HiPad: Text entry for Head-Mounted Displays Using Circular Touchpad

Haiyan Jiang*
Beijing Institute of Technique

Dongdong Weng†
Beijing Institute of Technique
AICFVE of Beijing Film Academy

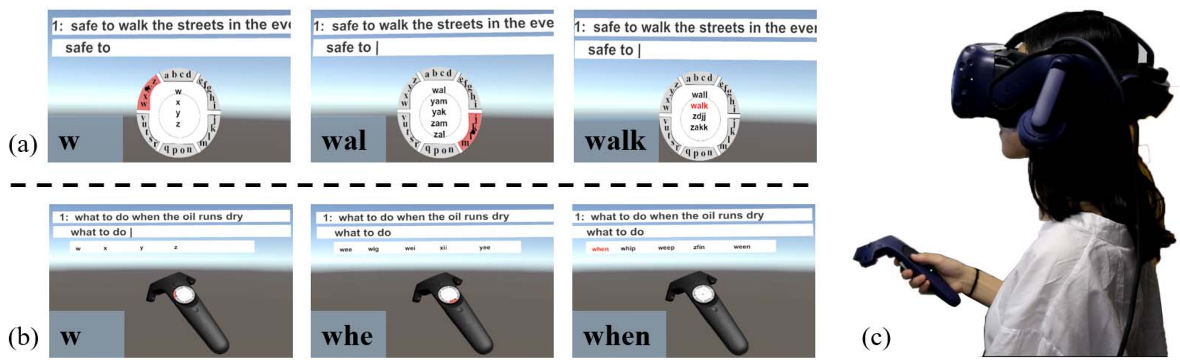


Figure 1: (a) Part of the entry process for ”walk” with VE-layout; (b) Part of the entry process for ”when” with TP-layout; (c) A user is typing with Hipad.

ABSTRACT

Text entry in virtual reality (VR) is currently a common activity and a challenging problem. In this paper, we introduce HiPad, leveraging a circular touchpad with a circular virtual keyboard, to support the one-hand text entry in mobile head-mounted displays (HMDs). The design of HiPad's layout is based on a circle and a square with rounded corners, where the outer circle is subdivided into six keys' regions containing letters. This technique input text by a common hand-held controller with a circular touchpad for HMDs and disambiguates the word based on the sequence of keys pressed by the user. In our first study, three potential layouts are considered and evaluated, leading to the design containing six keys. By analyzing the touch behavior of users, we optimize the 6-keys layout and conduct the second study, showing that the optimized layout has better performance. Then the third study is conducted to evaluate the performance of 6-keys HiPad with VE-layout and TP-layout and to study the learning curves. The results show that novices can achieve 13.57 Words per Minute (WPM) with VE-layout and 11.60 WPM with TP-layout and the speeds increase by 74.42% for VE-layout users and by 81.53% for TP-layout users through a short 60-phrase training.

Index Terms: Human-centered computing—Human computer interaction—Interaction paradigms—Virtual reality; Human-centered computing—Interaction techniques—Text input

1 INTRODUCTION

Text entry in virtual reality for mobile head-mounted devices (HMDs) remains an open problem. Many HMDs, such as HTC VIVE, Oculus rift and SAMSUNG HMD Odyssey, have now marketed to the masses. Hand-held controller devices for HMDs have become an important modality for interaction and for text entry in VR. However, aim-and-shoot technique [1, 2], the currently controller-based text entry technique, is tedious, cumbersome [3] and not efficient enough, in particular for the entry of long text. Besides, it is prone to cause fatigue quickly due to the movements in large space for selecting the characters [4].

Although other techniques such as speech [5, 6], mid-air typing [7–10] and head-based techniques [11–13] have been explored, there is still no common and efficient technique for virtual environments (VEs) in HMDs. In some situations, speech technique is error-prone and socially awkward. Mid-air typing usually relies on extra devices such as cameras or gloves, and is prone to cause fatigue for the long-time use. Head-based technique types by the head movements, increasing the possibility of motion sickness. More research about easy-of-use, easy-of-learning, and efficient text entry techniques for VEs in HMDs are needed.

In this paper, we introduce HiPad, a circular layout with multi-letter keys based one-hand text entry technique. This technique takes advantage of the shape of the circular touchpad on the controller and a circular virtual keyboard keeps consistent with it. Since the space of touchpad is limited, letters are condensed into several keys for HiPad, which is inspired by the work [14–18]. HiPad disambiguates the words typed based on the keys-pressed sequences by using word frequencies calculated from an English corpus and the list disambiguated words inspired by [14, 18, 19].

After designing the potential layouts, the first study is conducted for determining the layout with better performance, leading to the 6-keys layout, the speed of which could reach 9.14 WPM. Then, through analyzing the touch behavior of participants in the first study, the 6-keys layout is optimized by changing the size and position of keys and the second study is conducted to evaluate the performance of these optimized layouts. To address the occlusion problem of the virtual keyboard in VEs, we conduct the third study where we display the virtual keyboard on the virtual controller which corresponds with the physical controller and compare the effect of the circular layout's position. Besides, in the third study, both the learnability of HiPad with virtual keyboard displayed in virtual environments (VE-layout) and on the virtual touchpad (TP-layout) are explored. The results show the novices achieve an average speed of 13.57 WPM with VE-layout and 11.60 WPM with TP-layout through a short training with 60 phrases, and the uncorrected errors are 0.24% and 0.33% respectively. Through the short training, the speeds increase by 74.42% for the VE-layout users and by 81.53% for the TP-layout

*e-mail: jianghybit@163.com
†e-mail: crgj@bit.edu.cn

2642-5254/20/$31.00 ©2020 IEEE
DOI 10.1109/VR46266.2020.00092

users. This performance can be considered superior to the other techniques based on the circular layout in HMDs.

The contributions in this work include: (1) propose a text entry technique, HiPad, for mobile HMDs, using a circular touchpad on the controller; (2) optimize the 6-keys HiPad based on the touch behavior of users; (3) address the occlusion problem of the layout and explore the learnability of the two display methods of the keyboard that can demonstrate the possibility of placing the keyboard anywhere in VEs.

2 RELATED WORK

This work is relevant to text entry techniques for VEs in HMDs, circular layout and layouts with multi-letter keys.

2.1 Text entry in HMDs

Many research focuses on the text entry for VEs in HMDs with the development of VR technology and the widespread of the VR applications. For the text entry in HMDs, one problem is visual occlusion which causing users cannot use periphery devices such as physical keyboard conveniently, which also cannot be used in the mobile scenes.

One current solution is speech-based technique. Bowman [5] compares the input effects of pinch gloves, a one-hand chord keyboard, a soft keyboard using a pen and tablet, and speech, finding that the entry speed of speech is fastest at around 14 WPM, while the least error is generated in the pen and tablet keyboard technology. However, none of these techniques exhibit high levels of performance, availability, and user satisfaction. SWIFTER [6] is a speech-based multimodal text-entry system, which can achieve an entry rate of 23.6 WPM. However, speech-based text entry techniques are likely to suffer from problems in noisy environments [4]. Meanwhile, speech recognition has privacy and possible obtrusiveness in public environments and the correction of its errors is also a challenge [20]. It might interfere with the cognitive processes of tasks [21].

QWERTY physical keyboard, the most efficient text entry technique for computers, is used for HMDs [22–26]. For this technique, the challenge is the physical keyboard and the user's hands cannot to be saw in HMDs. James Walker proposes VelociTap for the QWERTY keyboard improving the entry efficiency when the keyboard is unseen but touchable [23]. In 2007, he modifies the system by adding the assistance of a virtual keyboard [24]. McGill [25] proposes an augmented virtual system for HMDs, which segments the image of hand and keyboard from the green background. The visual occlusion for real environments would cause input difficulties for users [23, 24], leading to higher error rates. Besides, the presentation of the keyboard and virtual hands would affect the performance [25,27–29]. HawKEY [26] is a text-entry system based on physical keyboard, which can be used while users stand, sit, walk in VR, and its speed can reach 77.7 WPM for experts and 44.1 WPM for novices.

Touchscreen-based techniques [4, 30–32] can be used in mobile scenes. However, it is difficult for users to sense their hands before the first press [30]. Besides, the extra devices with touchscreen such as smartphones and tablets are necessary. Mid-air typing [7–10] and gesture-based techniques have also proposed for HMDs. Other techniques using other sensors, such as RotoSwype [33], ThumbText [34] and HiFinger [35], are also used for HMDs. RotoSwype inputs text by word-gesture typing using a ring with a motion-tracking sensor worn the index finger in HMDs and it can achieve at least 14 WPM after a five-day study. ThumText inputs text by a ring-sized touch surface and it allows for the eyes-free condition, achieving 11.41 WPM after training. HiFinger inputs text using thumb-to-fingertips gestures with pressure sensors, achieving 9.82 WPM after 25-minute training. However, these techniques might require expensive extra sensors or devices like cameras or sensor-equipped gloves.

(a) The facade (b) The side

Figure 2: The controller of HTC VIVE PRO

Head-based techniques are also proposed for HMDs. Yu [11] compares three different head-based techniques, and the results show the best performance of about 24.73 WPM is from the GestureType technique. Marco Speicher [12] compares the head pointing, controller pointing, controller tapping, freehand, discrete and continuous cursor techniques and reported hand-held controllers technique outperformed all others. RingText [13] selects letters by using a virtual cursor controlled by the user's head movements, with expert users reaching an entry speed of 13.24 WPM after 90 minutes of training. However, the frequent movements of the head might increase the possibility of motion sickness [3].

For the current HMDs, the hand-held controller is an important modality for interaction. Although there are text entry methods based on a hand-held controller such as drum-Like VR Keyboard [36], some current methods with controllers might make users feel fatigued quickly and cost much time to learn [3]. The text entry techniques with controllers for VR remain underexplored. Thus, we hope to explore a text entry technique with the hand-held controller, providing a convenient, comfortable, efficiency text entry method. We explore the technique by taking advantage of the touchpad of the controller. In this paper, we use the HTC VIVE controller (Figure 2), which is a common controller with a touchpad. Since the split keyboard is the current default text input method like in [3, 12], we adopt this method in our technique.

2.2 Circular Layout

The circular layout is first proposed for the pen-based input (*e.g.*, Cirrin [37], T-Cube [38]). It can be used for small screens and might benefit the input of experts [38]. With TUP [39] all characters are assigned to fixed positions in circle layout, with an entry speed of novices reaching about 6-7 WPM. Circular layout is often used for text entry by combining with the gaze. PEYEwrite [40] selects letters by gazing at a hierarchical circular interface and the input speed could reach 7.85 WPM. SliceType [41] inputs text by gazing at a inner-outer circle layout where a language prediction model is applied to merge keys, with the entry speed reaching 3.45 WPM. Besides, the circular layout is also used for walk-up tabletop installations [42] and very large wall displays [43]. The circular layout can provide some potential extra benefits in interacting with circular interfaces [3]. It often used for the text entry of some devices with circular interfaces such as smartwathes. Users can input text using WrisText [44] by rotating the wrist in six directions with the input speed achieving 9.9 WPM and reaching 15.2 WPM after training.

Besides, the circular layout is also used for text entry in VR. RingText [13] combines the circular layout and gaze, achieving the speed of 11.30 WPM for novices and 13.24 WPM for experts after training. PizzaText [3] combines the circular layout and the Xbox One controller, achieving the speed of 8.95 WPM for novice users and 15.85 WPM for experts. However, this technique cannot be used by one-hand. Therefore, we explore the text entry technique by taking advantage of the circular layout and the circular touchpad on the controller.

2.3 Layouts with multi-letter keys

The multi-letter keys layout design is usually adopted to address the problem of the limited space (such as mobile phone and smartwatch), which enables to reduce the number of buttons required to support a full alphabetic keyboard [15–18]. The most common layout is T9 on smartphones or smartwatches. The novices can input text at 9 WPM and experts at 20 WPM for physical phone T9 entry (James and Reischel's study [45]). Invisiboard [46] can input text at 10.6 WPM on the smartwatch. FingerT9 [47] achieves 5.42 WPM for same-side-hand text entry on Smartwatches by using T9 layout and thumb-to-finger interaction. Komninos and Dunlop [48] proposes an ambiguous keyboard, which includes six keys containing three to six letters each, with entry speed reaching 8 WPM. UniWatch [49] containing a three-key keypad can input text at 9.84 WPM. The 1Line Keyboard [14] condenses the three rows of character keys in the QWERTY layout into a single line with eight keys for portable touchscreens, which can achieve 30 WPM after five 20-minute typing sessions.

For multi-letter keys input, it is necessary to adopt word disambiguation techniques for the word-level input. We achieve the text entry in the touchpad whose area is limited by reducing the number of keys and condensing multiple letters into one key.

3 HIPAD

As the hand-held controller is common modality for interaction for HMDs, We propose the text entry technique with circular layout - HiPad - based on the controller of HTC VIVE with a circular touchpad, which could provide potential extra benefits when it combines with circular layout. Figure 2 shows the controller, which contains a circular touchpad (the diameter is 0.04m) , menu button and trigger. The three buttons would be used for the text entry.

3.1 Layout

Since users are rarely willing to spend the time needed to learn a new keyboard layout [50, 51], we organize the letters in alphabetical order, taking advantage of the familiarity of the alphabetical order of users to minimize the learning time.

In the first study, we design three circular keyboard layouts (Figure 3). Figure 3(a)(b)(c) are the virtual keyboard users see in VEs and figure 3(d)(e)(f) are the real touch areas in the touchpad. In all layouts, we assume the diameters of the virtual keyboard and touchpad are 1 and the coordinates of the center of touchpad are (0,0); the diameter of the center square with rounded corners is 70% of the diameter of circle (Figure 3(a)); In the touchpad, the diameter of the center square is 50% of the diameter of touchpad (Figure 3(d)) and the center of the square is in the point (0,0). The real touch areas of keys are larger than the visual areas of keys, which leads users to touch the edge of touchpad and benefits the input accuracy. The first layout contains four keys. As shown in figure 3(d), when the key-1 on the touchpad is pressed, the key containing letters "a, b, c, d, e, f" is chosen. The second layout contains six keys and the third layout contains eight keys. Hereinafter, the center square area is called as the center-key. In experiments, virtual circular keyboards are rendered in the center of the field of view and far away from users (3m). The diameter of the keyboard was 0.5m. The diameter of the physical touchpad is 0.04m (figure 2(a)).

In the second and third studies, according to the touch points data in the first study, we optimize the touch areas in the physical touchpad and change the surface of the virtual circular keyboard (figure 4). The shape of the center-key is changed to a circle. In the virtual circular keyboard, a circle is added to help users confirm the touch area of center-key.

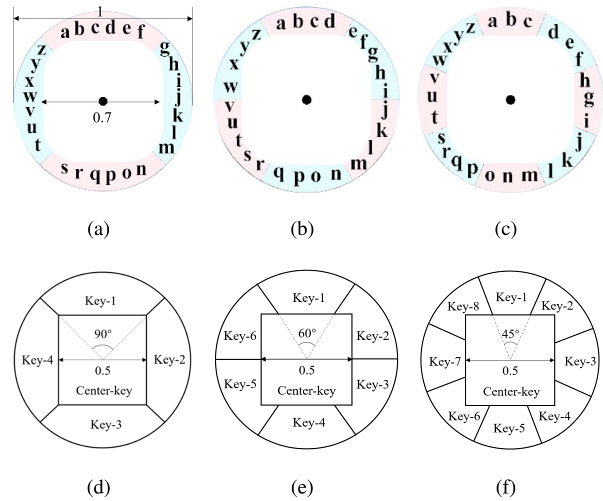
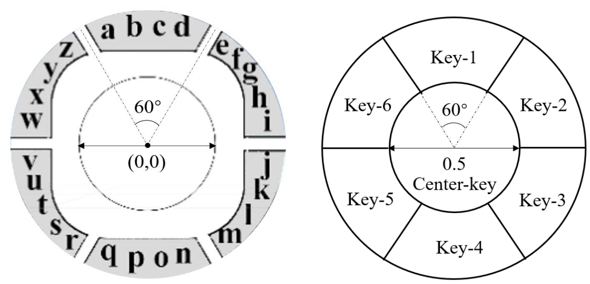


Figure 3: The possible keyboard layouts of HiPad. (a) Virtual keyboard of 4-keys layout; (b) Virtual keyboard of 6-keys layout; (c)Virtual keyboard of 8-keys layout; (d)Touch areas of keys of 4-keys layout; (e)Touch areas of keys of 6-keys layout; (f)Touch areas of keys of 8-keys layout.



(a) Virtual keyboard (b) Touch areas of keys

Figure 4: The optimized 6-keys layout

3.2 Word disambiguation

Since each key is associated with multiple letters (*e.g.*, for the 4-keys layout, the key corresponding to the key-1 can type the letters a, b, c, d, e, f), it is necessary to adopt word disambiguation for successful typing. We implement our algorithm for word disambiguation based on the American National Corpus (ANC) [52] containing a total of 239,208 unique words with respective frequencies. We first remove all entries in the ANC containing non-alphabetic characters. Then a hash table of all remaining words is built using the key sequence as the key and the words as the value. The candidate list ranks the candidates according to frequencies used. Figure 5 shows the coverage of the alphabetic only words in ANC in the top 10000 words. We count the percentage of words that appear as the most likely word for its key sequence and the percentage of words that appear in the top one, two, three, fourth and fifth most likely words for its key sequence. In the top 10000 words, 99.91%, 98.80% and 94.99% of those can be typed without the need for disambiguation in the top five most likely words for any key sequence in 8-keys, 6-keys and 4-keys respectively.

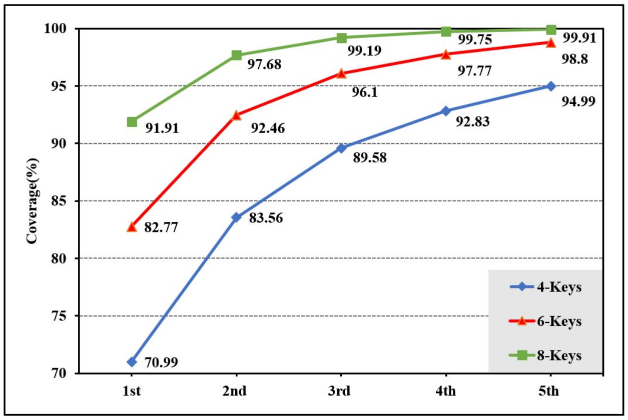
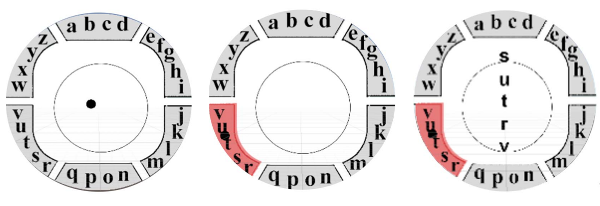


Figure 5: Coverage of the disambiguation algorithm for three layouts. Each line shows the proportion of the first, second, third, fourth and fifth words of the word list of the corresponding layouts.



(a) (b) (c)

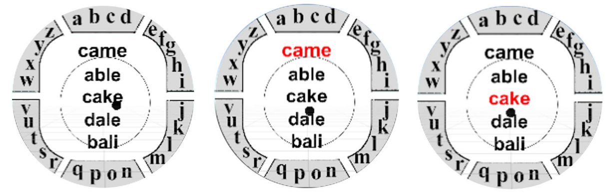
Figure 6: The process of letter selection. (a) The thumb touches on the center-key; (b) The thumb touch on one key; (c) The key was pressed and chosen.

3.3 Selection technique

Letter selection When the thumb of users touch on the physical touchpad, a black point would be rendered on the position of the touch point on the virtual circular keyboard corresponding to the touch position of the touchpad (figure 6(a)). Each area corresponds to the key containing the letters in this area; When the thumb touches the area of one key, the color of this key would change from gray to pink (figure 6(b)) as feedback and in this situation, if the user presses touchpad, the key would be selected (figure 6(c)). When the thumb touch on the center-key, the color would not change. Herein, the selection mechanism in three studies is same.

Word selection When a word sequence is pressed, the disambiguation word list would appear in the square in the center of the virtual keyboard. In this time, the color of all words is black (figure 6(a)) and if the user presses the center-key, the color of the first word in the list would change to red - we call this situation as Word-select-mode (figure 7(b)); the word with red color is called as the candidate-word. If the user presses the center-key again in the Word-select-mode, the candidate-word would be input. After one word is select, a space would be added automatically. But if the word only contains one letter, a space would not be added. Thus, out-of-vocabulary (OOV) can be input. Specially, When there is no word in the central square, pressing twice center-key can input "Space".

- **Next:** When in the Word-select-mode, press the key beneath the virtual keyboard (key-3 for 4-keys layout; key-4 for 6-keys layout; key-5 for 8-keys layout) to choose the next word as



(a) (b) (c)

Figure 7: The process of word selection. (a) The word list for one pressed sequence; (b) The first word is the candidate-word; (c) The third word is the candidate-word.

the candidate-word. For example, form figure 7(b) to 7(c), the "next" is pressed twice.

- **Last:** When in the Word-select-mode, press the key above the virtual keyboard (key-1 for all layouts) to choose the last word as the candidate-word.

The candidate list can show at most five prediction results (figure 7). If the candidate word is the fifth word in the list and the "next" is pressed, the candidate list shows the next five results if them exists; if the candidate word is the first word in the list and the "last" is pressed, the candidate list shows the last five results if them exists.

Delete The "menu" button on the controller (figure 2(a)). If an word sequence is pressed and there are predicted words in the center of the virtual keyboard, pressing the button once would delete the last input of the sequence and generate a new sequence. If there is no input sequence, pressing the button once would delete one letter in the input box.

Enter The "trigger" button on the controller (figure 2(b)).

This keyboard can be scaled to different sizes and it also can be put on the virtual model of controller since it only a visual cue for users who need to touch different area of the touchpad on the controller, which might be helpful for the limited field of view in VR.

4 THE FIRST STUDY: LAYOUT DESIGN

Since the area of the touchpad of the controller is limited, dividing the area into more keys might lead to high error rates and increase of positioning time which also might decrease the entry speed. However, more keys means fewer letters each key contained and the number of words corresponding to the input sequence would decrease and it is might easier for users to remember, which might benefit the entry speed. Thus, the goal of this study is to evaluate the effect of different numbers of keys in the fixed touchpad (4-keys, 6-keys, 8-keys).

4.1 Hypotheses

We formulate three hypotheses for the experiment:

H.1. The 6-keys layout will have a higher text entry speed than 4-keys and 8-keys. Since 4-keys layout contains more candidate words for one touch sequences, while 6-keys areas are big enough to touch as 4-keys. Meanwhile, too many letters in each key of 4-keys layout may lead to memory difficulty for users and the search time may rise. However, the touch area of each key of 8-keys are smaller, leading to more time for pointing target keys.

H.2. The 6-keys layout will have a lower error rate than the other two layouts. Since each key of 4-keys layout contains too many letters, resulting in memory difficulty, which may lead to more wrong pressed. Also, the area of each key of the 8-keys layout is too small, resulting in more pressed errors.

H.3. The 6-keys layout will have a lower NASA-TLX workload than the other two layouts. Since each key of 8-keys layout is more difficult to press precisely and quickly and for 4-keys layout, it is more difficult to find the key of one letter for users because one key contains more letters and locate the word needed to be input cost more time because it contains more candidate words for one touch sequence.

4.2 Participants

Ten right-handed participants (5 males; 5 females) between the ages of 21-29 (M=23.2) were recruited from our university campus to participate in this study. All participants input using right-hand in the study. Eight participants had some experiences with VR. Six participants were familiar with the T9 software keyboard on smartphones.

4.3 Apparatus and materials

The experiment was conducted on a computer with the Intel Core i7 processor and the NVIDIA GTX 1080Ti graphics card, and the software was implemented with C# in Unity 2019.2.1.f1. Our virtual environments were displayed in the HTV VIVE PRO that allowed participants to be completely immersed into the virtual world.

4.4 Experiments design and procedure

The experiment used the within-subjects design with one independent variable (three layouts). The whole experiment was divided into three sessions and each session included the evaluation of one kind of layout. Every participant completed the three sessions in a random order. We adopted the transcript method to evaluate the effect of different layouts. The transcript phrases were randomly generated from the MacKenzie phrase set [53].

Before each session, every participant needed to complete a practice trail with 10 phrases with the current layout. The whole experiment lasted approximately 70 minutes for each participant. In every session, every participant needed to type 15 phrases and after every session, participants were asked to complete the the NASA-TLX questionnaire [54] for the current layout. We adopted a 7-point ordinal scale to design NASA-TLX questionnaire. After every session, participants were also asked to give an preference of each layout on the scale of 0 (very dislike) to 7 (very like). After the whole experiment, we asked and collected the comments and advice about the layouts.

For the whole experiment, we collected 3 (layouts) $\times$15 (phrases) $\times$10 (participants) = 450 phrases.

4.5 Results

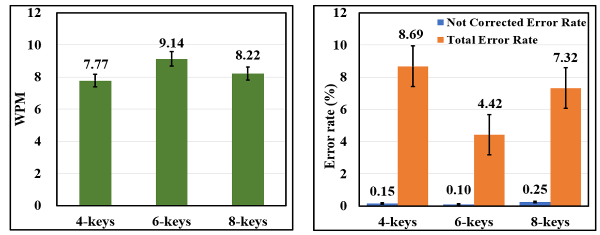
We calculated the text entry speed following Mackenzie [55]:

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

where T is the target string and S is the elapsed time in seconds from the first to the last pressed in the sentence. We calculated the total error rate (TER) [56], including the not corrected error rate (NCER) and corrected error rate (CER). We analyzed the data using a paired T-test for every two layouts and a GLM Repeated Measures ANOVA for three layouts, with Greenhouse-Geisser correction.

4.5.1 Text entry speed

Figure 8(a) shows mean text entry speed for each layout. The GLM Repeated Measures ANOVA results show there is no significant difference among the three layouts ($F_{1.399,12.593} = 4.177,\ p = 0.052,\ \eta_p^2 = 0.317$). But the paired T-test results shows there are significant differences between 4-keys layout and 6-keys layout ($t_9 = 2.700,\ p = 0.024$) and between 6-keys layout and 8-keys layout ($t_9 = 3.053,\ p = 0.014$).



(a) (b)

Figure 8: (a) Average text entry speed across three designs of HiPad; (b) Average total error rate (TER) and not corrected error rate (NCER) across three designs of HiPad.

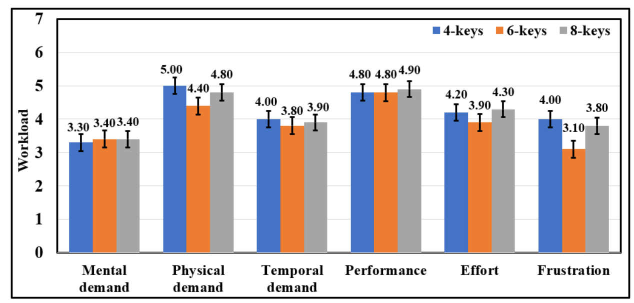


Figure 9: Average NASA-TLX scores across three designs of HiPad.

4.5.2 Error rate

Figure 8(b) shows the total error rate (TER) and not corrected error rate (NCER) across three layouts.

For TER, the GLM Repeated Measures ANOVA results show there is no significant effect among the three layouts ($F_{1.432,12.891} = 4.177,\ p = 0.317,\ \eta_p^2 = 0.161$). The paired T-test results also show there are no significant effects between any two layouts (4-keys and 6-keys layouts: $t_9 = 2.100,\ p = 0.065$; 4-keys and 8-keys layouts: $t_9 = 0.455,\ p = 0.660$; 6-keys and 8 keys layouts: $t_9 = 1.565,\ p = 0.152$).

For NCER, the GLM Repeated Measures ANOVA results show there is no significant effect among the three layouts ($F_{1.402,12.617} = 4.177,\ p = 0.355,\ \eta_p^2 = 0.104$). The paired T-test results also show there are no significant effects between any two layouts (4-keys and 6-keys layouts: $t_9 = 0.803,\ p = 0.443$; 4-keys and 8-keys layouts: $t_9 = 0.800,\ p = 0.444$; 6-keys and 8 keys layouts: $t_9 = 1.263,\ p = 0.238$).

4.5.3 Workload and preference

Figure 9 shows the NASA-TLX workload scores across the three layouts. The GLM Repeated Measures ANOVA results show there is no significant effect on all six measures among the three layouts. The paired T-test results reveal that there is no significant difference between any pairs of six measures.

Figure10 shows the preference of each keyboard. The mean score for 6-keys layout is the highest.

4.5.4 The effect of familiarity of T9 keyboard

Some participants are familiar with the T9 software keyboard on smartphones. However, the independent T-test results show there is no effect on their performance ($t_8 = 1.183,\ p = 0.271$). For participants who are familiar with the T9 keyboard, the average text entry speed is 7.78$\pm$1.68 WPM. For participants who are not, the average

Table 1: The proportion of the times of keys pressed by mistake.

Layout	Proportion of the times of keys pressed by mistake							
	key-1	key-2	key-3	key-4	key-5	key-6	key-7	key-8
4-keys	27.38%	23.35%	36.69%	14.49%				
6-keys	15.63%	32.30%	10.51%	18.46%	22.30%	9.38%		
8-keys	11.56%	18.22%	15.11%	4.44%	13.78%	20.89%	11.11%	4.89%

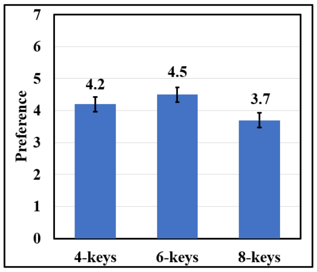
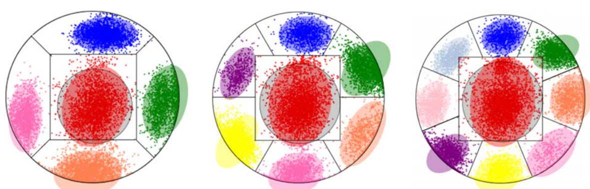
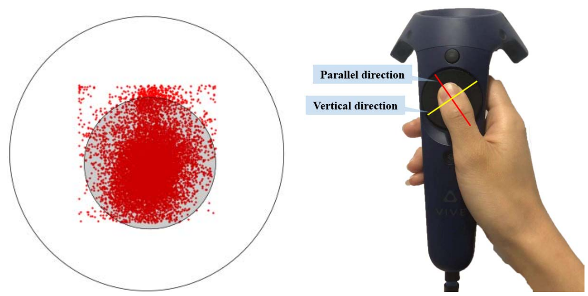


Figure 10: Average preference scores across three designs of HiPad.



(a) (b) (c)

Figure 11: Scatter plots with 95% confidence ellipses of touch points of each design of HiPad. (a) Touch points of 4-keys layout; (b) Touch points of 6-keys layout; (c) Touch points of 8-keys layout.



(a) (b)

Figure 12: (a) Touch points of the center-key in all designs; (b) The touch behavior of thumb on touchpad.

text entry speed is 8.88±0.96 WPM. This would indicate the familiarity of the T9 keyboard would not affect the user's performance.

Besides, some participants commended that the keys of 8-keys layout was small and it cost them more time to locate. Also, some participants mentioned that the margin among keys is not very clear.

4.6 Touch behavior analysis

Figure 11 shows scatter plots with 95% confidence ellipses the touch points of all participants for each layout, where the gray translucent circles in the center area are the minimum circles which contains 95% points of the center-key touch points and the center of each circle is the average coordinate of all center-key touch points for each layout.

Figure 12(a) shows the touch points of the center-key of all layouts, which includes 10369 points. Hereinafter, we assume that the diameter of the touchpad is 1, and thus, the diameter of this gray translucent circle is 0.479 and the center of circle is (0.023, -0.070).

The key pressed by mistake is the one users press actually when they intend to press other keys. The more the key is pressed by mistake, the key is prone to be pressed and the press accuracy will be higher. Meanwhile, the key with fewer press by mistake is more difficult for users to press and it is easier for them to press the adjacent keys. Thus, the press accuracy of these keys is lower.

For the 4-keys layout (figure 11(a)), the times of the key-1 pressed by mistake accounts for 27.38% of all times; the times of the key-2 pressed by mistake accounts for 23.35%; the times of the key-3 pressed by mistake accounts for 36.69%; the times of the key-4 pressed by mistake accounts for 14.49% (table 1). For the times: key-3 > key-1 > key-2 > key-4. In this study, since participants type by the right hand, the key-4 is the farthest away from the thumb, which might lead to a lower press accuracy.

For the 6-keys layout (figure 11(b)), the times of the key-1 pressed by mistake accounts for 15.63% of all times; the times of the key-2 pressed by mistake accounts for 32.30%; the times of the key-3 pressed by mistake accounts for 10.51%; the times of the key-4 pressed by mistake accounts for 18.46%; the times of the key-5 pressed by mistake accounts for 22.30%; the times of the key-6 pressed by mistake accounts for 9.38% (table 1). For the times: key-2 > key-5 > key-4 > key-1 > key-3 > key-6. The press accuracy of key-3 and key-6 is low.

For the 8-keys layout (figure 11(c)), the times of the key-1 pressed by mistake accounts for 11.56% of all errors; the times of the key-2 pressed by mistake accounts for 18.22%; the times of the key-3 pressed by mistake accounts for 15.11%; the times of the key-4 pressed by mistake accounts for 4.44%; the times of the key-5 pressed by mistake accounts for 13.78%; the times of the key-6 pressed by mistake accounts for 20.89%; the times of the key-7 pressed by mistake accounts for 11.11%; the times of the key-8 pressed by mistake accounts for 4.89% (table 1). For the times: key-6 > key-2 > key-3 > key-5 > key-1 > key-7 > key-8 > key-4. The press accuracy of key-8 and key-4 are low.

For all layouts, as shown in figure 12(b), the press accuracy of the keys that are parallel to the direction of the thumb movement is low, and have more errors of pressing by mistake. The press accuracy of the keys that are vertical to the direction of the thumb movement is higher. That also can be found in the scatter plots with 95% confidence ellipses.

4.7 Discussion

The results support our hypotheses. The 6-keys layout might have the largest potential for use. Users can press the keys of it more quickly and precisely than the 8-keys layout and it is easier for the users to remember the letters in each key of 6-keys layout than in

8-key layout, so the text entry speed of the 6-key layout high and the error rate and workload low. We conducted the experiment in a random order and from the results, and we find that there is a learning effect where participants usually performed slightly better in the second and third sessions. As 3 participants completed the 6-keys layout session as the first session while 3 participants for 4-keys layout and 4 participants for 8-keys layout, we believe 6-keys layout has better performance.

Although some participants mentioned that they are more familiar with the 8-keys layout since the letters in each key are similar to the layout of T9, we did not find that it has an effect on the performance. It might result from that we changed the traditional T9 layout to a circular layout. Thus, the final layout we choose was the 6-keys layout.

Moreover, according to the analysis of touch points, we find the touch positions of center-key are concentrated in a circle, whose center is close toward the palm. Thus, changing the position and shape of the center area might benefit typing performance. Besides, it is more difficult for participants to press the keys which in the direction parallel to the thumb. Thus, increasing the area of keys in this direction may lead to a better typing performance. It requires a further study.

5 THE SECOND STUDY: TOUCH AREA OPTIMIZED DESIGN

According to the results of the first study, we optimized the design of HiPad with 6-keys layout.

5.1 Layout optimized

We optimized the layout from three aspects.

The area of the center-key: the shape of the center-key is changed from the rounded square to circle. The diameter of the circle is 0.5 (assuming the diameter of the virtual keyboard is 1) and the center position of it is in the point (0.023, -0.070) (figure 13). The change also leads to the increase the area of each key.

The angles of key-3 and key-6: according to the scatter plots with 95% confidence ellipses of touch points of each key, increasing the angles of the key-3 and key-6 benefits the press accuracy of the two keys. For the original distribution of keys, each key accounts for 60°. Then, the angles of key-3 and key-6 increase to 70°, while the angles of other keys decrease to 55°14(c).

The layout: although the touch areas of keys are conterminal, the interspace is added among keys in the virtual keyboard. When participants touch the edge of the keys, it is easier for them to touch ambient keys, so the interspace in the virtual keyboard would not encourage them to touch the edge of keys (figure 13). Besides, to help the users confirm the position of center-key, we add a circle in the layout. In this study, the center position of the circle is changed with the position of center-key in different sessions in the virtual keyboard, while the shapes of keys on the virtual keyboard keep same (figure 13).

5.2 Hypotheses

In this study, we have two hypotheses.

H.4. The change of angle and position of the center-key would lead to faster entry speed.

H.5. The effect of the angle change would higher than the effect of the center-key's position change.

5.3 Participants and apparatus

Ten new right-handed participants (5 males; 5 females) between the ages of 22-27 (M=24.2) were recruited from our university campus to participate in this study. All participants typed using right hand in the study. The apparatus and devices in this study were the same as in the first study.

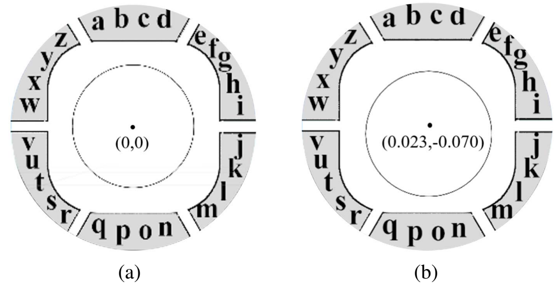


Figure 13: The virtual keyboard of optimized 6-keys HiPad. (a) The center of the circle of the center-key in the point (0,0); (b) The center of the circle of the center-key in the point (0.023,-0.070).

5.4 Experiments design and procedure

The experiment used a within-subjects design with one independent variable (four different touch areas). The whole experiment was divided into four sessions and each session included the evaluation of one kind of touch area. "NoChange (NC)" session (figure 14(a)): the center of the center-key circle is located in the point (0,0), and the angles of each key are same (60°); "AngleChange (AC)" session (figure 14(c)): the center of the center-key circle is located in the point (0,0), and the angles of keys in the key-2 and key-6 are 70°, the angles of other keys are 55°; "CircleChange (CC)" session (figure 14(b)): the center of the center-key circle is located in the point (0.023, -0.070), and the angles of each key are same (60°); "TwoChange (TC)" session (figure 14(d)): the center of the center-key circle is located in the point (0.023, -0.070), and the angles of keys in the key-2 and key-6 are 70°, the angles of other keys are 55°. Every participant completed the four sessions in a random order. The transcript phrases were randomly generated from the MacKenzie phrase set [53].

Before the experiment, every participant needed to complete a practice trail with 10 phrases. The whole experiment lasted approximately 45 minutes for each participant. In every session, every participant needed to type 12 phrases. In the experiment, participants were not told the difference between different sessions.

For the whole experiment, we collected 4 (sessions) $\times$ 12 (phrases) $\times$ 10 (participants) = 480 phrases.

5.5 Results

The date was analyzed using paired T-tests for every two sessions and a GLM Repeated Measures ANOVA for four sessions, with Greenhouse-Geisser correction.

5.5.1 Text entry speed

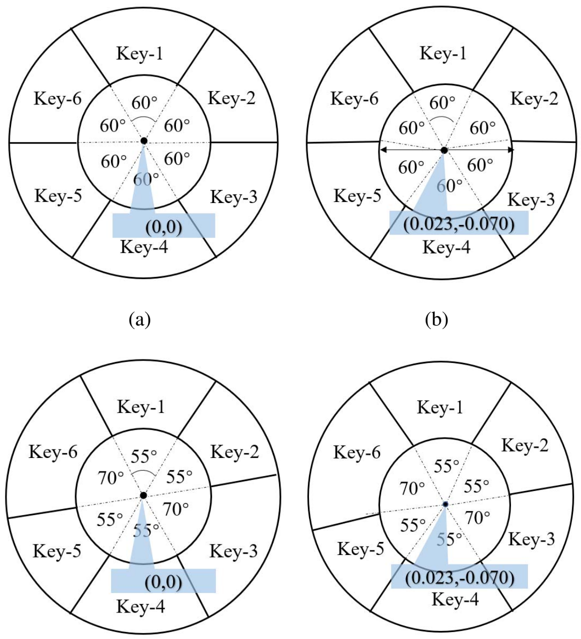
Figure 15(a) shows the text entry speed across the four sessions. The average speed of "TwoChange" session is higher than other sessions, but the GLM Repeated Measures ANOVA results show there is no significant difference among the four sessions ($F_{2.422,21.979} = 4.177,\ p = 0.300,\ \eta_p^2 = 0.125$). And the paired T-test results also do not yield any significant effect.

5.5.2 Error rate

Figure 15(b) shows the total error rate (TER) and not corrected error rate (NCER) across four sessions.

For TER, the GLM Repeated Measures ANOVA results show there is no significant effect among the four sessions ($F_{2.051,18.463} = 0.926,\ p = 0.416,\ \eta_p^2 = 0.093$). The paired T-test results show there are no significant effects between any two sessions.

For NCER, the GLM Repeated Measures ANOVA results show there is no significant effect among the three layouts ($F_{2.205,19.845} =$



(a) (b)

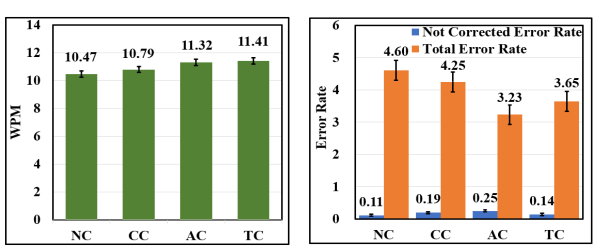
(c) (d)

Figure 14: The distribution of touch areas of optimized 6-keys HiPad. (a) NoChange; (b) CircleChange; (c)AngleChange; (d)TwoChange.

$0.671,\ p = 0.536,\ \eta_p^2 = 069$). The paired T-test results show there are no significant effects between any two sessions.

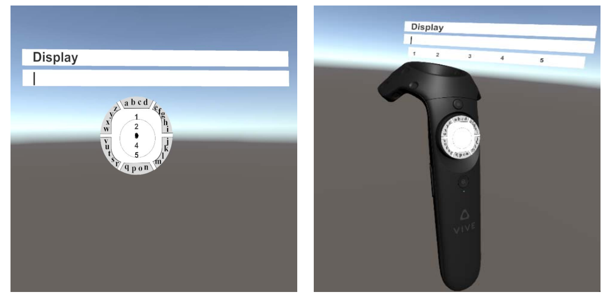
5.6 Discussion

Our results slightly support the H.4 and H.5., as the participants in TC and AC sessions performed a little better than in NC and CC sessions. Since the input modes in the different sessions are similar, there might be a learning effect and we conducted the four sessions experiments in a random order, leading to insignificant results. The experimental order of sessions would affect the performance of four input modes since 10 participants are not enough to make the design fully counterbalanced and participants performed relatively worse in the first session due to unfamiliarity. As 2 or 3 participants completed the one of the four input modes in the first session, the results, to some extent, can demonstrate the AC and TC optimized layouts perform better. Besides, for the left-hand participants, if they input using the left hand, the change of keys will be symmetrical, which means the center of the center circle should move toward palm in the direction parallel to the thumb. The angles of the closest and farthest key to the palm should be increased.



(a) (b)

Figure 15: (a) Average text entry speed across four optimized 6-keys HiPad; (b) Averege total error rate (TER) and not corrected error rate (NCER) across four optimized 6-keys HiPad.



(a) (b)

Figure 16: (a) HiPad with VE-layout; (b) HiPad with TP-layout.

6 THE THIRD STUDY: THE EFFECT OF KEYBOARD POSITION

When this technique is used in some virtual environments, the virtual keyboard may cause occlusion. The virtual keyboard can be placed on the touchpad of the virtual controller (figure 16(b)). We separate the place where the word candidates are displayed and the place where the virtual keyboard is displayed. When the circular virtual keyboard is in the virtual environments, we call that as VE-layout (figure 16(a)); when the virtual keyboard is in the touchpad of the virtual controller, we call that as TP-layout. In the TP-layout, the display method of word candidates is changed (figure 16(b)), since this method is more line with people's habits and the eyes of humans have a larger field of view in the horizontal direction than in the vertical direction. While in the VE-layout, the display method of word candidates remains the same to decrease the range of viewpoint changes. Thus, the goal of this study is to evaluate the effect of layout position. Besides, we are interested in how the two different layout users' performance would change over time. In this study, we choose the TC optimized layouts for the experiment.

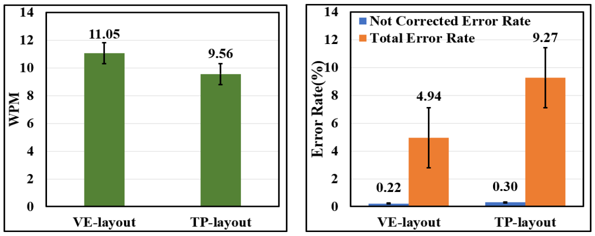
6.1 Participants and apparatus

Fifteen right-handed participants (10 males; 5 females) between the ages of 20-27 (M=22.5) were recruited from our university campus to participate in this study. All participants input using right hand in the experiment. Participants were divided into two groups (group one included 7 participants; group two included 8 participants). None participants participated in the two before studies. The apparatus and devices in this study were the same as in the first study.

6.2 Experiments design and procedure

The experiment used a between-subjects design. The participants of group one input phrases with VE-layout, while the participants of group two input phrases with TP-layout. In the whole experiment, except for the difference in the display of the virtual keyboard, the other procedure was same for the two groups. In order to evaluate the learnability, there was no practice. Before the experiment, participants were told how to input using this technique. The whole experiment was divided into six blocks with each block containing 10 phrases.

The whole experiment lasted approximately 50 minutes for each participant. Between two blocks, participants could have a break. After the last block, participants were asked to complete the NASA-TLX questionnaire.



(a) (b)

Figure 17: (a) Average text entry speed across the two types of HiPad; (b) Averege total error rate (TER) and not corrected error rate (NCER) across the two types of HiPad.

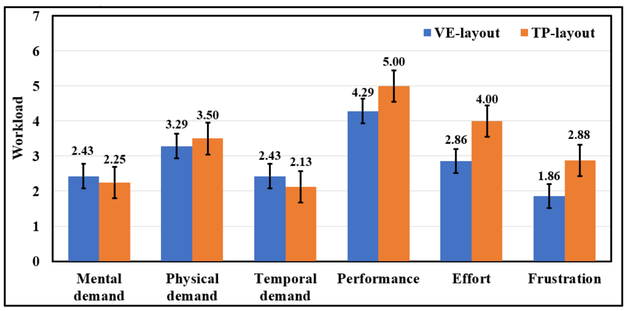


Figure 18: Average NASA-TLX scores across the two types of HiPad.

For the whole experiment, we collected 2 (layouts) $\times$ 6 (block) $\times$ 10 (phrases) $\times$15 (participants) = 1800 phrases.

6.3 Results

In this study, the data were analyzed using the independent T-test and the GLM Repeated Measures ANOVA methods.

6.3.1 Text entry speed

Figure 17(a) shows the text entry speed of VE-layout group and TP-layout group. The speed of the VE-layout group is higher than the speed of the TP-layout. However, the independent T-test results show that there is no significant difference between the two groups ($t_{13} = 1.718,\ p = 0.110$).

6.3.2 Error rate

Figure 17(b) shows the mean error rate of the two groups. According to the results of independent T-test, both NCER and TER of VE-layout group are significantly lower than the TP-layout (for NCER: $t_{13} = 6.237,\ p < 0.000$, for TER: $t_{13} = 6.547,\ p < 0.000$).

6.3.3 Workload

Figure 18 shows the NASA-TLX workload scores of the two groups. No significance is observed between any pairs of the six measures.

6.3.4 Learnability

In order to understand the general learning curve of the two display methods of virtual keyboard, we analyze the average speeds of two types changed over time (figure 19).

The GLM Repeated Measures ANOVA results show that for both the VE-layout and the TP-layout groups , there are significant differences across six blocks (for the VE-layout: $F_{2.351,14.108} = 25.254,\ p < 0.000,\ \eta_p^2 = 0.808$, for the TP-layout: $F_{2.039,14.271} = 17.927,\ p < 0.000,\ \eta_p^2 = 0.719$). Moreover, polynomial contrasts demonstrate that there are significant linear rising trends for both groups (for the VE-layout: $F_{1,6} = 80.520,\ p < 0.000,\ \eta_p^2 = 0.931$, for the TP-layout: $F_{1,7} = 45.681,\ p < 0.000,\ \eta_p^2 = 0.867$). For each block, the independent T-test results show that there are no significant differences between two groups (block1: $t_{13} = 2.101,\ p = 0.056$; block2: $t_{13} = 2.364,\ p = 0.034$; block3: $t_{13} = 1.516,\ p = 0.153$; block4: $t_{13} = 1.200,\ p = 0.251$; block5: $t_{13} = 0.925,\ p = 0.372$; block6: $t_{13} = 1.570,\ p = 0.140$).

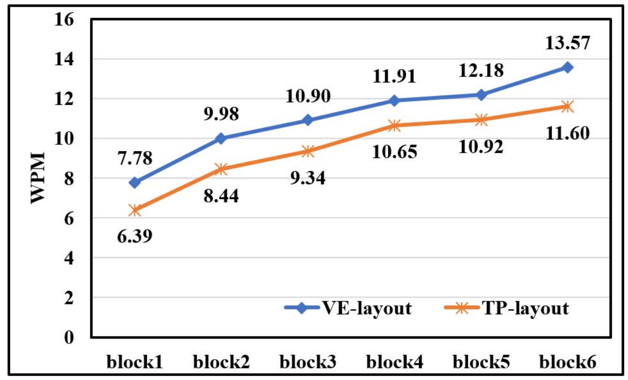


Figure 19: Average speed of of the two types HiPad across six blocks.

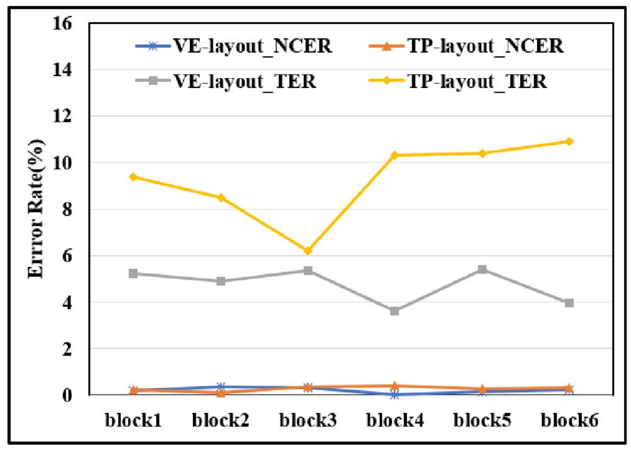


Figure 20: Average total error rate (TER) and not corrected error rate (NCER) of the two types HiPad across six blocks.

In the last block, the average speed of VE-layout group improves to 13.57 WPM from the 7.78 WPM in the first block through a short training with 60 phrases; the speed of TP-layout improves to 11.60 WPM from 6.39 WPM in the first block. Through the short training, the speed increased by 74.42% for VE-layout group and by 81.53% for TP-layout group. Moreover, the speed of one participant using VE-layout achieved 18.72 WPM and the speed of one participant using TP-layout achieved 14.16 WPM.

The TER and NCER change over time are shown in figure 20 .

For NCER, the GLM Repeated Measures ANOVA results show that for both the VE-layout and the TP-layout groups, there is no significant difference across six blocks (for the VE-layout: F2.155,12.930=0.879, p=0.446,=0.128, for the TP-layout: $F_{5,35} = 0.971,\ p = 0.449,\ \eta_p^2 = 0.122$). Polynomial contrasts demonstrate that there is no significant linear trend for both groups (for the VE-layout: $F_{1,6} = 0.369,\ p = 0.934,\ \eta_p^2 = 0.001$, for the TP-layout:

$F_{1,7} = 0.779,\ p = 0.407,\ \eta_p^2 = 0.100$). For each block, the independent T-test results show that except the block3 ($t_{13} = -2.989,\ p = 0.010$), there is no significant difference between two groups (block1: $t_{13} = 0.264,\ p = 0.796$; block2: $t_{13} = 1.308,\ p = 0.214$; block4: $t_{13} = 1.907,\ p = 0.079$; block5: $t_{13} = 0.877,\ p = 0.397$; block6:$t_{13} = 0.430,\ p = 0.674$).

For TER, the GLM Repeated Measures ANOVA results show that for both the VE-layout and the TP-layout groups, there is no significant difference across six blocks (for the VE-layout: $F_{2.593,15.557} = 0.461,\ p = 0.802,\ \eta_p^2 = 0.071$; for the TP-layout: $F_{2.393,16.754} = 1.680,\ p = 0.165,\ \eta_p^2 = 0.194$). Polynomial contrasts demonstrate that there is no significant linear trend for both groups (for the VE-layout: $F_{1,6} = 2.143,\ p = 0.194,\ \eta_p^2 = 0.263$, for the TP-layout: $F_{1,7} = 1.691,\ p = 0.235,\ \eta_p^2 = 0.195$). For each block, the independent T-test results show that except the block6 ($t_{13} = 2.306,\ p = 0.038$), there is no significant difference between two groups (block1: $t_{13} = 1.356,\ p = 0.198$; block2: $t_{13} = 1.521,\ p = 0.152$; block3: $t_{13} = 0.490,\ p = 0.632$; block4: $t_{13} = 2.085,\ p = 0.057$; block5: $t_{13} = 1.253,\ p = 0.232$).

6.4 Discussion

For novices, the average speed of 6-keys Hipad with VE-layout can reach 13.57 WPM and the average speed of it with TP-layout can reach 11.60 WPM through a short training with 60 phrases. These results seem to indicate that the HiPad technique performs better, comparing with some current text entry techniques using in HMDs such as PizzaText [3] and RingText [13]. The speeds of novices with PizzaText can reach 8.59 WPM after two more hours of training and RingText can reach 11.30 WPM after one hour training. But more strict comparison is needed. For the HiPad technique, the NCER is very low (0.22% for VE-layout and 0.30% for TP-layout).

In terms of learnability, we believe both HiPad with VE-layout and TP-layout are easy-to-learn according to the learning curves. Through the short training, the text entry speed increased by 74.42% for the VE-layout group and by 81.53% for the TP-layout group.

Across six blocks, the average speed of TP-layout lower than the VE-layout. For novices, they could not remember the distribution of the letters and they might need search time for some letters' choosing and for the TP-layout group, the transfer of fixations of eyes cost some time, might leading to the decrease of users' performance. More training might benefit the memory of the distribution of letters and decrease the frequency of the transfer of eyes' fixations. Besides, no significance is observed in the statistical results. In terms of the subjective rating, there is no significance difference between the two groups. According to the results, we can speculate the circular layout containing letters can be placed in the other place in VEs. But the practical applications require more research.

Overall, we believe the HiPad with TP-layout and with VE-layout both have potential use in HMDs, although the entry speed of HiPad with TP-layout is lower than that with VE-layout. The HiPad with TP-layout addresses the occlusion problem of keyboard in VEs.

7 Limitation and future work

There are some limitations, although this work has successfully produced a novel text entry technique in VEs for HMDs. For example, adding physical points in the center of the keys in the touchpad would benefit the users' confirm of touch when their thumb on keys, which might be beneficial to the entry speed. Besides, the entry performance of OOV may be limited due to multi-letter keys and needs more study.

Also, in the future, we will improve this technique form these aspects as following:

Special symbols This research only considers the entry of letters. However, a complete text entry technique should contain the entry of special symbols and numbers. Future work could explore the input mode of these symbols and numbers.

The layout of the letters Because the text entry of HiPad is based on word-level, the layout of the letters will affect the word collision (words with identical tapping sequences). Besides, for a same word, different layouts would have different input sequences, which would theoretically affect the text entry speed according to the Fitts' law [57]. The distribution of letters of HiPad can be optimized by combining the word clarity [17, 58] with speed predicted by the Fitts-Digraph model [59, 60] in the future.

8 Conclusion

In this paper, we introduce the HiPad, a circular touchpad based text entry technique that enables one-handed typing in HMDs. This technique allows users to press the virtual keys in touchpad for the ambiguous text entry. The first user study is conducted to determine the number of keys of this technique and the results demonstrate that the 6-keys layout has better performance compared to the 4-keys and 8-keys layouts. Then we analyze the data of touch points and found that it is more difficult for participants to touch the keys which in the direction parallel to the thumb. According to the results, we optimize the touch areas of keys and conduct the seconded study, finding that changing the center-keys position toward palm in the direction parallel to the thumb and the angles of the closest and farthest keys to the palm benefit the text entry. To address the occlusion problem of the virtual keyboard, the third study is conducted, which evaluates the final layout of 6-keys HiPad with VE-layout and TP-layout. The results of the study show that novice users can achieve 13.57 WPM with VE-layout and 11.60 WPM with TP-layout and the speed increased by 74.42% for VE-layout group and by 81.53% for TP-layout group through a short training with 60 phrases. We believe the HiPad is an efficient and easy-to-use technique for text entry in HMDs, especially for mobile scenes.

9 Acknowledgments

This work was supported by the National Key Research and Development Program of China (No.2017YFB1002504) and the National Natural Science Foundation of China (No.61960206007) and the 111 Project (B18005).

Acknowledgments

References

[1] Joseph J LaViola Jr, Ernst Kruijff, Ryan P McMahan, Doug Bowman, and Ivan P Poupyrev. *3D user interfaces: theory and practice*. Addison-Wesley Professional, 2017.

[2] Jakob Olofsson. Input and display of text for virtual reality head-mounted displays and hand-held positionally tracked controllers, 2017.

[3] Difeng Yu, Kaixuan Fan, Heng Zhang, Diego Monteiro, Wenge Xu, and Hai-Ning Liang. Pizzatext: Text entry for virtual reality systems using dual thumbsticks. *IEEE transactions on visualization and computer graphics*, 24(11):2927–2935, 2018.

[4] Jens Grubert, Lukas Witzani, Eyal Ofek, Michel Pahud, Matthias Kranz, and Per Ola Kristensson. Text entry in immersive head-mounted display-based virtual reality using standard keyboards. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 159–166. IEEE, 2018.

[5] Doug A Bowman, Chadwick A Wingrave, JM Campbell, VQ Ly, and CJ Rhoton. Novel uses of pinch gloves™ for virtual environment interaction techniques. *Virtual Reality*, 6(3):122–129, 2002.

[6] Sebastian Pick, Andrew S Puika, and Torsten W Kuhlen. Swifter: Design and evaluation of a speech-based text input metaphor for immersive virtual environments. In *2016 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 109–112. IEEE, 2016.

[7] Francine Evans, Steven Skiena, and Amitabh Varshney. Vtype: Entering text in a virtual world. *submitted to International Journal of Human-Computer Studies*, 1999.

[8] Manuel Pratorius, Ulrich Burgbacher, Dimitar Valkov, and Klaus Hinrichs. Sensing thumb-to-finger taps for symbolic input in vr/ar environments. *IEEE computer graphics and applications*, 2015.

[9] Xin Yi, Chun Yu, Mingrui Zhang, Sida Gao, Ke Sun, and Yuanchun Shi. Atk: Enabling ten-finger freehand typing in air based on 3d hand tracking data. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 539–548. ACM, 2015.

[10] Doug A Bowman, Christopher J Rhoton, and Marcio S Pinho. Text input techniques for immersive virtual environments: An empirical comparison. In *Proceedings of the human factors and ergonomics society annual meeting*, pages 2154–2158. SAGE Publications Sage CA: Los Angeles, CA, 2002.

[11] Chun Yu, Yizheng Gu, Zhican Yang, Xin Yi, Hengliang Luo, and Yuanchun Shi. Tap, dwell or gesture?: Exploring head-based text entry techniques for hmds. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 4479–4488. ACM, 2017.

[12] Marco Speicher, Anna Maria Feit, Pascal Ziegler, and Antonio Krüger. Selection-based text entry in virtual reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 647. ACM, 2018.

[13] Wenge Xu, Hai-Ning Liang, Yuxuan Zhao, Tianyu Zhang, Difeng Yu, and Diego Monteiro. Ringtext: Dwell-free and hands-free text entry for mobile head-mounted displays using head motions. *IEEE transactions on visualization and computer graphics*, 25(5):1991–2001, 2019.

[14] Frank Chun Yat Li, Richard T Guy, Koji Yatani, and Khai N Truong. The 1line keyboard: a qwerty layout in a single line. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 461–470. ACM, 2011.

[15] Brian Frey, Caleb Southern, and Mario Romero. Brailletouch: mobile texting for the visually impaired. In *International Conference on Universal Access in Human-Computer Interaction*, pages 19–25. Springer, 2011.

[16] Mikael Goldstein, Robert Book, Gunilla Alsiö, and Silvia Tessa. Non-keyboard qwerty touch typing: a portable input interface for the mobile user. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 32–39. ACM, 1999.

[17] Ryan Qin, Suwen Zhu, Yu-Hao Lin, Yu-Jung Ko, and Xiaojun Bi. Optimal-t9: An optimized t9-like keyboard for small touchscreen devices. In *Proceedings of the 2018 ACM International Conference on Interactive Surfaces and Spaces*, pages 137–146. ACM, 2018.

[18] Nathan Green, Jan Kruger, Chirag Faldu, and Robert St Amant. A reduced qwerty keyboard for mobile text entry. In *CHI'04 extended abstracts on Human factors in computing systems*, pages 1429–1432. ACM, 2004.

[19] http://www.t9.com/.

[20] Keith Vertanen. *Efficient correction interfaces for speech recognition*. PhD thesis, Citeseer, 2009.

[21] Ben Shneiderman. The limits of speech recognition. *Communications of the ACM*, 43(9):63–65, 2000.

[22] Haiyan Jiang, Dongdong Weng, Zhenliang Zhang, Yihua Bao, Yufei Jia, and Mengman Nie. Hikeyb: High-efficiency mixed reality system for text entry. In *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pages 132–137. IEEE, 2018.

[23] J Walker, S Kuhl, and K Vertanen. Decoder-assisted typing using an hmd and a physical keyboard. In *Extended Abstracts of the the ACM Conference on Human Factors in Computing Systems, CHI*, volume 16, 2016.

[24] James Walker, Bochao Li, Keith Vertanen, and Scott Kuhl. Efficient typing on a visually occluded physical keyboard. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 5457–5461. ACM, 2017.

[25] Mark McGill, Daniel Boland, Roderick Murray-Smith, and Stephen Brewster. A dose of reality: Overcoming usability challenges in vr head-mounted displays. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 2143–2152. ACM, 2015.

[26] Duc Minh Pham and Wolfgang Stuerzlinger. Hawkey: Efficient and versatile text entry for virtual reality. pages 1–11, 11 2019.

[27] Pascal Knierim, Valentin Schwind, Anna Maria Feit, Florian Nieuwenhuizen, and Niels Henze. Physical keyboards in virtual reality: Analysis of typing performance and effects of avatar hands. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 345. ACM, 2018.

[28] Jens Grubert, Lukas Witzani, Eyal Ofek, Michel Pahud, Matthias Kranz, and Per Ola Kristensson. Effects of hand representations for typing in virtual reality. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pages 151–158. IEEE, 2018.

[29] Jia-Wei Lin, Ping-Hsuan Han, Jiun-Yu Lee, Yang-Sheng Chen, Ting-Wei Chang, Kuan-Wen Chen, and Yi-Ping Hung. Visualizing the keyboard in virtual reality for enhancing immersive experience. In *ACM SIGGRAPH 2017 Posters*, page 35. ACM, 2017.

[30] Jan Gugenheimer, David Dobbelstein, Christian Winkler, Gabriel Haas, and Enrico Rukzio. Facetouch: Enabling touch interaction in display fixed uis for mobile virtual reality. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 49–60. ACM, 2016.

[31] Youngwon R Kim and Gerard J Kim. Hovr-type: Smartphone as a typing interface in vr using hovering. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*, pages 333–334. ACM, 2016.

[32] Yiqin Lu, Chun Yu, Xin Yi, Yuanchun Shi, and Shengdong Zhao. Blindtype: Eyes-free text entry on handheld touchpad by leveraging thumb's muscle memory. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(2):18, 2017.

[33] Aakar Gupta, Cheng Ji, Hui-Shyong Yeo, Aaron Quigley, and Daniel Vogel. Rotoswype: Word-gesture typing using a ring. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, page 14. ACM, 2019.

[34] Junhyeok Kim, William Delamare, and Pourang Irani. Thumb-text: Text entry for wearable devices using a miniature ring. In *Proceedings of Graphics Interface*, pages 18–25, 2018.

[35] Haiyan Jiang, Dongdong Weng, Zhenliang Zhang, and Feng Chen. Hifinger: One-handed text entry technique for virtual environments based on touches between fingers. *Sensors*, 19(14):3063, 2019.

[36] Costas Boletsis and Stian Kongsvik. Text input in virtual reality: A preliminary evaluation of the drum-like vr keyboard. *Technologies*, 7(2):31, 2019.

[37] Jennifer Mankoff and Gregory D Abowd. Cirrin: a word-level unistroke keyboard for pen input. In *ACM Symposium on User Interface Software and Technology*, pages 213–214, 1998.

[38] Dan Venolia and Forrest Neiberg. T-cube: a fast, self-disclosing pen-based alphabet. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 265–270. ACM, 1994.

[39] Morten Proschowsky, Nette Schultz, and Niels Ebbe Jacobsen. An intuitive text input method for touch wheels. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 467–470. ACM, 2006.

[40] Anke Huckauf and Mario H Urbina. Gazing with peyes: towards a universal input for various applications. In *Proceedings of the 2008 symposium on Eye tracking research & applications*, pages 51–54. ACM, 2008.

[41] Burak Benligiray, Cihan Topal, and Cuneyt Akinlar. Slicetype: fast gaze typing with a merging keyboard. *Journal on Multimodal User Interfaces*, pages 1–14, 2018.

[42] Uta Hinrichs, Holly Schmidt, Tobias Isenberg, Mark S Hancock, and Sheelagh Carpendale. Bubbletype: Enabling text entry within a walk-up tabletop installation. 2008.

[43] Garth Shoemaker, Leah Findlater, Jessica Q Dawson, and Kellogg S Booth. Mid-air text input techniques for very large wall displays. In *Proceedings of Graphics interface 2009*, pages 231–238. Canadian Information Processing Society, 2009.

[44] Jun Gong, Zheer Xu, Qifan Guo, Teddy Seyed, Xiang'Anthony' Chen, Xiaojun Bi, and Xing-Dong Yang. Wristext: One-handed text entry on smartwatch using wrist gestures. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, page 181. ACM, 2018.

[45] Christina L James and Kelly M Reischel. Text input for mobile devices: comparing model prediction to actual performance. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages

365–371. ACM, 2001.

[46] Aske Mottelson, Christoffer Larsen, Mikkel Lyderik, Paul Strohmeier, and Jarrod Knibbe. Invisiboard: maximizing display and input space with a full screen text entry method for smartwatches. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*, pages 53–59. ACM, 2016.

[47] Pui Chung Wong, Kening Zhu, and Hongbo Fu. 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*, page 178. ACM, 2018.

[48] Andreas Komninos and Mark Dunlop. Text input on a smart watch. *IEEE Pervasive Computing*, 13(4):50–58, 2014.

[49] Franck Poirier and Mohammed Belatar. Uniwatch: A soft keyboard for text entry on smartwatches using 3 keys. In *International Conference on Human-Computer Interaction*, pages 341–349. Springer, 2016.

[50] I Scott MacKenzie, Shawn X Zhang, and R William Soukoreff. Text entry using soft keyboards. *Behaviour & information technology*, 18(4):235–244, 1999.

[51] Xiaojun Bi, Barton A Smith, and Shumin Zhai. Quasi-qwerty soft keyboard optimization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 283–286. ACM, 2010.

[52] Nancy Ide and Catherine Macleod. The american national corpus: A standardized resource of american english. In *Proceedings of corpus linguistics*, volume 3, pages 1–7. Lancaster University Centre for Computer Corpus Research on Language . . . , 2001.

[53] I Scott MacKenzie and R William Soukoreff. Phrase sets for evaluating text entry techniques. In *CHI'03 extended abstracts on Human factors in computing systems*, pages 754–755. ACM, 2003.

[54] Sandra G Hart. Nasa-task load index (nasa-tlx); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*, pages 904–908. Sage publications Sage CA: Los Angeles, CA, 2006.

[55] I Scott MacKenzie. A note on calculating text entry speed. *Unpublished work. Available online at http://www. yorku. ca/mack/RN-TextEntrySpeed. html*, 2002.

[56] R William Soukoreff and I Scott 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*, pages 113–120. ACM, 2003.

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

[58] Xin Yi, Chun Yu, Weinan Shi, Xiaojun Bi, and Yuanchun Shi. Word clarity as a metric in sampling keyboard test sets. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 4216–4228. ACM, 2017.

[59] Xiaojun Bi, Barton A Smith, and Shumin Zhai. Multilingual touchscreen keyboard design and optimization. *Human–Computer Interaction*, 27(4):352–382, 2012.

[60] Shumin Zhai, Alison Sue, and Johnny Accot. Movement model, hits distribution and learning in virtual keyboarding. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 17–24. ACM, 2002.