ThumbStroke: A Virtual Keyboard in Support of Sight-Free and One-Handed Text Entry on Touchscreen Mobile Devices

JIANWEI LAI, Illinois State University
DONGSONG ZHANG, The University of North Carolina at Charlotte
SEN WANG, Compeer Financial
ISIL DOGA YAKUT KILIC, The University of Maryland, Baltimore County
LINA ZHOU, The University of North Carolina at Charlotte

11

The QWERTY keyboard on mobile devices usually requires users' full visual attention and both hands, which is not always possible. We propose a thumb-stroke-based keyboard, ThumbStroke, to support both sight-free and one-handed text entry. Text entry via ThumbStroke completely relies on the directions of thumb strokes at any place on the screen of a mobile device. It does not require physical press on any specific keys, thus eliminating the need for visual attention and reducing errors due to tiny key size, fat thumbs, limited thumb reachability, and visual occlusion. We empirically evaluated ThumbStroke through a 20-session longitudinal controlled lab experiment. ThumbStroke shows advantages in typing accuracy and user perceptions in comparison to the Escape and QWERTY keyboards and results in faster typing speed than QWERTY in sight-free and one-handed text entry. This study provides novel research contributions to mobile HCI, advancing the design of soft keyboards for one-handed interaction with mobile devices and mobile accessibility.

CCS Concepts: • **Human-centered computing** → **Keyboards**; **Text input**; **Gestural input**;

Additional Key Words and Phrases: ThumbStroke, keyboard, text entry, one-handed, sight-free, stroke

ACM Reference format:
Jianwei Lai, Dongsong Zhang, Sen Wang, Isil Doga Yakut Kilic, and Lina Zhou. 2019. ThumbStroke: A Virtual Keyboard in Support of Sight-Free and One-Handed Text Entry on Touchscreen Mobile Devices. *ACM Trans. Manage. Inf. Syst.* 10, 3, Article 11 (September 2019), 19 pages.
https://doi.org/10.1145/3343858

1 INTRODUCTION

Today, there are approximately 5 billion users of mobile handheld devices (e.g., smartphones and tablets) worldwide. Despite the convenience resulting from their portability and mobility,

This material is based on work supported by the National Science Foundation (NSF) (award #: CNS 1704800). Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.
Authors' addresses: J. Lai, School of Information Technology, Illinois State University, Normal, IL 61790; email: jlai12@ilstu.edu; D. Zhang and L. Zhou, Department of Business Information Systems & Operations Management, The University of North Carolina at Charlotte, 9201 University City Blvd., Charlotte, NC 28223; emails: {dzhang15, lzhou8}@uncc.edu; S. Wang, Compeer Financial, 2000 Jacobssen Dr., Normal, IL 61704; email: ws21cn@gmail.com; I. D. Y. Kilic, Department of Information Systems, University of Maryland, Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250; email: yakut1@umbc.edu.

2158-656X/2019/09-ART11 $15.00
https://doi.org/10.1145/3343858

mobile devices face significant usability and accessibility challenges due to their physical constraints, such as small screen size and the fat finger problem (Lai and Zhang 2015). Designing effective text entry methods for improving accessibility and usability of mobile devices has been not only a major research issue in the mobile HCI (Human-Computer Interaction) field but also a part of information system research (e.g., Adipat et al. (2011), Hoehle and Venkatesh (2015), and Steinbart et al. (2016)). Hoehle and Venkatesh (2015) suggested that user interface input, including keyboard, would have a significant impact on users' continued intention to use mobile apps. Middleton et al. (2014) argue that because users interact with mobile apps through a variety of input mechanisms (including voice, text, and location), designing effective mobile access is essential.

Current touchscreen mobile devices do not support text entry very well. First, text entry with QWERTY on mobile devices suffers from common challenges of interactions with touchscreen mobile devices, including visual occlusion, inaccurate key selection due to tiny keys and a fat finger, and limited reachability of a thumb (Lai and Zhang 2015; Roudaut et al. 2008). Second, users often use mobile devices in motion, in which they cannot devote all of their visual attention to their devices. Using mobile phones while walking leads to increased cognitive distraction, reduced situation awareness, and unsafe behaviors (Lamberg and Muratori 2012). There have been increasing incidents, some leading to injuries or even death of users, because users focused on their mobile phones while walking (Tatro and Fleming 2015). According to the UK National Accident Helpline (2016), 46% of users have put themselves in danger due to distraction while walking or driving. Therefore, a sight-free text entry technique can be beneficial. This need is much stronger and more obvious for users with visual impairments. According to the World Health Organization (WHO), 285 million people are visually impaired worldwide. Third, although people generally like one-handed interaction with mobile devices (Gold et al. 2012; Karlson and Bederson 2007), current text entry on mobile devices largely requires users to use both hands. One-handed text entry is especially beneficial for users with arm or hand disabilities or situational impairments. The latter refers to users' temporary difficulty in accessing mobile devices due to specific context or situations that they are in (Korhonen et al. 2007). For example, when a user is holding a cup of coffee, he or she has only one hand available to hold and interact with a mobile device. According to the U.S. Centers for Disease Control and Prevention (CDC), there are approximately 2 million people in the United States living with limb loss, and more than 500 Americans lose a limb every day. Furthermore, there are some benefits of supporting one-handed and sight-free interaction at the same time. For example, visually impaired users usually use canes to help them walk. A keyboard that supports both one-handed and sight-free text entry could be beneficial for this group of users while they are on the go. Users may want to interact with a phone without looking at it when overt use of mobile devices is socially inappropriate (Yi et al. 2012). In addition, enabling one-handed text entry in a sight-free manner provides users with protection from eavesdropping (Fukatsu et al. 2013). Luke Wroblewski, product director at Google, states that the best form of interaction with a phone should enable mobile users to engage in one-handed use with short spans of partial attention (Siang 2019).

Despite the benefits, one-handed text entry also introduces new usability challenges because users need to secure a device with their palm and four fingers while reaching keys with the thumb, which has limited flexion and extension (Lai and Zhang 2015). There have been some studies on one-handed or sight-free text entry techniques for mobile devices. Braille-based techniques, such as Holibraille (Nicolau et al. 2015) and EdgeBraille (Mattheiss et al. 2014), are only helpful for people who are familiar with Braille. Although some existing keyboards have a built-in one-handed mode, such as Google Keyboard 5.0, in which a keyboard is shifted to one side of the screen to make distant keys easier to be reached with a thumb, the fundamental problem of tiny keys is not fixed.

There is a need for more effective techniques to support one-handed and sight-free text entry on mobile devices.

The overarching goal of this research is to design, develop, and evaluate a novel keyboard in support of effective one-handed and sight-free text entry on touchscreen mobile devices by addressing the above-mentioned challenges and limitations of existing approaches. Design science is an important research paradigm in information systems (Niederman and March 2012; Nunamaker and Briggs 2011). By following the design science research paradigm (Hevner et al. 2004), the proposed ThumbStroke, a new artifact, enables users to enter text characters on a mobile device based on thumb strokes on the device screen without the need to physically press any specific keys on the keyboard. It offers several unique design advantages in comparison to the common QWERTY keyboard that is widely used in current mobile devices. First, using ThumbStroke, users enter characters by making strokes using a thumb at any place toward certain directions on a device screen, instead of reaching out to and pressing any keys physically, which solves the problem of limited thumb accessibility. Second, when users press on keys on QWERTY, their thumb will cover the content underneath, causing the visual occlusion problem (Scheibel et al. 2013). ThumbStroke avoids this problem completely by not requiring physical press on keys during text entry, also eliminating the negative effect of small key size on text entry. Third, ThumbStroke does not require visual attention to keys, thus supporting sight-free text entry. This research provides several novel contributions to the design principles of effective and accessible input mechanisms for improvement of usability and accessibility of touchscreen mobile devices. The proposed ThumbStroke keyboard also offers practical benefits and implications for mobile device users.

The rest of the article will be organized as follows. First, we will introduce the literature on sight-free and one-handed text entry methods for mobile devices in Section 2. Then, the design of the proposed ThumbStroke keyboard will be presented in detail in Section 3, followed by the description of an empirical evaluation of ThumbStroke in Section 4. The results of the evaluation will be presented in Section 5. Finally, the article will discuss the major findings, research contributions, practical implications, and limitations and future research in Section 6.

2 RELATED WORK

2.1 Sight-Free Text Entry

Screen reader software such as VoiceOver on iPhones and TalkBack on Android phones reads out the corresponding letter when users press a key on a regular QWERTY keyboard. This type of text entry method requires a user to press keys, which is especially challenging for a blind user. Speech input methods like Siri raise privacy and security concerns. Braille-based techniques (e.g., EdgeBraille (Mattheiss et al. 2014), Holibraille (Nicolau et al. 2015), LêBraille (Façanha et al. 2014), BrailleTouch (Southern et al. 2012), and BrailleKey (Subash et al. 2012)) are designed for visually impaired users. They are not suitable for those who do not use Braille and for people without visual impairments.

Some keyboards have been designed for people with visual impairments or situational impairments. For example, Escape (Banovic et al. 2013) consists of several isolated areas with a flower-shaped keypad in each area. In each keypad, there are several letters located at different directions from the center of the area (Figure 1(a)). It requires users to select an area first and then make a selection of a character in that area. Users enter the letter in the center of a flower by tapping on anywhere in the corresponding area. For the letters in the petals, users need to reach to the area first and then flip toward the corresponding directions. Users need to remember the locations of those areas on the screen to make it possible to reach them accurately without looking at the screen. No-Look Notes (Bonner et al. 2010) is a menu-based technique. It divides a device screen

(a) Escape (b) T9 (c) Thick Buttons (d) QWERTY in the Right-Handed Mode

Fig. 1. Some soft keyboards for mobile devices.

into small segments with characters in each segment. The user first needs to select a segment, which will take him or her to another screen in which the segment's characters are presented. Users then need to select the target character. This two-step approach is inefficient.

2.2 One-Handed Text Entry

Some keyboards have been designed to address the problem of tiny keys. For example, the T9 keyboard (Figure 1(b)) has multiple characters on each key, aiming to reduce the number of keys on a keyboard. Although it increases the size of keys, it introduces selection ambiguity due to multiple characters on the same key. Thick Buttons (Figure 1(c)) (Page 2013) enlarges and highlights keys that are most likely to be pressed next and shrinks other keys to improve typing accuracy. However, it is unsuitable for sight-free text entry because the sizes of the keys change dynamically. Some keyboards, such as Google Keyboard 5.0, have a one-handed mode, in which QWERTY is scaled down and shifted toward one side of the screen, but the size of keys becomes even smaller (Figure 1(d)). FingerT9 (Wong et al. 2018) maps a T9 keyboard layout to the finger segments and allows users to enter text by tapping on the sensors attached to their fingers with the thumb of the same hand. It has the potential to support both sight-free and one-handed text entry. However, its requirement of attaching sensors to fingers may limit its usage and user adoption.

2.3 Stroke/Gesture-Based Keyboards

There exist a few finger-stroke- and gesture-based text entry methods (e.g., Banovic et al. (2013), Costagliola et al. (2011), Wobbrock et al. (2003), and Zhai et al. (2009)). Instead of tapping on individual keys, a user using ShapeWriter (Zhai et al. 2009) can enter a word by sliding a finger through all the letters in the word consecutively. The keyboard approximately traces all letters slid through and analyzes them using a statistical model. This keyboard can be quite efficient but not practical for sight-free text entry. KeyScretch (Costagliola et al. 2011) allows users to type via both taps and strokes. Still, it does not address the limited thumb reachability and visual occlusion problems. Swipeboard (Chen et al. 2014) uses two separate gestures to select a letter, with the first swipe to choose the area and the second swipe to choose a letter, which is actually quite common in menu-based text entry methods, such as No-Look Notes (Bonner et al. 2010). They both have the potential to be used for sight-free text entry. However, users have to switch between keypads and may be confused about which keypad they are using at a specific time.

Despite prior studies on soft keyboards for sight-free and/or one-handed text entry on mobile devices, as summarized in Table 1, they have some limitations. More importantly, few of the existing keyboards can support sight-free and one-handed text entry simultaneously, which motivates this research.

Table 1. Summary of Existing Sight-Free/One-Handed Keyboards

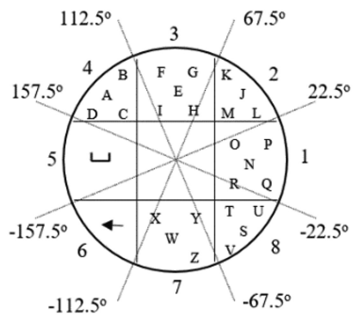
	Methods		Text Entry Support	Limitations
Sight-free	Voice based: VoiceOver, TalkBack		Voice feedback	Users need to accurately locate the characters
	Braille based (Façanha et al. 2014; Mattheiss et al. 2014; Nicolau et al. 2015; Southern et al. 2012; Subash et al. 2012)		Simulating typing with Braille	Keyboard layout is not letter based and only for visually impaired users
	Stroke/gesture based:	Escape (Banovic et al. 2013)	Avoid accurate pressing on keys	Users need to reach out to keypads
		EdgeWrite (Wobbrock et al. 2003)		Not for one-handed use
	Menu based: No-Look Notes (Bonner et al. 2010) Swipeboard (Chen et al. 2014)			Two steps to select a character/word
One-handed	Google Keyboard 5.0 and Microsoft's Word Flow keyboard		Shift keyboards to one side of the screen or in an arc	Does not solve the problem of tiny keys and finger visual occlusion
	Escape (Banovic et al. 2013)			Users need to reach out to keypads
	FingerT9 (Wong et al. 2018)		Tapping on sensors attached to fingers	Need to have sensors on fingers

3 DESIGN OF THUMBSTROKE

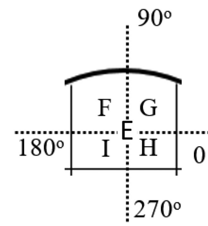
According to the Fitts's law (Fitts 1954), the speed of moving to a target on a screen is influenced by the distance to the target and target size. Hence, there are two ways to reduce the difficulty in a pointing task, namely enlarging a target and bringing it closer (Blanch et al. 2004). Shifting a keyboard to the reachable area of a thumb will make keys smaller, which will make text entry more error-prone. Indirect input methods can extend the reach of a thumb to a distant element (Pfeuffer et al. 2017), which is equivalent to bringing a target closer. Therefore, we are interested in designing ThumbStroke as an indirect input method so that users will not be constrained by the positions of keys.

To minimize the impact of key size on text entry, we use a direction-based key selection mechanism for ThumbStroke. The direction-based selection has been used in marking menus (Kurtenbach and Buxton 1994), in which finger strokes toward different directions are used to select menu items. Bragdon et al. (2011) found that users could perform precise stroke gestures on a mobile phone with little visual attention and some gestures could be used in sight-free and one-handed interaction. Specifically, gestures in eight directions, including up, down, left, right, upper-left, lower-left, upper-right, and lower-right, are simple, quick to execute, and potentially tolerant of imprecision due to rapid execution (Bragdon et al. 2011). Hence, the design of ThumbStroke combines both indirect input and directional stroke-based selection, making ThumbStroke independent of both position and size of keys.

ThumbStroke is a virtual keyboard with a single round key, which is divided into eight small areas around its center (Figure 2(a)). With the center of the key as the default starting reference point, each of the eight small areas is located within a certain direction range (i.e., between two adjacent dotted lines shown in Figure 2(a)). In each small area, a character is located at the center as the area reference point, surrounded by three or four other characters located at a certain direction from the center. For example, as shown in Figures 2(a) and 2(b), "A" is located in the center of the



(a) Direction range of small areas



(b) Direction range in a small area

Fig. 2. The design of ThumbStroke.

small area 4, and “G” is located in the direction of 0° to 90° from its area center “E.” According to Lai and Zhang (2014), the angle interval between any two adjacent areas or keys should be no less than 45° for effective selection. Therefore, we adopt this guideline in the design of ThumbStroke, in which each area has a 45° range from the keyboard center, while each character in any area has a 90° range from its area center. The letters in ThumbStroke are generally organized into small areas in a clockwise manner in alphabetical order, aiming to make it easy for users to remember the location of individual letters.

The most fundamental design feature of ThumbStroke lies in that individual characters are selected for text entry solely based on one continuous thumb stroke at any location on a screen. The length of the strokes is scale independent. Figure 3 illustrates how to enter text with ThumbStroke:

- When a user touches a text field, ThumbStroke will automatically appear in the middle of the screen as default. A long press on the keyboard enables users to move it to any location that they prefer. The center of the keyboard will be activated automatically as the starting reference point.
- A user moves his or her thumb on the screen in the direction toward an intended small area where a target character is located. The small area in that direction will be chosen as the current focus area. The character located in the center of that focus small area will be automatically activated as the current reference point, which is highlighted in bold and changed to the red color from the original white color (i.e., the letter “E” in Figure 3(a)). If the user lifts his or her finger away from the screen now, the currently activated letter (i.e., “E”) will be entered into the text field.
- If the user changes the moving direction toward the lower-right corner without lifting his or her thumb away from the screen, the letter “H” will be activated (Figure 3(b)). The user then lifts his or her thumb away from the screen to enter “H.”
- After a letter is selected and entered, ThumbStroke will immediately reset the center of the keyboard as the current reference point.

The moving direction of a thumb is dynamically captured and calculated. When ThumbStroke appears on a device screen, if a user double taps anywhere on the screen, the keyboard will switch between a letter keypad (i.e., Figure 2(a)) and a symbol/number keypad. The layout and entry of the numbers and symbols on the ThumbStroke keyboard are similar to those of the letter keypad. To support sight-free text entry, a selected character can be read out to the user in the same way as Talkback does.

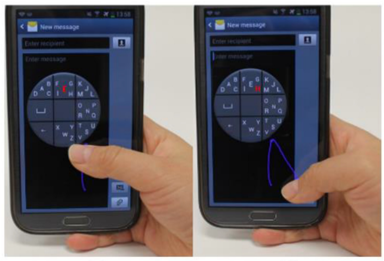

(a) (b)

Fig. 3. Text entry via ThumbStroke.

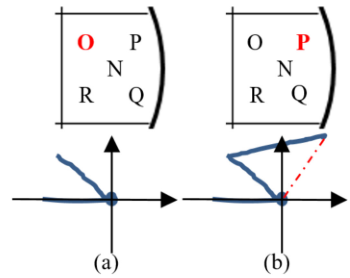


Fig. 4. Error correction with ThumbStroke.

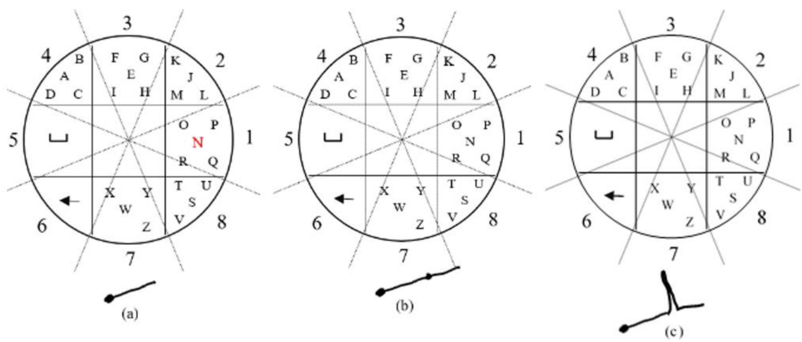


Fig. 5. Area selection cancelation.

Furthermore, ThumbStroke provides an error correction feature. If a user selects a wrong character, he or she can correct it by moving the thumb toward the right direction. For example, when the intended target character is “P” but a user moves his or her thumb in the wrong direction and mistakenly activates “O,” instead of lifting the thumb away from the screen to enter “O,” the user continuously moves the thumb to the right. A new direction, which is represented by the red dotted line, is calculated, and “P” will be activated and selected (see Figure 4). If a user selects a wrong area, he or she can cancel the selection by continuing to move the thumb in the previous direction after a pause (Figure 5(b)) or after selecting a wrong letter (Figure 5(c)).

4 EVALUATION

We conducted a controlled laboratory experiment with a 3*2 (3 keyboards * 2 smartphone sizes) within-subjects design to evaluate ThumbStroke, with Escape (Banovic et al. 2013) and QWERTY as the baseline methods. We chose those two baseline keyboards because the former is a similar stroke-based approach that supports both sight-free and one-handed text entry, while the latter is the most widely used keyboard on mobile devices. Screen size may influence users’ one-handed interaction with mobile devices (Lai and Zhang 2015). Hence, we also examined the potential moderating effect of screen size on one-handed text entry using two smartphones with different screen sizes.

4.1 Participants

The learning curve of a new keyboard makes it very difficult to recruit a large number of participants. Participants typically need to take part in multiple sessions within a certain period of time, which requires a lot of time and effort commitment. Because of this, many previous studies, such as Banovic et al. (2013), Costagliola et al. (2013), Han and Kim (2015), Lyons et al. (2006), and Mackenzie and Felzer (2010), only involved 6 to 12 participants. In this study, we recruited 13 sighted participants (5 male, 8 female) from an East Coast university in the United States, and each participated in 20 individual experiment sessions. They were undergraduate and graduate students with a major in information systems. Five were between 18 and 25 years old; seven were between 26 and 30 years old; and one was over 30 years old. They were all right-handed and had prior experience with touchscreen mobile phones. They all frequently used QWERTY on their mobile phones. They had an average hand length of 17.2cm (SD = 0.9), thumb length of 6.5cm (SD = 0.6), and hand breadth of 9.5cm (SD = 0.7). Each participant received $200 for participating in the experiment.

4.2 Apparatus

ThumbStroke, Escape, and QWERTY were developed in Java and installed on two touchscreen Android phones. One was a Samsung Galaxy Note 2 phone with a 5.5" screen. The other was a Kyocera Event phone with a 3.5" screen. When participants interacted with those phones, the time and pixel coordinates of every interaction would be recorded. To enable the participants to reach individual areas on Escape or specific keys of QWERTY, by following the guideline provided by Banovic et al. (2013), we anchored Escape in the bottom-right corner of the Galaxy Note 2 phone without scaling. The Galaxy Note 2 phone offers a one-handed interaction mode, in which a keyboard can be aligned to the right or left of the screen for right- or left-handed users. We adopted this mode by aligning QWERTY to the right side of the screen (Figure 1(d)) because in a pilot study, 6 out of 31 participants reported difficulty in reaching the keys on QWERTY that were distant from their thumb. For the Kyocera Event phone, which had a smaller screen, Escape and QWERTY were made to fit the width of the screen. The size of Escape was the same as that in Banovic et al. (2013) for both phones.

4.3 Independent and Dependent Measures

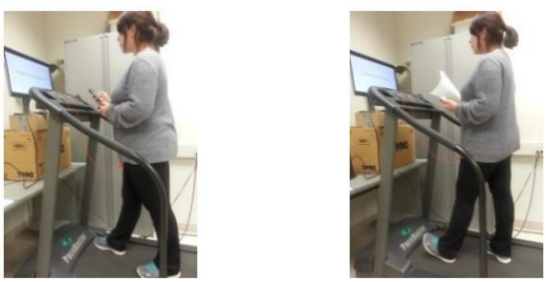
The independent variables are keyboards, phones, and lab sessions. The dependent variables include participants' text entry performance, measured by words per minute (WPM) and error rate, and user perceptions. In WPM, a "word" is defined as five characters, which is the average number of characters in a word (Millet 2009). Regarding error rate, text entry keystrokes can be categorized into four groups (Soukoreff and MacKenzie 2003): Correct (C), Incorrect but Fixed (IF), Incorrect and Not Fixed (INF), and Fixed (F) keystrokes (e.g., backspace). Corrected error rate (CER) refers to the percentage of errors that the participants made and then corrected during text entry, which is calculated as IF/(C+INF+IF). Uncorrected error rate is the percentage of errors that are not corrected (Millet 2009), which is calculated as INF/(C+INF+IF). Participants' perceptions, including perceived ease of use (PEOU), perceived effectiveness, and overall satisfaction, were assessed via a questionnaire consisting of nine 7-point Likert scale questions (Table 2) adapted from the IBM Post-Study System Usability Questionnaire (Lewis 1995).

4.4 Experiment Design

Each participant completed 20 sessions in total. The participants performed text entry tasks in the sighted condition in sessions 1 to 10, in which they were allowed to look at the screen of the phones during text entry. Sessions 11 to 20 were used to evaluate the keyboards for sight-free text entry, in which the screens of mobile phones were covered with a paper cone attached to the participants' wrists with medical tapes (see Figure 6(b)). In each session, the participants were asked to enter 60 different short phrases displayed on a 24-inch monitor in front of them as quickly and accurately

Table 2. Questions of User Perception Factors

Factors	Items (1 = Totally Disagree, 4 = Neutral, 7 = Totally Agree)
Perceived ease of use	Overall, I am satisfied with how easy it is to use this keyboard. It was simple to use this keyboard. It was easy to learn to use this keyboard. I felt comfortable using this keyboard.
Perceived effectiveness	I could effectively complete the tasks using this keyboard. I was able to complete the tasks quickly using this keyboard. I was able to efficiently complete the tasks using this keyboard. I believe I could become productive quickly using this keyboard.
Overall satisfaction	Overall, I am satisfied with this keyboard.



(a) The sighted condition

(b) The sight-free condition

	1	2	3	4	5	6
1	A	B	C	D	E	F
2	B	C	D	E	F	A
3	C	D	E	F	A	B
4	D	E	F	A	B	C
5	E	F	A	B	C	D
6	F	A	B	C	D	E

(c) Latin-square design

Fig. 6. The experiment setup.

as possible. The phrases were randomly selected from the phrase set created by Mackenzie and Soukoreff (2003), which originally included 500 phrases varied from 16 to 43 characters in length. Similar to Banovic et al. (2013), the participants were not allowed to complete more than three sessions within the same day. We did not include the entry of symbols and numbers in the experiment for several reasons. First, including them would increase the workload and cognitive efforts of the participants in the experiment greatly and cause fatigue. Second, it is common in keyboard studies to focus on character entry only. The phrase set created by Mackenzie and Soukoreff (2003) has been widely used in keyboard studies (Kano et al. 2006). It does not include numbers or symbols. Finally, there is no evidence showing that users would enter symbols and numbers much differently from letters.

During each session, the participants entered 10 phrases under each condition, with a total of 60 phrases (i.e., 2 phones * 3 keyboards * 10 phrases). We used a 6 * 6 Latin Square (Grant 1948) to balance out the order of two phones and three keyboards (Figure 6(c), in which letters A to F represent six experimental treatments). For example, if a participant was randomly assigned to order 1 in the first session, he or she would do the second session in order 2, and so on and so forth till order 6 before repeating them again. There was no repetition of phrases used within each session. Because different keyboards apply different autocorrection and word prediction algorithms, autocorrection and word prediction were disabled to minimize possible confounding effects.

To simulate situational impairments and mobility of users in the real world, the participants were asked to enter text phrases while walking on a treadmill (Bergstrom-Lehtovirta et al. 2011; Schabrun et al. 2014) (Figures 6(a) and 6(b)). Following Bergstrom-Lehtovirta et al. (2011), the moving speed of the treadmill was set by individual participants according to their normal walking

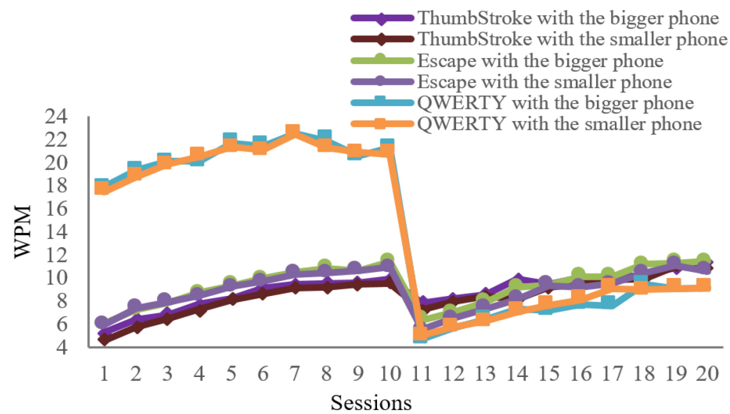


Fig. 7. Means of WPM of the three keyboards.

speed while interacting with a mobile device. The mean of the participants' selected treadmill speed was 2.0km/h (SD = 0.7 km/h). To ensure one-handed interaction, the participants were required to hold a phone and interact with it using their dominant hand only while holding a monitor remote control in the other hand to navigate phrases on the monitor.

4.5 Procedure

The participants received training prior to session 1 to get familiar with ThumbStroke, Escape, and QWERTY. After they felt comfortable with the keyboards and experimental tasks, the sessions started. The participants finished the first 10 sessions in the sighted condition without audio feedback and completed sessions 11 to 20 in the sight-free condition with audio feedback. At the end of the first and last sessions of the sighted and sight-free conditions, the participants filled out the questionnaire about their perceptions (Table 2).

5 RESULTS

The metrics of NotCorrectedErrorRate and CorrectedErrorRate of StreamAnalyzer (Wobbrock and Myers 2006) were modified to calculate UER and CER. Repeated measures ANOVA was applied to evaluate the effects of keyboards, phones, and sessions on WPM, UER, CER, and user perceptions. Greenhouse-Geisser correction was used when data failed the test for sphericity.

5.1 Typing Speed

The means of WPM are shown in Figure 7. In the sighted condition, the main effects of keyboard ($F(2, 24) = 188.77, p < 0.001$) and session ($F(2.98, 35.76) = 39.44, p < 0.001$) were significant, but the main effect of phone ($F(1, 12) = 2.32, p > 0.05$) was not. Using QWERTY (mean = 20.58, SD = 3.94) for phrase entry was significantly faster than using ThumbStroke (mean = 8.01, SD = 1.32, $p < 0.001$) and Escape (mean = 9.18, SD = 2.34, $p < 0.001$). There was no significant difference in typing speed between ThumbStroke and Escape ($p > 0.05$).

In the sight-free condition, the main effects of keyboard ($F(1.38, 16.59) = 7.07, p < 0.01$), session ($F(3.18, 38.15) = 40.90, p < 0.001$), and phone ($F(1, 12) = 7.48, p < 0.05$) were significant. Using ThumbStroke (mean = 9.43, SD = 2.18) and Escape (mean = 9.10, SD = 2.53) for phrase entry was significantly faster than QWERTY (mean = 7.52, SD = 2.89, $p < 0.05$), but there was no significant difference between ThumbStroke and Escape ($p > 0.05$). The overall

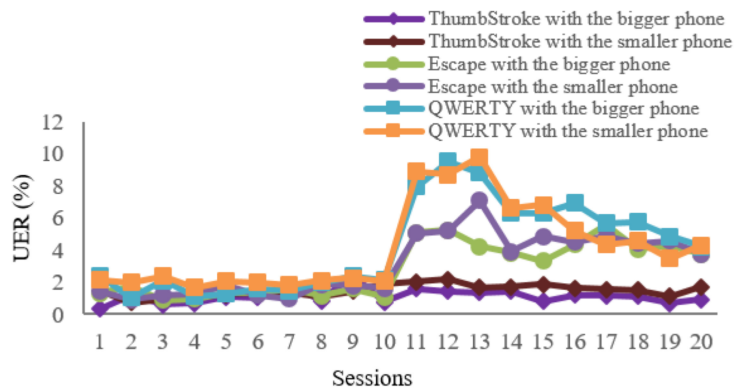


Fig. 8. Means of UER of three keyboards.

WPM of the bigger phone (mean = 8.83, SD = 2.69) was significantly larger than that of the smaller phone (mean = 8.52, SD = 2.67, $p < 0.05$).

In session 11, the average means of WPM with ThumbStroke, Escape, and QWERTY with the bigger phone were 7.83 (SD = 2.04), 6.32 (SD = 1.57), and 4.72 (SD = 1.66), respectively, and those with the smaller phone were 7.23 (SD = 2.19), 5.53 (SD = 1.09), and 5.02 (SD = 1.97). In the last sight-free session, the average means of WPM with ThumbStroke, Escape, and QWERTY were 10.50 (SD = 1.30), 11.45 (SD = 1.48), and 9.15 (SD = 3.13) with the bigger phone, and 10.76 (SD = 1.65), 10.57 (SD = 1.81), and 9.13 (SD = 2.66) with the smaller phone, respectively. ThumbStroke and Escape led to significantly faster entry than QWERTY ($p < 0.05$) in both the first and last sessions. The participants also achieved faster speed with the bigger phone than with the smaller one ($p < 0.05$) in the last session.

5.2 Error Rate

5.2.1 Uncorrected Error Rate. The means of UER are presented in Figure 8. In the sighted condition, the main effects of keyboard ($F(2, 24) = 4.57$, $p < 0.05$) and phone ($F(1, 12) = 8.30$, $p < 0.05$) were significant, while the main effect of session was not ($F(3.02, 36.23) = 1.09$, $p > 0.05$). ThumbStroke (mean = 1.16, SD = 1.17) had significantly lower UER than QWERTY (mean = 1.89, SD = 1.65, $p < 0.05$). There was no significant difference between ThumbStroke and Escape (mean = 1.35, SD = 1.53) or between QWERTY and Escape ($p > 0.05$). The UER while typing with the bigger phone was significantly lower than that of the smaller phone ($p < 0.05$).

In the sight-free condition, the main effects of keyboard ($F(2, 24) = 7.70$, $p < 0.05$) and session ($F(1.77, 21.18) = 4.88$, $p < 0.05$) on UER were significant. The main effect of phone was not ($F(1, 12) = 0.72$, $p > 0.05$). ThumbStroke (mean = 1.43, SD = 1.32) achieved significantly lower UER than Escape (mean = 4.60, SD = 5.01, $p < 0.05$) and QWERTY (mean = 6.43, SD = 6.41, $p < 0.05$). No significant difference existed between QWERTY and Escape ($p > 0.05$). In session 11, ThumbStroke had significantly lower UER than QWERTY and Escape ($p < 0.01$). In session 20, ThumbStroke achieved significantly lower UER than QWERTY ($p < 0.01$). There was no significant difference between QWERTY and Escape, between ThumbStroke and Escape, and between two phones ($p > 0.05$).

5.2.2 Corrected Error Rate. The means of CER are presented in Figure 9. In the sighted condition, the main effects of keyboard ($F(1.13, 13.55) = 9.15$, $p < 0.01$) and session ($F(2.93, 35.13) = 3.71$, $p < 0.05$) were significant, but the main effect of phone was not ($F(1, 12) = 7.21$, $p > 0.05$).

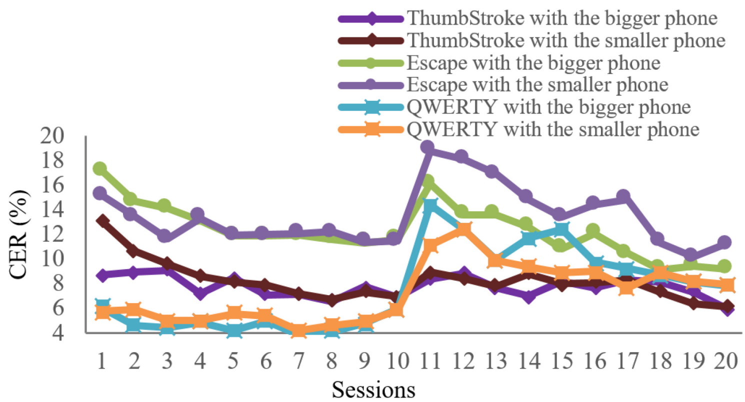


Fig. 9. Means of CER of the three keyboards.

Overall, ThumbStroke (mean = 8.20, SD = 3.67) and QWERTY (mean = 4.99, SD = 2.68) had significantly lower CER than Escape (mean = 12.70, SD = 8.81, $p < 0.05$). There was no significant difference between ThumbStroke and QWERTY ($p > 0.05$). In session 1, QWERTY had significantly lower CER than ThumbStroke and Escape, and there was no significant difference between the latter two keyboards. In session 10, ThumbStroke and QWERTY had significantly lower CER than Escape ($p < 0.05$). There was no significant difference between ThumbStroke and QWERTY ($p > 0.05$).

In the sight-free condition, the main effects of keyboard ($F(2, 22) = 3.69$, $p < 0.05$) and session ($F(2.58, 28.35) = 7.97$) were significant ($p < 0.01$), but the main effect of phone was not ($F(1, 11) = 4.67$, $p > 0.05$). Overall, with ThumbStroke (mean = 7.78, SD = 3.92), the participants achieved significantly lower CER than with Escape (mean = 13.05, SD = 4.94, $p < 0.05$). No significant difference was found between ThumbStroke and QWERTY (mean = 9.90, SD = 10.27) and between Escape and QWERTY ($p > 0.05$). In sessions 11 and 20, the CER of ThumbStroke was significantly lower than that of Escape ($p < 0.001$). There was no significant difference between Escape and QWERTY, between ThumbStroke and QWERTY, and between two phones ($p > 0.05$).

5.3 User Perceptions

The Cronbach alphas for PEOU and perceived effectiveness constructs were 0.92 and 0.96, respectively. The means of user perception factors (ranging from 1 to 7, with 1 = Totally Disagree, 4 = Neutral, and 7 = Totally Agree) are presented in Table 3. They are the overall perceptions of the three keyboards regardless of phone size.

Repeated measures ANOVA results of user perceptions are reported in Table 4. In the sighted condition, QWERTY resulted in significantly higher levels of PEOU and perceived effectiveness than Escape ($p < 0.05$), but there was no significant difference between QWERTY and ThumbStroke or between Escape and ThumbStroke ($p > 0.05$). For Overall Satisfaction, the mean of QWERTY was significantly higher than that of Escape ($p < 0.01$). The mean of ThumbStroke was also significantly higher than that of Escape ($p < 0.05$). There was no significant difference in overall satisfaction between QWERTY and ThumbStroke ($p > 0.05$).

In the sight-free condition, overall, ThumbStroke resulted in significantly higher PEOU than Escape ($p < 0.05$) and QWERTY ($p < 0.001$). Escape also had higher PEOU than QWERTY ($p < 0.05$). In session 11, ThumbStroke had higher PEOU than QWERTY ($p < 0.001$) and Escape ($p < 0.01$), and Escape had higher PEOU than QWERTY ($p < 0.01$). In session 20, ThumbStroke still

Table 3. User Perception Factors of Three Keyboards

Factors	Session	ThumbStroke Mean (SD)	Escape Mean (SD)	QWERTY Mean (SD)
PEOU	1	5.25 (0.74)	4.83 (0.96)	6.08 (1.46)
	10	5.98 (1.06)	5.27 (1.35)	6.04 (0.55)
	11	6.00 (0.75)	4.62 (1.37)	3.08 (1.37)
	20	6.29 (0.74)	5.38 (1.09)	4.48 (1.70)
Perceived Effectiveness	1	5.04 (1.18)	4.54 (1.15)	6.13 (1.58)
	10	5.73 (1.10)	5.15 (1.37)	6.02 (0.65)
	11	5.92 (0.98)	4.37 (1.39)	2.75 (1.38)
	20	6.27 (0.98)	5.35 (1.24)	4.19 (1.60)
Overall Satisfaction	1	5.38 (1.39)	4.62 (1.39)	6.08 (1.61)
	10	6.00 (1.15)	5.15 (1.21)	6.08 (0.64)
	11	6.00 (0.82)	4.31 (1.49)	2.85 (1.52)
	20	6.31 (0.95)	5.38 (1.33)	4.0 (1.53)

Table 4. Repeated Measures ANOVA Results of User Perceptions

		Main Effect		Keyboard * Session
Factors	Session	Keyboard F(2, 24)	Session F(1, 12)	F(2, 24)
PEOU	1, 10	6.28*	2.30	1.70
	11, 20	24.59***	7.50*	3.78
Perceived Effectiveness	1, 10	8.22*	1.92	1.57
	11, 20	23.25***	6.74*	3.92*
Overall Satisfaction	1, 10	7.20*	1.55	0.74
	11, 20	23.13***	6.01*	2.47

Note: * 0.05 significance level; *** 0.001 significance level.

had higher PEOU than QWERTY ($p < 0.01$) and Escape ($p < 0.05$), and Escape had higher PEOU than QWERTY ($p < 0.05$).

In the sight-free condition, ThumbStroke received significantly higher perceived effectiveness than Escape ($p < 0.05$) and QWERTY ($p < 0.001$). Perceived effectiveness of Escape was also higher than that of QWERTY ($p < 0.05$). In session 11, ThumbStroke had higher perceived effectiveness than QWERTY ($p < 0.001$) and Escape ($p < 0.01$), and Escape was higher than QWERTY ($p < 0.01$). In session 20, ThumbStroke led to higher perceived effectiveness than QWERTY ($p < 0.01$) but not Escape ($p > 0.05$), and Escape had higher perceived effectiveness than QWERTY ($p < 0.05$).

ThumbStroke also received significantly higher overall satisfaction than Escape ($p < 0.05$) and QWERTY ($p < 0.001$) in the sight-free condition. Overall satisfaction with Escape was also significantly higher than that with QWERTY ($p < 0.05$). In session 11, ThumbStroke led to higher overall satisfaction than QWERTY ($p < 0.001$) and Escape ($p < 0.01$), and Escape was higher than QWERTY ($p < 0.01$). In session 20, ThumbStroke resulted in higher overall satisfaction than QWERTY ($p < 0.001$) but not Escape, and Escape had higher perceived effectiveness than QWERTY ($p < 0.01$).

6 DISCUSSION

In this research, we designed, developed, and empirically evaluated a thumb-stroke-based virtual keyboard called ThumbStroke for one-handed and sight-free text input on mobile devices. It is aimed to address the limited thumb reachability, visual occlusion, and low accuracy problems of one-handed text entry and to support sight-free text entry on mobile devices. We are not intending to suggest that ThumbStroke should replace QWERTY or other input methods, such as voice input, under all circumstances. They can certainly coexist as different options for users. The major characteristics of ThumbStroke lie in the following aspects:

- ThumbStroke enables users to hold and interact with a touchscreen mobile phone with one hand only.
- It does not require precise key tapping or pressing, thus supporting sight-free text entry.
- Text input with ThumbStroke relies on thumb stroke directions on a device screen rather than physical press on specific keys. Therefore, the thumb reachability problem is eliminated.
- The location of ThumbStroke on a device screen is flexible. The size of ThumbStroke can also be adjusted by users. It will not block content as traditional keyboards do and therefore will avoid the visual occlusion problem.
- Different from menu-based keyboards, which often require users to select an area on one keypad and then a character separately on another keypad, ThumbStroke combines area and character selection within one continuous stroke on a single keypad, which can be less confusing for sight-free text entry because users do not need to remember which keypad they are interacting with.
- We did not use tapping in ThumbStroke to avoid entering some letters by mistake whenever users touch the screen. This is quite important for sight-free text entry. In contrast, Siwpeboard (Chen et al. 2014) uses tapping to select both an area and a letter in the area. Users could accidentally enter characters whenever they touch the screen. This is another major design advantage of ThumbStroke compared to other existing keyboards such as Escape (Banovic et al. 2013) and Swipeboard (Chen et al. 2014).

6.1 Major Findings

6.1.1 Sighted Condition. The participants achieved significantly better performance in WPM and perceived ease of use using QWERTY than using the other two keyboards, while ThumbStroke achieved lower UER than QWERTY and lower CER than Escape. We believe that the participants' high familiarity with QWERTY is the major reason that the participants achieved the highest WPM and perceived ease of use with it. To make the QWERTY keyboard accessible to the participants in one-handed interaction, we had to shift it to the right-hand side of the screen. As a result, the problem of limited thumb reachability associated with QWERTY was intentionally addressed. Otherwise, all of the measures with QWERTY should be significantly worse than what we observed in this experiment. In addition, direct pointing to keys in the sighted condition is faster than making strokes, which is in line with the findings in Lai and Zhang (2015).

ThumbStroke yielded significantly lower UER than QWERTY because QWERTY always requires accurate key press, which can cause more erroneous entries than a stroke-based text entry method. The bigger phone led to lower UER than the smaller phone, likely because on the smaller phone, QWERTY had even smaller keys than on the bigger phone, which made text entry even more difficult.

6.1.2 Sight-Free Condition. ThumbStroke outperformed QWERTY in WPM and UER and achieved better results than Escape in UER and CER. ThumbStroke also received the best user perceptions. The typing speed of QWERTY dropped dramatically and was significantly lower than those of ThumbStroke and Escape. It could be because QWERTY requires a user to accurately press on keys, while the other two do not. ThumbStroke and Escape achieved similar levels of typing speed. Typing with the bigger phone was also faster than with the smaller phone because the size of keys was bigger, especially when the problem of limited thumb reachability was alleviated by shifting the keyboard to one side.

The participants using QWERTY performed better than using Escape in terms of CER, but worse in UER. It could be because the participants were less certain about the location of an aimed character with QWERTY, and thus tended to hear the character to make sure the correct key was pressed before releasing the thumb from the screen to enter it. Moreover, to correct an error while using QWERTY, the participants would need to move their thumb to switch to the "Backspace" key, which could be more challenging than swiping toward a certain direction with Escape. As a result, when the participants made a mistake with QWERTY, they might be less willing to fix it. The "Back," "Home," and "Menu" buttons of the smaller phone are located along the bottom border of the screen. Based on our observation during the experiment, those buttons were likely to be clicked by accident in the sight-free condition, especially while using Escape, which had two segments right above the bottom border of the screen. It could also be the reason there was an interaction effect between keyboard and phone size for Escape as shown in Figure 9.

As shown in Figures 8 and 9, compared to Escape and QWERTY, ThumbStroke has fewer steep lines from sessions 11 to 20, indicating that the learning curve of ThumbStroke is not as steep as those of the other two keyboards in the sight-free condition. Furthermore, QWERTY was the worst among the three keyboards in terms of PEOU, perceived effectiveness, and overall satisfaction for sight-free text entry, while ThumbStroke was the best for all user perceptions.

Overall, ThumbStroke significantly outperformed QWERTY in text entry speed (i.e., WPM), accuracy (i.e., CER), perceived ease of use, perceived effectiveness, and overall satisfaction in the sight-free condition.

6.2 Research Contributions and Practical Implications

This research and its findings provide multiple insights for mobile interface and keyboard designers. First, directional-stroke-based character selection shows benefits for sight-free text entry in both typing speed and accuracy compared to the traditional pressing-based text entry method like QWERTY. After some practice, the participants were able to move their thumb toward any of those eight directions without looking at the screen with high accuracy. This finding implies that from a thumb-stroke-based keyboard design perspective, dividing directions of thumb strokes into eight directions, each with a 45-degree "zone," is sufficient and effective for distinguishing different stroke directions to identify key selections. It shows that such a design, which fundamentally addresses several common problems with QWERTY, such as thumb accessibility, fat finger, and visual occlusion, is practically feasible and can be mastered quickly. This finding can be applied to thumb-stroke-based user interactions at large and provides conceptual guidelines for improving the accessibility features of smartphones that can support visually impaired users better. Currently, users with severe visual impairments rely on screen reader software, which usually only differentiates four finger movement directions, including up, down, left, and right. Based on our results, the other four directions, that is, upper-left, lower-left, upper-right, and lower right, have the potential to be effective interaction gestures and to give more flexibility and options to interact with touchscreen devices. In addition, current gestures with angles used on mobile phones, such as "up then down" and "up then left," did not consider those four directions either. Our study showed that the participants were able to change the direction accurately while sliding a thumb on the screen. Moreover, the gestures with turns can be used with other techniques such as the back patterns used by BackMirror (Wong et al. 2016) and Bezel Cursor (Li et al. 2016) to further increase the gesture vocabulary for one-handed mobile interaction.

Second, directional-stroke-based location-independent text entry can improve the accuracy of sight-free text entry, but strokes may slow down text entry speed in the sighted condition compared to direct key pressing. According to the Fitts's law (Fitts 1954), bringing a target closer could make it easier and faster to move to and select a target. An indirect input method can bypass the impact of the target's distance on interaction speed. However, interaction behavior that is more complicated than pointing, such as making strokes in this study, can compromise the benefit of indirect input in terms of speed in the sighted condition. Nevertheless, the benefit of indirect input was prominent in the sight-free condition, because the participants did not have to locate and reach the target accurately, which could be very challenging to do without looking at the screen. ThumbStroke has the potential to be used by and benefit visually impaired users, which will be assessed in a future study.

Third, unlike traditional keyboards such as QWERTY, ThumbStroke is based on stroke directions. Thus, it is independent of key size. This feature may be useful for devices with small screen sizes (e.g., smartwatches).

Fourth, this research also provides benefits to general users of touchscreen mobile devices. We observed that the participants' interaction patterns (e.g., location, length, angle of strokes, etc.) were very different even when they entered the same content. The unique interaction patterns of individual users can be used as physical biometrics for continuous mobile user authentication, which warrants future studies.

Another research contribution is that we evaluated ThumbStroke on touchscreen mobile phones in both sighted and sight-free conditions. The findings demonstrate that using a directional-stroke-based keyboard like ThumbStroke can lead to better performance and user perceptions for sight-free text entry.

This research also sheds some light for the design of mobile apps involving one-handed and/or sight-free interaction: (1) The freedom of interaction anywhere on the screen can benefit one-handed interaction because users do not have to reach remote objects. It also can avoid awkward areas that are hard to interact with; (2) Users are able to make distinctive strokes in eight different directions and then turn to four directions in the sight-free condition. Directional strokes with and without turns could be generalized for sight-free interaction on mobile phones; (3) Selecting objects with directional strokes is useful for sight-free interaction. However, it may not be efficient in the sighted condition compared to direct touch; (4) Indirect input methods, which are target location independent, are useful for sight-free interaction; and (5) Directional strokes with a turn can be used to select an area and a character in the area with only one gesture. This design may also apply to menu selections.

6.3 Limitations

There are several limitations of this study that provide future research opportunities. First, to simulate situational impairments, we asked the participants to walk on a treadmill when completing the experimental tasks. We did not test ThumbStroke in other motor conditions. Second, the character arrangement on ThumbStroke does not map to that on a regular QWERTY keyboard, with which most users are familiar. Currently, we used alphabetical order, which was reported by some participants to be beneficial for them to remember the characters' locations. Other arrangements, such as those based on character usage frequency in English words (Banovic et al. 2013), will be examined in the future.

ACKNOWLEDGMENTS

We want to thank Dr. Nikola Banovic and Dr. Jacob O. Wobbrock for kindly providing us with the source code of Escape and StreamAnalyzer.

REFERENCES

B. Adipat, D. Zhang, and L. Zhou. 2011. The effects of tree-view based presentation adaptation on mobile web browsing. *MIS Quarterly* 35 (2011), 1.

N. Banovic, K. Yatani, and K. N. Truong. 2013. Escape-keyboard: A sight-free one-handed text entry method for mobile touch-screen devices. *International Journal of Mobile Human Computer Interaction (IJMHCI)* 5, 3 (2013), 42–61.

J. Bergstrom-Lehtovirta, A. Oulasvirta, and S. Brewster. 2011. The effects of walking speed on target acquisition on a touchscreen interface. In *13th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 143–146.

R. Blanch, Y. Guiard, and M. Beaudouin-Lafon. 2004. Semantic pointing: Improving target acquisition with control-display ratio adaptation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 519–526.

M. N. Bonner, J. T. Brudvik, G. D. Abowd, and W. K. Edwards. 2010. No-look notes: Accessible eyes-free multi-touch text entry. In *International Conference on Pervasive Computing*. 409–426.

A. Bragdon, E. Nelson, Y. Li, and K. Hinckley. 2011. Experimental analysis of touch-screen gesture designs in mobile environments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 403–412.

X. A. Chen, T. Grossman, and G. Fitzmaurice. 2014. Swipeboard: A text entry technique for ultra-small interfaces that supports novice to expert transitions. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. 615–620.

G. Costagliola, V. Fuccella, and M. Di Capua. 2011. Text entry with keyscretch. In *Proceedings of the 16th International Conference on Intelligent User Interfaces*. 277–286.

G. Costagliola, V. Fuccella, and M. Di Capua. 2013. Interpretation of strokes in radial menus: The case of the keyscretch text entry method. *Journal of Visual Languages & Computing* 24, 4 (2013), 234–247.

A. R. Façanha, W. Viana, M. C. Pequeno, M. de Borba Campos, and J. Sánchez. 2014. Touchscreen mobile phones virtual keyboarding for people with visual disabilities. In *International Conference on Human-Computer Interaction*. 134–145.

P. Fitts. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*. 47, 6 (1954), 381–391.

Y. Fukatsu, B. Shizuki, and J. Tanaka. 2013. No-look flick: Single-handed and eyes-free japanese text input system on touch screens of mobile devices. In *Proceedings of the 15th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 161–170.

J. Gold, J. Driban, N. Thomas, T. Chakravarty, V. Channell, and E.-G. Komaroff. 2012. Postures, typing strategies, and gender differences in mobile device usage: An observational study. *Applied Ergonomics*. 43, 2 (2012), 408–412.

D. A. Grant. 1948. The Latin square principle in the design and analysis of psychological experiments. *Psychological Bulletin* 45, 5 (1948), 427.

B. Han and K. Kim. 2015. Typing performance evaluation with multimodal soft keyboard completely integrated in commercial mobile devices. *Journal on Multimodal User Interfaces* 9, 3 (2015), 173–181.

A. Hevner, S. T. March, J. Park, and S. Ram. 2004. Design science in information systems research. *MIS Quarterly* 28, 1 (2004), 75–105.

H. Hoehle and V. Venkatesh. 2015. Mobile application usability: Conceptualization and instrument development. *MIS Quarterly* 39 (2015), 2.

A. Kano, J. C. Read, and A. Dix. 2006. Children's phrase set for text input method evaluations. In *Proceedings of the 4th Nordic Conference on Human-Computer Interaction: Changing Roles*. 449–452.

A. Karlson and B. Bederson. 2007. Thumbspace: Generalized one-handed input for touchscreen-based mobile devices. In *Human-Computer Interaction (INTERACT'07)*. 324–338.

H. Korhonen, J. Holm, and M. Heikkinen. 2007. Utilizing sound effects in mobile user interface design. *Human-Computer Interaction (INTERACT'07)*. 283–296.

G. Kurtenbach and W. Buxton. 1994. User learning and performance with marking menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 258–264.

J. Lai and D. Zhang. 2014. A study of direction's impact on single-handed thumb interaction with touch-screen mobile phones. In *CHI'04 Extended Abstracts on Human Factors in Computing Systems*. 2311–2316.

J. Lai and D. Zhang. 2015. Extendedthumb: A target acquisition approach for one-handed interaction with touch-screen mobile phones. *IEEE Transactions on Human-Machine Systems* 45, 3 (2015), 362–370.

E. M. Lamberg and L. M. Muratori. 2012. Cell phones change the way we walk. *Gait & Posture* 35, 4 (2012), 688–690.

J. R. Lewis. 1995. IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use. *International Journal of Human-Computer Interaction* 7, 1 (1995), 57–78.

W. H. A. Li, H. Fu, and K. Zhu. 2016. Bezelcursor: Bezel-initiated cursor for one-handed target acquisition on mobile touch screens. *International Journal of Mobile Human Computer Interaction (IJMHCI)* 8, 1 (2016), 1–22.

K. Lyons, T. Starner, and B. Gane. 2006. Experimental evaluations of the twiddler one-handed chording mobile keyboard. *Human-Computer Interaction* 21, 4 (2006), 343–392.

I. S. Mackenzie and T. Felzer. 2010. Sak: Scanning ambiguous keyboard for efficient one-key text entry. *ACM Transactions on Computer-Human Interaction (TOCHI)* 17, 3 (2010), 11.

I. S. MacKenzie and R. W. Soukoreff. 2003. Phrase sets for evaluating text entry techniques. In *CHI'03 Extended Abstracts on Human Factors in Computing Systems*. 754–755.

E. Mattheiss, G. Regal, J. Schrammel, M. Garschall, and M. Tscheligi. 2014. Dots and letters: Accessible braille-based text input for visually impaired people on mobile touchscreen devices. In *International Conference on Computers for Handicapped Persons*. 650–657.

C. Middleton, R. Scheepers, and V. K. Tuunainen. 2014. When mobile is the norm: Researching mobile information systems and mobility as post-adoption phenomena. *European Journal of Information Systems* 23, 5 (2014), 503–512.

B. Millet. 2009. Design and evaluation of three alternative keyboard layouts for a five-key text entry technique. *Open Access Dissertations*. 513. Retrieved from https://scholarlyrepository.miami.edu/oa_dissertations/513.

National Accident Helpline. 2016. Everyday Tech Dangers Driving Brits to Distraction. Retrieved from https://www.national-accident-helpline.co.uk/accident-claims/accident-awareness-week.

H. Nicolau, K. Montague, T. Guerreiro, A. Rodrigues, and V. L. Hanson. 2015. Holibraille: Multipoint vibrotactile feedback on mobile devices. In *Proceedings of the 12th Web for All Conference*. 30.

F. Niederman and S. T. March. 2012. Design science and the accumulation of knowledge in the information systems discipline. *ACM Transactions on Management Information Systems (TMIS)* 3, 1 (2012), 1.

J. F. Nunamaker and R. O. Briggs. 2011. Toward a broader vision for information systems. *ACM Transactions on Management Information Systems (TMIS)* 2, 4 (2011), 20.

T. Page. 2013. Usability of text input interfaces in smartphones. *Journal of Design Research* 11, 1 (2013), 39–56.

K. Pfeuffer, K. Hinckley, M. Pahud, and B. Buxton. 2017. Thumb+ pen interaction on tablets. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 3254–3266.

A. Roudaut, S. Huot, and E. Lecolinet. 2008. Taptap and magstick: Improving one-handed target acquisition on small touch-screens. In *Proceedings of the Working Conference on Advanced Visual Interfaces*. 146–153.

S. M. Schabrun, W. van den Hoorn, A. Moorcroft, C. Greenland, and P. W. Hodges. 2014. Texting and walking: Strategies for postural control and implications for safety. *PLoS One* 9, 1 (2014), e84312.

J.-B. Scheibel, C. Pierson, B. Martin, N. Godard, V. Fuccella, and P. Isokoski. 2013. Virtual stick in caret positioning on touch screens. In *Proceedings of the 25 Ième Conférence Francophone on l'Interaction Homme-Machine*. 107.

T. Siang. 2019. Using mobile apps—the one thumb, one eyeball test for good mobile design. Retrieved from https://www.interaction-design.org/literature/article/using-mobile-apps-the-one-thumb-one-eyeball-test-for-good-mobile-design.

R. W. Soukoreff and I. S. MacKenzie. 2003. Metrics for text entry research: An evaluation of MSD and KSPC, and a new unified error metric. In *SIGCHI Conference on Human Factors in Computing Systems*. 113–120.

C. Southern, J. Clawson, B. Frey, G. Abowd, and M. Romero. 2012. An evaluation of brailletouch: Mobile touchscreen text entry for the visually impaired. In *14th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 317–326.

P. J. Steinbart, M. J. Keith, and J. Babb. 2016. Examining the continuance of secure behavior: A longitudinal field study of mobile device authentication. *Information Systems Research* 27, 2 (2016), 219–239.

N. S. Subash, S. Nambiar, and V. Kumar. 2012. Braillekey: An alternative Braille text input system: Comparative study of an innovative simplified text input system for the visually impaired. In *2012 4th International Conference on Intelligent Human Computer Interaction (IHCI'12)*. 1–4.

S. Tatro and O. Fleming. 2015. Man, distracted by electronic device, identified after falling to death at sunset cliffs. Retrieved from https://www.nbcsandiego.com/news/local/Man-Dies-After-Falling-Off-Cliff-at-Sunset-Cliffs-Lifeguards-363534491.html.

J. O. Wobbrock and B. A. Myers. 2006. Analyzing the input stream for character-level errors in unconstrained text entry evaluations. *ACM Transactions on Computer-Human Interaction (TOCHI)* 13, 4 (2006), 458–489.

J. O. Wobbrock, B. A. Myers, and J. A. Kembel. 2003. Edgewrite: A stylus-based text entry method designed for high accuracy and stability of motion. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology*. 61–70.

P. C. Wong, H. Fu, and K. Zhu. 2016. Back-mirror: Back-of-device one-handed interaction on smartphones. In *SIGGRAPH ASIA 2016 Mobile Graphics and Interactive Applications*. 10.

P. C. Wong, K. Zhu, and H. Fu. 2018. Fingert9: Leveraging thumb-to-finger interaction for same-side-hand text entry on smartwatches. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 178.

B. Yi, X. Cao, M. Fjeld, and S. Zhao. 2012. Exploring user motivations for eyes-free interaction on mobile devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2789–2792.

S. Zhai, P. O. Kristensson, P. Gong, M. Greiner, S. A. Peng, L. M. Liu, and A. Dunnigan. 2009. Shapewriter on the iPhone: From the laboratory to the real world. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*. 2667–2670.

Received January 2019; revised May 2019; accepted July 2019