

A Thumb Stroke-Based Virtual Keyboard for Sight-Free Text Entry on Touch-Screen Mobile Phones

Jianwei Lai
School of Information Technology
Illinois State University
Normal, IL 61790, USA
jlai12@ilstu.edu

Dongsong Zhang, Sen Wang,
Isil Yakut Kilic, Lina Zhou
Department of Information Systems
University of Maryland, Baltimore County
Baltimore, MD 21250, USA
zhangd, senwang1, yakut1, zhou@umbc.edu

Abstract

The use of QWERTY on most of the current mobile devices for text entry usually requires users' full visual attention and both hands, which is not always possible due to situational or physical impairments of users. Prior research has shown that users prefer to hold and interact with a mobile device with a single hand when possible, which is challenging and poorly supported by current mobile devices. We propose a novel thumb-stroke based keyboard called ThumbStroke, which can support both sight-free and one-handed text entry on touch-screen mobile devices. Selecting a character for text entry via ThumbStroke completely relies on the directions of thumb movements at anywhere on a device screen. We evaluated ThumbStroke through a longitudinal lab experiment including 20 sessions with 13 participants. ThumbStroke shows advantages in typing accuracy and user perceptions in comparison to Escape and QWERTY and results in faster typing speed than QWERTY for sight-free text entry.

1. Introduction

Many current mobile interfaces are designed based on GUIs of desktop computers for users with full visual attention on screens [1]. However, mobile phone users are often in motion, such as walking, in which they cannot devote all of their visual attention to mobile phones [2]. Therefore, sight-free interaction techniques that enable users to interact with mobile devices, such as entering text, without visual attention can be very beneficial. For example, when a user walks on a busy street and needs to send a text message, it would be safer if she does not need to always look at the screen while typing. Also, such sight-free interaction can improve the accessibility of mobile devices, especially for users with

visual impairments. According to WHO, 285 million people are estimated to be visually impaired worldwide - 39 million are blind and 246 million have low vision. The National Federation of the Blind reported that over 7.35 million people in the United States had a visual disability in 2014.

Another desired support for mobile devices is the support of one-handed interaction, which allows users to hold and interact with a mobile device using one hand so as to free another hand for other activities. One-handed interaction is especially beneficial for users with arm or hand disabilities or in situational impairments. Situational impairments refer to users' temporary difficulty in accessing mobile devices due to specific context or situations that they are in [3]. For example, when a user is holding a cup of coffee, 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), in the United States, there are approximately 2 million people living with limb loss, and more than 500 Americans lose a limb every day. It is necessary to develop effective mechanisms for one-handed interaction with mobile devices for people with hand/upper limb loss or disabilities [4]. Ideally, mobile interaction should just require one hand [5]. There has been increasing research on one-handed interaction with mobile devices in the past decade. Prior studies suggested that users would prefer using mobile devices with a single hand when possible [6]. However, one-handed interaction introduces usability problems [7]. Mobile phones have tiny buttons and crowded keypads, which are difficult to select and press accurately with a finger, let alone a fat one [8]. There are also areas on the touch screen of a mobile device that are difficult to reach by a thumb in one-handed interaction [4].

Texting is an essential function of mobile communication and connectivity. The standard QWERTY keyboard is available on the majority of mobile devices, but its size is ill-suited to the mobile

paradigm [9]. Although typing with one hand is common [10], it is not always easy with QWERTY because the user needs to secure a device with her palm and four fingers while reaching and pressing keys with the thumb, which has limited flexion and extension. Some of the keys can be difficult to reach by a thumb [4].

To address the above challenges, in this research, we propose and empirically evaluate a stroke-based keyboard called ThumbStroke for effective sight-free and one-handed text entry on touch-screen mobile devices. This keyboard has several distinct features: 1) instead of tapping on specific keys on a traditional keyboard accurately to enter characters, users using ThumbStroke can select and enter characters by making strokes toward certain directions at any place on a device screen, which solves the problem of limited thumb accessibility; 2) users do not need to reach for keys, and the keyboard position on the device screen is not constrained by the mobility of a thumb; 3) when users press on keys on a traditional keyboard, their thumb will cover the content underneath, causing the visual occlusion problem [11]. ThumbStroke avoids this problem completely by not requiring physical selection of specific keys during text entry, eliminating the negative effects of small key size on text entry; and 4) ThumbStroke does not require visual attention on keypads, thus supporting sight-free text entry. The results of an empirical evaluation through a longitudinal controlled laboratory experiment demonstrate that ThumbStroke is effective and well perceived by the participants.

The rest of the paper will be organized as follows. We will first introduce the literature on existing methods for text entry on mobile devices. Then, we will present the design of ThumbStroke, followed by the description of our empirical evaluation methodology. Next, we will present results. Finally, the paper will conclude with discussions and future research.

2. Related work

2.1. Sight-free text entry

Sight-free text entry is not only needed by people with visual impairments but also desired by sighted users with or even without situational impairments. There have been some commercial tools developed for assisting visually impaired users with interacting with mobile devices, such as VoiceOver on iPhones and TalkBack on Android phones. They read out the letter when users press a key during text entry. Braille-based techniques, such as BrailleTouch [12], TypeInBraille [13], BrailleType [14], and BrailleKey [15], are also

designed for visually impaired users who are familiar with Braille, but not suitable for well-sighted people.

Some specially designed keyboards can be useful for people with visual impairments or situational impairments. For example, the Escape keyboard [16] enables a user to enter letters by pressing the thumb on designated areas on a device screen to enter letters in the center of the areas, or by flicking into different directions to enter characters surrounding the center. It requires a user to select an area first. No-Look Notes [17] divides a device screen into small segments and presents characters inside those segments. The user first needs to put a finger on a segment that contains the target character, then selects the segment by keeping one finger on it and tapping the screen with another. Selecting a segment takes the user to another screen on which the segment's characters are presented. Users then select the desired character by putting a finger in the area that contains it and tapping on the screen with a second finger. This two-step approach can be tedious and time-consuming. In addition, a user needs to put a finger on an intended segment correctly, which can be error-prone for sight-free text entry. The challenge for a graffiti-based keyboard [18] lies in the need of accurate recognition of users' handwriting input and the requirement of users to remember all the Graffiti characters. Speech-to-text recognition can be used for sight-free text entry. However, the voice input may be inconvenient, disturbing (e.g., in a library or on the street), and even cause privacy and security concerns (e.g., password entry).

2.2. One-handed text entry

Low thumb accessibility, visual occlusion, and low accuracy are common problems in one-handed interaction with touch-screen mobile phones [4, 5]. Some keys on a traditional soft keyboard for mobile devices such as QWERTY are difficult to reach due to limited thumb accessibility. When users tap on keyboard keys on a touch screen, the thumb will occlude the content underneath. In addition, the tiny keys of the QWERTY keyboard on mobile phones make key selection difficult and error-prone. Hence, the accuracy and speed of text entry with such a keyboard are severely affected in one-handed interaction. Stick [19] and Half-QWERTY [20] keyboards have multiple characters on each key, aiming to reduce the number of keys on a keyboard and increase key size. As a result, they may address the issue caused by tiny keys of QWERTY on mobile phones, but meanwhile, they introduce the selection ambiguity problem. Some keyboards, such as Google Keyboard 5.0, have a one-handed mode, in which the QWERTY keyboard is scaled down and moved to one side of the screen so that

it is easier for users to reach the keys in one-handed interaction. Nevertheless, they did not solve the issue of tiny keys.

2.3. Stroke-based keyboards

There also exist some stroke-based text entry methods. For example, instead of tapping on individual keys, a user using ShapeWriter [21] can enter a word by sliding a finger through all the letters in the word consecutively. The keyboard approximately traces all letters slid through, regardless of their locations, and analyzes them using a statistical model. The statistically most likely word will then be selected [21, 22]. This keyboard is quite efficient but very challenging for sight-free text entry. Also, reaching out to a specific key may not always be easy. Quickwriting [23] and Cirrin [24] are also stroke-based keyboards, but they do not support sight-free text entry well because they require users to find the locations of characters on a screen.

Despite prior studies on software keyboards, sight-free and/or one-handed text entry on touch-screen mobile handheld devices remains not well-supported by existing keyboards, which motivates this research.

3. Design of ThumbStroke

The objective of this research is to design, develop, and evaluate ThumbStroke, a novel thumb-stroke-direction based keyboard, to support sight-free and one-handed text entry on mobile devices while addressing the common problems of low thumb accessibility, visual occlusion, and error proneness simultaneously.

ThumbStroke is a virtual keyboard with a single round key, which is divided into eight small areas around its center, as shown in Figure 1. 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 (between two adjacent dotted lines shown in Figure 1). With the center of each small area as the starting area reference point, a character is either located in the center of a small area or in a certain direction range from the center. For example, as shown in Figure 1, 'E' is located in the center of the small area 3, and 'G' is located in the direction of 0-90° from the area center 'E'. According to [25], angle intervals of thumb moving directions (i.e., the angle between two adjacent but different directions) influence a thumb's movement speed and accuracy in one-handed interaction. It is suggested that the angle interval between any two adjacent areas or keys should be no less than 45°. Therefore, we followed this guideline in the design of ThumbStroke. A similar design has been used in marking menus [26, 27] for menu item selection, but not

for text entry on touch-screen devices. We organize the letters in ThumbStroke into small areas in a clock-wise manner generally based on their alphabetical order, aiming to make it easy for new users to learn and remember the positions of individual letters easily.

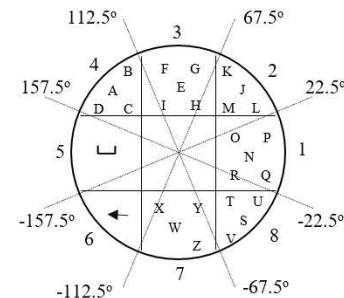
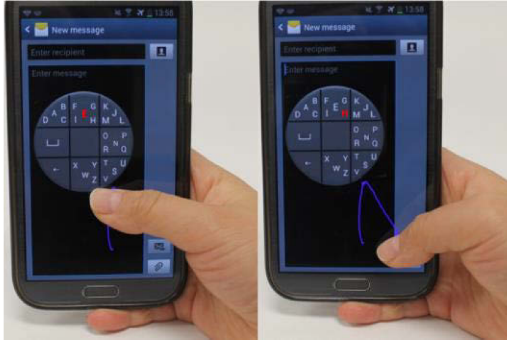


Figure 1. The design of ThumbStroke

The fundamental unique feature of ThumbStroke lies in that individual characters are selected for text entry completely based on one continuous thumb stroke at any location on the touch screen of a mobile device, not by physically pressing any keys on the keyboard.

Figure 2 illustrates how to enter text with ThumbStroke:

- When a user touches a text field, such as the address bar of an Internet browser, ThumbStroke will automatically appear on the screen. A long press in the center of the keypad enables users to move the keyboard to any location that they prefer. The center of the keyboard will be activated automatically as the starting reference point.
- A user moves her thumb on the screen in the direction toward an intended small area. The moving direction is calculated and one of the eight surrounding small areas in that direction will be identified and 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 2(a)). If the user lifts 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 towards the lower-right corner without lifting her thumb away from the screen, the letter 'H' will be activated (Figure 2(b)). The user now lifts her thumb away from the screen, the activated letter 'H' will be entered.
- After a letter is selected and entered, ThumbStroke automatically sets the center of the keyboard as the reference point.



(a) (b)

Figure 2. Text entry via ThumbStroke

To support sight-free text entry, a selected character will be read out just in the same way as Talkback does. The moving direction of the thumb is dynamically captured and calculated during the text entry process. 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 1) and a symbol/number keypad.

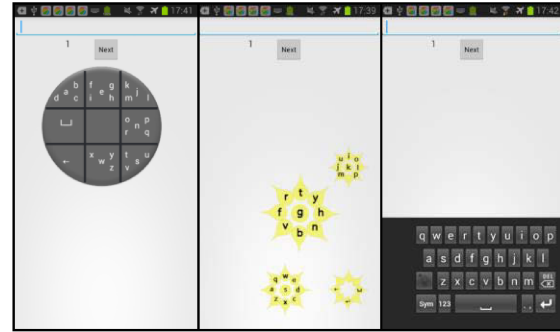
Furthermore, ThumbStroke provides a novel error correction feature for users. If a user mistakenly selects a wrong character, she can correct it by moving the thumb toward the right direction before lifting the thumb away from the screen. If a user selects a wrong area, she can cancel the selection by continuing moving the thumb in the previous direction after a pause.

4. Evaluation

We conducted a controlled laboratory experiment with a 3*2 (3 keyboards * 2 smartphones) within-subjects design to evaluate ThumbStroke, with Escape [16] and QWERTY used as baseline keyboards. With Escape (Figure 3(b)), users can enter the letter in the center of a flower by tapping on one of the areas. For the letters in the petals, users need to reach to the area and flip toward the corresponding directions. We selected Escape because it is also a stroke-based keyboard aiming to support both one-handed and sight-free text entry. One of the major differences between ThumbStroke and Escape is that the latter requires users to reach to a small area to select a character, but with ThumbStroke, users can make strokes anywhere on the screen. The QWERTY keyboard is the most commonly used keyboard on mobile phones for users. Hence, it was included as a baseline.

Screen size may influence users' one-handed interaction with mobile phones. A bigger screen of a mobile phone, which is popular nowadays, increases the

difficulty for users to grasp it with one hand, despite more space for interaction, and has more areas on its screen that are difficult to reach by a thumb. Hence, to evaluate the potential moderating effect of mobile device screen size on one-handed text entry, two smartphones with different screen size were used in this study.



(a) ThumbStroke (b) Escape (c) QWERTY

Figure 3. Keyboard layouts

4.1. Participants

The learning curve is a common phenomenon and challenge in learning a new keyboard. Usually, participants need to use a new keyboard for multiple sessions over a period of time [28], making it very difficult to have a large sample size. That is why many previous studies [16, 29-34] only included 6~12 participants. In this study, 13 participants (5 male, 8 female) at an east-coast university in the United States were recruited and participated in the evaluation. They were undergraduate and graduate students with a major in information systems. Among them, 5 were between 18 and 25 years old, 7 between 26 and 30 years old, and 1 was over 30 years old. They were all right-handed and had prior experience with touch-screen mobile phones. Each participant received \$200 for successfully completing the experiment.

4.2. Apparatus

The ThumbStroke, Escape, and QWERTY keyboards were implemented in Java using the Android SDK in Eclipse for user evaluation. They were installed on two touch-screen smartphones. One was a Samsung Galaxy Note 2 phone with a 5.5" HD Super AMOLED display. The other one was a Kyocera Event phone with a 3.5" capacitive touch screen. When a participant interacted with those phones, logs in the mobile phones recorded the time and pixel coordinates of the interactions.

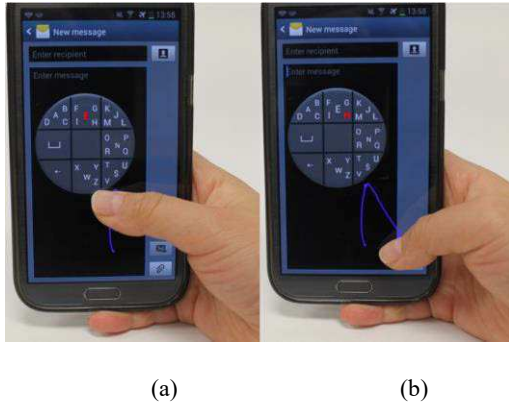


Figure 2. Text entry via ThumbStroke

To support sight-free text entry, a selected character will be read out just in the same way as Talkback does. The moving direction of the thumb is dynamically captured and calculated during the text entry process. 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 1) and a symbol/number keypad.

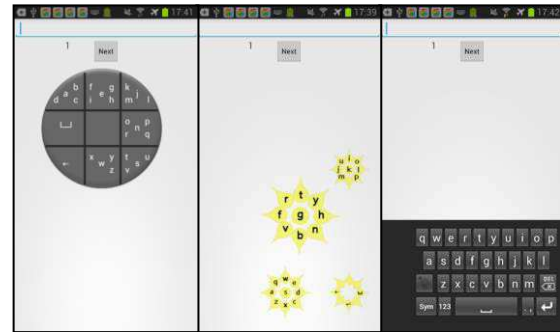
Furthermore, ThumbStroke provides a novel error correction feature for users. If a user mistakenly selects a wrong character, she can correct it by moving the thumb toward the right direction before lifting the thumb away from the screen. If a user selects a wrong area, she can cancel the selection by continuing moving the thumb in the previous direction after a pause.

4. Evaluation

We conducted a controlled laboratory experiment with a 3*2 (3 keyboards * 2 smartphones) within-subjects design to evaluate ThumbStroke, with Escape [16] and QWERTY used as baseline keyboards. With Escape (Figure 3(b)), users can enter the letter in the center of a flower by tapping on one of the areas. For the letters in the petals, users need to reach to the area and flip toward the corresponding directions. We selected Escape because it is also a stroke-based keyboard aiming to support both one-handed and sight-free text entry. One of the major differences between ThumbStroke and Escape is that the latter requires users to reach to a small area to select a character, but with ThumbStroke, users can make strokes anywhere on the screen. The QWERTY keyboard is the most commonly used keyboard on mobile phones for users. Hence, it was included as a baseline.

Screen size may influence users' one-handed interaction with mobile phones. A bigger screen of a mobile phone, which is popular nowadays, increases the

difficulty for users to grasp it with one hand, despite more space for interaction, and has more areas on its screen that are difficult to reach by a thumb. Hence, to evaluate the potential moderating effect of mobile device screen size on one-handed text entry, two smartphones with different screen size were used in this study.



(a) ThumbStroke (b) Escape (c) QWERTY

Figure 3. Keyboard layouts

4.1. Participants

The learning curve is a common phenomenon and challenge in learning a new keyboard. Usually, participants need to use a new keyboard for multiple sessions over a period of time [28], making it very difficult to have a large sample size. That is why many previous studies [16, 29-34] only included 6~12 participants. In this study, 13 participants (5 male, 8 female) at an east-coast university in the United States were recruited and participated in the evaluation. They were undergraduate and graduate students with a major in information systems. Among them, 5 were between 18 and 25 years old, 7 between 26 and 30 years old, and 1 was over 30 years old. They were all right-handed and had prior experience with touch-screen mobile phones. Each participant received \$200 for successfully completing the experiment.

4.2. Apparatus

The ThumbStroke, Escape, and QWERTY keyboards were implemented in Java using the Android SDK in Eclipse for user evaluation. They were installed on two touch-screen smartphones. One was a Samsung Galaxy Note 2 phone with a 5.5" HD Super AMOLED display. The other one was a Kyocera Event phone with a 3.5" capacitive touch screen. When a participant interacted with those phones, logs in the mobile phones recorded the time and pixel coordinates of the interactions.

By following the guideline proposed by [16], we anchored Escape in the bottom-right corner of the Galaxy Note 2 phone without scaling as shown in Figures 3(b). In the Galaxy Note 2 phone, a one-handed (interaction) mode can be enabled when necessary, in which a keyboard is aligned to the right/left of the screen for right/left-handed users to make it easier for them to use the keyboard with only one hand. We adopted this mode by aligning QWERTY to the right side of the screen during the experiment (Figure 3(c)) because all participants were right-handed. The width of the QWERTY keyboard is the same as the one-handed QWERTY keyboard provided by the Galaxy Note 2 phone itself.

For the Kyocera Event phone, which had a smaller screen, Escape and QWERTY fitted the width of the screen. The size of Escape was the same as in [16] for both phones. ThumbStroke was presented in the center of the screen by default for both phones (Figure 3(a)), and users could adjust the position as they liked.

4.3. Independent and dependent measures

The independent variables are keyboard, phone, and session. We included session as an independent variable to assess participants' progress as practice increased (i.e., learning curve). The dependent variables include participants' text entry performance and perception. Participants' performance of text entry tasks was assessed by typing speed and error rates.

Words per minute (WPM): WPM measures typing speed. Here a "word" is defined as five characters, which is the average number of characters in a word, including spaces [35, 36].

Error rate: keystrokes are categorized into 4 groups based on Soukoreff and MacKenzie's error metrics [37]: Correct (C), Incorrect but Fixed (IF), Incorrect and Not Fixed (INF), and Fixed (F) keystrokes (e.g., backspace). Corrected error rate (CER) and uncorrected error rate (UER) are calculated based on those four groups of keystrokes.

CER refers to the percentage of errors that the participants committed and then corrected during text entry, which is calculated as $IF/(C+INF+IF)$. Corrected errors are not reflected in the final text, yet they are still an important aspect of the accuracy of text entry [38]. UER is the percentage of errors that are not corrected [36], which is calculated as $INF/(C+INF+IF)$ [36].

User perceptions: Participants' perceptions of the three keyboards, including perceived ease of use (PEOU), perceived effectiveness, and overall satisfaction, were

assessed through a post-study questionnaire. They were assessed through eight 7-point Likert scale questions (Table 1). Those questions were adapted from the IBM Post-Study System Usability Questionnaire [39] and were grouped into three factors.

Table 1. Questions of user perception factors

Factors	Items (1 = "Totally Disagree", 4 = "Neutral", 7 = "Totally Agree")
PEOU	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.

4.4. Experiment design

We asked the participants to enter 60 short phrases displayed on a desktop monitor in front of them as fast and accurately as possible using ThumbStroke, Escape and QWERTY and two phones in each session. The phrase set was adopted from [40], originally including 500 phrases, which varied from 16 to 43 characters in length (mean = 28.61). Symbols and numbers were not included in the phrases.

Since there was a learning curve for new keyboards, each participant completed 20 sessions in total in this study. Sessions 1 to 10 were practice sessions, which allowed participants to practice with the three keyboards on mobile phones. In the practice sessions, participants were allowed to look at the screen of the phones during text entry. Sessions 11 to 20 were used to evaluate the three keyboards for sight-free text entry. In those sessions, the screens of mobile phones were blocked with a paper cone attached to the participants' wrists with medical tapes so that they could not see the screens. Audio feedback was provided to the participants as TalkBack does. Whenever a character was selected, it would be read out to the participants. With QWERTY, the participants could slide their fingers over keys until the right character was announced and then lift their finger off the screen. Similar audio feedback was used in the study of No-Look Notes [17].

Depending on the participants' availability, any two consecutive sessions for both practice and evaluation

were scheduled with a 2-72-hour interval. Similar to other studies (e.g., [16]), the participants in this study were not allowed to complete more than three sessions within the same day.

Phrases were randomly picked from the original phrase set and grouped into sets of 10 phrases, without any repeated phrases in any single experiment session. During each session, the participants entered one set of phrases using each keyboard and a total of 60 phrases in each session (2 phones * 3 keyboards * 10 phrases). The order of keyboards, mobile phones, and phrase sets were all balanced out to minimize potential learning effects. Since different keyboards usually apply different auto-correction and word prediction algorithms, auto-correction and word prediction were disabled for all conditions to minimize possible confounding effects.

To simulate situational impairments and the mobility of users in the real world, the participants entered text phrases while walking on a treadmill. Following a previous study [41], the moving speed of the treadmill was set by individual participants according to their normal walking speed when interacting with a keyboard on a touch-screen mobile device. The mean of participants' selected treadmill speed was 2.0 km/h (SD = 0.7 km/h). To ensure one-handed interaction, participants were required to hold a phone and interact with it using their dominant hand only, while holding a remote controller in the other hand to display the next phrase on the screen after they finished entering the current phrase.

4.5. Procedure

After signing a consent form, the participants went through a 15-minute training session prior to the first practice session to get familiar with ThumbStroke, Escape, and QWERTY. The participants were explained how the three keyboards worked and practiced with several sample phrase entry tasks similar to those used in the formal experiment using the three keyboards. After they were comfortable with the keyboards and tasks, the practice sessions would start. Participants finished sessions 1-10 in the sighted condition without audio feedback for training purpose and completed sessions 11-20 in the sight-free condition with audio feedback. Before the first session in the sight-free condition (i.e., session 11), the participants had 15-minute training with the audio feedback using the three keyboards and several sample phrases similar to those used in the formal experiment. The participants filled out questionnaires about their perceptions at the end of the first and last sessions in the sight-free condition.

5. Results

We used StreamAnalyzer [30] to analyze text entry data collected during the study. We modified the metrics of NotCorrectedErrorRate and CorrectedErrorRate of StreamAnalyzer 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.

The means of WPM, UER and CER during the first and last sight-free session are presented in Table 2.

Table 2. Means of WPM, UER, and CER in the first and last sight-free sessions

Session	K	P	WPM	UER (%)	CER (%)
First	E	B	6.32±1.57	5.10±1.31	16.10±1.48
		S	5.53±1.09	5.05±1.22	18.79±1.47
	T	B	7.83±2.04	1.59±0.54	8.37±0.81
		S	7.23±2.19	2.02±0.51	8.93±1.29
	Q	B	4.72±1.66	7.96±2.20	14.40±2.85
		S	5.02±1.97	8.91±2.42	11.11±2.05
Last	E	B	11.45±1.48	3.76±1.17	9.21±0.95
		S	10.57±1.81	3.74±1.21	11.23±1.04
	T	B	10.50±1.30	0.91±0.27	5.99±0.85
		S	10.76±1.65	1.70±0.32	6.12±0.81
	Q	B	9.15±3.13	4.24±0.70	7.83±3.02
		S	9.13±2.66	4.29±1.09	7.97±2.86

Note: K (Keyboard): T (ThumbStroke), E (Escape), and Q (QWERTY); P (Phone): B (Big) and S (Small)

5.1. Typing speed

The average WPM of ThumbStroke and Escape in the first training session was 4.95 (SD = 0.97) and 6.00 (SD = 1.97), while in the last training session, they achieved WPM of 9.73 (SD = 1.47) and 11.17 (SD = 2.34) respectively. Since all participants were familiar with QWERTY, there was no obvious learning curve for it. The WPM of QWERTY was 21.00 (SD = 4.46) in the last training session.

The means of WPM of the three keyboards and two phones in the first and last sessions under the sighted-free condition are presented in Table 2. 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 ($F(1, 12) = 7.48, p < 0.05$) are all significant. The interaction effect between keyboard and phone is significant ($F(2, 24) = 5.00, p < 0.05$) as well. We did not find a significant interaction effect between keyboard and session, between phone and session, or among the three factors ($p > 0.05$). Text entry with ThumbStroke (mean = 9.43, SD = 2.18) and Escape (mean = 9.10, SD = 2.53) are significantly faster than text entry with QWERTY (mean = 7.52, SD = 2.89, and $p < 0.05$), but we did not find significant difference between ThumbStroke and

Escape ($p > 0.05$). The overall WPM of the big phone (mean = 8.83, SD = 2.69) is also significantly larger than that of the small phone (mean = 8.52, SD = 2.67, and $p < 0.05$).

ThumbStroke and Escape were significantly faster than QWERTY ($p < 0.05$) in the last session. The big phone also achieved faster speed than the small phone ($p < 0.05$) in the last session.

5.2. Error rate

5.2.1. Uncorrected error rate (UER). For UER, the means of the three keyboards and the two phones in the first and last sight-free sessions are presented in Table 2. 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 are significant. The main effect of phone is insignificant ($F(1, 12) = 0.72$, $p > 0.05$). No significant interaction effect was found ($p > 0.05$).

Using ThumbStroke resulted in significantly lower UER than using Escape (mean difference = - 3.17, $p < 0.05$) and QWERTY (mean difference = - 5.02, $p < 0.05$). No significant difference was found between QWERTY and Escape ($p > 0.05$). ThumbStroke had significantly lower UER than QWERTY in the last session ($p < 0.01$). There are no significant differences between QWERTY and Escape, ThumbStroke and Escape, and between big and small phones ($p > 0.05$).

5.2.2. Corrected error rate (CER). The means of CER of the three keyboards and two phones in the first and last sessions under the sight-free condition are presented in Table 2. The main effect of keyboard ($F(2, 22) = 3.69$, $p < 0.05$), and session ($F(2.58, 28.35) = 7.97$, $p < 0.01$) are significant. The main effect of phone is not ($F(1, 11) = 4.67$, $p > 0.05$). There is a significant interaction effect between keyboard and phone ($F(2, 22) = 5.25$, $p < 0.05$), but no significant interaction effect between keyboard and session, between phone and session, and among all three factors ($p > 0.05$).

Using ThumbStroke has significantly lower CER than using Escape (mean difference = - 5.27, $p < 0.05$). No significant difference was found between ThumbStroke and QWERTY and between Escape and QWERTY ($p > 0.05$).

ThumbStroke has significantly fewer CER than Escape in the last session ($p < 0.001$). There are no significant differences between Escape and QWERTY, ThumbStroke and QWERTY, and between the big and small phones ($p > 0.05$).

5.3. User perceptions

The Cronbach's Alphas for PEOU and perceived effectiveness constructs are 0.91 and 0.93, respectively.

The means of user perception factors (ranging from 1-7, with 1 representing the lowest perceptions and 7 representing the highest perceptions) are presented in Table 3. They are the overall evaluation of three keyboards with both phones. The main effects of keyboard and session on PEOU are significant (keyboard: $F(2, 24) = 24.59$, $p < 0.001$; session: $F(1, 12) = 7.50$, $p < 0.05$). The interaction effect between them is insignificant ($F(1.24, 14.82) = 3.78$, $p > 0.05$). ThumbStroke results in significantly higher PEOU than Escape ($p < 0.05$) and QWERTY (mean difference = 2.37, $p < 0.001$). Escape also has higher PEOU than QWERTY ($p < 0.05$). After the last evaluation session (session 20), the PEOU with all three keyboards is significantly higher than that after the first evaluation session (session 11) ($p < 0.05$).

Table 3. Means of user perceptions of three keyboards

Factors	Session	E	T	Q
PEOU	First	4.62±1.37	6.00±0.75	3.08±1.37
	Last	5.38±1.09	6.29±0.74	4.48±1.70
PE	First	4.37±1.39	5.92±0.98	2.75±1.38
	Last	5.35±1.24	6.27±0.98	4.19±1.60
OS	First	4.31±1.49	6.00±0.82	2.85±1.52
	Last	5.38±1.33	6.31±0.95	4.00±1.53

Note: T (ThumbStroke), E (Escape), and Q (QWERTY); PEOU (Perceived Ease of Use), PE (Perceived Effectiveness), and OS (Overall Satisfaction)

The main effects of keyboard and session on perceived effectiveness are significant (keyboard: $F(2, 24) = 23.25$, $p < 0.001$; session: $F(1, 12) = 6.74$, $p < 0.05$). The interaction effect between the two factors is also significant ($F(2, 24) = 3.92$, $p < 0.05$). ThumbStroke receives significantly higher perceived effectiveness than Escape ($p < 0.05$) and QWERTY ($p < 0.001$). Perceived effectiveness of Escape is also higher than that of QWERTY ($p < 0.05$). After the last evaluation session, perceived effectiveness with all three keyboards is significantly higher than that after the first session ($p < 0.05$).

The main effects of keyboard and session on overall satisfaction are significant (keyboard: $F(2, 24) = 23.13$, $p < 0.001$; session: $F(1, 12) = 6.01$, $p < 0.05$). The interaction effect between the two factors is not ($F(2, 24) = 2.47$, $p > 0.05$).

ThumbStroke receives significantly higher overall satisfaction than Escape ($p < 0.05$) and QWERTY ($p < 0.001$). Overall satisfaction with Escape is also significantly higher than that with QWERTY ($p < 0.05$). After the last evaluation session, overall satisfaction

with all three keyboards is significantly higher than that after the first session ($p < 0.05$).

6. Discussion

The uniqueness of ThumbStroke, in comparison to existing keyboards used on mobile devices, lies in the following aspects:

- ThumbStroke enables users to hold and interact with a touch-screen mobile phone with one hand only.
- It does not require precise tapping or pressing as traditional keyboards, such as QWERTY, do. It supports sight-free text entry.
- Existing soft keyboards used on mobile phones may suffer from the limited thumb accessibility problem. Text input with ThumbStroke relies on thumb strokes performed anywhere on a device screen rather than physical press on specific keys. So the thumb accessibility problem is eliminated with ThumbStroke.
- Almost all existing soft keyboards are located at the bottom of a touch-screen mobile device. Research has shown that placing a keyboard at the top or middle of a display can lead to lower error rates and higher user satisfaction than placing at the bottom of the display [42]. The location of ThumbStroke is flexible and can be moved by users as they like. The size of the keyboard can be adjusted. Because a user does not need to press keys to enter text, it will not cause the visual occlusion problem.
- Different from menu-based keyboards, which often require users to select an area and then a character separately, ThumbStroke combines area selection and character selection within one single stroke, which is more efficient.

The evaluation results show that QWERTY was outperformed by ThumbStroke and Escape in typing speed in the sight-free condition. It could be because QWERTY requires users to accurately press keys on it while the other two do not. ThumbStroke and Escape achieved a similar level of typing speed. Typing using the big phone was also faster than using the small phone, which is understandable. The larger screen size of a big phone should make the interactions with a keyboard easier.

ThumbStroke was significantly better than both Escape and QWERTY in UER for sight-free text entry. It may be because QWERTY requires accurate press on keys and Escape requires users to physically reach an area and then flick toward a direction, which could be challenging in one-handed interaction. In contrast, ThumbStroke does not have those limitations.

ThumbStroke is significantly better than Escape in CER for sight-free text entry. The possible reason is that

ThumbStroke provides a relatively easy and straightforward way for error correction in comparison to Escape. ThumbStroke and QWERTY achieved similar levels of CER. In addition, the participants using QWERTY performed better than using Escape in terms of CER, but worse in UER. It could be because with QWERTY, participants were less certain about the location of an aimed character, thus tended to hear the character to make sure the correct one was pressed, and then release the thumb from the screen to enter it. Moreover, in order to correct an error while using QWERTY, the participants would need to switch to the delete key by moving their thumbs around on the screen, which could be more challenging than swiping into a certain direction with Escape. As a result, when the participants made a mistake with QWERTY, they might be less willing to fix it.

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 in all these three aspects.

Unlike traditional keyboards, such as QWERTY, ThumbStroke is based on stroke directions. As a result, it is key-size independent. In addition, different from Escape, ThumbStroke has little restriction on its size. This feature may be particularly useful for devices with limited screen sizes, such as smartwatches. We plan to evaluate ThumbStroke on a smartwatch in the future.

There are some limitations of this study that lead to several future research opportunities. First, the character arrangement on ThumbStroke does not map to that on a regular QWERTY keyboard, with which most users are familiar. Currently, we used the alphabetical order, which was reported by the participants to be beneficial for them to remember characters' locations. Some other arrangements, such as based on character usage frequency in English words, are worth further investigations. For example, most commonly used letters in English, i.e., e, t, a, o, i, and n, as well as space and backspace, can be placed in the center of small areas. We also plan to investigate the difficulty of making strokes in different directions, and apply easy strokes to the most frequently used letters.

Second, the current ThumbStroke is designed for text entry in English. We have not explored its generalizability to other languages. Some languages have complex characters, such as Chinese. Chinese words can be entered with Pinyin, which has 26 characters almost identical to the English alphabet. As shown in Figure 1, there are still spaces on ThumbStroke for eight extra characters. Thus, we believe our keyboard can also be used for the entry of text in other languages that can be entered with no more than 34 distinct characters. The generalizability of

ThumbStroke is definitely worthy of future investigation.

Third, we did not assess the potential economic value of ThumbStroke, such as the amount of money that the participants felt willing to pay to use ThumbStroke. It would be interesting to examine this issue in future studies.

Acknowledgments

This material is based upon work supported by the National Science Foundation (awards #: CNS 1704800, SES 1527684). Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. We want to thank Mr. Nikola Banovic and Dr. Jacob O. Wobbrock for kindly providing us with the source code of Escape and StreamAnalyzer.

References

- [1] S. Brewster, J. Lumsden, M. Bell, M. Hall, and S. Tasker, "Multimodal/eyes-free interaction techniques for wearable devices," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2003, pp. 473-480.
- [2] S. Brewster, "Overcoming the lack of screen space on mobile computers," *Personal and Ubiquitous Computing*, vol. 6, pp. 188-205, 2002.
- [3] H. Korhonen, J. Holm, and M. Heikkinen, "Utilizing sound effects in mobile user interface design," in *Proceedings of the Human-Computer Interaction-INTERACT 2007*, pp. 283-296, 2007.
- [4] J. Lai and D. Zhang, "ExtendedThumb: A target acquisition approach for one-handed interaction with touch-screen mobile phones," *IEEE Transactions on Human-Machine Systems*, vol. 45, pp. 362-370, 2015.
- [5] A. Roudaut, S. Huot, and E. Lecolinet, "TapTap and MagStick: improving one-handed target acquisition on small touch-screens," in *Proceedings of the Working Conference on Advanced Visual Interfaces*, Napoli, Italy, pp. 146-153, 2008.
- [6] A. Karlson and B. Bederson, "ThumbSpace: generalized one-handed input for touchscreen-based mobile devices," in *Proceedings of the Human-Computer Interaction-INTERACT 2007*, pp. 324-338, 2007.
- [7] Y. S. Park and S. H. Han, "One-handed thumb interaction of mobile devices from the input accuracy perspective," *International Journal of Industrial Ergonomics*, vol. 40, pp. 746-756, 2010.
- [8] S. Boring, D. Ledo, X. A. Chen, N. Marquardt, A. Tang, and S. Greenberg, "The fat thumb: using the thumb's contact size for single-handed mobile interaction," in *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pp. 39-48, 2012.
- [9] I. S. MacKenzie and R. W. Soukoreff, "Text entry for mobile computing: models and methods, theory and practice," *Human-Computer Interaction*, vol. 17, pp. 147-198, 2002.
- [10] S. Azenkot and S. Zhai, "Touch behavior with different postures on soft smartphone keyboards," in *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices And Services*, pp. 251-260, 2012.
- [11] J.-B. Scheibel, C. Pierson, B. Martin, N. Godard, V. Fuccella, and P. Isokoski, "Virtual stick in caret positioning on touch screens," in *Proceedings of the 25 Ième Conférence Francophone On l'Interaction Homme-Machine*, p. 107, 2013.
- [12] C. Southern, J. Clawson, B. Frey, G. Abowd, and M. Romero, "An evaluation of BrailleTouch: mobile touchscreen text entry for the visually impaired," in *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pp. 317-326, 2012.
- [13] S. Mascetti, C. Bernareggi, and M. Belotti, "TypeInBraille: a braille-based typing application for touchscreen devices," in *Proceedings of the 13th International ACM SIGACCESS Conference on Computers and Accessibility*, pp. 295-296, 2011.
- [14] J. Oliveira, T. Guerreiro, H. Nicolau, J. Jorge, and D. Gonçalves, "BrailleType: unleashing braille over touch screen mobile phones," in *Proceedings of the IFIP Conference on Human-Computer Interaction*, pp. 100-107, 2011.
- [15] N. S. Subash, S. Nambiar, and V. Kumar, "BrailleKey: an alternative Braille text input system: comparative study of an innovative simplified text input system for the visually impaired," in *Proceedings of the Intelligent Human Computer Interaction (IHCI), 2012 4th International Conference on*, pp. 1-4, 2012.
- [16] N. Banovic, K. Yatani, and K. N. Truong, "Escape-keyboard: a sight-free one-handed text entry method for mobile touch-screen devices," *International Journal of Mobile Human Computer Interaction*, vol. 5, pp. 42-61, 2013.
- [17] M. N. Bonner, J. T. Brudvik, G. D. Abowd, and W. K. Edwards, "No-Look Notes: accessible eyes-free multi-touch text entry," in *Proceedings of the International Conference on Pervasive Computing*, pp. 409-426, 2010.
- [18] H. Tinwala and I. S. MacKenzie, "Eyes-free text entry on a touchscreen phone," in *Proceedings of the IEEE Toronto International Conference Science and Technology for Humanity TIC-STH*, pp. 83-88, 2009.

- [19] N. Green, J. Kruger, C. Faldu, and R. S. Amant, "A reduced QWERTY keyboard for mobile text entry," in *Proceedings of the CHI '04 Extended Abstracts on Human Factors in Computing Systems*, Vienna, Austria, pp. 1429-1432, 2004.
- [20] E. Matias, I. S. MacKenzie, and W. Buxton, "Half-QWERTY: A one-handed keyboard facilitating skill transfer from QWERTY," in *Proceedings of the INTERACT'93 and CHI'93 Conference on Human Factors in Computing Systems*, pp. 88-94, 1993.
- [21] S. Zhai, P. O. Kristensson, P. Gong, M. Greiner, S. A. Peng, L. M. Liu, *et al.*, "Shapewriter on the iphone: from the laboratory to the real world," in *CHI'09 Extended Abstracts on Human Factors in Computing Systems*, pp. 2667-2670, 2009.
- [22] S. Zhai and P. O. Kristensson, "The word-gesture keyboard: reimagining keyboard interaction," *Communications of the ACM*, vol. 55, pp. 91-101, 2012.
- [23] K. Perlin, "Quikwriting: continuous stylus-based text entry," in *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*, pp. 215-216, 1998.
- [24] J. Mankoff and G. D. Abowd, "Cirrin: a word-level unistroke keyboard for pen input," in *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*, pp. 213-214, 1998.
- [25] J. Lai and D. Zhang, "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*, pp. 2311-2316, 2014.
- [26] G. Kurtenbach and W. Buxton, "The limits of expert performance using hierarchic marking menus," in *Proceedings of the INTERACT'93 and CHI'93 Conference on Human Factors in Computing Systems*, pp. 482-487, 1993.
- [27] S. Zhao and R. Balakrishnan, "Simple vs. compound mark hierarchical marking menus," in *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, pp. 33-42, 2004.
- [28] K. Vertanen and P. O. Kristensson, "Complementing text entry evaluations with a composition task," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 21, p. 8, 2014.
- [29] G. Costagliola, V. Fuccella, and M. Di Capua, "Interpretation of strokes in radial menus: The case of the KeyScratch text entry method," *Journal of Visual Languages & Computing*, vol. 24, pp. 234-247, 2013.
- [30] J. O. Wobbrock and B. A. Myers, "Analyzing the input stream for character-level errors in unconstrained text entry evaluations," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 13, pp. 458-489, 2006.
- [31] B. Han and K. Kim, "Typing performance evaluation with multimodal soft keyboard completely integrated in commercial mobile devices," *Journal on Multimodal User Interfaces*, vol. 9, pp. 173-181, 2015.
- [32] N. P. Oo and N. L. Thein, "Design and Evaluation of Android Slide Keyboard for Myanmar Language," *International Journal of Informatics and Communication Technology (IJ-ICT)*, vol. 1, pp. 119-125, 2012.
- [33] I. S. Mackenzie and T. Felzer, "SAK: Scanning ambiguous keyboard for efficient one-key text entry," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 17, p. 11, 2010.
- [34] K. Lyons, T. Starner, and B. Gane, "Experimental evaluations of the twiddler one-handed chording mobile keyboard," *Human-Computer Interaction*, vol. 21, pp. 343-392, 2006.
- [35] J. O. Wobbrock, "Measures of text entry performance," in *Text Entry Systems: Mobility, Accessibility, Universality*, I.S. MacKenzie and K. Tanaka-Ishii, Eds., ed: San Francisco: Morgan Kaufmann, pp. 47-74, 2007.
- [36] B. Millet, "Design and evaluation of three alternative keyboard layouts for a five-key text entry technique," Doctor of Philosophy Dissertation, Industrial Engineering, University of Miami, 2009.
- [37] R. W. Soukoreff and I. S. MacKenzie, "Metrics for text entry research: an evaluation of MSD and KSPC, and a new unified error metric," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 113-120, 2003.
- [38] R. W. Soukoreff and I. S. MacKenzie, "Recent developments in text-entry error rate measurement," in *Proceedings of the CHI'04 Extended Abstracts on Human Factors in Computing Systems*, pp. 1425-1428, 2004.
- [39] J. R. Lewis, "IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions for use," *International Journal of Human-Computer Interaction*, vol. 7, pp. 57-78, 1995.
- [40] I. S. MacKenzie and R. W. Soukoreff, "Phrase sets for evaluating text entry techniques," in *Proceedings of the CHI'03 Extended Abstracts on Human Factors in Computing Systems*, pp. 754-755, 2003.
- [41] J. Bergstrom-Lehtovirta, A. Oulasvirta, and S. Brewster, "The effects of walking speed on target acquisition on a touchscreen interface," in *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, pp. 143-146, 2011.
- [42] T. Nakagawa and H. Uwano, "Usability evaluation for software keyboard on high-performance mobile devices," in *Proceedings of the HCI International 2011-Posters' Extended Abstracts*, pp. 181-185, 2011.