International Journal of Human–Computer Interaction

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/hihc20

Climbing Keyboard: A Tilt-Based Selection Keyboard Entry for Virtual Reality

Junfeng Huang, Minghui Sun, Jun Qin, BoYu Gao & Guihe Qin

INTERNATIONAL JOURNAL OF HUMAN–COMPUTER INTERACTION
https://doi.org/10.1080/10447318.2022.2144120

Climbing Keyboard: A Tilt-Based Selection Keyboard Entry for Virtual Reality

Junfeng Huang[a,b], Minghui Sun[a,b], Jun Qin[c], BoYu Gao[d], and Guihe Qin[a]

[a]College of Computer Science and Technology, Jilin University, China; [b]Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, China; [c]College of Computer Science and Technology, Changchun University of Science and Technology, China; [d]College of Computer Science and Technology, Jinan University, China

ABSTRACT

Text input is one of the common interaction tasks in virtual environments. However, current inputting methods (e.g., laser-input: aim-and-shoot technique) have many limitations, such as inefficiency, lack of precision, and fatigue of long-text inputting. We propose a *Climbing Keyboard* method to allow easier, faster, and more accurate text input, and use tilt instead of precise aiming. The selected target changes from a specific letter to a group of letters with such a tilt-based interaction based on the QWERTY layout, which aims to reduce the learning cost, especially for novice users. Meanwhile, expert users can focus on the screen without looking at the keyboard. We designed three user studies to evaluate the performance of proposed method, including the verification of the usability of the tilt interaction method in the first study, optimization of the tilt angle range in the second study, and evaluation of the learning curve of Climbing keyboard in the last study. Our results showed that participants can reach 16.48 words per minute after an hours of training.

1. Introduction

Text input is one of essential tasks for many applications. For example, typing account name and password when a user logs into a system. The performance of text inputting can be reached 40 WPM MacKenzie (2002) with computer or 33 WPM Azenkot and Zhai (2012) with smartphone. However, the performance of text inputting in VR is only about 13 WPM, and the user gets tired quickly. There are some common methods to support text input in VR (e.g., laser-pointing, gestures, speech-based). However, some challenges occurred when using these common methods to input for VR. On one hand, the well-known method of text inputting in VR is the aim-and-shoot technique, however, jitter issues affects users' performance in this method D. Yu et al. (2018). When users press the button on the controller to confirm selection, the direction and position of the controller will be changed, which is known as Heisenberg effect Bowman et al. (2001). Meanwhile, users can feel mental fatigue and physical fatigue quickly, because they need to aim the letters accurately by laser means they need to pay more attention to their body control. On the other hand, with development of artificial intelligence, some researchers have proposed speech Pick et al. (2016) and mid-air input Gupta et al. (2019); Hoste et al. (2012) to avoid the shortcomings of typing with a handle controller. However, speech is not suitable for noisy scenes, meanwhile, it makes it difficult to protect users' privacy as well. The input method for Head-Mounted Displays (HMDs) Jimenez (2017) brings high physical demand and motion sickness, which makes users dizzy. Mid-Air input González et al. (2009) requires additional cameras or gloves to capture the movements on the hand, cumbersome wearable devices are not suitable for long-text input. Some of them are slow and enable low accuracy, high learning costs, and even a low immersion of VR. Therefore, it is important to investigate a text entry method with the features of efficient, precise, and easy to learn.

In response to the mentioned challenges, this article proposes a novel technique, Climbing Keyboard, that enables users to enter texts efficiently and easily. It is a tilt-based selection keyboard using two controllers with a QWERTY layout. We divide the characters on the QWERTY keyboard into 6 parts, each controller mapping three parts (Figure 1). Compared to aiming the letters on the keyboard accurately, tilting controllers at different approximate angles makes users feel less fatigue and more fluently. After users choose one part by tilting the controller, the color of characters of this part changed to red to facilitate participants to choose. Record the parts selected by the user in order, and generate candidate words according to this order. Users can input the intended word by touching the corresponding area of the touchpad. The design of Climbing Keyboard is underpinned by three solution principles for VR text entry (efficient, easy to learn, comfortable to use), which we have distilled from the literature and prior experience in text entry design for VR. In our opinion, using the QWERTY keyboard layout can reduce the user's learning cost-effectively. But there is no clear evidence that the QWERTY

CONTACT Minghui Sun smh@jlu.edu.cn College of Computer Science and Technology, Jilin University, China

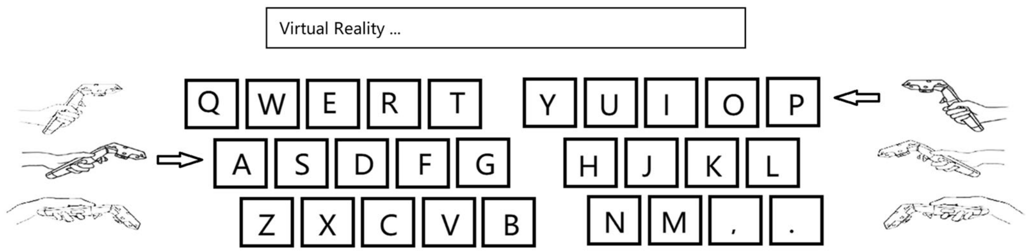

Figure 1. The characters on QWERTY keyboard be divided into 6 parts, each controller mapping three parts.

layout is effective for text input on a circular and small screen. Therefore, we designed a circular layout which is often used in small screen interaction for comparison. Finally, the experimental results show that the QWERTY layout is better than the circular layout in terms of speed and accuracy.

To evaluate Climbing Keyboard, we designed four candidate layouts, and carry out experiments to select the proper layout. The result shows users can input the intended word accurately and fast. Even novice users can reach 11.21 WPM. We are concerned that different tilt angles may affect the user's typing performance. After analyzing the tilt operation of users in the first experiment, we conducted the second experiment with different angles. We divide the 90° into three angles as "up," "middle," and "down." The first angle range is equally divided by 90°. The second angle range is based on dual hands. Thirdly, the tilt range of each handle is based on the tilt habit of each hand. Compare the three angle ranges through experiments. In terms of speed and accuracy, Finally, we chose 0°–26°, 26°–56°, 56°–90° as the optimal tilt angle of the layout. We conducted a five-block user study to evaluate the performance of the Climbing Keyboard (the best performing design found in the second study). The results showed that users could achieve 16.48 WPM after a short training. All the studies were granted ethical approval by the Jilin University College of Computer Science and Technology.

Our contributions in this work include: (1) Integration of text entry in Virtual Reality. (2) Climbing Keyboard design and implementation. (3) optimize the Climbing Keyboard based on the tilt habits of users.

2. Related work

2.1. Typing in virtual environments

There are various kinds of researches on text entry in virtual reality Environments. The most popular and conventional way of text entry in VR is the aim-and-shoot style, in which the user holds the handle and cast a virtual laser to select the letters on the keyboard accurately and make the final confirmation using a trigger of the controller Lee and Kim (2017); Olofsson (2017). Jigger affects users' performance in this method. Another kind of technique, called Google Drum Keys, taps the virtual keyboard like a drum, which was proposed by Google Daydream Labs, but this technique is easy to cause fatigue quickly because it needs users to make a large spatial movement to type one character. Lee and Kim (2017) added some buttons on the handles to remind users of their muscle memory in the usual touch typing, However, some characters must be aimed by ray. When users input a word, they need to push the button as well as aim the letter on a virtual keyboard. Recently, D. Yu et al. (2018) presented a PizzaText, a circular layout-based text entry technique for VR systems using the two thumbsticks of an Xbox controller. By rotating the two joysticks of the game controller, users could easily enter text by using this circular keyboard. Instead of using controllers, several studies focus on typing with a physical keyboard in VR Knierim et al. (2018); Lin et al. (2017); Schneider et al. (2019); Walker et al. (2017). However, HMDs, such as HTC Vive, PlayStation, and Oculus Rift, which are used to display virtual scenes, obscures the position of the keyboard Jimenez (2017). Despite McGill et al. (2015), using a dedicated device a keyboard, they marked it to ease tracking by the virtual reality system. Kim and Kim (2017) used the screen of a smartphone as a typing interface, users could type on the phone. From our point of view, using an extra device for inputting text will interrupt users' immersion.

Some researchers focus on Mid-air text-entry without handheld devices. Tap Systems proposed a new type of VR typing. The user wore a finger tap that could track the movement of the hand for text entry. Aakar et al. proposed a word-gesture typing using a ring technique called RotoSwype Gupta et al. (2019) by rotating users' wrists to draw a line on a virtual keyboard. These methods were lack of haptic feedback and required a lot of training costs. Tying in the air could be tiring for the user and was not suitable for long periods of time and a large amount of text input. Speech techniques Adhikary and Vertanen (2021); Hoste et al. (2012); Pick et al. (2016) were undoubtedly the fastest and the most convenient of all input methods. However, it was not practical when entering passwords or speaking in public. It might be difficult to correct mistakes. Speicher et al. (2018) evaluated several methods of text-entry in VR, such as Head pointing C. Yu et al. (2017), Eye gaze selection Majaranta and Räihä (2007), Finger gestures González et al. (2009), and Physical keyboard Lee and Chung (2008). His conclusion showed that pointing using tracked hand-held controllers outperformed all other methods.

2.2. Tilt-based text-entry for handheld devices

The possibility of using tilt handles to text entry is given by an accurate tracker in VR. There are many types of research about tilt-based text-entry for the handheld device. Tilt is always used to be explored for interaction for replacing some specific gestures Hinckley and Song (2011; as shake to confirm) or solving the problem with one-handed interaction Chang et al. (2015); Yeo, Phang et al. (2017). Partridge et al. (2002) proposed TiltType which was the first research for using tilt in text-entry. TiltTpye combined the buttons of the device, direction, and the angle of tilt to the intended character. GesText Jones et al. (2010) used the accelerometer to select the characters. To solve the problem of inconvenience with a single hand, Yeo et al. proposed SWIM technique Yeo et al. (2017) which retrieved the orientation of the device and mapped it to absolute tilt magnitude and screen coordinates (x,y) for controlling a pointer to infer the intended word. Wigdor and Balakrishnan (2003) proposed a tilt-based selection keyboard. Users pressed the buttons of mobile phones to select one part, then tilted the device in different directions to complete twice selection and confirm the intended characters. Aakar et al. proposed the RotoSwype technique Gupta et al. (2019), which was a word-gesture tying method using a ring in VR and AR. Users needed to rotate their wrist and draw a line that contained the intended characters according to the QWERTY keyboard layout. The user's action was tracked by a ring that was worn on the index finger. Tilt gave us a second dimension to extend the method of text entry.

However, most of the current tilt-based text input methods are designed as gesture input. Users need to memory the specific gesture and it increases the users' memory load. Our method only requires the user to input words by simply tilting the handle.

2.3. Layouts with multi-letter keys

The multi-letter key layout design is usually used to solve the problem of limited space. For example, devices with small screens such as smartwatches. The layouts to solve this problem are circular layout and T9 layout. Few types of research apply a circular layout for text entry and menu selection. Arranging letters or buttons on a ring by a specific order is called a circular layout. Zhao et al. (2007) proposed earPod, a touch-based auditory menu technique, which used a circle layout. And it proved that when compared with the iPod-like visual linear menu, earPod performed better. TUP Proschowsky et al. (2006) was early research about text entry using a circular layout. Inspired by shorthand gesture recognizers, Mankoff presented a new system using a circle keyboard for pen-input called Currin Mankoff and Abowd (1998). However, that was difficult to learn for users. Uta proposed a BubbleType Hinrichs et al. (2008) technique by typing on the tabletop which designed a BubbleCircle layout using a circle keyboard. Anke proposed the pEYEs Huckauf and Urbina (2008) technique which made two selections to input the intended word. Yu et al. (2018) used a circle layout for text entry in VR. Novice users achieved an average of 8.59 WPM, while expert users achieved 15.85 WPM. Yi et al. (2017) proposed a compass technique that allowed users to enter text on a smartwatch using a circle layout keyboard. Wong et al. (2018) achieved One-handed T9 layout input using a smartwatch, and its speed reached 5.42 WPM. As mentioned above, we find that the circle layout is more suitable for text input on screens with a small size or circle shape. The circular keyboard can arrange the distribution of letters reasonably but requires a lot of learning costs.

QWERTY keyboard layout is the most common typing layout in a century. Beginners prefer to use the QWERTY layout compared to other layouts. Once users learn QWERTY layout typing, it is hard for them to forget. Some researchers want to make some changes to the original QWERTY layout, hoping to achieve better results. For example, MacKenzie (1992) proposed the half-QWERTY keyboard which divided the keyboard into two parts. Every single part was operated by a single hand. Liebowitz and Margolis (1990) studied and compared a variety of keyboard layouts with QWERTY layout and concluded that other layouts did not perform well. They studied the history of the QWERTY layout and concluded that it was difficult to change the habit of using the QWERTY layout. In this way, novice users could master this technique. For multi-letter key inputthere are many candidate characters should be chosen as a final result. Therefore, a reasonable word disambiguation algorithm can effectively improve input performance. Variants of word disambiguation algorithms have been proposed Çetiner et al. (2021); Gogoi et al. (2021); Kaddoura and Ahmed (2022) and they reduced segment error, diacritic error and word error rates.

3. Climbing keyboard

In order to improve input efficiency and user satisfaction, a climbing keyboard is proposed. A keyboard layout based on QWERTY enables text entry by tilting the controller of two HTC controllers. Because no other equipment is used, this kind of design allows text input not to destroy the user's immersion. Typing by tilting controllers is like a climber climbing a mountain, so we call it CLIMBING KEYBOARD. In the following, we will show the details of this method. First, Keyboard layout is described and the reason why we choose this layout is given. Secondly, a word disambiguation method is given to handle the problem of multiple candidate words and tries to decrease the workload of user. This part is very important for the user satisfaction. Finally, although word disambiguation can automatically choose the suitable candidate, the correct rate is not 100%. So the final selection method is discussed and designed.

3.1. Keyboard layout

In daily text input, most people use the QWERTY keyboard layout on the computer or mobile phone. Most people do not like to learn a new layout, so according to the principle of easy learning. We chose to use the QWERTY keyboard layout in this technology. As shown in Figure 1, we arrange the 26 letters according to the QWERTY layout and divide the QWERTY layout into two parts, corresponding to the left hand and the right hand. Three rows of letters can be

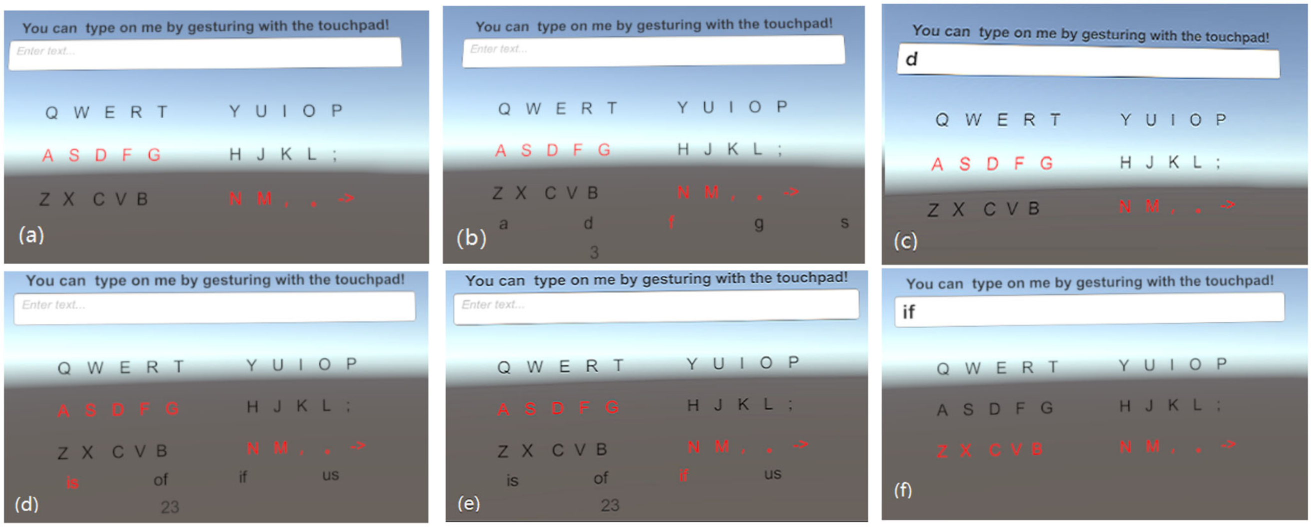

Figure 2. The screen shows interaction after YUIOP has been entered and working through the steps (d–f). (a) tilts the handle to choose a part. (b) Candidate letters appear after pressing touch-pad. (c) Enter the intended letter after pressing the trigger. (d–f) is the process of inputting words.

selected by each hand. For example, the first row corresponding to the left hand contains "Q W E R T," the second row contains "A S D F G," and the third row contains "Z X C V B"…. We divide the angle of inclination into three degrees corresponding to three rows of letters. Take the left hand as an example. Tilting the handle up means that the line "Q W E R T" is selected, and tilting the handle down means that the line "Z X C V B" is selected. Keeping the handle horizontal means the line "A S D F G" is selected. According to the letter sequence of the word, select and confirm the corresponding part. After the entry sequence of the intended word, five candidate words will appear under the QWERTY keyboard. Finally, we touch different parts of the touch-pad to select the intended word.

3.2. Selection

Letter selection: When the user tilts the handle to a suitable position, the corresponding part of the climbing keyboard will turn red. When we press the touch pad with our thumb, the candidate letters will appear below the keyboard. When the thumb moves left or right on the touchpads, the different letters will turn red (Figure 2(b)). We only need to press the trigger key when the intended letter turns red, and the letter will be entered into the input box (Figure 2(c)). There will be no spaces after the input of letters because the input of non-dictionary words is taken into account. Especially, If space is needed, the user can press the left trigger to input a "Space."

Word selection: When a word sequence is an input by tilting the handles, five candidate words will be listed under the keyboard. The five candidate words are ranked by frequency of use. Swipe left or right on the touchpad with the user's thumb. Different words will turn red. (Figure 2(d)) When the word we want to enter turns red, press the trigger of the left-hand key, and the word will be entered in the input box (Figure 2(f)). Then our algorithm detects that a word has been entered, and space will be automatically added after the word. Our algorithm also provides an implementation of auto-complete by using data structure trie.

Delete: If the sequence of a word is entered incorrectly, we can use the trigger on the right handle to delete the sequence just entered one by one. Or we can use the menu button of the left handle to delete the sequence just entered. When the input sequence is deleted, press the trigger button on the right handle again, and the words in the input box will be deleted one by one. Or we can press the Grip button on the left handle to clear the input box.

Confirm: When the user finishes inputting, just press the grip button on the right handle to complete the text submission. In actual application scenarios, this button can switch between typing and VR interaction.

3.3. Word disambiguation

Each character group contains multiple characters, for example, a user selects a row, he may want to enter a letter from A, S, D, F, G. Entry sequence is input by titling handles. There are several different potential input results based on entry sequence. In order to ensure the correctness of word input, it is necessary to use a word disambiguation. We use our algorithm to encode each word. For example, the word "entry" is composed of letters "e, n, t, r, y." we only need to enter the group number of each letter in the word. like "1,6,1,1,2." In order to implement this algorithm, we have to encode a large number of words. The implementation of our algorithm is mainly based on ANC (American National Corpus) Ide and Macleod (2001) which contains over 22,000,000 words of written and spoken American English. We choose the most frequently used top 5000 words in ANC as our corpus. After we encode these words, we will get a hash table of these words. When we enter a sequence, we get candidate words that satisfy this sequence. The result shows that as the number of candidate words increases, the number of words that our algorithm cover is increasing, as shown in Figure 3. We list five candidate words to cover 98.14% of the words. If the number of candidate words is up to six, the word coverage rate can only increase by 0.84%, but the user needs to choose among the six candidate words. The more candidate words, the more time user may spend.

4. The first study: Tilt or no-tilt

In order to evaluate the effectiveness of the tilt interaction for typing, we designed two interaction methods, namely NTI (no-tilt inputting), TI (tilt inputting). Both methods used word selection with the same disambiguation engine.

4.1. Layouts implementation

4.1.1. Non-tilt inputting (NTI)

We divide the QWERTY keyboard into 6 parts which contain 5 characters, and each part is corresponding to a slice of the touch-pad on the handle (see Figure 4). The interaction is as follows.

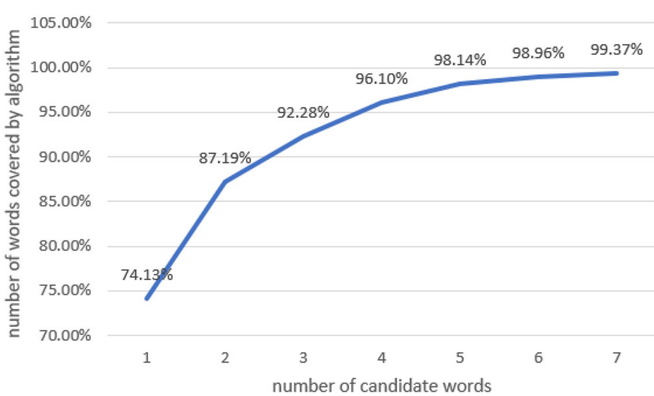

Figure 3. With the increase of candidate words, the algorithm can cover the proportion of ANC words.

According to the ordinate position on the touchpad that the user touches, the corresponding part is highlighted. Users select the intended part by pressing the touchpad down. The word disambiguation algorithm will calculate the candidate words that meet the conditions based on the input sequence, and including autocomplete of current word. According to the abscissa of the touchpad of the user touching the handle, the candidate word is highlighted. Press the trigger key to enter candidate words.

4.1.2. Tilt inputting (TI)

The layout of handles is the same as the NTI layout. Compared with the NTI, the interaction method is different, using a tilt handle instead of fingers touching different positions of the touchpad. For the specific interaction process, see the Section 3.3.

4.2. Hypotheses

We formulate three hypotheses for the experiment:

H.1. In terms of typing speed, the use of touch selection on the touch-pad is better than tilted interaction. The touch action on the touchpad is smaller than the tilt handle.

H.2. In terms of accuracy, the use of tilted interaction is better than touch selection on the touch-pad. Setting three key positions on a touchpad with a diameter of 4cm is too close, and accidental touches are inevitable.

4.3. Participants

A total of 12 undergraduates and postgraduates from a local university were recruited for the experiment (7 males; 5 females). All participants with a mean age of

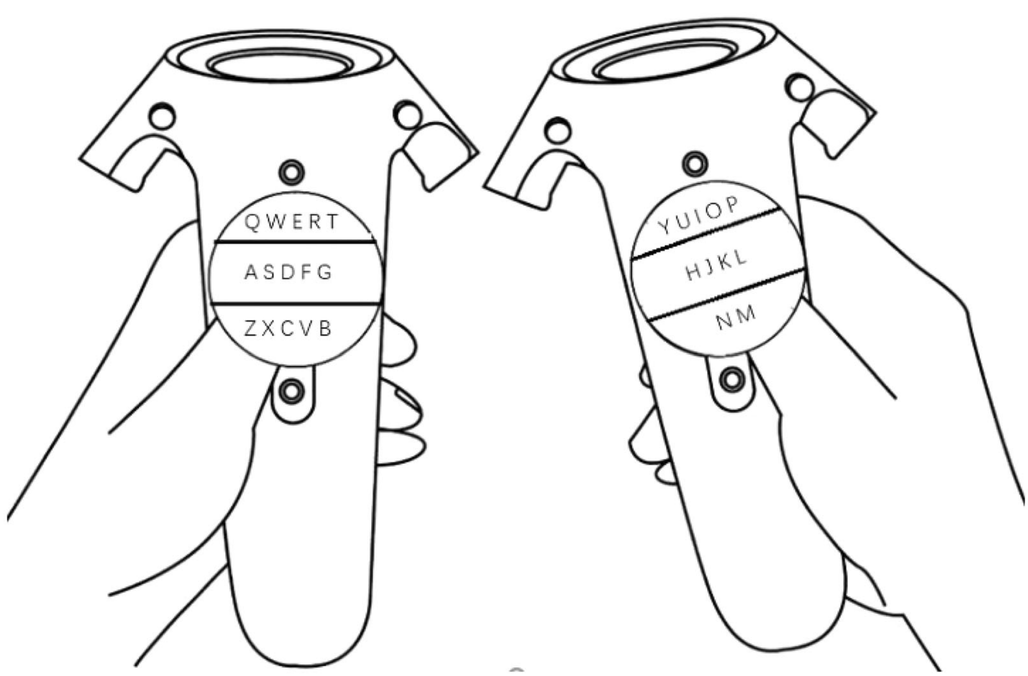

Figure 4. Handle key distribution of NTI layout.

24 (range: 19–26, SD = 2.53) had normal vision or corrective lenses and were right-handed.

4.4. Apparatus

The experiment used the following equipment: An Intel Core i7 processor PC with a dedicated NVIDIA GTX 1080TI graphics card. The program was developed in C#. NET and run on the Unity3D platform. A VR head show (HTC Vive, HTC Inc.), which includes a VR HMD helmet, two laser positioners, and brackets, was used for building and displaying VR environments.

4.5. Experiments design and procedure

The study used a within-subject design with one factors: interactions (touch and tilt). For the whole experiment, we gathered 2(interaction) × 10(phrase) × 12(participant) × 3(block)=720 timed trails. The experiment lasted approximately 1 hour for one participant. After each trial, subjects had a possibility to have a break if they felt tired. The experiment meets the Latin square design. Each participant was first introduced to word typing and need to be trained by completing some learning tasks before the formal experiment. Tasks include entering words and sentences using different layouts and interactions based on prompt. If they can complete the tasks of training without difficulty, they can conduct formal experiments. In the formal experiment, participants need to enter the phrase given above the input box. The phrases used in experiments were all from the phrases provided by MacKenzie and Soukoreff (2003). Participants were told in advance to entry the phrases as soon as possible. They were asked to correct the error if they notice the error immediately but ignore the error if they type the next word. Participants can rest for 5 minutes after completing 10 phrases. During the experiment, the time accuracy and the number of backspaces used by participants will be recorded. After the experiment is over, participants need to complete a NASA-TLX questionnaire, and answer their feelings about different interactions and make some suggestions about improving the way of interaction. We will record all experimental data during the experiment for subsequent experimental analysis.

4.6. Results

WPM (words per minute) is used to measure participant typing speed, formally:

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

where S is the time (in seconds) from the first to the last key press and |T| is the number of characters in the transcribed text.

The error rate is reported based on total error rate (TER) and not corrected error rate (NCER) Levenshtein (1966)

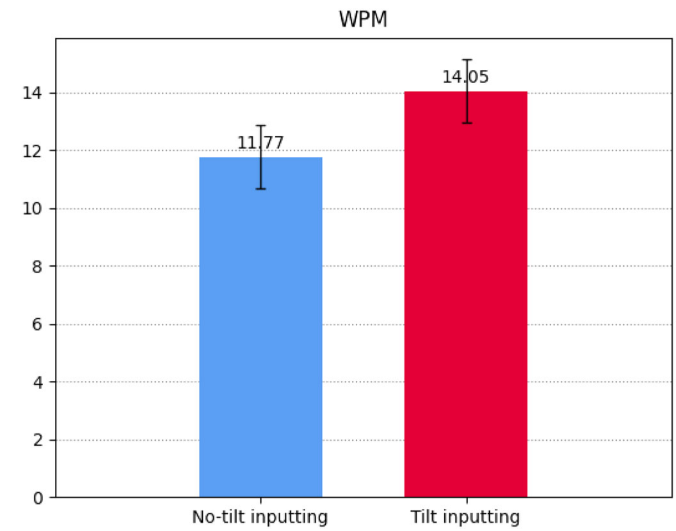

Figure 5. Demonstrates the performance of different technologies at typing speed. Error bars indicate 95% confidence intervals.

Uncorrected error rate is used to measure participant typing accuracy, formally:

$$\text{NCER} = \frac{\text{MSD}(\text{P}, \text{T})}{\max(\text{P}, \text{T})} \times 100\%$$

where P is the present phrase, T is the transcribed phrase. MSD is a method using Levenshtein's algorithm Levenshtein (1966). Max is a method to calculate the maximum number of characters for two strings.

Total error rate is used to measure participant typing accuracy, formally:

$$\text{TER} = \frac{\text{B}}{\text{P} + \text{B}} \times 100\% + \frac{\text{MSD}(\text{P}, \text{T})}{\max(\text{P}, \text{T})} \times 100\%$$

where P is the lengthy of present phrase; and B is the number of backspaces used. T is the transcribed phrase.

4.6.1. Text entry speed

There were significant effects of Interaction ($F1,22 = 7.022$, $p = 0.015$) on WPM. We can see from Figure 5 that using the tilt method was significantly faster than using the selection method. The average input speed of users using Tilt inputting can reach 14.05 WPM. But users can only reach 11.77 WPM with No-tilt Inputting.

4.6.2. Error rate

There were no significant effects of Interaction ($F1,22 = 0.171$, $p = 0.68$) on not corrected error rate. Furthermore, no significant interaction effect on Interaction ($F1,22 = 0.651$, $p = 0.426$) was found. We can see from Figure 6 that the two interaction methods have little difference in error rate.

4.6.3. Workload

As shown in Figure 7, participants rated both interactions after the experiment. The full score is 5, the higher the score, the worse the performance. It is particularly important to note that the Temporal demand here are not obtained

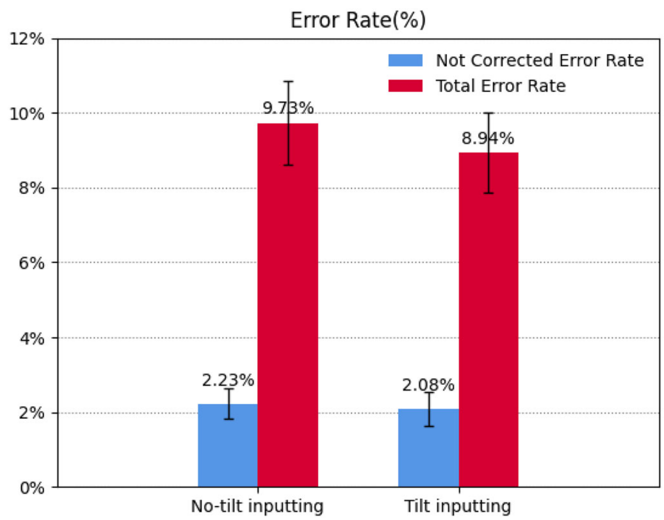

Figure 6. Demonstrates the performance of different technologies at the error rate. Error bars indicate 95% confidence intervals.

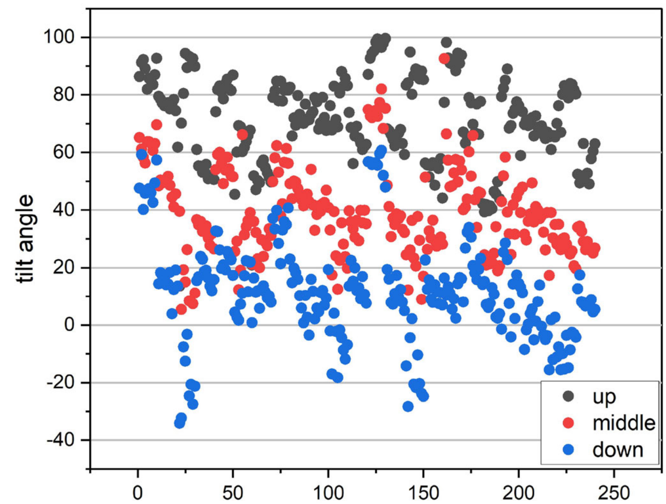

Figure 8. The points recorded by the participant by tilting the both handles.

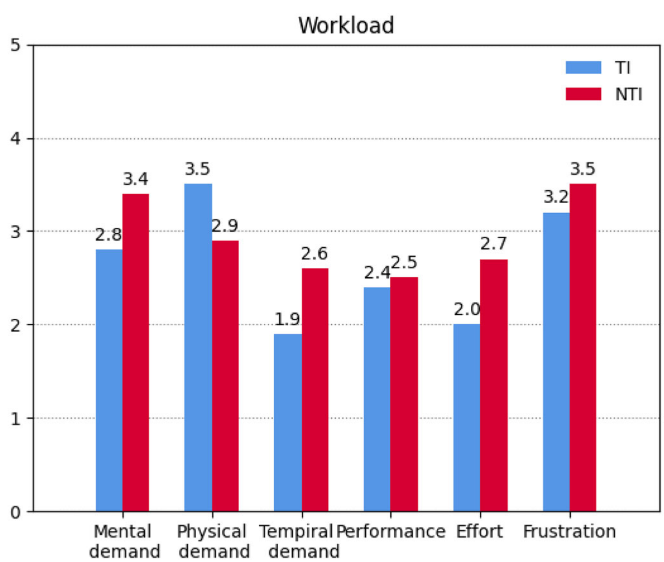

Figure 7. Demonstrates the performance of different technologies at the error rate.

through the recorded time and accuracy, but the subjective feelings of the experimental participants. It can be seen from the results that NTI is not as good as TI in terms of mental fatigue. According to the participants in the experiment, NTI used the touchpad to select blocks, and the selection of candidate words also used the touchpad to confuse them. While TI does not perform as well as NTI on physical fatigue, we will adjust the tilt angle below.

4.7. Discussion

The results support **H.1.** In terms of typing speed, the use of tilted interaction is better than touch selection on the touch-pad. For the **H.2.**, the results of ANOVA do not show there are significant between the two groups. This result shows that our word disambiguation algorithm is stable. According to subjective evaluation, TI is better than NTI.

Moreover, according to the description of the participants, the current setting of the tilt angle is the cause of the error. And the tilt range is relatively large, not only need to rotate the wrist, sometimes need to swing the arms. Therefore, changing the tilt angle may benefit typing performance. It requires a further study.

5. The second study: Tilt angle

When using tilt for text input, the size of the angle range is particularly important. When the setting angle range is too large, the user needs to swing his arm to increase fatigue. When the set angle is too small, the error rate will increase. The purpose of this experiment is to find a reasonable range of angles.

5.1. Pilot study

We conducted a small, follow-up experiment (with 12 participants from the first study) to find a comfortable tilt setting. We asked twelve experiment participants to tilt the handle up 10 times and tilt the handle down 10 times and keep the handle in a horizontal position 10 times with their left hands. Afterward, the participants did this task again with their right hands. Press the trigger button when the user thinks the tilt is at a suitable angle. The tilt angle of the handle will be recorded. Figure 8 records the tilt handle angle of both hands. Figure 9(a) records the tilt angle of the left hand, and Figure 9(b) records the tilt angle of the right hand. We divide the tilt range of the handle into three angles as "up," "middle" and" "down." We calculate the average value for each range based on the data collected above. From Table 1, we can see that 60°–90° for up, 30°–60° for middle, and 0°–30° for down is not the best comfortable layout. And we can find that the tilt angle of the left hand is lower than the tilt angle of the right hand. And judging from the degree of dispersion of the data, the left hand has a greater range of activity, which has a clear dividing line compared with the right hand.

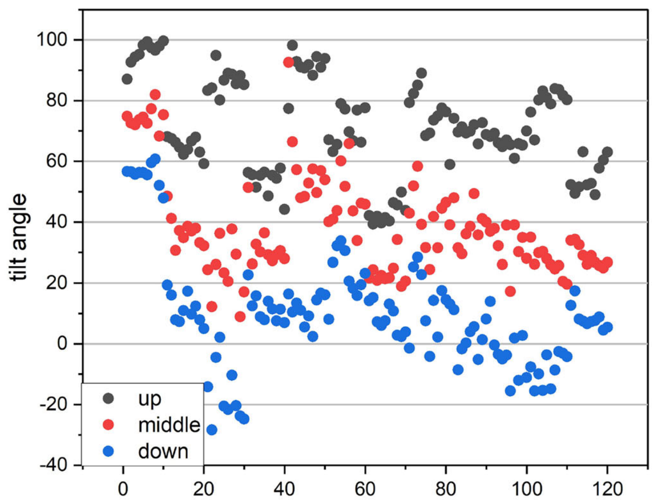

Figure 9. (a) The points recorded by the participant by tilting the left handles. (b) The points recorded by the participant by tilting the right handles.

Table 1. the average value for each range

	Up	Middle	Down
dual hands	72.0°	39.5°	12.2°
left hand	71.6°	38.8°	9.9°
right hand	72.5°	39.8°	14.5°

Table 2. Angle range of three layouts

	Up	Middle	Down
Bisect 90°	>60°	30°–60°	<30°
both hands average angle	>56°	26°–56°	<26°
left angle	>55°	24°–55°	<24°
right angle	>56°	27°–56°	<27°

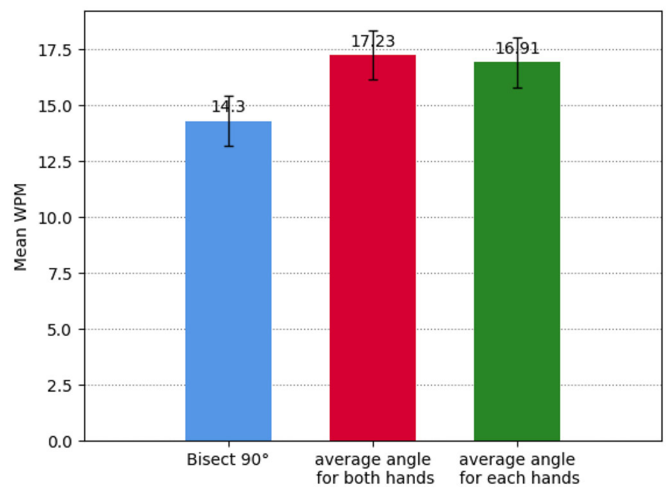

Figure 10. Demonstrates the performance of different angle ranges at typing speed.

5.2. Layout optimized

Just like the conclusions drawn from the preliminary experiment, the left and right hands may have different control over tilt. The tilt angle for the left and right hands may also be different. We wanted to find a general conclusion for this property to enhance its usability for most users. Thus, the goal of this study was to investigate the performance of different angle range of tilt. We have designed three angle ranges (Bisect 90°, both hands average angle, the left and right-hand angles were calculated separately). This experiment is dedicated to finding the best angle. The results were shown in Table 2.

5.3. Hypotheses

We formulate two hypotheses for the experiment:

H.3. We believe that the angle range obtained from the scatter plot obtained by the user tilting the wrist is better than dividing 90° into three levels equally.

H.4. We believe that the comfort angle of the right hand is higher than the comfort angle of the left hand. Therefore, we think it is better to calculate the angle range of both hands separately than to calculate the angle range of both hands together.

5.4. Experiment design and procedure

The experiment requires each participant to do three input tests with different angle ranges. And this time the technology added Word disambiguation. The required input phrases are from the phrase test set in the first experiment. Participants in the experiment were allowed to practice as long as they wanted. The test task needs to be repeated in 3 groups of 10 sentences each. The experiment meets the Latin square design. Each experimental participant will practice the system set up from different angles. Each set of practice time is up to them. The whole experiment lasted approximately 40 minutes for each participant. Between two layouts, participants could have a break.

5.5. Result

5.5.1. Text entry speed

Figure 10 showed the text entry speed of different angle ranges. The speed of angle range 2 was higher than the others. We used one-way ANOVA analysis with Bonferroni post-hoc test, which allowed us to discover which specific

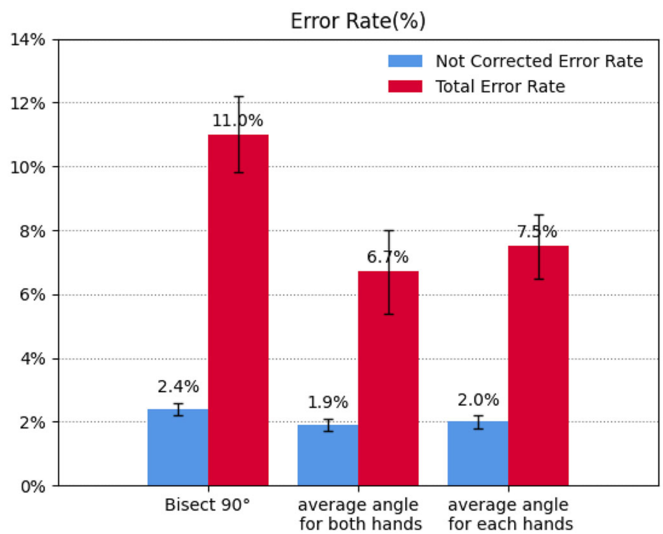

Figure 11. Demonstrates the performance of different angle ranges at an error rate.

means differed. The results revealed that there were significant effects of angle ranges (F 2,33 = 4.642, $p = 0.017$) on WPM. The significance between range 1 and range 2 ($p = 0.013$), range 1 and range 3($p = 0.012$) was found. But the difference between range 2 and range 3($p = 0.968$) was not significant.

5.5.2. Error rate

13 shows the text entry error rate of different angle ranges. No significant differences were found between any ranges. And all NCER is not high. But for TER, there was significant effects of angle ranges (F 2,33 = 3.568, $p = 0.039$). The difference between range 1 and range 2 ($p = 0.024$) was found. And significant differences between range 1 and range 3($p = 0.030$) were found too. No significant differences were found between range 2 and range 3($p = 0.929$). The error rate of range 1 is higher than the other two groups.

5.6. Discussion

According to the experimental results, we can see that Hypothesis 3 is correct. It is unreasonable to divide 90° into three levels equally. The tilt angle we designed according to the user's habit of tilting the wrist is more reasonable. It can achieve faster speed and less error rate. According to what a participant said after the experiment "To satisfy typing accuracy and wrist comfort. I must raise my forearm to tilt in Range 1. And other angle ranges don't need." For Hypothesis 4, there is no significant difference between Range 2 and Range 3. No matter that the participant did Range 2 first or Range 3 first. Most users did not notice the change in angle. Therefore, we believe that only using the tilt angle change within 3° will not affect the performance. We choose Range 2 with the best performance in Figures 10 and 11 as the angle range of the Climbing keyboard.

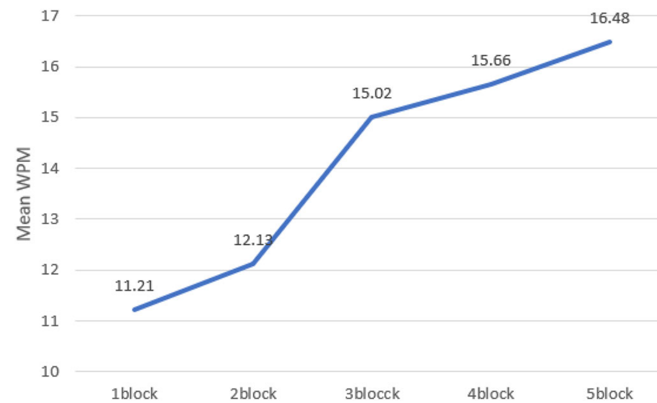

Figure 12. Text entry speed over 5 blocks.

6. The third study: Performance evaluation

We conducted a five-block user study to evaluate the performance of Climbing Keyboard (the best performing design found in the second study). The goal of the experiment is to measure the learning cost of Climbing Keyboard and how well users could perform text entry using Climbing keyboard. We were also interested in how the performance would be improved after a period of practice.

6.1. Participants and apparatus

Ten right-handed participants (8 males; 2 females) between the ages of 22–25 ($M = 23.5$) were recruited from our university campus to participate in this study. All participants input using the right hand in the experiment. The apparatus and devices in this study were the same as in the first study.

6.2. Experiment design and procedure

To evaluate the learnability, there were no participants from the second study. Before the experiment, participants were told how to input by using the Climbing keyboard. The whole experiment was divided into five blocks. In each block, participants were asked to enter 10 phrases as accurately and quickly as possible. Between the two blocks, participants can take a break. We asked participants to type words as accurately and fast as possible. The whole experiment lasted approximately 40 minutes for each participant. In all, we collected 10(participant) $\times$ 5(block) $\times$ 10(phrase) $=$ 500 phrases.

6.3. Result

We employed an ANOVA with Sessions (from block one to block five) as the within-subject.

6.3.1. Text entry speed

Figure 12 shows the change in the typing speed of participants over five blocks. We can see that the average speed of 11.21wpm can be achieved for the first block using this technology, and after five blocks of practice, the speed can reach 16.48wpm. The results of ANOVA show us the

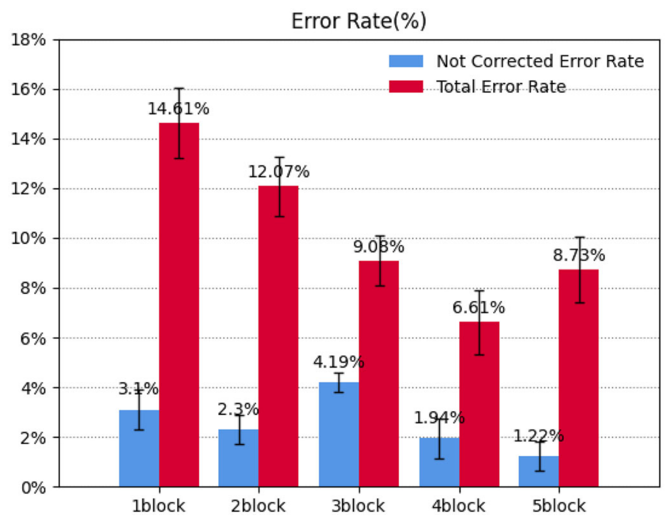

Figure 13. Mean total error rate (TER) and not corrected error rate (NCER) over 5 blocks.

significant differences between blocks (F 4.538 = 23.95, $p = 0.000$). Meanwhile, the result also revealed significant differences between Block 2 and Block 3($p = 0.000$). No significant differences were found between Block 3 and Block 4 ($p = 0.358$), nor between Block 4 and Block 5 ($p = 0.232$). However, significant differences were found between Block 3 and Block 5 ($p = 0.035$). This trend shows that the learning curve was getting more and more stable after each block.

6.3.2. Total and uncorrected error rate

Figure 13 shows the NCER and TER changed over five blocks. For NCER, there were no significant differences between any blocks (F 4.538 = 0.718, $p = 0.58$). For TER, there were significant differences between blocks (F 4.538 = 2.807, $p = 0.025$). There were no significant differences between Block 1 and Block 2, Block 2, and Block 3, Block 3 and Block 4, Block 4 and Block 5. However, Significant differences were found between Block 1 and Block 3, Block 1, and Block 4, Block 1and Block 5. It can be seen from Figure 13 that the error rate is declining. This trend combined with the ANOVA results can show that the participants are making steady progress.

6.4. Discussion

With 40 mins of typing phrases, participants reached speeds exceed 16 WPM, with a NCER near 2%. This outperforms the existing typing techniques in VR using handles. Table 3 shows the comparison of existing technology focused on text-entry by controller with Climbing Keyboard in terms of speed and accuracy. For each technology, we take the average of novices and experts. Climbing keyboard speed is higher than most input methods, and NCER is at a medium level. In terms of learnability, we believe Climbing Keyboard is easy-to-learn according to the learning curves. After entering 40 phrases, participants' typing speed can be increased to 16.48WPM. At the same time, the error rate is reduced to 1.22%. Compared with the method of inputting in letters Speicher et al. (2018), inputting in words is faster. Same as the method of inputting words as units Jiang and Weng (2020), QWERTY's layout is more reasonable and suitable for beginners to learn. The method of using Pointing as the input method Speicher et al. (2018) is fast but feels tired when typing for a long time. This table is only for meta comparison in different surveys, the degree of formality is far less than in Direct comparative study. We think that the climbing keyboard performs well mainly because the climbing keyboard uses two hands to operate. Two-hand interaction technology, relative to one-hand interaction technology, can improve text input performance and accuracy to some extent Boletsis and Kongsvik (2019), and is more convenient when performing complex and bimanually demanding tasks. One-handed input is only used when both hands are not free. However, given the standard indicators of WPM and NCER, we think this is a reasonable way to correlate Climbing's keyboard performance.

Table 3. the average value for each range

Technique	WPM	NCER%
Climbing Keyboard	16.48	1.22
Hipad Jiang and Weng (2020)	11.41	0.14
Controller Pointing Speicher et al. (2018)	15.4	1

7. Future work

There are some limitations, although this work has successfully produced a novel text entry technique in VR. For example, after using the tilt handle to input the word sequence, the way to select candidate words is not convenient. The problem is that the input speed of words out of the dictionary is slow. In the future, we will improve this technique from two aspects as follows:

Candidate words: There are only five candidate words listed, and the order of the candidate words is determined by their frequency of use. Increasing the time of use, will play a big role, and we will enrich our corpus.

Special symbols: For a complete input method, in addition to input letters and words, it is also possible to input special characters and numbers. In future work, we will consider completing.

Dominant hand: In this study, all users were right-handed. In future work, we will ask left-handed users to join the study and evaluate the system.

We will put the Climbing keyboard on the App Store if possible. With the user's permission, collect individual user's tilt preferences and make adaptive adjustments.

8. Conclusion

In this article, we introduced a *Climbing keyboard*, a text input technology for virtual reality without the help of other devices. This technology allows users to tilt the handle and touch the touchpad on the handle to enter the intended letters. Its design follows three design standards: fast input speed, high accuracy, and easy to learn. We conducted two user studies to determine the best design capabilities of the technology before evaluating its performance and usability. The first study compared several potential layouts of the Climbing keyboard to provide a basis for its final design. We performed a second experiment on the tilt angle of the handle. Through the experiments conducted by 12 participants, the final layout of the Climbing keyboard was further evaluated through tilted interaction and linear layout. The third study shows that novice users can reach 16.48 words/minute (WPM) in one and a half hours' training, while experts Users can reach 18.53 WPM. The fastest speed of Pizzatext D. Yu et al. (2018) is 15.86 WPM. An additional device dual thumb sticks of a hand-held game controller is needed, while Climbing keyboard can be implemented with the default VR controller. We believe that the Climbing keyboard is an interaction metaphor with excellent user experience (high-speed, low error rate, and easy to learn) for many VR applications.

Acknowledgements

We thank the participants in the user study for their availability and significant comments.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This study has been partially supported by National Natural Science Foundation of China [61872164] and Program of Science and Technology Development Plan of Jilin Province of China [20220201147GX].

ORCID

Junfeng Huang http://orcid.org/0000-0003-1829-4193
Minghui Sun http://orcid.org/0000-0002-1809-8187

References

Adhikary, J., & Vertanen, K. (2021). Text entry in virtual environments using speech and a midair keyboard. *IEEE Transactions on Visualization and Computer Graphics*, *27*(5), 2648–2658. https://doi.org/10.1109/TVCG.2021.3067776

Azenkot, S., & Zhai, S. (2012). Touch behavior with different postures on soft smartphone keyboards. In *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services* (pp. 251–260). https://doi.org/10.1145/2371574.2371612

Boletsis, C., & Kongsvik, S. (2019). Text input in virtual reality: A preliminary evaluation of the drum-like vr keyboard. *Technologies*, *7*(2), 31. https://doi.org/10.3390/technologies7020031

Bowman, D., Wingrave, C., Campbell, J., & Ly, V. (2001). Using pinch gloves (TM) for both natural and abstract interaction techniques in virtual environments. Technical Report TR-01-23, Computer Science, Virginia Tech.

Çetiner, M. Y., Ild Ir Im, A., Onay, B., & Öksüz, C. (2021). Word sense disambiguation using kenet. In *2021 29th Signal Processing and Communications Applications Conference (SIU)* (pp. 1–4). https://doi.org/10.1109/SIU53274.2021.9477816

Chang, Y., L'Yi, S., Koh, K., Seo, J. (2015). Understanding users' touch behavior on large mobile touch-screens and assisted targeting by tilting gesture. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (pp. 1499–1508).

Gogoi, A., Baruah, N., & Nath, L. J. (2021). Assamese word sense disambiguation using cuckoo search algorithm. *Procedia Computer Science*, *189*, 142–147. https://doi.org/10.1016/j.procs.2021.05.110

González, G., Molina, J. P., García, A. S., Martínez, D., & González, P. (2009). Evaluation of text input techniques in immersive virtual environments. In *New Trends on Human–Computer Interaction* (pp. 109–118). Springer.

Gupta, A., Ji, C., Yeo, H.-S., Quigley, A., & Vogel, D. (2019). Rotoswype: Word-gesture typing using a ring. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (pp. 1–12).

Hinckley, K., & Song, H. (2011). Sensor synaesthesia: Touch in motion, and motion in touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 801–810).

Hinrichs, U., Schmidt, H., Isenberg, T., Hancock, M. S., & Carpendale, S. (2008). *Bubbletype: Enabling text entry within a walk-up tabletop installation.* Research Report 2008-893-06, Department of Computer Science, University of Calgary.

Hoste, L., Dumas, B., & Signer, B. (2012). Speeg: A multimodal speech- and gesture-based text input solution. In *Proceedings of the International Working Conference on Advanced Visual Interfaces* (pp. 156–163).

Huckauf, A., & Urbina, M. H. (2008). Gazing with peyes: Towards a universal input for various applications. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications* (pp. 51–54).

Ide, N., Macleod, C. (2001). The american national corpus: A standardized resource of american english. In *Proceedings of Corpus Linguistics* (Vol. 3, pp. 1–7).

Jiang, H., & Weng, D. (2020). Hipad: Text entry for head-mounted displays using circular touchpad. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (pp. 692–703). https://doi.org/10.1109/VR46266.2020.00092

Jimenez, J. G. (2017). *A prototype for text input in virtual reality with a swype-like process using a hand-tracking device.* [Unpublished doctoral dissertation].

Jones, E., Alexander, J., Andreou, A., Irani, P., & Subramanian, S. (2010). Gestext: Accelerometer-based gestural text-entry systems. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 2173–2182).

Kaddoura, S. D. Ahmed, R. (2022). A comprehensive review on arabic word sense disambiguation for natural language processing applications. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, e1447.

Kim, Y. R., & Kim, G. J. (2017). Hovr-type: Smartphone as a typing interface in vr using hovering. In *2017 IEEE International Conference on Consumer Electronics (ICCE)* (pp. 200–203). https://doi.org/10.1109/ICCE.2017.7889285

Knierim, P., Schwind, V., Feit, A. M., Nieuwenhuizen, F., & Henze, N. (2018). 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* (pp. 1–9).

Lee, K. C., & Chung, N. (2008). Empirical analysis of consumer reaction to the virtual reality shopping mall. *Computers in Human Behavior*, *24*(1), 88–104. https://doi.org/10.1016/j.chb.2007.01.018

Lee, Y., Kim, G. J. (2017). Vitty: Virtual touch typing interface with added finger buttons. In *International Conference on Virtual, Augmented and Mixed Reality* (pp. 111–119).

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *In Soviet Physics Doklady*, *10*, 707–710.

Liebowitz, S. J., & Margolis, S. E. (1990). The fable of the keys. *The Journal of Law and Economics*, *33*(1), 1–25. https://doi.org/10.1086/467198

Lin, J.-W., Han, P.-H., Lee, J.-Y., Chen, Y.-S., Chang, T.-W., & Chen, K.-W. (2017). Visualizing the keyboard in virtual reality for enhancing immersive experience. In *ACM SIGGRAPH 2017 Posters* (pp. 1–2). https://doi.org/10.1145/3102163.3102175

MacKenzie, I. S. (1992). Fitts' law as a research and design tool in human-computer interaction. *Human–Computer Interaction*, *7*(1), 91–139. https://doi.org/10.1207/s15327051hci0701_3
MacKenzie, I. S. (2002). Kspc (keystrokes per character) as a characteristic of text entry techniques. In *International Conference on Mobile Human-Computer Interaction* (pp. 195–210).
MacKenzie, I. S., & Soukoreff, R. W. (2003). Phrase sets for evaluating text entry techniques. In *CHI'03 Extended Abstracts on Human Factors in Computing Systems* (pp. 754–755). https://doi.org/10.1145/765891.765971
Majaranta, P., & Räihä, K.-J. (2007). Text entry by gaze: Utilizing eye-tracking. *Text Entry Systems: Mobility, Accessibility, Universality*, 175–187. https://doi.org/10.1145/3313831.3376317
Mankoff, J., & Abowd, G. D. (1998). 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).
McGill, M., Boland, D., Murray-Smith, R., & Brewster, S. (2015). 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* (pp. 2143–2152).
Olofsson, J. (2017). *Input and display of text for virtual reality head-mounted displays and hand-held positionally tracked controllers* (Vol. 15). Lulea University of Technology, Department of Computer Science, Electrical and Space Engineering.
Partridge, K., Chatterjee, S., Sazawal, V., Borriello, G., & Want, R. (2002). Tilttype: Accelerometer-supported text entry for very small devices. In *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology* (pp. 201–204).
Pick, S., Puika, A. S., & Kuhlen, T. W. (2016). Swifter: Design and evaluation of a speech-based text input metaphor for immersive virtual environments. In *2016 IEEE Symposium on 3D User Interfaces (3DUI)* (pp. 109–112). https://doi.org/10.1109/3DUI.2016.7460039
Proschowsky, M., Schultz, N., & Jacobsen, N. E. (2006). An intuitive text input method for touch wheels. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 467–470).
Schneider, D., Otte, A., Gesslein, T., Gagel, P., Kuth, B., Damlakhi, M. S., Dietz, O., Ofek, E., Pahud, M., Kristensson, P. O., Muller, J., & Grubert, J. (2019). Reconviguration: Reconfiguring physical keyboards in virtual reality. *IEEE Transactions on Visualization and Computer Graphics*, *25*(11), 3190–3201. https://doi.org/10.1109/TVCG.2019.2932239
Speicher, M., Feit, A. M., Ziegler, P., & Krüger, A. (2018). Selection-based text entry in virtual reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (pp. 1–13).
Walker, J., Li, B., Vertanen, K., & Kuhl, S. (2017). Efficient typing on a visually occluded physical keyboard. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (pp. 5457–5461).
Wigdor, D., & Balakrishnan, R. (2003). Tilttext: Using tilt for text input to mobile phones. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology* (pp. 81–90).
Wong, P. C., Zhu, K., & Fu, H. (2018). Fingert9: Leveraging thumb-to-finger interaction for same-side-hand text entry on smartwatches. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (pp. 1–10).
Yeo, H.-S., Phang, X.-S., Castellucci, S. J., Kristensson, P. O., & Quigley, A. (2017). Investigating tilt-based gesture keyboard entry for single-handed text entry on large devices. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (pp. 4194–4202). https://doi.org/10.1145/3025453.3025520
Yi, X., Yu, C., Xu, W., Bi, X., & Shi, Y. (2017). Compass: Rotational keyboard on non-touch smartwatches. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (pp. 705–715).
Yu, C., Gu, Y., Yang, Z., Yi, X., Luo, H., Shi, Y. (2017). 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* (pp. 4479–4488).
Yu, D., Fan, K., Zhang, H., Monteiro, D., Xu, W., & Liang, H.-N. (2018). Pizzatext: Text entry for virtual reality systems using dual thumbsticks. *IEEE Transactions on Visualization and Computer Graphics*, *24*(11), 2927–2935. https://doi.org/10.1109/TVCG.2018.2868581
Zhao, S., Dragicevic, P., Chignell, M., Balakrishnan, R., & Baudisch, P. (2007). Earpod: Eyes-free menu selection using touch input and reactive audio feedback. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 1395–1404).

About the authors

Junfeng Huang is a postgraduate student of the Jilin University on his topic of target selection techniques in Virtual Reality. He has a background in computer science and technique. He wanted to gain expertise in Human Computer Interaction and VR.

Minghui Sun is currently an Associate Professor with the College of Computer Science and Technology, Jilin University, China. He is interested in using HCI methods to solve challenging real-world computing problems in many areas, including tactile interface, pen-based interface, and tangible interface.

Jun Qin is currently a Lecturer of Changchun University of Science and Technology. His main researches focus on image processing and tracking with both deep learning methods and traditional methods, especially in the field of image segmentation and image fusion.

Boyu Gao is currently an Associate Professor of Jinan University. His researches mainly lie on the intersection of Virtual Reality (VR), Human-Computer Interaction (HCI), and Artificial Intelligence (AI), specifically, focusing on designing multimodal feedback, evoking believable experiences in virtual environments and perceptual-based user interfaces.

Guihe Qin is currently a Professor with the College of Computer Science and Technology, Jilin University, China. His research interests include intelligent control and embedded systems, computer vision, and intelligent connected vehicle.