		1					1	2	1		1		
ER	4		3	9		1	1	1	1	7	3	2		1	
T		5		3	3		1	1	1		7			1	
ASD	8	7	1		7			2		1	5	1	3		1
FG		1	1	7				3		1	2	9		1	
ZX				1			1	1						1	
C				7						1					1
VB		1		1											
Y		1	2	1			1					1			
UI		4		3	1						4		2	2	
OP		4	5	1	1					6			4		
HJ		3		1	9								1	3	
KL		2		2	2									1	
N		3			1	2	2	3		1	1				1
M									3		1		1	3	

According to these error metrices, Table 5.4 illustrates the details of errors in eyes-on mode and eyes-free mode. Accuracy errors associated with accidentally selecting the neighboring direction of the desired direction. More specifically, the accuracy errors of eyes-on mode consist of left neighbor error rate (0.27%) and right neighbor error rate (0.21%). And eyes-free mode's left neighbor accuracy error rate is 4.27% and right neighbor accuracy error rate is 2.04%. Left-to-right errors and right-to-left errors related to incorrect hand usage, i.e., when the left hand is used to retrieve a character instead of the right hand and vice versa. There was a slight bias between left hand and right hand in incorrect hand usage. Furthermore, the left-to-left errors and right-to-right errors refer to retrieve a character in wrong direction with correct hand. The Table 5.4 shows for both eyes-on mode and eyes-free mode, the left-to-left error rates (0.74% and 5.39%) are higher than right-to-right error rates (0.31% and 2.58%), which may be caused by all the participants are right handed. Besides that, it is natural to expect a left-hand bias as more characters are assigned to the left hand (see Figure 3.2 and Figure 3.3) and consequently more errors will occur. Apart from that, the mirror error rates are the percentages of errors caused by the user dragging his/her thumbs in the incorrect mirror symmetric direction. For example, a mirror error occurs when the user drags his/her right thumb west to retrieve the *f* character. In generally, the eyes-on mode's error rates are lower than error rates with eyes-free mode as words suggestion providing in eyes-on mode and thus may have less error occurrence with eyes-on mode.

The overall error rates results illustrate that the error rate is high when no visual feedback is provided. While, detecting entering errors is challenge for individuals under the eyes-free mode. Hence, a robust automatic error correction mechanism such as language model based on word disambiguation could be resolved in the future work of this study.

Table 5.4 Details of Errors in Eyes-on Mode and Eyes-free Mode

	Accuracy Errors	Left-to-Left Errors	Right-to-Right Errors	Left-to-Right Errors	Right-to-Left Errors	Mirror Errors
Eyes-on Mode	0.48%	0.74%	0.31%	0.50%	0.48%	0.38%
Eyes-free Mode	6.31%	5.39%	2.58%	1.91%	2.08%	2.01%

5.3 Learning

The extent to which the participants referred help hints (see Figure 3.7 Visual hints in the Eyes-free Mode) varied greatly among the participants. The box plot in Figure 5.10 displays the distribution of normalized help hints per character data for the four training sessions. The mean number of reference consultations for the four sessions were 1.91% (session1), 1.37% (session2), 0.55% (session3), and 0.43% (session4), respectively (see the four crosses in Figure 5.9). And the spread is small, namely $SD = 0.0191$, $SD = 0.0128$, $SD = 0.0067$, and $SD = 0.0048$ for each session. Furthermore, training had a significant effect on practice ($F(3, 27) = 3.058$, $p = .045$. After training, even in eyes-free mode, the participants were familiar with the QWERTY layout, and the frequency of using the visual hints became reduced.

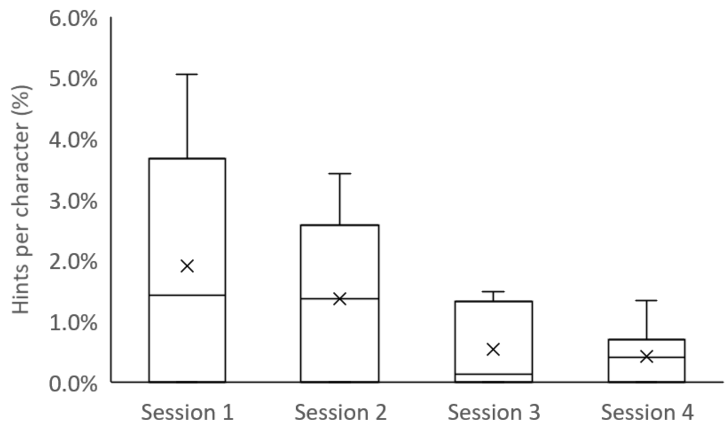


Figure 5.10 Percentage visual help hints per character.

Figure 5.11 shows the distribution of hints associated with characters. Clearly, the group *VB* is associated with the largest number of hints (20.37%) suggesting that this group is the hardest to remember. This is followed by *C* (13.89%) and then the north (*UI*), east (*KL*) and northeast (*OP*) direction for the right hand. No help was requested for the *HL* group.

QW 1.85%	ER 2.78%	T 4.63%	Y 2.78%	UI 12.04%	OP 8.33%
ASD 6.48%		FG 2.78%	HJ 0.00%		
ZX 9.26%	C 13.89%	VB 20.37%	N 3.70%	M 0.93%	KL 10.19%

Figure 5.11 Help hints associated with characters.

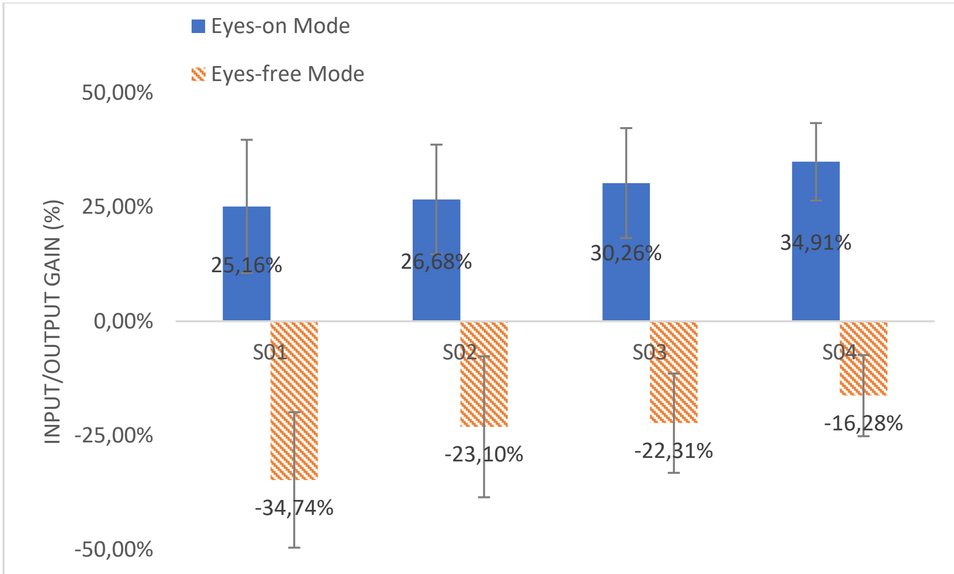


Figure 5.12 Character Input/output gains. Error bars show standard deviation.

5.4 Character input/output gains

Eyes-on mode application provides some appropriate alternative word suggestions for users when they input text. Thus, the participants in eyes-on were able to select the desired word when it appears in the suggestion word list instead of entering all the characters. To access the effect of the word suggestions, the character input/output gains was calculated for both eyes-on and eyes-free mode. The formula for computing the character input/output gain shows in Equation 5.1, where T_{input} is in terms of total number of input characters including BACKSPACE and corrected characters, T_{output} refers to the total number of output characters. The experiment data shows that all the eyes-on group participants used word suggestions during performing the experiment tasks. Although the T_{input} also includes BACKSPACE, the percentages of BACKSPACE in eyes-on mode are lower than 7% (see Figure 5.7). Therefore, the $T_{output}$is greater than the T_{input} , and the I/O_{gain} for the eyes-on mode is positive. While, the BACKSPACE ratios in eyes-free mode are relatively high, between 8%-18%. The T_{input} hence is greater than the T_{output} in eyes-free mode and the T_{output} is negative. Clearly, the training had a significant effect on character input/output gain ($F(3, 54) = 8.626$, $p < .001$). As Figure 5.12 shows, during the four sessions, the I/O_{gain} increased for both eyes-on and eyes-free mode.

Specifically, the I/O_{gain} in the eyes-free mode application is -34.74% in the first session and raises to -16.28% in the last session. Moreover, during the four training sessions, the I/O_{gain} in eyes-on mode raises from 25.16% in the first session to 34.91% in the fourth session. Apart from that, there was also a significant difference in the character input/out gains between eyes-on and eyes-free mode ($F(1, 18) = 151.1$, $p < .001$).

$$I/O_{gain} = \frac{|T_{output}|-|T_{input}|}{|T_{input}|} \times 100\% \qquad (5.1)$$

Analysis of the character input/output gains show that the word suggestions in the eyes-on mode was able to help participants reduce inputting gestures, thereby the BACKSPACE operations were effectively falls down in eyes-on mode. This might be one of the reasons to explains the eyes-on mode has a higher text entry speed and lower error rates compared to the eyes-free mode. Furthermore, reflection about the evaluation experiment indicates that to keep the experimental condition constant for both groups, the word suggestions condition should not be provided to participants in the eyes-on mode. Apart from that, retrieving and hitting the specific word in the suggestions list is heavily depends on users' visual feedback and advanced motor function. Thus, the further research of this study needs to explore a new modality for employing word suggestions in eyes-free text entry. For instance, audio feedback and switches assistive technology on smartphone can be used to help users to retrieve the desired word in the suggestions list.

5.5 Thumb thinking time

The eyes-on mode and eyes-free mode's direction transition matrices with size of 15 × 15, were generated based on the collected experiment data. These matrices summarize the average thumb thinking time for transitions between the various directions. After analyzing these two matrices, the smallest 20 direction transition time pairs and the largest 20 direction transition time pairs for eyes-free mode and eyes-on mode are shown in Table 5.5 and Table 5.6, respectively. For example, the shortest mean thumb thinking time in eyes-on mode is transition from north (*UI*) direction for right hand to southwest (*N*) direction for right hand (669 milliseconds) (see Table 5.5).

Table 5.5 The direction transition Matrix for Eyes-free Mode

Small Pairs				Large Pairs			
From	To	Thinking Time (milliseconds)	Hands	From	To	Thinking Time (milliseconds)	Hands
UI	N	669	Right	ZX	VB	2343	Left
KL	KL	676	Right	T	C	2347	Left
T	KL	677	Mixed	T	FG	2444	Left
T	HJ	705	Mixed	T	VB	2451	Left
OP	N	716	Right	FG	Y	2492	Mixed
VB	ER	718	Left	QW	FG	2527	Left
QW	N	772	Mixed	FG	M	2601	Mixed
QW	KL	776	Mixed	C	VB	2643	Left
M	VB	784	Mixed	M	FG	2661	Mixed
N	N	791	Right	Y	HJ	2715	Right
KL	UI	801	Right	ZX	T	2721	Left
N	C	812	Mixed	Y	M	2921	Right
UI	M	819	Right	VB	QW	2958	Left
UI	KL	823	Right	M	HJ	3395	Right
N	FG	834	Mixed	HJ	C	3406	Mixed
C	QW	834	Left	M	ZX	3588	Mixed
HJ	UI	835	Right	QW	C	3952	Left
C	T	839	Left	QW	ZX	4132	Left
VB	KL	839	Mixed	KL	VB	4181	Mixed
ASD	N	840	Mixed	Y	Y	5198	Right

According to Table 5.5, there are 9 pairs of mixed hand and 11 pairs of mono hand (8 pairs of right hand and 3 pairs of left hand) in the eyes-free mode's 20 small pairs, and 6 pairs of mixed hand and 14 pairs of mono hand (4 pairs of right hand and 10 pairs of left hand) in the eyes-on free mode's 20 large pairs. There is a significant difference in the thumb thinking time between the mono hand and the mixed hand in eyes-free mode ($U = 4874$, $p = .003$). As expected, since none of the subjects reported left-hand dominance in eyes-free mode group, the number of right-hand pairs are more than the number of left-hand pairs in the 20 small pairs, while in the 20 large pairs, except the 6 pairs of mixed hand, the left-hand pairs occupy 71% (10/14) of the total single-handed pairs. However, there is no difference in the thumb thinking time between the single left hand and the single right hand ($U = 1743$, $p = .312$), or no difference between left-to-right hand and right-to-left hand transitions in thumb thinking time ($U = 1469$, $p = .566$). One explanation could be that the participants in eyes-free mode experiment had less experience in using two thumbs to operate smartphones and the mean self-assessed two thumb typing skill is 2.8 ($SD = 1.03$) (see Figure 4.3).

As for the eyes-on mode, there are 10 pairs involving both hands and 10 pairs involving a single hand (4 pairs with the right hand and 6 pairs with the left hand) in the eyes-on mode's 20 small pairs, and 6 pairs with both hand and 14 pairs with a single hand (7 pairs with the right hand and 7 pairs with the left hand) in the eyes-on mode's 20 large pairs (see Table 5.6). Although, all the participants in the eyes-on mode are right handed, the results have a slight bias, which may be caused by words suggestion in eyes-on mode. Also, there is no difference in thumb thinking time between the single hand and the mixed hand ($U = 5869$, $p = .347$), or no difference between the single-left hand and the single-right hand in terms of thumb thinking time ($U = 1495$, $p = .672$). While, there is a significant difference in thumb thinking time between the left-to-right hand transitions and the right-to-left hand transitions ($U = 1112$, $p = .008$). Furthermore, regardless of whether in eyes-on mode or eyes-free mode, the number of mixed hand pairs are dominant in small pairs and the number of mono hand pairs is dominant in large pairs, which further illustrates that two hands cooperation is more efficient than single hand manipulation when entering text.

Table 5.6 The direction transition Matrix for Eyes-on Mode

Small Pairs				Large Pairs			
From	To	Thinking Time (milliseconds)	Hands	From	To	Thinking Time (milliseconds)	Hands
FG	UI	607	Mixed	HJ	M	2717	Right
QW	HJ	616	Mixed	M	N	2721	Right
T	HJ	622	Mixed	ZX	C	2805	Left
VB	UI	640	Mixed	FG	QW	2887	Left
QW	UI	687	Mixed	FG	N	2903	Mixed
QW	FG	707	Left	M	QW	2905	Mixed
M	Y	761	Right	Y	HJ	3037	Right
QW	ER	789	Left	ZX	ER	3066	Left
C	C	804	Left	C	QW	3101	Left
UI	N	811	Right	KL	ZX	3181	Mixed
FG	OP	814	Mixed	T	VB	3193	Left
HJ	OP	821	Right	KL	HJ	3294	Right
HJ	ER	830	Mixed	T	FG	3329	Left
M	ASD	833	Mixed	VB	HJ	3515	Mixed
ER	ER	838	Left	QW	KL	3619	Mixed
FG	KL	853	Mixed	N	HJ	3882	Right
ER	ZX	854	Left	VB	N	4403	Mixed
C	N	860	Mixed	HJ	Y	4405	Right
VB	ER	886	Left	M	KL	4672	Right
N	OP	891	Right	ZX	ZX	5882	Left

According to the angle formed by the pair of directions, these directions transition pairs could be divided into five categories: repeat (0°), near-repeat (45°), perpendicular (90°), near-opposite (135°), and opposite (180°). For example, the repeat direction pair refers to the left or right thumbs move in the same direction twice, for instance, directions [‘FG’, ‘FG’]. Near-repeat direction pairs are two neighboring directions, such as, [‘QW’, ‘ER’], [‘ER’, ‘T’], and [‘ASD’, ‘Y’].

The two modes' direction transition matrix has the same number of these pairs, namely, 29 repeat direction pairs, 56 near-repeat direction pairs, 56 perpendicular direction pairs, 56 near-opposite pairs, and 28 opposite direction pairs. There are no differences of these five thinking time categories in Eye Mode ($F(4, 220) = 0.381$, $p = .822$), and Eye Free Mode ($F(4, 220) = 0.889$, $p = .471$).

Table 5.7 Mean radian of angle and mean thumb holding time for each direction

		Eyes-free Mode		Eyes-on Mode
Direction	Mean Angle (radian)	Mean Holding Time (milliseconds)	Mean Angle (radian)	Mean Holding Time (milliseconds)
QW	2.32	175.12	2.47	699.13
ER	1.72	216.18	1.63	606.77
T	0.77	238.55	0.75	630.55
ASD	3.02	227.87	3.10	645.22
FG	-0.08	228.92	-0.05	738.93
ZX	-2.42	268.86	-2.48	987.67
C	-1.50	241.06	-1.53	840.76
VB	-0.78	240.42	-0.77	723.90
Y	2.42	222.49	2.43	656.18
UI	1.42	196.64	1.48	545.94
OP	0.73	234.15	0.74	578.09
HJ	-3.01	214.87	-3.07	603.09
KL	0.10	235.44	0.02	668.25
N	-2.35	225.01	-2.36	606.79
M	-1.66	233.12	-1.62	663.49
KL	-0.61	185.88	-0.54	1143.33

5.6 Thumb holding time and direction angle

As noted the thumb holding time is used to show how long the thumb is held down during each directional gesture when entering text, which is important information for analyzing the performance of this text entry strategy. Based on the logged data from the experiment, the mean thumb holding time and the mean direction angle for each direction with eyes-free mode and eyes-on mode are shown in Table 5.7. The direction (angle in radians) corresponds to the direction of the participants dragging gesture. Since this experiment employs a between-subjects design, it was expected that the mean radian of each direction angle would exhibit a slight difference between eyes-on mode and eyes-free mode. The mean holding time with eyes-on mode is much greater than the mean holding time with eyes-free mode, for instance, the mean thumb holding time with eyes-on mode (699.13 milliseconds) is more than three times of the mean thumb holding time with eyes-free mode (175.12 milliseconds) in the northwest (*QW*) direction for the left hand. This result is reasonable because the eyes-on mode provides visible virtual keyboards (see Figure 3.2 and Figure 3.3) to participants, therefore, the participant in eyes-on mode could drag their thumb according to the visible visual keyboards to choose the correct characters. During this process the participants may make some incorrect moves, and the correct character can be retrieved when they move their finger in the direction of the desired character and release their thumb (see figure 3.5). While, the participants in eyes-free mode group were not able to refer to the left and right visual keyboard, therefore, they had to rely on their spatial memory of QWERTY layout to entering character, which resulted in the eyes-free mode mean thumb holding time being shorter than the mean thumb holding time in the eyes-on mode.

Figure 5.13 and Figure 5.14 illustrate the radial bar plots of the thumb holding time and the direction angle with eyes-free mode and eyes-on mode, respectively. These radial plots present the distribution of the thumb holding time for each direction angle. Results are almost symmetrical-across the two hands. The thumb holding time is nearly uniform in most directions, but for some directions the thumb holding times are much longer. For instance, the average thumb holding time in the southwest direction for the left hand and in the southeast direction for the right hand were more than two times the mean thumb holding time for the other directions in eyes-on mode (see Figure 5.14).

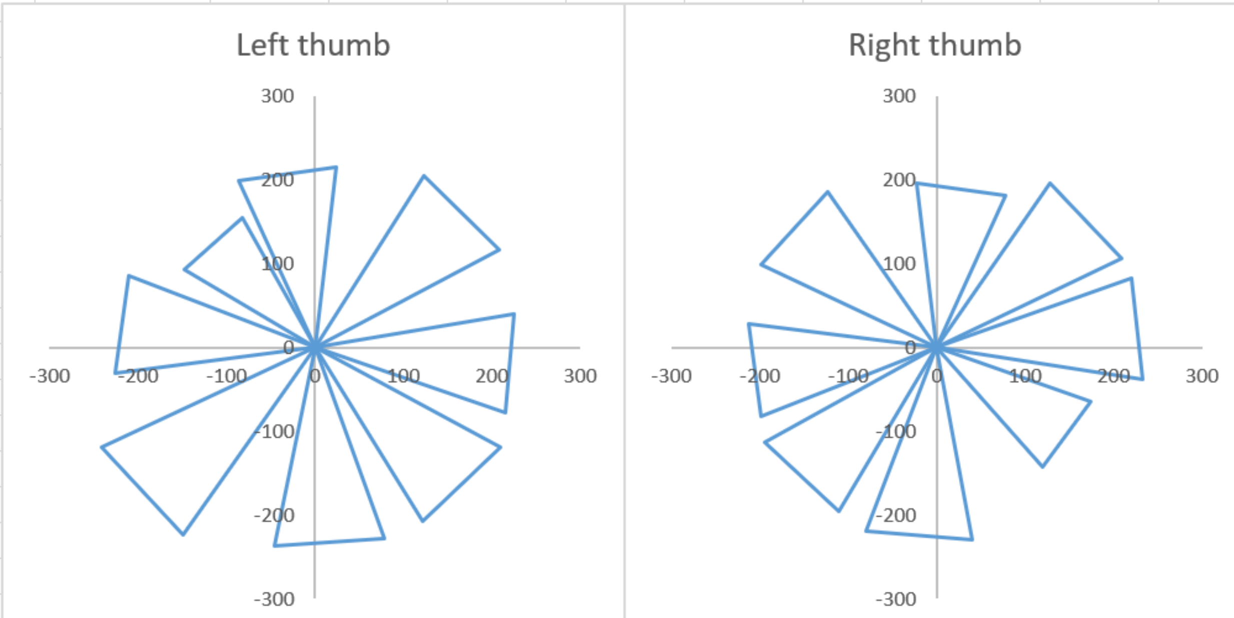


Figure 5.13 Breakdown of thumb holding time with the different direction angles in Eyes-free Mode

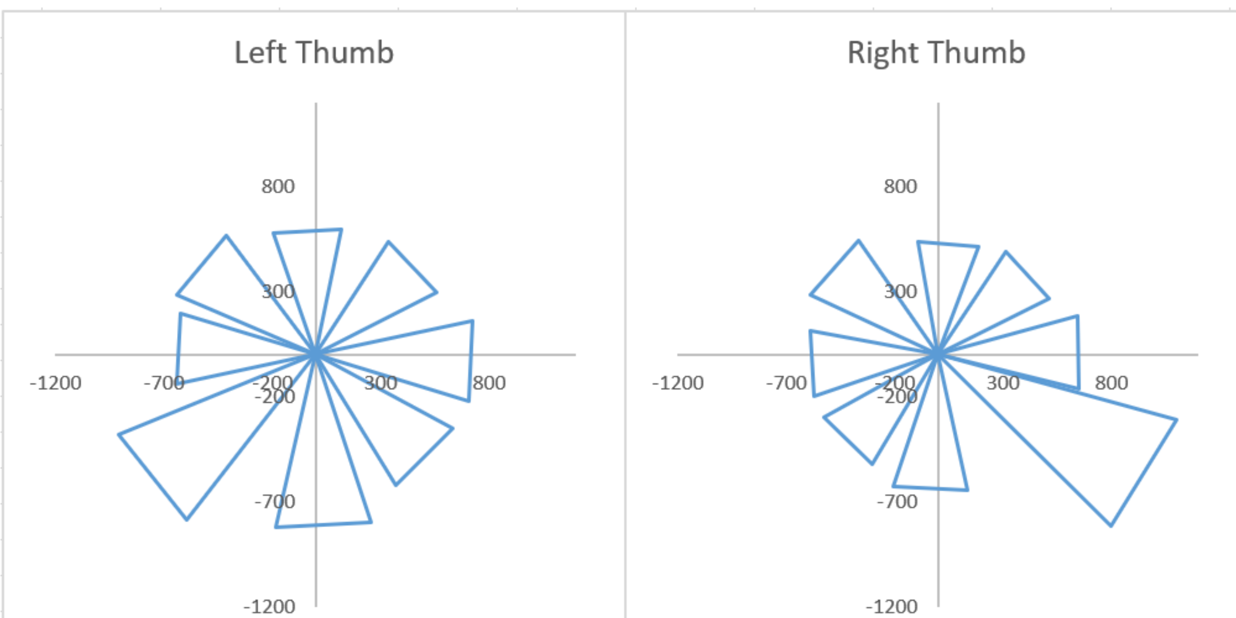


Figure 5.14 Breakdown of thumb holding time with the different direction angles in Eyes-on Mode

Table 5.8 Mean thumb move distance for each direction

	Eyes-free Mode	Eyes-on Mode
Direction	Mean move length (pixel)	Mean move length (pixel)
QW	174.25	137.98
ER	144.76	99.33
T	115.75	121.63
ASD	143.87	113.32
FG	125.40	100.12
ZX	162.39	137.82
C	152.93	112.76
VB	167.55	129.18
Y	110.37	123.76
UI	133.32	98.34
OP	160.43	128.84
HJ	125.75	105.41
KL	132.08	103.64
N	151.65	127.10
M	144.03	106.86
KL	107.61	132.329

5.7 Direction angle and length

The thumb move length in each direction facilitate another way to analyze the performance of this text entry approach. Table 5.8 shows the mean length of the thumb move distance for each direction in the two modes, and its corresponding angle is displayed in Table 5.7. For most directions the thumb move distance in eyes-free mode are greater than the thumb move distance in eyes-on mode. This may be explained by the visual keyboard in eyes-on mode. The character can be retrieved when the participants drag their thumbs to the direction of the desired character. More specifically, if the thumb move distance is greater than the threshold distance (60 pixels), the background of the chosen letters group will become darker. In eyes-on mode this visual feedback helps participants control their thumb moving distance. While in eyes-free mode the participants subconsciously move their thumb distance longer without the visual feedback conditions to ensure they can retrieve letters.

Figures 5.15 and 5.16 show the polar plots of the thumb move distance with each direction angle for the two modes, respectively. The results illustrate that the distribution of thumb move distances for the eight directions are in mirrored symmetrically across the hands. Without the left and right visual keyboards (see Figure 3.2 and Figure 3.3) in eyes-free mode, the distribution of thumb move distance resembles oblique ellipses. This result is as expected, based on the physiological structure of the human’s thumb, the lateral movement range of the thumb joint is greater than the vertical movement range of the thumb joint. For example, in Figure 5.13, the left thumb move distance in northwest (174.25 pixels) and southeast (167.55 pixels) directions are greater than the move distance in northeast (115.75 pixels) and southwest (162.39 pixels) directions for left thumb. Besides that, the right thumb move distance in northeast (160.43 pixels) and southwest (151.65 pixels) directions are greater than the right thumb move distance in northwest (110.37 pixels) and southeast (107.61 pixels) directions. While, due to the visual feedback in eyes-on mode, the distribution of thumb move distance (see Figure 5.16) just like the distribution of letter groups in visual keyboards (see Figure 3.2 and Figure 3.3).

5.8 Subjective assessment

A face-to face post experiment questionnaire (on a 6-point Likert scale) about the subjective assessment of this proposed text entry method were conducted after the participants completed all the four training sessions. Firstly, each participant was asked whether they found the tested application was easy to use. And then the two groups of participants were inquired about their self-assessed performance using the tested applications. All the responses were in the upper groups of the Liker scale which are from 3 to 6. At last, a Mann Whitney U test was used to show that there was no significant difference in terms of usability of the two applications between the eyes-on and eyes-free groups (U = 33, p = .195), and there was no significant difference in self-assessed performance between the two groups (U = 32.5, p = .170).

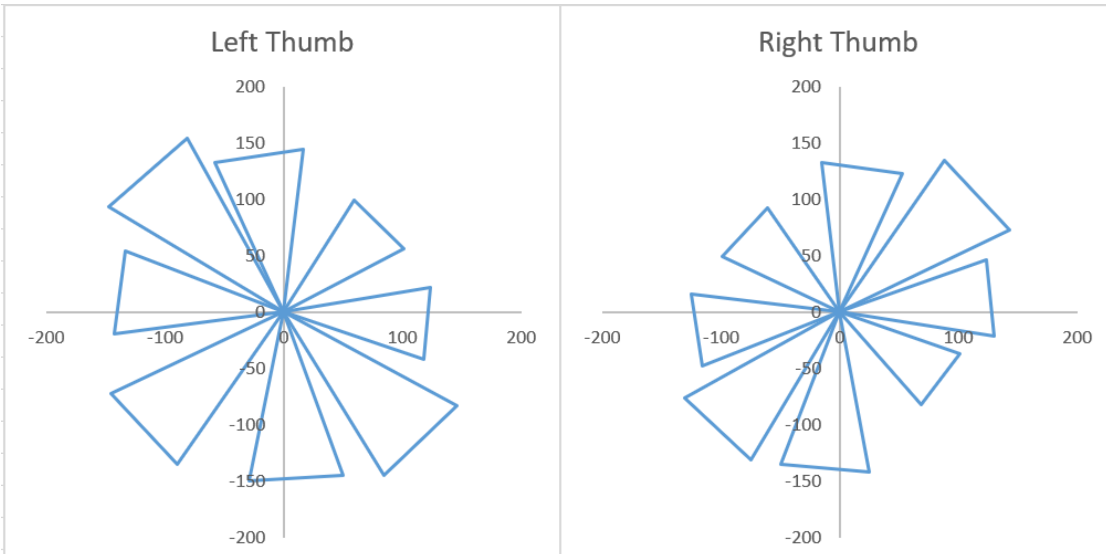


Figure 5.15 Breakdown of thumb move distance in the different direction angles in Eyes-free Mode

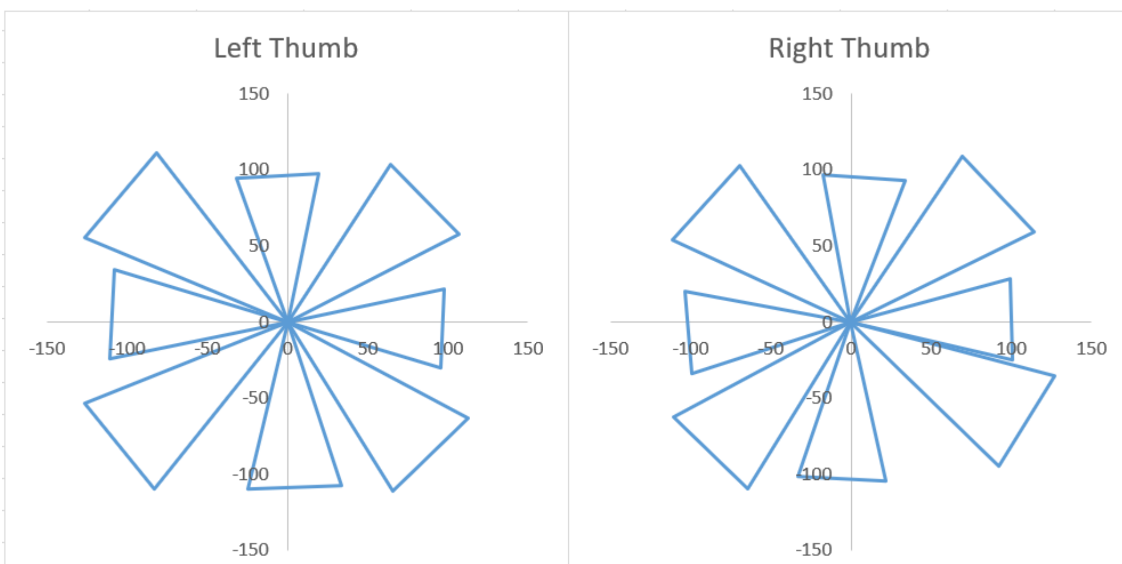


Figure 5.16 Breakdown of thumb move distance in the different direction angles in Eyes-on Mode

6. Conclusion

This study utilized the user familiar QWERTY keyboard layout and simple bimanual gestures to explore the eyes-free text entry strategy on smartphone. According to human hands symmetry, the QWERTY keyboard layout split into two symmetric sections and each part consist of seven or eight groups of characters (see Figure 3.2 and 3.3). The specific character was retrieved by moving left or right thumb to the direction of the required characters group box. A longitudinal user study showed that the users achieved a text entry speed of 11.1 WPM in eyes-free mode and 14.1 WPM in eyes-on mode, and the error rate was 9.9% for eyes-free mode and 3.9% for eyes-on mode after four training sessions. Furthermore, an expert evaluation shows the peak text entry rate at 24.85 WPM with an error rate of 2.27% without referencing help in the eyes-free mode. The results are comparable in performances to other text entry approaches. And the longitudinal data suggests that the text entry performance can be improved with further training. The results also demonstrated that the visual keyboard in the eyes-on mode application provides a finger movement pattern to the users which constrained the users' thumb motion. Besides that, all participants relied on the word suggestions list to entry text in eyes-on mode application, which yielded 34.9% of output/input gain during the four training sessions. Since the proposed text entry approach can be used eyes-free, it holds potential for visually impaired or/and blind users. However, there are still some limitations in this study, for instance the error rates were high without providing appropriate word disambiguating mechanisms in eyes-free mode. A robust error correction mechanism such as language model based on automatic word disambiguating could be resolved in the future work of this study. Moreover, this user study does not include any blind or/and visually impaired users, hence, some visually impaired or/and blind users should be involved in the further user studies.

Reference

Azenkot, S., Bennett, C. L., & Ladner, R. E. (2013). *DigiTaps: eyes-free number entry on touchscreens with minimal audio feedback.* Paper presented at the Proceedings of the 26th annual ACM symposium on User interface software and technology.

Azenkot, S., & Zhai, S. (2012). *Touch behavior with different postures on soft smartphone keyboards.* Paper presented at the Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services.

Bi, X., Chelba, C., Ouyang, T., Partridge, K., & Zhai, S. (2012). *Bimanual gesture keyboard.* Paper presented at the Proceedings of the 25th annual ACM symposium on User interface software and technology.

Bi, X., Ouyang, T., & Zhai, S. (2014). *Both complete and correct?: multi-objective optimization of touchscreen keyboard.* Paper presented at the Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.

Bi, X., Smith, B. A., & Zhai, S. (2010). *Quasi-qwerty soft keyboard optimization.* Paper presented at the Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.

Boissiere, P., & Dours, D. (2003). *An overview of existing writing assistance systems.* Paper presented at the Proceedings of the IFRATH Workshop.

Buxton, W., & Myers, B. (1986). *A study in two-handed input.* Paper presented at the ACM SIGCHI Bulletin.

Buzzi, M. C., Buzzi, M., Leporini, B., & Trujillo, A. (2014). *Designing a text entry multimodal keypad for blind users of touchscreen mobile phones.* Paper presented at the Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility.

Card, S. K., Newell, A., & Moran, T. P. (1983). *The Psychology of Human-Computer Interaction*: L. Erlbaum Associates Inc.

Cassingham, R. C. (1986). The dvorak keyboard. *Freelance Communications*.

Castellucci, S. J., & MacKenzie, I. S. (2008). *Graffiti vs. unistrokes: an empirical comparison.* Paper presented at the Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.

Chau, D. H., Wobbrock, J. O., Myers, B. A., & Rothrock, B. (2006). *Integrating isometric joysticks into mobile phones for text entry.* Paper presented at the CHI'06 extended abstracts on Human factors in computing systems.

COLEMAN, S. (2013). Colemak keyboard layout.

Cuaresma, J., & MacKenzie, I. S. (2013). *A study of variations of Qwerty soft keyboards for mobile phones.* Paper presented at the Proceedings of the International Conference on Multimedia and Human-Computer Interaction-MHCI.

Darragh, J. J., Witten, I. H., & James, M. L. (1990). The reactive keyboard: A predictive typing aid. *Computer, 23*(11), 41-49.

Dunlop, M. D., Durga, N., Motaparti, S., Dona, P., & Medapuram, V. (2012). *QWERTH: an optimized semi-ambiguous keyboard design.* Paper presented at the Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services companion.

Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of experimental psychology, 47*(6), 381.

Fredkin, E. (1960). Trie memory. *Communications of the ACM, 3*(9), 490-499.

Fuglerud, S. (2014). *Inclusive design of ICT: The challenge of diversity.* PhD dissertation. Oslo: Department of Media and Communication, University of Oslo. Available full-text DOI: 10.13140/2.1. 4471.5844.

Goel, M., Jansen, A., Mandel, T., Patel, S. N., & Wobbrock, J. O. (2013). *ContextType: using hand posture information to improve mobile touch screen text entry.* Paper presented at the Proceedings of the SIGCHI conference on human factors in computing systems.

Goh, T., & Kim, S. W. (2014). *Eyes-free text entry interface based on contact area for people with visual impairment.* Paper presented at the Proceedings of the adjunct publication of the 27th annual ACM symposium on User interface software and technology.

Goldberg, D., & Richardson, C. (1993). *Touch-typing with a stylus.* Paper presented at the Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems.

Goodman, J., Venolia, G., Steury, K., & Parker, C. (2002). *Language modeling for soft keyboards.* Paper presented at the Proceedings of the 7th international conference on Intelligent user interfaces.

Guerreiro, T., Lagoá, P., Nicolau, H., Santana, P., & Jorge, J. (2008). *Mobile text-entry models for people with disabilities.* Paper presented at the Proceedings of the 15th European conference on Cognitive ergonomics: the ergonomics of cool interaction.

Gunawardana, A., Paek, T., & Meek, C. (2010). *Usability guided key-target resizing for soft keyboards.* Paper presented at the Proceedings of the 15th international conference on Intelligent user interfaces.

Harper, S., & Yesilada, Y. (2008). *Web accessibility: a foundation for research*: Springer Science & Business Media.

Hopkins, W. G. (2008). Quantitative research design.

Jain, M., & Balakrishnan, R. (2012). *User learning and performance with bezel menus.* Paper presented at the Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.

Jones, P. E. (1998). *Virtual keyboard with scanning and augmented by prediction.* Paper presented at the Proceedings of European Conference series on Disability, Virtual Reality and Associated Technologies. The university of Reading. available from http://www. icdvrat. reading. ac. uk/1998/papers/1998.

Kane, S. K., Bigham, J. P., & Wobbrock, J. O. (2008). *Slide rule: making mobile touch screens accessible to blind people using multi-touch interaction techniques.* Paper presented at the Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility.

Karat, C.-M., Halverson, C., Horn, D., & Karat, J. (1999). *Patterns of entry and correction in large vocabulary continuous speech recognition systems.* Paper presented at the Proceedings of the SIGCHI conference on Human Factors in Computing Systems.

Kim, H., Row, Y.-k., & Lee, G. (2012). *Back keyboard: a physical keyboard on backside of mobile phone using qwerty.* Paper presented at the CHI'12 Extended Abstracts on Human Factors in Computing Systems.

Kristensson, P.-O., & Zhai, S. (2005). *Relaxing stylus typing precision by geometric pattern matching.* Paper presented at the Proceedings of the 10th international conference on Intelligent user interfaces.

Kurauchi, A., Feng, W., Joshi, A., Morimoto, C., & Betke, M. (2016). *EyeSwipe: Dwell-free text entry using gaze paths.* Paper presented at the Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems.

Kristensson, P. O., & Vertanen, K. (2012). *Performance comparisons of phrase sets and presentation styles for text entry evaluations.* Paper presented at the Proceedings of the 2012 ACM international conference on Intelligent User Interfaces.

Lee, P. U.-J., & Zhai, S. (2004). Top-down learning strategies: can they facilitate stylus keyboard learning? *International Journal of Human-Computer Studies, 60*(5-6), 585-598.

Leporini, B., Buzzi, M. C., & Buzzi, M. (2012). *Interacting with mobile devices via VoiceOver: usability and accessibility issues.* Paper presented at the Proceedings of the 24th Australian Computer-Human Interaction Conference.

Lewis, J. R., Potosnak, K. M., & Magyar, R. L. (1997). Keys and keyboards *Handbook of Human-Computer Interaction (Second Edition)* (pp. 1285-1315): Elsevier.

MacKenzie, I. S. (2012). *Human-computer interaction: An empirical research perspective*: Newnes.

MacKenzie, I. S., & Ashtiani, B. (2011). BlinkWrite: efficient text entry using eye blinks. *Universal Access in the Information Society, 10*(1), 69-80.

MacKenzie, I. S., & Soukoreff, R. W. (2002a). *A character-level error analysis technique for evaluating text entry methods.* Paper presented at the Proceedings of the second Nordic conference on Human-computer interaction.

MacKenzie, I. S., & Soukoreff, R. W. (2002b). A model of two-thumb text entry. *SPACE, 67*, 18-43.

MacKenzie, I. S., & Soukoreff, R. W. (2002c). Text entry for mobile computing: Models and methods, theory and practice. *Human–Computer Interaction, 17*(2-3), 147-198.

MacKenzie, I. S., & Soukoreff, R. W. (2003). *Phrase sets for evaluating text entry techniques.* Paper presented at the CHI'03 extended abstracts on Human factors in computing systems.

MacKenzie, I. S., Soukoreff, R. W., & Helga, J. (2011). *1 thumb, 4 buttons, 20 words per minute: Design and evaluation of H4-Writer.* Paper presented at the Proceedings of the 24th annual ACM symposium on User interface software and technology.

MacKenzie, I. S., & Zhang, S. X. (1999). *The design and evaluation of a high-performance soft keyboard.* Paper presented at the Proceedings of the SIGCHI conference on Human Factors in Computing Systems.

MacKenzie, I. S., Zhang, S. X., & Soukoreff, R. W. (1999). Text entry using soft keyboards. *Behaviour & information technology, 18*(4), 235-244.

Mankoff, J., & Abowd, G. D. (1998). *Cirrin: a word-level unistroke keyboard for pen input.* Paper presented at the Proceedings of the 11th annual ACM symposium on User interface software and technology.

Martin, D. W. (2007). *Doing psychology experiments*: Cengage Learning.

Matias, E., MacKenzie, I. S., & Buxton, W. (1993). *Half-QWERTY: A one-handed keyboard facilitating skill transfer from QWERTY.* Paper presented at the Proceedings of the INTERACT'93 and CHI'93 Conference on Human Factors in Computing Systems.

Matias, E., MacKenzie, I. S., & Buxton, W. (1994). *Half-QWERTY: typing with one hand using your two-handed skills.* Paper presented at the Conference companion on Human factors in computing systems.

Mizobuchi, S., Chignell, M., & Newton, D. (2005). *Mobile text entry: relationship between walking speed and text input task difficulty.* Paper presented at the Proceedings of the 7th international conference on Human computer interaction with mobile devices & services.

Munteanu, C., & Penn, G. (2015). *Speech-based interaction: myths, challenges, and opportunities.* Paper presented at the Proceedings of the 20th International Conference on Intelligent User Interfaces.

Nicolau, H., & Jorge, J. (2012). *Touch typing using thumbs: understanding the effect of mobility and hand posture.* Paper presented at the Proceedings of the SIGCHI conference on human factors in computing systems.

Noyes, J. (1983). The QWERTY keyboard: A review. *International Journal of Man-Machine Studies, 18*(3), 265-281.

Oliveira, J., Guerreiro, T., Nicolau, H., Jorge, J., & Gonçalves, D. (2011). *Blind people and mobile touch-based text-entry: acknowledging the need for different flavors.* Paper presented at the The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility.

Pedrosa, D., Pimentel, M. d. G., & Truong, K. N. (2015). *Filteryedping: A dwell-free eye typing technique.* Paper presented at the Proceedings of the 33rd annual acm conference extended abstracts on human factors in computing systems.

Perlin, K. (1998). *Quikwriting: continuous stylus-based text entry.* Paper presented at the Proceedings of the 11th annual ACM symposium on User interface software and technology.

Plamondon, R., & Srihari, S. N. (2000). Online and off-line handwriting recognition: a comprehensive survey. *IEEE Transactions on pattern analysis and machine intelligence, 22*(1), 63-84.

Polacek, O., Sporka, A. J., & Slavik, P. (2017). Text input for motor-impaired people. *Universal Access in the Information Society, 16*(1), 51-72.

Pollmann, F., Wenig, D., & Malaka, R. (2014). *HoverZoom: making on-screen keyboards more accessible.* Paper presented at the CHI'14 Extended Abstracts on Human Factors in Computing Systems.

Raynal, M. (2014). *KeyGlasses: semi-transparent keys on soft keyboard.* Paper presented at the Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility.

Rodrigues, É., Carreira, M., & Gonçalves, D. (2016). Enhancing typing performance of older adults on tablets. *Universal Access in the Information Society, 15*(3), 393-418.

Sandnes, F. E., & Aubert, A. (2007). Bimanual text entry using game controllers: Relying on users' spatial familiarity with QWERTY. *Interacting with Computers, 19*(2), 140-150.

Sandnes, F. E., Tan, T. B., Johansen, A., Sulic, E., Vesterhus, E., & Iversen, E. R. (2012). Making touch-based kiosks accessible to blind users through simple gestures. *Universal Access in the Information Society, 11*(4), 421-431.

Sears, A., & Zha, Y. (2003). Data entry for mobile devices using soft keyboards: Understanding the effects of keyboard size and user tasks. *International Journal of Human-Computer Interaction, 16*(2), 163-184.

Shi, W., Yu, C., Yi, X., Li, Z., & Shi, Y. (2018). TOAST: Ten-Finger Eyes-Free Typing on Touchable Surfaces. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 2*(1), 33.

Silfverberg, M., MacKenzie, I. S., & Kauppinen, T. (2001). *An isometric joystick as a pointing device for handheld information terminals.* Paper presented at the Graphics Interface.

Soukoreff, R. W., & MacKenzie, I. S. (2003). *Metrics for text entry research: an evaluation of MSD and KSPC, and a new unified error metric.* Paper presented at the Proceedings of the SIGCHI conference on Human factors in computing systems.

Soukoreff, R. W., & MacKenzie, I. S. (2004). *Recent developments in text-entry error rate measurement.* Paper presented at the CHI'04 extended abstracts on Human factors in computing systems.

Tinwala, H., & MacKenzie, I. S. (2010). *Eyes-free text entry with error correction on touchscreen mobile devices.* Paper presented at the Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries.

Ward, D. J., Blackwell, A. F., & MacKay, D. J. (2000). *Dasher—a data entry interface using continuous gestures and language models.* Paper presented at the Proceedings of the 13th annual ACM symposium on User interface software and technology.

Wigdor, D., & Balakrishnan, R. (2003). *TiltText: using tilt for text input to mobile phones.* Paper presented at the Proceedings of the 16th annual ACM symposium on User interface software and technology.

Wobbrock, J. O. (2007). *Measures of text entry performance*: San Francisco: Morgan Kaufmann.

Wobbrock, J. O., Aung, H. H., Myers, B. A., & Lopresti, E. F. (2005). Integrated text entry from power wheelchairs. *Behaviour & information technology, 24*(3), 187-203.

Wobbrock, J. O., Myers, B. A., Aung, H. H., & LoPresti, E. F. (2004). *Text entry from power wheelchairs: EdgeWrite for joysticks and touchpads.* Paper presented at the ACM SIGACCESS Accessibility and Computing.

Wobbrock, J. O., Myers, B. A., & Kembel, J. A. (2003). *EdgeWrite: a stylus-based text entry method designed for high accuracy and stability of motion.* Paper presented at the Proceedings of the 16th annual ACM symposium on User interface software and technology.

Yamada, H. (1980). *A historical study of typewriters and typing methods, from the position of planning Japanese parallels*: Journal of Information Processing.

Yeo, H.-S., Phang, X.-S., Castellucci, S. J., Kristensson, P. O., & Quigley, A. (2017). *Investigating tilt-based gesture keyboard entry for single-handed text entry on large devices.* Paper presented at the Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems.

Zhai, S., Kristensson, P.-O., & Smith, B. A. (2004). In search of effective text input interfaces for off the desktop computing. *Interacting with Computers, 17*(3), 229-250.

Zhai, S., & Smith, B. A. (2001). *Alphabetically biased virtual keyboards are easier to use: layout does matter.* Paper presented at the CHI'01 extended abstracts on Human factors in computing systems.