Efficient Flower Text Entry in Virtual Reality

Jiaye Leng, Lili Wang, Xiaolong Liu, Xuehuai Shi, and Miao Wang

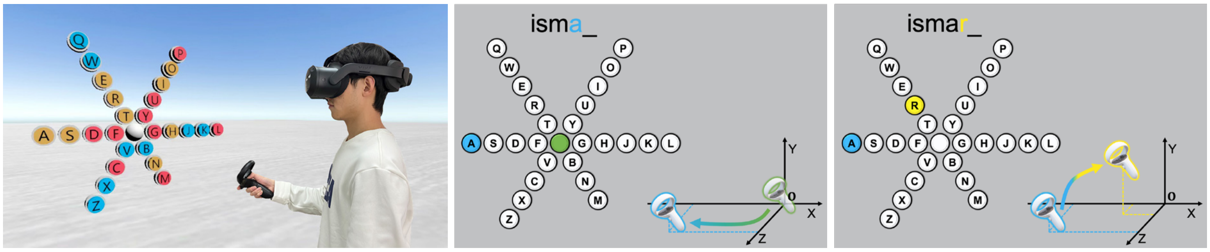


Fig. 1: In the virtual environment, the user holds a controller and applies our Flower Text Entry method to enter text (left). The process of the typing is illustrated in the middle and right images. When the controller is translated from the origin green position to the blue position, the letter 'a' is selected for input (middle); then the controller is translated to the yellow position, and the letter 'r' is selected for input (right).

Abstract— Text entry is a frequently used task in virtual reality (VR) applications, and controller is the most common interactive device in current VR systems. However, in terms of typing speed, there is still a gap between the existing controller-based text entry techniques and using a physical keyboard in reality, so it is important to improve the efficiency of the controller-based text entry. In this paper, we introduce Flower Text Entry, a single-controller text entry method based on a newly designed flower-shaped keyboard using hand 3D translation interaction for letters selection. We conduct user studies to optimize the keyboard design and the mapping between the interaction and selection, so as to evaluate our method. The results show that our method has high typing speed, lower error rate, and is very friendly to novices compared with the state-of-the-art controller-based text entry methods. After a short training, the novice group can type at 17.65 words per minute (WPM), and the potential expert group can type at 22.97 WPM. The highest typing speed is up to 30.80 WPM achieved by a potential expert participant.

Index Terms—Virtual reality, text entry, keyboard layout, hand interaction, controller

◆

1 INTRODUCTION

In many virtual reality (VR) scenarios, text entry is a common and important task, such as the communication between multiple users in collaboration and labeling scene information. Many existing text entry techniques require extra devices, which are complicated and expensive. Controller is currently the most common VR interaction device, and thus it is a potential candidate input device for text entry in the virtual environment (VE).

Many researchers have explored the efficiency, learnability, and usability of controller-based text entry techniques. PizzaText [49] is a dual-hand joystick-based text entry technique with a circular keyboard layout. After two hours of training, expert users can type at a speed of 15.85 words per minute (WPM). Ray-based method [36] involves dual-hand typing on a virtual QWERTY keyboard using the rays emitted from controllers. The drum-like keyboard method [3] is also a dual-hand one, which draws on the idea of ray selection, with a total error rate of 7.2%. Compared with dual-hand, single-hand techniques can liberate one hand to do other interactive tasks. HiPad [15] is a single-hand text entry technique that uses a controller with a touchpad for text input on a circular keyboard. Novices can reach a typing speed of 13.57 WPM after a 60-phrase training. We also focus on the single-hand text entry technique and consider the following aspects of design: high typing speed, low error rate, only using a simple controller with buttons (no joystick or touchpad), higher naturalness and simplicity for users to learn.

In this paper, we propose Flower Text Entry, an efficient text entry technique based on a new flower-shaped keyboard. First, we design a flower-shaped keyboard, which is a hybrid of the QWERTY and circular keyboard, and introduce a 3D hand translation interaction for single-hand text entry. Then we evaluate two types of letter layouts and four possible optimization options, and confirm the final form of the keyboard. After that, we optimize the mapping function of user interaction and letter selection by analyzing users' behavior. At last, we design a 6-day user study to evaluate the performance of our method. The results show that our method is efficient and accurate. The potential experts can achieve an average of 22.97 WPM (s.e.= 1.43) on the sixth day, and the average NCER and TER (NCER means errors left in the transcribed text, TER means all committed errors during typing) over six days are 0.04% (s.e. = 0.04%) and 3.42% (s.e. = 0.36%) respectively. Our method also has good learnability. After a 6-day practice, the typing speed of novices increases by 96.99%, and that of potential experts increases by 23.56%. The highest typing speed is 30.80 WPM, which is achieved by a potential expert. Compared with the state-of-the-art methods, PizzaText and HiPad require controllers with joysticks and touchpad respectively, while our method only uses a simple controller with buttons. Ray-based method and drum-like keyboard require dual hands to input text and easily fatigue the user, while our method only needs a single hand with more natural interaction.

In summary, the contributions of our Flower Text Entry are as follows: 1) Introduce 3D hand translation interaction into the controller-based text entry technique. 2) Design a new flower-shaped keyboard for text entry in VR. 3) Optimize the mapping function of user interaction and letter selection based on users' behavior. 4) Design a study to evaluate the performance of our Flower Text Entry.

- *Lili Wang is with State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China; Peng Cheng Laboratory, Shenzhen, China; and Beijing Advanced Innovation Center for Biomedical Engineering, Beihang University, Beijing, China. Lili Wang is the corresponding author. E-mail: wanglily@buaa.edu.cn.*
- *Jiaye Leng, Xiaolong Liu, Xuehuai Shi and Miao Wang is with State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing, China. E-mail: lengjiaye@buaa.edu.cn, liuxiaolong186@buaa.edu.cn, shixuehuaireal@buaa.edu.cn, miaow@buaa.edu.cn.*

Manuscript received 11 March 2022; revised 11 June 2022; accepted 2 July 2022. Date of publication 01 September 2022; date of current version 03 October 2022. Digital Object Identifier no. 10.1109/TVCG.2022.3203101

2 RELATED WORK

In this section, we review the existing text entry methods in VR, and text entry with circular keyboard.

2.1 Text Entry in VR

Many researchers have explored how to perform text entry efficiently and conveniently in the context of VR applications. The most straightforward way is to plug in a physical keyboard in the VE. Walker et al. [39] found that when people wore HMDs and could not see the keyboard, the speed and accuracy of text entry decreased dramatically. When a virtual keyboard assistant was given [40], the impact of wearing HMDs could be relieved. McGill et al. [22, 28] leveraged the concept of Mixed Reality (MR) to insert real-world typing hands and a keyboard into virtual scenes to improve typing performance in VR. Jiang et al. [16] reported that physical keyboard-based techniques with MR could roughly achieve the same typing efficiency as in the real world. Knierim et al. [20] used retro-reflective markers to track hand motion and visualized avatar hands and a keyboard in VR. These methods achieve fast typing speed, but all require a physical keyboard. They are not suitable for some VR applications with no place to put the keyboard, especially for mobile applications.

In order to solve this problem, Pham et al. [29] proposed HawKey, in which the user wore a physical keyboard on a hawker's tray-like tablet in front of himself. Touchscreen-based techniques perform better in mobile scenes [11, 19, 21, 23, 33]. Gugenheimer et al. [11] installed the touchscreen on the back of HMDs, so users could select the virtual content in their field of view by clicking the corresponding position of the touchscreen. Lu et al. [23] explored using a thumb for eyeless typing on the touchscreen, allowing users to tap an imaginary QWERTY keyboard. However, additional touchscreen devices (e.g. tablets, smartphones, or smartwatches) are required in these methods, which are not easy to integrate into existing VR systems.

Speech-based text entry techniques in VR have received much attention. Bowman et al. [4] compared the speech technique with the pinch keyboard using pinch gloves, the one-hand chord keyboard, and the soft keyboard using pen and tablet, and they found that the speech technique was the fastest. Pick et al. [30] proposed a speech-based multi-mode text entry method SWIFTER, which achieved a typing speed of 23.6 WPM. Adhikary et al. [1] combined speech techniques with a mid-air keyboard to increase the speed to 28 WPM. The speech techniques achieve good text entry performance, but they may encounter problems in some scenarios, such as noise and privacy in a shared environment [34] and error correction [38].

Mid-air typing techniques were also proposed, including glove-based [4, 6], motion-tracking [12, 31, 47], and sensor-based techniques [17, 18, 41, 43, 45]. A challenging issue of mid-air typing arises from the physiology of the hand, where intentional finger movements produce unintentional co-activation in other fingers, which can lead to spurious input events [7]. Moreover, these techniques require expensive extra devices or sensors, which may affect users' interaction with the VE and confine users to the vicinity of the installed sensor's location.

Head-based techniques also have been explored. Yu et al. [48] compared three different head-based techniques (TapType, DwellTpe, and GestureType) and found that the GestureType was the fastest. Speicher et al. [36] evaluated six different text entry methods and reported that head pointing was a viable method. RingText [44] was a dwell-free method and used the virtual cursor controlled by users' head movements for selection. However, frequent head movements are more likely to cause motion sickness in VR [49].

In current VR systems, controllers are interactive devices that have been widely used and familiar to users. Controller-based text entry techniques can be classified into dual-hand techniques and single-hand techniques. The existing dual-hand techniques are as follows. Inspired by real-world laser pointers, ray-based techniques [36] use rays generated from the position of the user's two controllers as the starting points and in the directions of the controllers to point and select objects. Their disadvantage is that when the keys on the keyboard are small, pressing the keys on the controller to select them can cause hand tremors, which may lead to a high error rate. Compared with the ray-based selection, controller tapping method [36] makes selections by flipping and holding the controller like a digital pen to tap the virtual key. Pad-based discrete/continuous cursor control method [36] uses two controllers to control two discrete/continuous cursors on a virtual QWERTY keyboard for character selections and confirms the input by the trigger button. Joysticks can also be used for text entry [10], PizzaText [49] divides a circular keyboard into several pizza blocks, places four letters in each block, and uses the omnidirectional dual-joystick controller to make two-step selections. The drum-like keyboard method [3] uses the rays emitted from the controllers as a drum stick and taps down on the keyboard for selection. However, this method may cause users to feel fatigued quickly, resulting in a high error rate. Compared with dual-hand techniques, single-hand techniques allow users to use one hand to do other interactive actions and may be easier to learn. HiPad [15] is a single-hand technique, which uses the touchpad on a single controller and selects letters on a partitioned keyboard by multiple pressing. Most efficient controller-based text entry methods require dual hands, which tend to fatigue the user and lead to high error rate. Some methods require complex controllers with joysticks or touchpad, and their interaction process is not smooth enough. In order to solve these problems, we propose a single-hand text entry method with a two-step interaction, which consists of a 3D hand translation in the first step and a single keystroke in the second step to select individual letters, involving only a small amount of spatial activity. It avoids multiple keystrokes for one selection, achieving high efficiency and lower error rate.

2.2 Text Entry with Circular Keyboards

The circular keyboard layout is often used in text entry and was first applied to pen-based text entry. Venolia et al. [37] reported that circular layouts could be used on small screens and might be beneficial to experts. Cirrin [27] and its expanding version [5] achieved the word-level gesture input with the circular layout. The Transparent User guided Prediction (TUP) method [32] with a language prediction algorithm was proposed to make users type text on a circular touch-sensitive wheel, and easy to select the characters with the highest probability. The circular keyboard layout is also used in gazed-based typing applications. Huckauf et al. [14] selected letters by gazing at the circular-interface keyboard. Benligiray et al. [2] applied language prediction model to optimize the gaze input of the inner-outer circle layout, which could help to use the screen area more effectively. Circular keyboards are also often seen in wearables such as smartwatches. COMPASS [46] was a non-touch bezel-based text entry technique of the circular keyboard on smartwatches. WrisText [9] allowed users to enter text by rotating the wrist of the watch hand towards six directions, each of which represented a key in the circular keyboard. The circular keyboards are also widely used in VR text entry. PizzaText [49] is a circular-keyboard-based technique, which divides a circle into several parts and each part contains four characters. RingText [44] makes letters selection in a circular keyboard by controlling head movements. HiPad [15] allows users to enter text on a segmented circular keyboard. Referring to the layout design idea of the above methods, we propose a new flower-shaped keyboard layout and explore it in VR.

3 DESIGN RATIONALE OF FLOWER TEXT ENTRY

Our goal is to design an efficient and easy-to-use text entry technique, so we explore design rationales from the following three aspects.

3.1 Familiarity

The keyboard layout that users are most familiar with is the layout of the physical QWERTY keyboard. So when we design the keyboard layout, a more intuitive way of thinking is to keep the features of the QWERTY keyboard letter distribution as much as possible, at least three features of which should be maintained. The first one is that the letters distributed in a row remain in a straight line in the design of the new keyboard. For example, 'QWERT' is in one row and 'ASDF' is in another row, so 'QWERT' should be arranged in a line, and 'ASDF' should be placed in another line in the new keyboard. The second feature is that the distribution of letters in the same row in the QWERTY keyboard has a certain order, which is well known. For example, the letters 'Q', 'W', 'E', 'R', and 'T' are arranged in sequence from left to right. We should keep this local letter distribution orderly when designing a new keyboard. The third feature is that when the user uses the QWERTY keyboard to input, the keyboard is usually divided into two parts, the left hand is used to type the letters on the left part of the keyboard, and the right hand is used to type the letters on the right side of the keyboard. Although our new keyboard is designed for single-hand operation, it still maintains this feature. The letters of the left area of the QWERTY keyboard are still placed on the left part of the new keyboard and vice versa. Moreover, when the user selects the keys, we can add the sound similar to the real mechanical keyboard to give the user an auditory experience, like using the physical keyboard in reality.

3.2 Efficiency

To improve the efficiency of inputting text, we can consider the following aspects. In the interaction process, we should try to reduce the operation steps involved in selecting each letter and provide a smooth experience. For selecting one letter on the keyboard, some methods require at least two keystrokes, such as PizzaText [49] and HiPad [15]. They use blocked keyboard layouts where the user has to first find the target block and then find the target letter from the block, which is not a smooth and familiar process for the user. Inspired by the use of physical keyboards, we can transfer its interactive process of first moving the finger over the target key and then pressing it to complete the selection to VR. We can use the 3D hand translation to move the controller to find the target key on a virtual keyboard, and then select it with a single keystroke. Although this is also a two-step selection process, it is more natural to the user. To do this, we design a block-shaped circular keyboard, and each block is expanded to ensure that a single key corresponds to a single letter. We have not used the dwell method to select the letter to avoid the minimum time interval of the selection [44]. Some visualizations can also be considered to improve the efficiency of text entry. One possible approach is to assign different colors to keys based on how often individual letters are used, highlighting those frequently used letters so users can quickly find them. Another one is to give some visual feedback when users select the key and help them quickly confirm the current selection. Word prediction can predict a complete word based on the input of the first few letters of the user, and perform word completion, thereby reducing the number of letters that need to be input and improving input efficiency. Therefore, when designing a new text entry technique, we should consider its ability to integrate the word prediction function easily. At last, the number of keys on the left and right parts of the keyboard should be balanced as much as possible, so that for the method based on 3D hand translation, the user's hand does not need to move in one direction all the time while typing. Otherwise, long time typing may cause fatigue and slow down text input speed.

3.3 Accuracy

Visual, auditory, and tactile feedback during text entry can help the user confirm whether the current selection is what he expects, thereby improving the accuracy rate. Therefore, we should consider including them when designing text entry techniques. Word correction can help users automatically correct spelling errors, so it can greatly reduce the rate of input errors. When designing the text entry technique, we should consider integrating a word correction function. Human perceptions of the distance of their hand motion are non-uniform, which means that people feel that moving hands near the neutral position is easier. The farther away from the neutral position, the harder it is to move their hands. This factor should be taken into account when designing our text entry technique.

4 FLOWER TEXT ENTRY

Based on the above design rationales, we design a flower-shaped keyboard and propose a single-controller interaction based on 3D hand translation to input text efficiently in the VE.

4.1 Keyboard Layout Design

Starting with the QWERTY keyboard, we divide the keys into six groups based on their left-right distribution and row arrangement on the keyboard (Figure 2 left). Then we rotate each group of keys to form a radial circular keyboard. The radial circular keyboard comprises a central point and six branches around in different directions. The space character is placed in the center, and the 26 letters are divided into six groups, each group of 3 to 5 letters, and are distributed on the six branches, like a blooming six-petal flower, so we call it the flower keyboard (Figure 2 right). In the flower keyboard, we maintain the three features of the QWERTY keyboard: linear pattern, keeping the letter order in a group and keeping the left/right parts of the QWERTY keyboard.

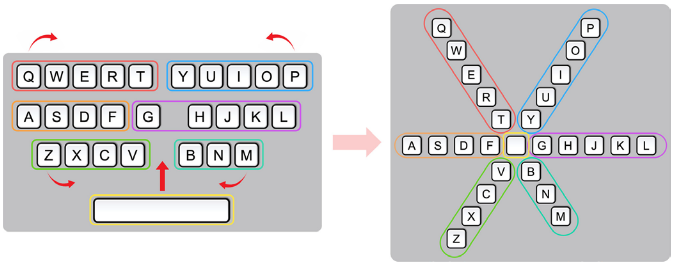


Fig. 2: The flower-shaped keyboard design.

Fig. 3: The functional buttons on the controller used in our method.

In order to increase the speed of text entry, we adopt a form in that only one letter is distributed in each key. So when each letter is selected, only one keystroke is required, i.e. the number of keystrokes per character (KSPC) [24] is 1, which is considered to be critical for increasing the speed of text entry [49]. While the state-of-the-art handheld text entry methods (e.g. PizzaText [49] and HiPad [15]) have a KSPC greater than 1. We also equalize the number of letters on the left and right sides of the flower to maintain the balance of the keyboard and eliminate the deviation in both two directions, therefore we move 'G' to the right side.

4.2 Interaction

We use HTC Vive Focus controller as the input device. Figure 3 shows the controller and the functional buttons we use. We introduce the interactive process of our method from the following three parts.

Reset Before entering text, we need to place a flower-shaped keyboard in the VE, which has a diameter of 1.56 in Unity, and the size of the keyboard is about 1/3 of the height of the screen. When the user presses the reset button, the flower-shaped keyboard is placed directly in front of him, i.e. the keyboard surface faces the user, and the distance to the user is 3.5 in Unity. The user's line of sight is usually forward when wearing the headset, and the keyboard will not block the virtual scene too much. At the same time, we build a coordinate system for interaction, in which we use the center of the keyboard as the origin O, the user's head facing forward as the z-axis negative direction, right vertically as the x-axis positive direction, and up vertically as the y-axis positive direction. Then we bind the controller's current position to the center of the keyboard. Since the user may change his position when entering text, the reset function can adapt the keyboard to position changes.

3D hand translation interaction As shown in Figure 4 left, the user holds the controller with one hand to translate in the 3D space. We project the vector $\overrightarrow{OP}$ (from the origin O of the interactive coordinate system to the controller position P) onto the XY plane to obtain the projected hand translation vector $\overrightarrow{OP'}$, and calculate the length of $\overrightarrow{OP}$ as the hand translation distance $\left\|\overrightarrow{OP}\right\|_2$, where $\|\cdot\|_2$ represents the

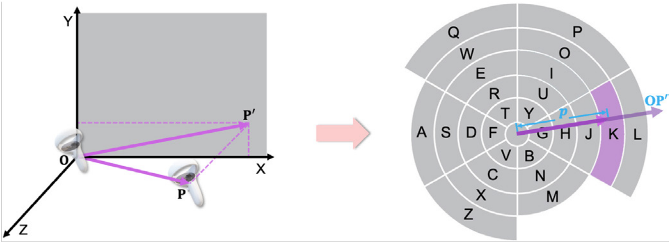


Fig. 4: Mapping from the interactive coordinate system (left) to the valid selection area of the keyboard (right).

Euclidean norm. Then we use the hand translation couple $(\overrightarrow{OP'}, \left\|\overrightarrow{OP}\right\|_2)$ to select keys on the keyboard by mapping to locations on our flower keyboard. Figure 4 right shows the valid selection area for each key of the flower-shaped keyboard. The projected hand translation vector $\overrightarrow{OP'}$ is used to indicate the direction on the keyboard directly. For distance mapping, we create a linear mapping function to scale hand translation distance $\left\|\overrightarrow{OP}\right\|_2$ to the range of the valid selection area of the keys on the keyboard. The mapped distance p is computed according to Equation 1, where k is the mapping coefficient (discussed in Section 6). According to Equation 1 of the mapping, the k value can be scaled with the change of keyboard size in the virtual scene, so the interaction experience is independent of the size of the keyboard and can eliminate the influence of the keyboard size.

$$p = k \times \left\|\overrightarrow{OP}\right\|_2 \qquad (1)$$

Selection & Backspace As shown in Figure 3, we use the ‘B’ button as the selection button and the ‘A’ button as the backspace button. These two function buttons can also be placed on any other type of controller with buttons. When the user selects a letter, he can switch the candidate letter key by translating the controller. The current candidate letter key is slightly enlarged to give the user a visual feedback. After pressing the selection button, the selection is completed, and the selected letter will be added to the input text box. Pressing the backspace button can delete one letter from the input text box. If the current input text box is empty, nothing is done.

4.3 Letter Layouts and Optimization Options

To achieve efficient typing, we consider two types of letter layouts and four options, which may optimize the performance of our method.

- Letter layouts. The default letter layout of our keyboard is designed in the same way as the QWERTY keyboard (row 2 in Figure 5). While the letter layouts of [15, 44, 49] are in alphabetical order. Thus, another possible letter layout of the Flower Text Entry is in alphabetical order. For example, ‘A’, ‘B’, ‘C’, ‘D’, ‘E’ are in one petal and ‘F’, ‘G’, ‘H’, ‘I’, ‘J’ are in another petal (row 1 in Figure 5).
- Colored keys. The default keyboard color is gray (column 1 in Figure 5). Another possible option is highlighting the keys according to the letter frequency in English [42]. Single color makes each key difficult to distinguish, which may reduce the efficiency of the user’s search for letters. Multiple bright colors can enhance the user’s distinction between keys and help users quickly find the letters they want to select. So we sort all letters by frequency of usage, divide them into three groups, and highlight the keys of high, mid, and low frequencies letters in yellow, blue, and red (column 2 in the Figure 5) respectively.
- Mapped hand position visualization. The default method only has the visual feedback that the candidate key will be enlarged slightly, and has not any hand position hint (column 1 in Figure 5). The visual feedback of changing the candidate letter key from one to another is a discrete visualization. While the mapped hand position visualization shows the exact mapped position of the user’s hand on the valid selection area of the keyboard, which is a continuous visualization. It may help the user maintain the perception of his hand motion and improve efficiency. For this reason, another option is to visualize the mapped hand position with a bright green dot (column 3 in Figure 5).
- Word prediction and correction. The previous techniques usually incorporate word prediction and correction [15, 44], which can improve typing speed and reduce error rate, so we consider adding them to the Flower Text Entry as well to maximize its potential. We use SymSpell [8] to predict the subsequent letters in the word based on the letters that the user has entered to perform word completion, which can also automatically correct spelling errors. When the user types, the first two predicted results are displayed in the candidate area (column 4 in Figure 5), which is always directly below the current cursor. After the user presses the trigger button of the controller, the horizontal hand translation is used to select candidate predicted words, i.e. moving to the left is to select the left one, and vice versa. Releasing the trigger button indicates that the current candidate word is selected. After that, the entered letter string will be replaced by it, and a space character is automatically added to the input text box. This process also includes a word correction function. Meanwhile, the vertical upward translation of the hand is used to cancel the prediction.
- Audio and tactile feedback. We consider adding more feedback during typing to help users select more accurately, including audio and tactile feedback. When pressing the selection button to choose a letter key, the user hears a sound similar to typing on the physical keyboard. Pressing the backspace button will have a Windows-style delete sound. Tactile feedback refers to the vibration of the controller when switching candidate keys.

5 PILOT USER STUDY 1: EVALUATE THE LETTER LAYOUTS AND OPTIMIZATION OPTIONS

We have proposed two types of letter layouts and four possible optimization options in Section 4.3, thus we design a pilot user study to evaluate them.

5.1 Pilot User Study Design

Hypotheses We formulate five hypotheses for the pilot user study:

H.1. Typing with the QWERTY letter layout would have a higher input speed than with the alphabetical order letter layout since users might be more familiar with the former.

H.2. Typing with multiple color keys would have a higher input speed than with single color keys. Since colors could give users strong visual feedback and might improve users’ attention to frequently used keys, giving keys different colors might help users find the target key faster when typing.

H.3. Typing with mapped hand position visualization would have a higher input speed than without it. Since mapped hand position visualization could help users identify their hand positions on the keyboard more quickly and intuitively.

H.4. Typing with word prediction and correction would have a higher input speed than without them. With the help of recommendations, word prediction could reduce the numbers of selection when users type a word, and word correction could help users avoid spelling mistakes.

H.5. Typing with audio and tactile feedback would have a higher input speed than without them. Since audio and tactile feedback could help users confirm that the letter key is selected, it avoids the requirement for users to keep staring at the input text box while typing.

H.6. The QWERTY letter layout and all optimization options could reduce workload since they could help users enter text more naturally than without them.

Participants Eighteen participants (twelve males and six females, aged between 22-30) from our university are recruited in this study, and thirteen of them had some VR HMDs experiences. They are all familiar with the alphabet but not native English speakers, and all participants are right-handed.

Hardware Setup The pilot user study is conducted on a computer with an Intel Core i7 processor and an NVIDIA GeForce RTX 2080 graphics card. The program is developed with C# in Unity 2020.1.12.f1c1 and is run in the Unity3D platform. We use HTC Vive Focus 3 as the experimental device to provide participants with an immersive experience.

Task and Procedure The experiment uses a 2 × 5 within-subjects design, i.e. all participants are required to attend 10 sessions. The

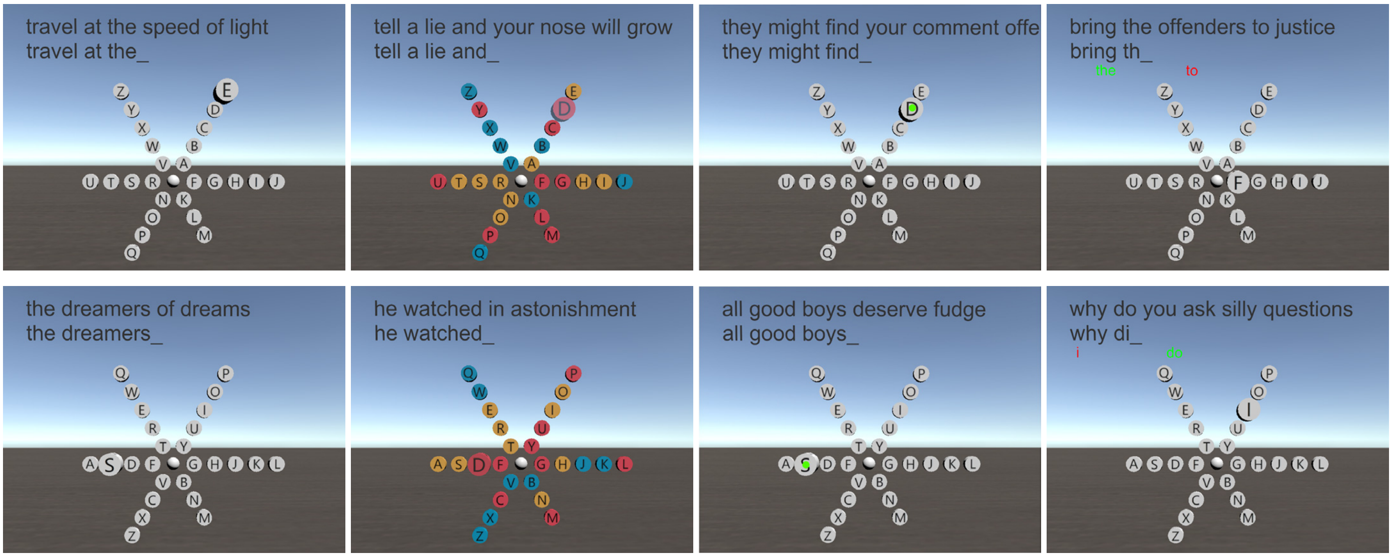


Fig. 5: Two types of letter layouts and three kinds optimization options of our flower keyboard. QWERTY letter layout is in row 2, alphabetical order letter layout is in row 1. Default design is in column 1, colored keys option is in column 2, mapped hand position visualization option is in column 3, word prediction and correction option is in column 4.

task is to type ten phrases in each session. The order of sessions is set randomly, and phrases in each session are randomly generated from the Mackenzie phrase set [26]. Participants are asked to enter the text 'quickly and accurately'. We label the sessions according to different combinations of conditions. Session 1, 3, 5, 7, 9 are with the alphabetical order letter layout, and session 2, 4, 6, 8, 10 are with the QWERTY letter layout. Session 1 and 2 are with the default keyboard design. Session 3 and 4 are with the option of colored keys. Session 5 and 6 are with the option of mapped hand position visualization. Session 7 and 8 are with the option of word prediction and correction. Session 9 and 10 are with the option of audio and tactile feedback. Before the whole experiment, we introduce how to use the keyboard based on the default design, and participants are given 5 minutes to get familiar with the interaction of the keyboard. Before each session, participants are required to complete one phrase exercise. After each session, participants will fill out the NASA-TLX questionnaire [13] of the current session and rest for two minutes. Each participant spends an average of 100 minutes. A total of 2 (layouts) × 5 (default + 4 options) × 10 (phrases) × 18 (participants) = 1800 phrases are collected. We use Equation 2 [25] to calculate WPM to measure the typing speed:

$$WPM = \frac{|T|-1}{S} \times 60 \times \frac{1}{5} \tag{2}$$

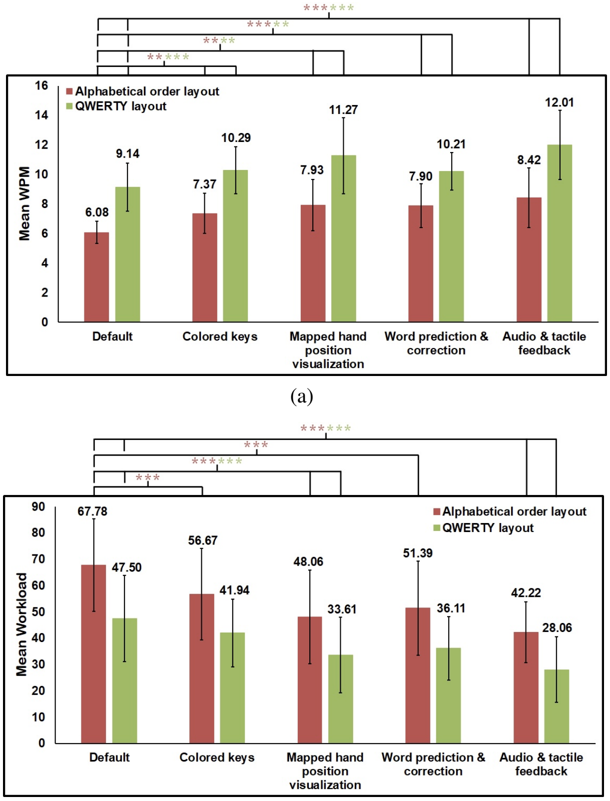
Where T is the target phrase and S is the time (in seconds) taken between the first and last press in each phrase. We also calculated the Total Error Rate (TER) = Not Corrected Error Rate (NCER) + Corrected Error Rate (CER) [35].

5.2 Results

We use a two-way repeated measure ANOVA and Bonferroni correction in pair-wise comparisons. We also use a Greenhouse-Geisser adjustment to correct violations of the spherical hypothesis. The following are the results of the analysis.

Typing Speed Significant effects on typing speed are found on 'letter layout' ($F_{1,17} = 251.454$, $p = 1.278 \times 10^{-11}$, $\eta_p^2 = .937$) and 'option' ($F_{1.936,32.904} = 35.186$, $p = 8.775 \times 10^{-9}$, $\eta_p^2 = .674$), and a significant interaction effect is also found on 'letter layout' × 'option' ($F_{2.327,39.567} = 3.972$, $p = .022$, $\eta_p^2 = .189$). The result of pair-wise comparisons show that QWERTY layout significantly improves typing speed ($p = 1.278 \times 10^{-11}$), and the application of optimization options has a significant effect on speed compared with the default design (colored keys: $p = 1.269 \times 10^{-5}$, mapped hand position visualization: $p = 2.015 \times 10^{-4}$, word prediction and correction: $p = 2.016 \times 10^{-7}$, audio and tactile feedback: $p = 1.792 \times 10^{-6}$), in which adding more feedback has the largest mean difference of 2.60 WPM (s.e. = 0.31).

Figure 6 (a) shows the mean typing speed over 10 sessions. The typing speed of the QWERTY letter layout is overall faster than that of the alphabetical order letter layout, and all typing speeds with optimization options are faster than those without them. Among all sessions, the QWERTY layout with audio and tactile feedback (session 10) achieves the highest typing speed of 12.01 WPM (s.e. = 2.42). Compared with the default design with the alphabetical order letter layout (session 1), speed of 6.08 WPM (s.e. = 0.77), session 10 has an increase of 97.53%.



(a)

(b)

Fig. 6: Mean typing speed (a) and workload (b) of the flower keyboard with two types of letter layouts and four options. Error bars indicate standard deviation. Asterisks denote statistical significance between different options in two layouts.

Workload Figure 6 (b) shows the mean NASA-TLX workload scores

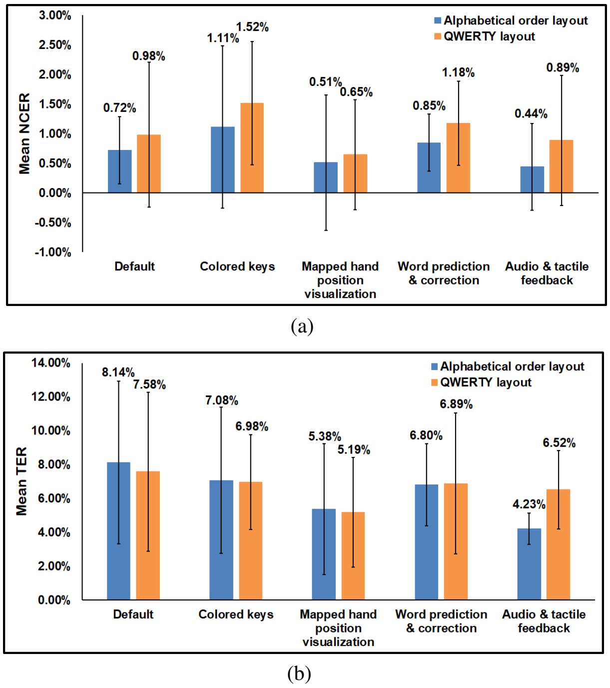


Fig. 7: Mean NCER (a) and TER (b) of the flower keyboard with two types of letter layouts and four options. Error bars indicate standard deviation.

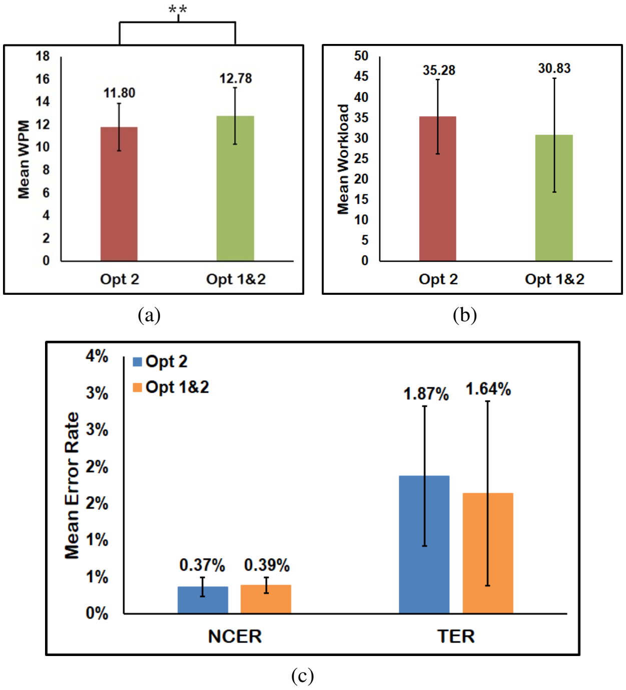


Fig. 8: The comparisons of mean typing speed (a), workload (b), and error rates (c) of the flower keyboard (with QWERTY letter layout) between with all options and without colored keys. Error bars indicate standard deviation. Asterisks denote statistical significance.

over 10 sessions. The results of ANOVA indicate significant effects on ‘letter layout’ ($F_{1,17} = 42.215$, $p = 5.472 \times 10^{-6}$, $\eta_p^2 = .713$) and ‘option’ ($F_{1.979,33.651} = 42.254$, $p = 7.279 \times 10^{-10}$, $\eta_p^2 = .713$), and a significant interaction effect on ‘letter layout’ × ‘option’ ($F_{2.143,36.438} = 4.247$, $p = .020$, $\eta_p^2 = .200$). The pair-wise comparisons show that the QWERTY layout significantly reduces the workload ($p = 5.472 \times 10^{-6}$), and the application of optimization options is also better than that of the default design (colored keys: $p = .001$, mapped hand position visualization: $p = 8.930 \times 10^{-10}$, word prediction and correction: $p = .001$, audio and tactile feedback: $p = 3.878 \times 10^{-10}$).

Error Rate Figure 7 shows the mean NCER (a) and TER (b) over 10 sessions. For NCER, significant effects are found on ‘letter layout’ ($F_{1,17} = 4.786$, $p = .043$, $\eta_p^2 = .220$) and ‘option’ ($F_{1.781,30.284} = 4.013$, $p = .033$, $\eta_p^2 = .191$), but no significant interaction effect on ‘letter layout’ × ‘option’ ($F_{2.612,44.402} = 2.281 \times 10^{-1}$, $p = .851$, $\eta_p^2 = .013$). The pair-wise comparisons show that the QWERTY layout causes a significant increase in the NCER ($p = .043$). Audio and tactile feedback has a nearly significant difference compared with colored keys ($p = .052$) and a significant difference compared with word prediction and correction ($p = .010$). For TER, there is no significant effect on ‘letter layout’ ($F_{1,17} = .847$, $p = .370$, $\eta_p^2 = .047$), but a slightly significant effect on ‘option’ ($F_{1.972,33.522} = 3.496$, $p = .042$, $\eta_p^2 = .171$), and a significant interaction effect on ‘letter layout’ × ‘option’ ($F_{2.651,45.071} = 3.488$, $p = .028$, $\eta_p^2 = .170$) are found. No significant difference is found in the pair-wise comparisons.

5.3 Discussion

The results fully support our six hypotheses. We combine the ANOVA results of speed, workload, and error rate to determine the final form of the keyboard.

For the letter layouts, the results of speed and workload show that the QWERTY layout can significantly accelerate the typing speed and reduce the fatigue of users. Although it led to an increase in the NCER, it is still within the acceptable range, which references to PizzaText (NCER of 1.59%) [49]. All participants claim that the QWERTY layout makes it easier for them to find the target letters and significantly reduces the workload. Therefore, we chose the QWERTY layout as the final letter layout.

For the optimization options, all four options perform well in terms of speed and workload, and their error rates are not significantly different from those of the original design. In these options, option 1 (colored keys) and 2 (mapped hand position visualization) are both related to the visual feedback, and option 2 has a better performance, so we add two additional sessions to explore whether they will work when combined: session 11 is with the QWERTY layout and all options except option 1, and session 12 is with the QWERTY layout and all options. The same eighteen participants from the previous 10 sessions participate in session 11 and session 12. Since option 3 (word prediction and correction) is in operational level and option 4 (audio and tactile feedback) utilizes tactile and auditory senses, we think that they and the visual level (option 1 and 2) improve typing performance through different perceptual channels and will not interact with each other, so we do not attend to test the other combinations. Figure 8 shows the mean (a) typing speed, (b) workload, and (c) error rate of session 11 and session 12. The ANOVA results show that there is a significant effect on typing speed ($F_{1,17} = 10.506$, $p = .005$, $\eta_p^2 = .382$), while no significant effects on workload ($F_{1,17} = 2.345$, $p = .144$, $\eta_p^2 = .121$), NCER ($F_{1,17} = .563$, $p = .463$, $\eta_p^2 = .032$), and TER ($F_{1,17} = 0.917$, $p = .352$, $\eta_p^2 = .051$). The results indicate that combining colored keys and mapped hand position visualization achieves significant improvement of text entry speed without significantly affecting workload and error rate. Fourteen participants report a preference for option 2 and option 4. One participant says *“option 2 allows me to quickly locate the mapping position of my hand, which gives me more security during typing”*. Another participant says *“option 4 allows me to quickly confirm that a key is selected. The combination of sound and vibration gives a rhythmic feel to the typing process”*. Ten participants also report that the function of automatically adding space in option 3 is convenient, which allows them to start spelling the next word directly. We finally choose all optimization options.

6 PILOT USER STUDY 2: OPTIMIZE THE DISTANCE MAPPING

After the first pilot user study, twelve participants report excessive hand translations and incorrect selections during letter selection, which limits typing speed and increases error rate. Since we use a linear mapping function to scale hand translation distance $\|OP\|_2$ to the range of the valid selection area of letter keys on the keyboard in Equation 1, we think that these problems can be alleviated by collecting and analyzing the hand translation data when users select letters and adjusting the mapping coefficient k in Equation 1. We design pilot user study 2 to optimize the distance mapping.

6.1 Pilot User Study Design

This study consists of two parts: 1) data collection and optimization; 2) evaluation of the optimized mapping. In the first part, we test the typing performance of different linear mapping coefficients. Similar to [50], we also collect click-points (hand translation couple data) when participants click keys during the experiment. Then we analyze the distribution of click-points of the coefficient with the best typing performance and optimize the mapping process based on it. In the second part, we verify the validity of the optimization through experiments.

Hypotheses We formulate one hypothesis for each of the two parts of this pilot user study:

H.1. Different mapping coefficients would affect users' typing performance.

H.2. Optimizing the distance mapping process could improve typing speed and reduce error rate.

Participants and Hardware Setup The same eighteen participants from pilot user study 1 participate in this study. The hardware setup of this study is also the same as that of pilot user study 1.

Task and Procedure This pilot user study uses a within-subjects design with one independent variable. The first part of the study includes five sessions (session 1-5), which test the different linear mapping coefficients, $k = 3.5, 4, 4.5, 5, 5.5$. The selection of these coefficients is based on the experience of the experts (participants from pilot user study 1). And the order of these five sessions is randomly set. We analyze the data collected in these five sessions, and then obtain the optimized mapping. The second part of the study has one session (session 6) to evaluate the optimized mapping. The task in each session was the same as that in the pilot user study 1. A total of 6 (sessions) × 10 (phrases) × 18 (participants) = 1080 phrases are collected.

Mapping Optimization with Click-points Analysis We optimize the distance mapping of hand translation by analyzing the click-points collected in the first five sessions. We select the click-points data of $k = 4.5$, which has the highest typing speed and lowest error rate, with a total of 11,402 click-points. Before analyzing the data, we remove the outliers that are more than three times the standard deviation from the centroid, which account for 3.5% of the total.

The data is visualized in Figure 9. The scatter plot (a) contains click-points and 95% confidence ellipses of all click-points for each key. The data of each key is rendered with different colors. The click-points locations in the plot are calculated using Equation 3:

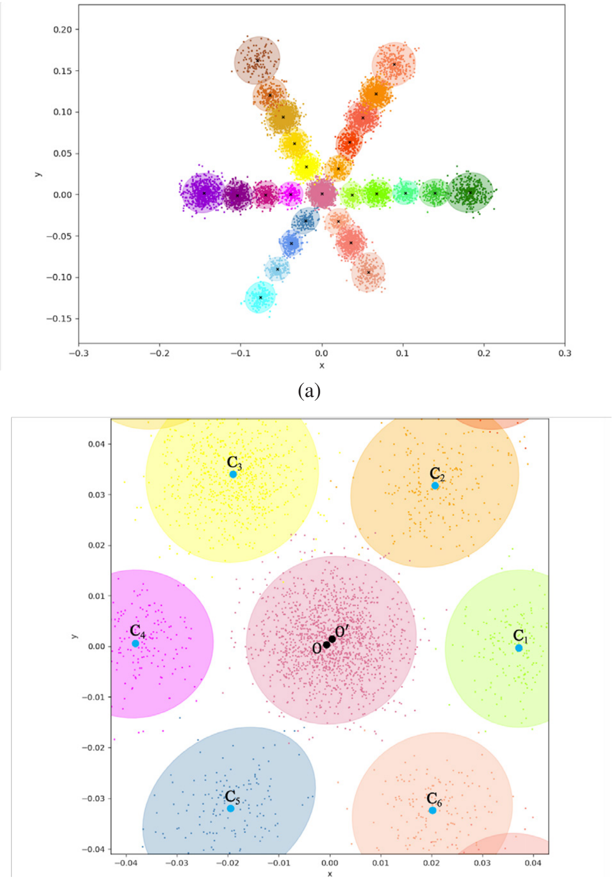
$$\begin{cases} x = \left\|\overrightarrow{OP}\right\|_2 \times \left(\hat{OP'} \cdot \hat{e_x}\right) \\ y = \left\|\overrightarrow{OP}\right\|_2 \times \left(\hat{OP'} \times \hat{e_x}\right) \end{cases} \tag{3}$$

Where $\hat{}$ represents the unit vector in a direction, and $\hat{e_x}$ is the unit vector in the positive direction of the x-axis. The data visualized in Figure 9 also includes the misselected points. For example, when users want to select 'H', they may misselect the adjacent key 'G', and then they have to delete 'G' and select 'H' again. We add such misselected points of 'G' to the click-points of 'H'.

We enlarge the central part of (a) to get (b), which shows the click-points and 95% confidence ellipse of the space key and the black dot O', which is the mean of all click-points belonging to the space key. We find that $O'(0.0003, 0.0012)$ is off the intended position $O(0, 0)$. Therefore, according to Equation 4, we recalculate the mapping coefficients for each key. $\overrightarrow{O'C_i}$ is the vector from the new origin O' of the keyboard to the center C_i (blue points) of the confidence ellipse for each key, and Len is the distance from the center of the keyboard to the center of each key in the virtual scene.

$$\begin{cases} k = \frac{Len}{\left\|\overrightarrow{O'C_i}\right\|_2} \\ i \in [1, 26] \end{cases} \tag{4}$$

Table 1 shows the values of k calculated in Equation 4. Direction 1 is the right horizontal direction, and Direction 2-6 are the other five directions in counterclockwise order. Key 1-4 are keys arranged from the center to the outside, i.e. 'G', 'Y', 'T', 'F', 'V', 'B' are key 1; 'H', 'U', 'R', 'D', 'C', 'N' are key 2; 'J', 'I', 'E', 'S', 'X' are key 3; 'K', 'O', 'W' are key 4. We do not use the data of the last key in each direction. These data are not informative because we truncate users' hand translation beyond a certain threshold, and map it to the last key, so their hand translation may be exaggerated when selecting the last key in each direction. The last column of Table 1 shows the average value of k for all keys except key 1 in each direction. For example, in direction 1, we average k values of 'H', 'J', 'K'. We can see that the mapping coefficients k of key 1 around the center of the keyboard are all less than 4.5, while k values of keys outside this range are all greater than or equal to 4.5. Thus in each direction, we use two values of k to scale the hand translation, i.e. within the range of key 1 and space key in the virtual scene, we use the k value of the first column in Table 1 as the mapping coefficient. Beyond this range, we use the k value of the last column in Table 1 as the mapping coefficient. We use this piece-wise mapping as the optimized one in session 6. We do not set k values for each keys because the k values of outer keys are very close, so we take the average to provide a smoother sense of operation for the user. It is also possible to make one for each person, but it will require a large number of experiments by a single person to collect the data, so we do not personalize this process but use a general method derived from multi-person data.



(a)

(b)

Fig. 9: (a) The scatter plot that contains click-points and 95% confidence ellipses for each key. (b) The central part of (a).

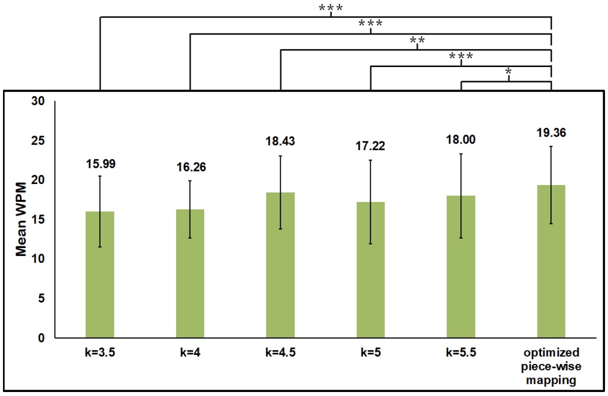
6.2 Results

We use a one-way repeated measure ANOVA to analyze the results. Bonferroni correction is used in pair-wise comparisons, and Greenhouse-Geisser adjustment is used for correction in case of violations of the spherical hypothesis.

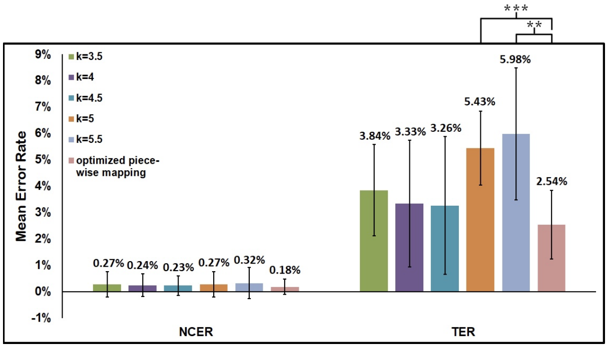
Typing Speed Figure 10 (a) shows the mean typing speed of 6 sessions. ANOVA results show that 'session' ($F_{2.134,36.280} = 18.470$, $p = 1.944 \times 10^{-6}$, $\eta_p^2 = .521$) has a significant effect on typing speed. In pair-wise comparison, we find that the typing speed of session 3 ($k = 4.5$) is the fastest within session 1-5, and session $1vs3$, $2vs3$, $3vs4$

Table 1: Recalculated mapping coefficients k of the 26 letter keys, and the last column is the average k value of all valid keys except key 1 in each direction.

k	Key 1	Key 2	Key 3	Key 4	Avg.(except key 1)
Direction 1	4.33	4.73	4.65	4.59	4.66
Direction 2	4.36	4.53	4.57	4.65	4.58
Direction 3	4.21	4.60	4.60	4.73	4.64
Direction 4	4.13	4.62	4.60	–	4.61
Direction 5	4.13	4.50	4.50	–	4.50
Direction 6	4.08	4.64	–	–	4.64



(a)



(b)

Fig. 10: Mean typing speed (a) and error rate (b) of 6 sessions in pilot user study 2. Error bars indicate standard deviation. Asterisks denote statistical significance.

are significantly different (all $p < .05$), which indicates that $k = 4.5$ performs better on typing speed than the other four coefficients. Meanwhile, session $1 vs 6$, $2 vs 6$, $3 vs 6$, $4 vs 6$, $5 vs 6$ are found significantly different (all $p < .05$) as well, which indicates that the optimized mapping improves the typing speed significantly.

Error Rate Figure 10 (b) shows the mean NCER and TER over 6 sessions. For NCER, no significant effect is found on 'session' ($F_{1.999,33.991} = 1.455$, $p = .248$, $\eta_p^2 = .079$), and in pair-wise comparisons. For TER, a significant effect is found on 'session' ($F_{1.991,33.854} = 8.677$, $p = .001$, $\eta_p^2 = .338$). In pair-wise comparisons, we find that TER of session 3 ($k = 4.5$) is the lowest within session 1-5, and session $3 vs 4$ is significantly different ($p < .05$), which indicates that $k = 4.5$ preforms better on TER than the other four coefficients. Meanwhile, TER of session 6 is the lowest over 6 sessions, and session $4 vs 6$ and $5 vs 6$ are found significantly different (all $p < .05$), which indicates that optimized mapping decreases TER. Session $1 vs 4$ and $2 vs 5$ are also found significantly different (all $p < .05$).

6.3 Discussion

The results of the experiment fully support our two hypotheses. Firstly, different mapping coefficients affect users' typing performance. The essence of distance mapping is the scaling of the hand translation on the valid selection area of the keyboard. The first five bars of Figure 10 show the performance of the first five sessions with different k. We find that session 3 with $k = 4.5$ has the highest typing speed and lowest error rate, session 1-3 has an increasing trend of typing speed and a decreasing trend of error rate, and vice versa in session 3-5. The reason for the above phenomenon may be that if the k value is too small, the user feels that the selection process is too slow. If the k value is too large, the user feels that the selection process is too sensitive. Both situations affect the typing speed and error rate of text entry. Twelve participants report that too sensitive operations cause higher workload and can not select accurately, which can also partly explain this phenomenon. Secondly, optimizing the mapping process improves typing speed and reduces error rate. Users have different perceptions of hand translation in different directions, which leads to low performance of setting k value with a constant value for all directions. We optimize the k value in each direction and achieve better performance. Furthermore, we find that even in the same direction, the user's perception of hand translation differs between the area near the center and the area far from the center. Hence, a piece-wise linear mapping is appropriate. Fifteen participants report that the optimized mapping provides the most comfortable experience compared with too small or too large k values. We use this optimized mapping in the following user study.

7 USER STUDY

We conduct a 6-day user study to evaluate the performance of Flower Text Entry and explore how the performance of the two groups, the novice group and the potential expert group, will improve over a short 6-day practice. We use the same hardware setup as pilot user studies.

7.1 User Study Design

Participants Ten participants (seven males and three females, aged between 21-25) are recruited for the user study. We divide them into two groups: the novice group and the potential expert group. None of the participants in the novice group participates in the pilot user studies, and they are also all right-handed. The participants of the potential expert group are from the pilot user studies. We rank all participants of the pilot user studies according to their performance, select the top five best performers, and invite them to continue participating in the 6-day experiment and form the potential expert group. All participants have experience with VR HMDs.

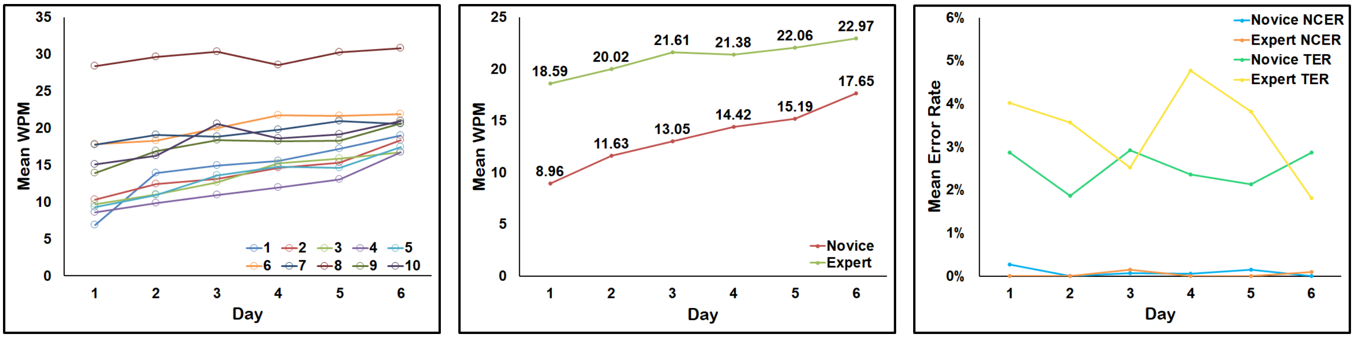
Task and Procedure The whole experiment consists of six sessions, one per day. The task is to require participants to type ten phrases in each session. Before each session starts, participants can appropriately practice 2-4 phrases. In each session, The phrases are randomly generated by the Mackenzie phrase set [26]. Same as required in the pilot user studies, participants are asked 'as fast and accurate as possible' to enter the text. Before the experiment starts, we show the novice group how to use Flower Text Entry and give them 5 minutes to try it. It takes an average of 30 minutes in six days for each participant to complete the whole procedure. The data of 5 (participants) × 2 (groups) × 6 (sessions) × 10 (phrases) = 600 phrases are collected.

7.2 Results

We use a mixed-design ANOVA with 'session' (1-6) as the within-subjects variable and 'group' (novice and potential expert) as the between-subjects variable. We use Bonferroni correction in pair-wise comparisons and Greenhouse-Geisser adjustment in violations of the spherical hypothesis.

Typing Speed The results indicate that 'session' ($F_{5,40} = 49.540$, $p = 4.337 \times 10^{-16}$, $\eta_p^2 = .861$) and 'session' × 'group' ($F_{5,40} = 5.697$, $p = 4.655 \times 10^{-4}$, $\eta_p^2 = .416$) have significant effects on typing speed. There is also a significant effect of 'group' ($F_{1,8} = 11.717$, $p = .009$, $\eta_p^2 = .594$) on typing speed. Overall, there is a significant difference in the typing speed over 6 days. The typing speed of the potential expert group is significant faster than the novice group, and the learning effects of the two groups are significantly different. The pair-wise comparisons reveal significant differences of session $1 vs 3$, $1 vs 4$, $1 vs 5$, $1 vs 6$, $2 vs 4$, $2 vs 5$, $2 vs 6$, $3 vs 6$, $4 vs 6$, $5 vs 6$ (all $p < .05$). This trend shows that after 6 sessions' practice, the learning curve is still on the rise.

Figure 11 (a) shows the mean typing speed of 10 participants. Overall, the average speed of all tests is 17.29 WPM. The average speed of all participants is 13.78 WPM (s.e. = 1.32) in the first session and reaches 20.31 WPM (s.e. = 1.01) in the last session, with an increase of 47.39%. The fastest speed is achieved by a participant from the



(a) (b) (c)

Fig. 11: (a) Mean typing speed of 10 participants, (b) mean typing speed of novice group and potential expert group, (c) mean NCER and TER of novice group and potential expert group of 6 sessions.

potential expert group in the sixth session, reaching 30.80 WPM, and his average speed is also the highest (29.65 WPM). Figure 11 (b) shows the mean typing speeds of two groups in each session. The average speed of the novice group is 13.48 WPM (s.e. = 1.57), and the average speed of the potential expert group is 21.10 WPM (s.e. = 1.57). The speed of potential expert group increases from 18.59 WPM (s.e. = 1.86) in the first session to 22.97 WPM (s.e. = 1.43) in the last session, with an increase of 23.56%. While the speed of the novice group soars from 8.96 WPM (s.e. = 1.86) in the first session to 17.65 WPM (s.e. = 1.43), with an increase of 96.99%. In particular, one participant in the novice group achieves 6.87 WPM in the first session, whereas in the second session, his speed increases rapidly to 13.90 WPM.

Error Rate For NCER, the results show that no significant effects are found on 'session' ($F_{2.851,22.809} = .691$, $p = .560$, $\eta_p^2 = .079$), 'session' × 'group' ($F_{2.851,22.809} = 1.361$, $p = .280$, $\eta_p^2 = .145$), or 'group' ($F_{1,8} = .912$, $p = .367$, $\eta_p^2 = .102$). For TER, ANOVA tests also yield no significant effects on 'session' ($F_{5,40} = .834$, $p = .534$, $\eta_p^2 = .094$), 'session' × 'group' ($F_{5,40} = 1.724$, $p = .151$, $\eta_p^2 = .177$) or 'group' ($F_{1,8} = 3.170$, $p = .113$, $\eta_p^2 = .284$). It indicates that the training has no significant impact on the error rate.

Figure 11 (c) shows the mean NCER and TER of novice group and potential expert group of 6 sessions. Overall, for all tests, the average NCER is 0.07%, and the average TER is 2.96%. For the average NCER, the novice group is 0.09% (s.e. = 0.04%), while the potential expert group is 0.04% (s.e. = 0.04%). For the average TER, the novice group is 2.50% (s.e. = 0.36%), while the potential expert group is 3.42% (s.e. = 0.36%). As shown in Figure 11 (c), we find that the TER of the potential expert group is higher than that of the novice group, which is due to their faster typing speed, thus increasing the probability of typing error, so that the TER inevitably rises in certain sessions.

7.3 Discussion

We compare our Flower Text Entry with the state-of-the-art methods, PizzaText [49], HiPad [15], ray-based QWERTY keyboard [36], and drum-like keyboard [3].

In terms of efficiency, novice users of Flower Text Entry reach 17.65 WPM after about 80 phrases of training, and potential expert users reach 22.97 WPM after the same training. Novice users of PizzaText reach 8.59 WPM after two hours of practice, and potential expert users reach 15.85 WPM after the same practice. Novice users of HiPad reach 13.57 WPM through a 60-phrase training. The typing speed of ray-based QWERTY keyboard can reach 15.44 WPM, and that of drum-like keyboard can reach 24.61 WPM. Compared with PizzaText, HiPad, and ray-based QWERTY keyboard, the typing speed of our method is faster, even for the dual-hand drum-like keyboard, our method achieves similar performance. The possible reason for this is that the interaction process of our method is more natural with a low workload, avoiding two or more keystrokes for each selection.

In terms of accuracy, both the NCER and TER of Flower Text Entry are low, with 0.09% and 2.50% of NCER and TER for novice users and 0.04% and 3.42% for potential expert users. The NCER and TER of novice users with PizzaText are 1.56% and 5.91% respectively, and those of potential expert users are 1.59% and 5.08% respectively. The NCER of HiPad is 0.22%. The NCER of ray-based QWERTY keyboard is 0.97%, and the TER of drum-like keyboard is 7.2%. The comparisons indicate that our method is more accurate in selection than the state-of-the-art methods. This shows that our optimized piece-wise mapping works well, which also indicates that hand interaction requires considering the impact of human hand activity.

In terms of learnability, after training for about 80 phrases, the speed of novice users increases by 96.99%, and that of potential expert users increases by 23.56%, which indicates that our Flower Text Entry is very friendly and easy to learn for users, since our method adopts small and natural hand movements for interaction.

8 CONCLUSION, LIMITATIONS, AND FUTURE WORK

In this paper, we propose Flower Text Entry, a single-controller text entry technique. This technique is based on a flower-shaped keyboard and introduces 3D hand translation interaction into the controller-based text entry techniques. Compared with the state-of-the-art methods, Flower Text Entry is a single-hand method with only a simple controller for interaction, and achieves high typing speed, lower error rate, and good learnability. After training about 80 phrases, novice users reach the average typing speed of 17.65 WPM, and potential expert users reach 22.97 WPM. The highest typing speed can reach 30.80 WPM.

Although Flower Text Entry has been proved to be efficient, accurate, and easy to learn, it still has some limitations. Firstly, it requires a certain amount of physical space to use, and according to the click-points we collect, the actual interaction area has an average diameter of about 33cm, so a very crowded environment may influence its performance. Secondly, it cannot be used directly in the state when the user is moving, because in Section 4.2 we mention that an interactive coordinate system needs to be established before starting typing, and the origin of the coordinate system is determined in three-dimensional space, i.e. when the user moves, the origin needs to be modified, which will affect the typing experience. Thirdly, it cannot be used while the user's hands are occupied or in a VR system without controllers. Finally, the participants of our user study are all right-handed typists, so in the future, more user studies need to be conducted for left-handed typists.

In the future, we will improve this method from the following four aspects. The first one is to adapt our method to mobile scenarios, we can automatically detect the change of user's position in space, and adaptively provide an appropriate origin of interactive coordinate system. One possible solution is to fix a tracker on the user's body and compute the relative transformation of the controller and the tracker. The second one is to explore if more hand motion can be integrated into text entry and improve keyboard design with more hand motion features. The third one is to explore whether using our keyboard in a crowded virtual scene would affect the efficiency of text entry. The last one is to find the relationship between colormap and letter frequency according to the characteristics of the human visual system, so as to better visualize our keyboard.

ACKNOWLEDGMENTS

This work was supported by National Key R&D plan 2019YFC1521102, by the National Natural Science Foundation of China through Projects 61932003 and 61772051.

REFERENCES

[1] J. Adhikary and K. Vertanen. Text entry in virtual environments using speech and a midair keyboard. *IEEE Transactions on Visualization and Computer Graphics*, 27(5):2648–2658, 2021.

[2] B. Benligiray, C. Topal, and C. Akinlar. Slicetype: fast gaze typing with a merging keyboard. *Journal on Multimodal User Interfaces*, 13(4):321–334, 2019.

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

[4] D. A. Bowman, C. J. Rhoton, and M. S. Pinho. Text input techniques for immersive virtual environments: An empirical comparison. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 46(26):2154–2158, 2002. doi: 10.1177/154193120204602611

[5] J. Cechanowicz, S. Dawson, M. Victor, and S. Subramanian. Stylus based text input using expanding cirrin. In *Proceedings of the working conference on Advanced visual interfaces*, pp. 163–166, 2006.

[6] F. Evans, S. Skiena, and A. Varshney. Vtype: Entering text in a virtual world. *submitted to International Journal of Human-Computer Studies*, 1999.

[7] C. R. Foy, J. J. Dudley, A. Gupta, H. Benko, and P. O. Kristensson. Understanding, detecting and mitigating the effects of coactivations in ten-finger mid-air typing in virtual reality. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–11, 2021.

[8] W. Garbe. SymSpell, 6 2012.

[9] J. Gong, Z. Xu, Q. Guo, T. Seyed, X. Chen, X. Bi, and X.-D. Yang. Wristext: One-handed text entry on smartwatch using wrist gestures. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1–14, 2018.

[10] Z. Gu, X. Xu, C. Chu, and Y. Zhang. To write not select, a new text entry method using joystick. In *International Conference on Human-Computer Interaction*, pp. 35–43. Springer, 2015.

[11] J. Gugenheimer, D. Dobbelstein, C. Winkler, G. Haas, and E. 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*, pp. 49–60, 2016.

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

[13] S. G. Hart. Nasa-task load index (nasa-tlx); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*, vol. 50, pp. 904–908. Sage publications Sage CA: Los Angeles, CA, 2006.

[14] A. Huckauf and M. H. Urbina. Gazing with peyes: towards a universal input for various applications. In *Proceedings of the 2008 symposium on Eye tracking research & applications*, pp. 51–54, 2008.

[15] H. Jiang and D. Weng. 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, 2020. doi: 10.1109/VR46266.2020.00092

[16] H. Jiang, D. Weng, Z. Zhang, Y. Bao, Y. Jia, and M. Nie. Hikeyb: High-efficiency mixed reality system for text entry. In *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 132–137, 2018. doi: 10.1109/ISMAR-Adjunct.2018.00051

[17] H. Jiang, D. Weng, Z. Zhang, and F. Chen. Hifinger: One-handed text entry technique for virtual environments based on touches between fingers. *Sensors*, 19(14):3063, 2019.

[18] J. Kim, W. Delamare, and P. Irani. Thumbtext: Text entry for wearable devices using a miniature ring. In *Graphics Interface*, 2018.

[19] Y. R. Kim and G. J. Kim. Hovr-type: Smartphone as a typing interface in vr using hovering. In *2017 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 200–203. IEEE, 2017.

[20] P. Knierim, V. Schwind, A. M. Feit, F. Nieuwenhuizen, and N. 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*, pp. 1–9, 2018.

[21] L. Lik-Hang, Z. Yiming, Y. Yui-Pan, T. Braud, S. Xiang, and P. Hui. One-thumb text acquisition on force-assisted miniature interfaces for mobile headsets. In *2020 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 1–10. IEEE, 2020.

[22] J.-W. Lin, P.-H. Han, J.-Y. Lee, Y.-S. Chen, T.-W. Chang, K.-W. Chen, and Y.-P. Hung. Visualizing the keyboard in virtual reality for enhancing immersive experience. In *ACM SIGGRAPH 2017 Posters*, pp. 1–2. 2017.

[23] Y. Lu, C. Yu, X. Yi, Y. Shi, and S. 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):1–24, 2017.

[24] I. S. MacKenzie. Kspc (keystrokes per character) as a characteristic of text entry techniques. In *International Conference on Mobile Human-Computer Interaction*, pp. 195–210. Springer, 2002.

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

[26] I. S. MacKenzie and R. W. Soukoreff. Phrase sets for evaluating text entry techniques. In *CHI'03 extended abstracts on Human factors in computing systems*, pp. 754–755, 2003.

[27] J. Mankoff and G. D. Abowd. Cirrin: A word-level unistroke keyboard for pen input. In *Proceedings of the 11th annual ACM symposium on User interface software and technology*, pp. 213–214, 1998.

[28] M. McGill, D. Boland, R. Murray-Smith, and S. 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*, pp. 2143–2152, 2015.

[29] D.-M. Pham and W. Stuerzlinger. Hawkey: Efficient and versatile text entry for virtual reality. In *25th ACM Symposium on Virtual Reality Software and Technology*, pp. 1–11, 2019.

[30] S. Pick, A. S. Puika, and T. 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)*, pp. 109–112, 2016. doi: 10.1109/3DUI.2016.7460039

[31] M. Prätorius, U. Burgbacher, D. Valkov, and K. Hinrichs. Sensing thumb-to-finger taps for symbolic input in vr/ar environments. *IEEE computer graphics and applications*, 35(5):42–54, 2015.

[32] M. Proschowsky, N. Schultz, and N. E. Jacobsen. An intuitive text input method for touch wheels. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pp. 467–470, 2006.

[33] R. Qin, S. Zhu, Y.-H. Lin, Y.-J. Ko, and X. 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*, pp. 137–146, 2018.

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

[35] R. W. Soukoreff and I. S. MacKenzie. Metrics for text entry research: An evaluation of msd and kspc, and a new unified error metric. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 113–120, 2003.

[36] M. Speicher, A. M. Feit, P. Ziegler, and A. Krüger. Selection-based text entry in virtual reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1–13, 2018.

[37] D. Venolia and F. Neiberg. T-cube: A fast, self-disclosing pen-based alphabet. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 265–270, 1994.

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

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

[40] J. Walker, B. Li, K. Vertanen, and S. Kuhl. Efficient typing on a visually occluded physical keyboard. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 5457–5461, 2017.

[41] E. Whitmire, M. Jain, D. Jain, G. Nelson, R. Karkar, S. Patel, and M. Goel. Digitouch: Reconfigurable thumb-to-finger input and text entry on head-mounted displays. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(3):1–21, 2017.

[42] Wikipedia contributors. Letter frequency — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Letter$_{f}requencyoldid = 1053878032, 2021. [Online; accessed7 - November - 2021]$.

[43] P. C. Wong, K. Zhu, and H. 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*, pp. 1–10, 2018.

[44] W. Xu, H.-N. Liang, Y. Zhao, T. Zhang, D. Yu, and D. 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. doi: 10.1109/TVCG.2019.2898736

[45] Z. Xu, P. C. Wong, J. Gong, T.-Y. Wu, A. S. Nittala, X. Bi, J. Steimle, H. Fu, K. Zhu, and X.-D. Yang. Tiptext: eyes-free text entry on a fingertip

keyboard. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, pp. 883–899, 2019.

[46] X. Yi, C. Yu, W. Xu, X. Bi, and Y. Shi. Compass: Rotational keyboard on non-touch smartwatches. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 705–715, 2017.

[47] X. Yi, C. Yu, M. Zhang, S. Gao, K. Sun, and Y. 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*, pp. 539–548, 2015.

[48] C. Yu, Y. Gu, Z. Yang, X. Yi, H. Luo, and Y. 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*, pp. 4479–4488, 2017.

[49] D. Yu, K. Fan, H. Zhang, D. Monteiro, W. Xu, and H.-N. Liang. Pizzatext: Text entry for virtual reality systems using dual thumbsticks. *IEEE Transactions on Visualization and Computer Graphics*, 24(11):2927–2935, 2018. doi: 10.1109/TVCG.2018.2868581

[50] S. Zhu, T. Luo, X. Bi, and S. Zhai. Typing on an invisible keyboard. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1–13, 2018.