• Article •

Eye-shaped Keyboard for Dual-hand Text Entry in Virtual Reality

Kangyu Wang[1], YangQiu Yan [1], Hao Zhang[1], Xiaolong Liu[1], Lili Wang[1*]

1. *State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China*

*** Corresponding author**, wanglily@buaa.edu.cn

Received: Accepted:

Abstract We propose the eye-shaped keyboard, a keyboard for high-speed text entry in VR. The eye-shaped keyboard has the shape of dual eyes with characters arranged along the curved eyelids, which ensures a low density and short spacing of the keys. The eye shaped keyboard references the QWERTY key sequence, allowing users to benefit from their experience with the QWERTY keyboard. The user interacts with the eye-shaped keyboard using rays controlled by dual hands. A character can be entered in one step by moving rays from the inner eye regions to the regions of the characters. A high-speed auto-complete has been designed for the eye-shaped keyboard. We conducted a pilot study to ascertain the best parameters for the eye-shaped keyboard. Then we carried out a user study to compare our eye-shaped keyboard with the QWERTY keyboard and the circular keyboard. For beginners, the eye-shaped keyboard performed significantly more efficiently, more accurately with less task load and less hand movement than circular keyboard. Compared with QWERTY keyboard, the eye-shaped keyboard is more accurate, significantly reduces hand translation while keeping similar efficiency. Finally, to evaluate the potential of eye-shaped keyboard, we conducted another user study. In this study, participants were asked to type continuously for 3 days using the eye-shaped keyboard, 2 sessions per day. In each session, participants needed to type for 20 minutes, and then their typing performance was tested. The eye-shaped keyboard has been proved to be efficient and potential, with an average speed of 19.89 words per minute (WPM), and a mean uncorrected error rate of 1.939%. The maximum speed reached 24.97 WPM after 6 sessions and still trending upwards.

Keywords Virtual reality; Human computer interaction; text entry; shaped keyboard; QWERTY-like key sequence; dual-hand cooperation

1 Introduction

Fast and accurate text entry is one of the most important functions that must be provided in computer systems. In virtual reality (VR), text entry is an extremely common interaction, such as entering account passwords, chatting via typing, and taking text notes. A well-designed text entry method needs to balance method compatibility with VR devices, input speed, input error rate and learning difficulty. But there are numerous challenges with text entry methods in VR. Firstly, variety of available VR devices are limited, with joysticks and headsets being the most reachable devices. Secondly, compared with non-VR environment, VR devices can not provide the sense of touch and lack the feedback provided by real objects when entering text. Thirdly, it is difficult for users to balance precision and speed with their movements. The current common input methods are physical keyboards and QWERTY virtual keyboards, novel virtual keyboards, etc. But they each have their own limitations. Traditional physical keyboards are difficult to introduce into VR. They are not compatible with current mainstream input methods for VR and reduce the immersion. The virtual QWERTY keyboard reduces learning difficulty but suffers from low input efficiency, high error rates and complex input process. Other novel virtual keyboards tend to have shortcomings in terms of learning difficulty, speed of text entry, input errors or task load of use.

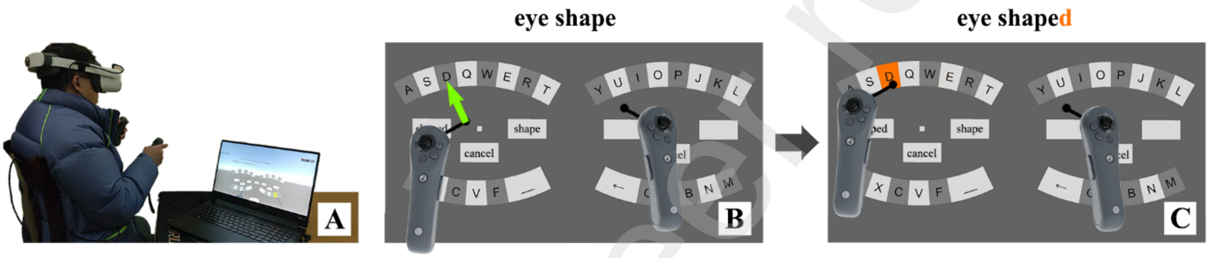

Figure 1 (A) Third-person perspective: the two controllers are used to enter text via the eye-shaped keyboard; (B and C) To complete the text "eye shaped" after the user has already entered "eye shape", the user moves the ray to the region of the letter "D" by moving the left-handed handle along the green arrow. The entered field is displayed at the top of the figure.

In this paper, we propose eye-shaped keyboard, a novel keyboard for text entry in VR. The eye-shaped keyboard is shown in Figure 1. The keyboard has a dual-eye layout, which provides a small distance between keys at a low key density. Our keyboard references the key sequence of the QWERTY keyboard to make it easy for the user to learn. The keyboard requires two hands for text entry, providing a well-designed cooperation between dual hands for input efficiency. By this method, one character can be entered in one step.

We evaluated the performance of our eye-shaped keyboard through three progressive studies. First, we designed a pilot user study to select the best parameters (upper and lower eyelid spacing, color scale marking for keys, recoloring of rays and key sequences) for the keyboard. We then tested our text entry method in a user study, where we compared the text entry performance of our technique, eye-shaped keyboard, with two other techniques: keyboard with QWERTY layout and with circular layout. In addition to typing speed, error rate, workload and SSQ, we propose the comparative measure of user hand movement, which reveals the mechanical load of user's hands. Based on the results, we found that for beginners, the eye-shaped keyboard performed significantly more efficiently, more accurately with less task load and less hand movement than circular keyboard. Compared with QWERTY keyboard, the eye-shaped keyboard is more accurate, and significantly reduces hand translation while keeping similar efficiency as the QWERTY keyboard. Then we conducted another user study to carry out an experiment of

6 sessions to evaluate the potential of eye-shaped keyboard on speed and error rates. Participants did a certain amount of exercise by typing with eye-shaped keyboard and got test for their typing performance in each session. The final experimental results show that the novice users (those who have no experience with eye-shaped keyboard) can achieve an average speed of 17.24 words per minute (WPM) (s.e. = 1.33) with 2.04% (s.e. = 1.57%) of the not corrected error rate. The four adept users (those who have participated in previous studies) can reach an average of 22.54 WPM (s.e. = 2.14) with 2.02% (s.e. = 1.01%) of the not corrected error rate. The maximum typing speed achieves 24.97 WPM in the last session. From this study, our eye-shaped keyboard is proved to be efficient and potential, and the performance still tends to be better.

The contributions of our paper include: (1) an eye-shaped layout keyboard for text entry in VR; (2) a new hand movement metric for comparison among manual text entry techniques in VR; (3) comparison among text entry keyboard for eye-shaped, circular and QWERTY layouts in VR; and (4) a manifestation of the utility of eye-shaped keyboard through a 3-day user study.

2 Related works

Designing an input method with high efficiency and low task load in virtual environment is a challenging problem. Shaped keyboard is one of the main ideas. The specific layouts of keyboards are various, and the circular layout shows satisfying performance, which also enlightened us the eye-shaped layout. Besides, ideas of transferring the QWERTY layout or similar key sequence from physical keyboard to virtual text entry environment and two-handed interaction method also promote our method design.

2.1 Circular layout

As early as the end of the nineteenth century, the circular layout was applied on text entry for pen-based computers or other equipments (such as T-Cube [1] and Cirrin[2]). Later, keyboards with circular layout were used with varied auxiliary input devices or even without them. Among them, watch or watch-like device is one alternative to the stylus which is also an excellent match with the circular layout. Katsuragawa et al. designed a text entry method for smartwatches using gestural interaction and visual feedback of circular layout entities[3]. User's typing speed reached as high as 8.42 WPM. Yi et al.[4] and Gong et al.[5] also applied the circular layout to their input methods on watches, called COMPASS and WrisText respectively. The average speed of COMPASS reaches 8.7 WPM, while for WrisText, the typing speed of novice reached up to 15.2 WPM and achieved as fast as 24.9 WPM after more training. They all reached a relatively high level of typing speed, but using them requires additional auxiliary equipment, which is not conducive to the promotion of the corresponding input methods.

Many of previous works about text entry method in virtual environment also employ the layout of keyboard in a circular way, with default devices instead of additional equipment. Yu et al. proposed PizzaText, with a circular virtual keyboard and a handle to input[6]. The user twists left joystick of the handle to the direction of the slice of pie containing the target letter, and then tilts right joystick to choose the target letter within the pie. Performance of PizzaText on text entry speed reaches 10.20 WPM by novices and 14.32 WPM by experts. RingText[7], designed by Xu et al., only uses ray of HMD to move the cursor to hit and select the letters, which lies between the inner and outer part of the circular keyboard. It reports an average text entry speed of 8.74 WPM and a maximum speed of 13.24 WPM after a period of training.

All of the above input methods using a circular layout have shown great potential in terms of text entry speed. In the circular layout, each key is at a similar distance from the default viewing focus and cursor position (i.e., the center of the circle), which may give the user a better grasp of the layout. In addition, the direction-based selection reduces the requirement for movement accuracy to some extent.

2.2 QWERTY key sequence

Lots of previous works tried to transfer the standard QWERTY keyboard into virtual environment. One natural approach is to use a physical QWERTY keyboard, and map it into a keyboard of the same (or similar) shape in virtual world to provide touch and visual feedback at the same time. Kim et al. used cameras or wired trackers to track the user's hands and the keyboard and mapped them to virtual ones[8]. Walker et al. developed an typing assistance system for users with HMD or invisible physical keyboard[9]. Instead of showing the entire hands, the system only highlights the keys pressed recently on the virtual keyboard. But with the increasing accuracy of motion tracking technology, some text entry methods employed advanced object positioning and motion tracking devices (Motive 1.10 motion capture software, Vive Lighthouse System etc.) to track the user's hands and the real keyboard and to reflect them to models in the virtual environment in different forms like skeleton or joint node of hands[10,11,12,13]. Some previous works discussed the feasibility of inlaying the video of hands and keyboard captured by camera on HMD rather than generating models of them in virtual environment[13,14].

Another approach is to build a completely virtual QWERTY keyboard. Yu et al. discussed some text entry methods that requires users to move their heads with HMDs to control the cursor tap or dwell on the target keys of the virtual keyboard to select letters[15]. Speicher et al. discussed several methods for the interaction between the controllers and the virtual QWERTY keyboard, such as head pointing, controller pointing, controller tapping, etc.[16]. Mehring et al. designed KITTY, which enables the user to type with the keyboard in virtual environment by touching with special gloves[17]. Xu et al. used the same virtual QWERTY keyboard to compare different selection methods of typing[18]. Boletsis et al. evaluated well-known Drum-Like VR Keyboard of QWERTY layout, which achieved a high text entry rate of 24.61 WPM as well as moderate-to-high total error rate of 7.2%[19].

Methods of physical QWERTY keyboard show great performance, but they requires additional peripherals--the keyboard, which is not expected in most VR application scenario. But whether it is a physical or virtual QWERTY keyboard, the main idea is to maintain the relative position of the keys of the QWERTY keyboard in the virtual environment to increase familiarity and enable users to transfer skills. The subjective feedback we received in our experiments corroborates this. Therefore, using a virtual QWERTY keyboard or maintaining a similar key sequence will better meet the needs of VR text entry, similar to the rational of QwertyRing by Gu et al.[20].

2.3 Interaction methods

Two-handed input is widely used in both VR and non-VR environments, and may be in line with users' habits. Most VR devices also provide hardware support for two-handed input. Compared with other interaction methods, well-designed two-handed methods, taking advantage of the dexterity and parallelism of both hands, are more accurate and efficient. PizzaText of great performance, as we mentioned above, used two dual joysticks of one handle controller to select the letters, whose control requires user's two hands or two thumbs[6]. Similarly, 2-Thumbs Typing (2TT) by Zhang et al. also requires users to move their thumbs to enter texts[21]. But the difference is that 2TT uses two touchpads of HTC VIVE controllers and users need to make single stroke gestures on them to input letters. Average text entry speed with this method reaches 6.35 WPM. One possible explanation for the relatively low speed is that mapping single letter to multi-step two-finger gesture may reduce the efficiency. The various input methods mentioned above using physical QWERTY keyboards occupy all the fingers of the user's two hands, such as the method proposed by Knierim et al.[12], methods compared by Grubert et al.[13], qVRty proposed by Hoppe et al.[11], method developed by Bovet et al.[10], etc. Most virtual QWERTY keyboard occupies two hands as two wholes, and one hand operates a device (such as a handle) to select keys and input characters. The interaction methods of Controller Pointing, Controller Tapping, Discrete & Continuous Cursor discussed by Speicher et al. are examples[16].

Text entry by head motion or gazing are also two popular subjects in previous research. Yu et al. discussed performance of the HMD-based text entry techniques with different confirmation methods[15]. RingText is another well-performed text entry method only using HMD[7]. However, under the current technical conditions, the weight of VR HMDs is still relatively high. For example, although the weight of the HMD of Oculus Quest 2 is reduced to about 500 grams, it is still very heavy compared to about 10 grams of ordinary glasses. Using VR HMD to select characters by head translation and rotation still causes a large task load.

Besides the gaze-based text entry technique we mentioned before (pEYEs[22]), Ma et al. proposed a system combining steady-state visual evoked potentials (SSVEP) and eye tracking for text entry in virtal reality, which reported a text entry speed of around 10 WPM[23]. Majaranta et al. improved dwell-to-choose gaze-based input method by using a regulable dwell time, which increased the text entry speed of this method to 19.89 WPM[24]. Rajanna et al. examined the impact of keyboard size (without-view or within-view), selection method (dwell or click) and motion (sitting or in motion) on typing performance and user experience of gaze-based text entry technique, only to find that within-view keyboard performs better (10.15 WPM) than without-view keyboard (9.15 WPM)[25]. Click is better than dwell and stationary environment cultivates user's input performance rather than motion environment. However, there are two limitations of gaze-based input method. One is that eye-tracking devices are expensive, limiting their widespread use. The other is that, for this technique, gazing is not only used for confirmation, but also used when users check the target before they decide to select. In this case, dwell is very necessary to separate these two processes to lower the error rate, which innately causes low typing speed.

3 Methods

In this section, we propose the eye-shaped keyboard for text entry. We first introduced some design rationales for VR text entry in section 3.1. Subsequently, we introduced the keyboard layout of our eye-shaped keyboard in detail in section 3.2. The text entry process using eye-shaped keyboard and other interactions of the eye-shaped keyboard are introduced in section 3.3.

3.1 Design rationales

We have developed eye-shaped keyboard for text entry in VR environment and the selection of eye-shaped layout is based on the following rationales.

Circular-like Layout

Circular keyboards are widely used in VR environments and many of them have good performance in terms of text entry speed and task load. We think that the direction-based selection contributes a lot and have chosen a curved key layout, similar to the circular layout.

QWERTY-like Key Sequence

In order to make use of user's experience with the QWERTY keyboard, many keyboards in the VR environment retain the QWERTY sequence. We also use a QWERTY-like key sequence, which is suitable for curved keyboard layout.

Greater Horizontal Width

When users move the rays and enter text in VR environment, they prefer to move horizontally[15]. Therefore, making the horizontal width of the keyboard larger than the vertical height may help to improve typing efficiency.

Two-handed Input

Two-handed input is the majority in QWERTY keyboards, with higher speed and lower task load for each hand. Because two hands have a greater range of movement compared to single hand, two-handed keyboards can be designed with a lower key density, which may improve text entry accuracy.

3.2 Keyboard layout

We designed the eye-shaped keyboard based on the above rationales. First, we placed two circular keyboards in the left and right hand positions, which users can operate with both hands separately. Next, based on the users' preference for horizontal motion, we adjusted the circles to ellipses whose width is greater than their height. Since the leftmost and rightmost parts of the ellipses are too far from the center, these parts are discarded. Finally, we designed the corresponding QWERTY-like key sequences for the above eye-shaped layout.

In user's view, the eye-shaped keyboard is floating in front of the user. The eye-shaped keyboard is divided into two parts, the left eye and the right eye. Both two eyes are divided into the upper eyelid part and the lower eyelid part. The upper eyelid and the lower eyelid are arcs on two different circles with the same radius, and the line connecting the centers of the two circles is perpendicular to the ground. The angle between the two planes where the eyes are located is less than 180 degrees.

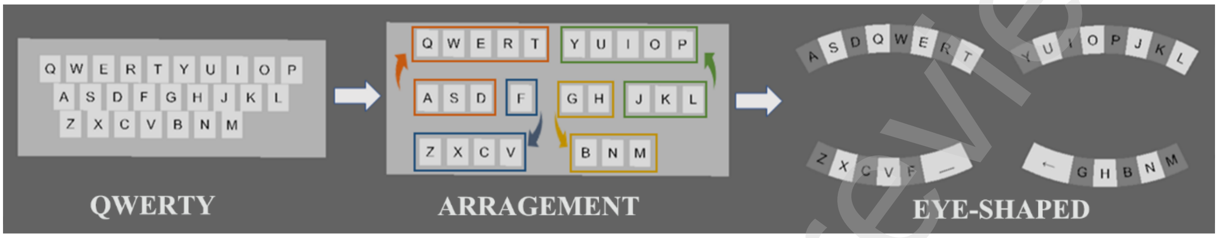

Figure 2 (A) Transition from QWERTY key sequences to the sequences of eye-shaped keyboard. QWERTY key sequences are divided into 8 parts and then reassembled into the 4 parts of the eye-shaped keyboard.

As is shown in Figure 2, the key sequence of the upper left eyelid is a-s-d-q-w-e-r-t, including the three leftmost keys a-s-d in the second row of the QWERTY keyboard, and the five leftmost keys q-w-e-r-t in the first row of the QWERTY keyboard, maintaining the horizontal adjacent relationship in QWERTY key sequence. Similarly, the key sequence of the lower left eyelid is z-x-c-v-f-g and space, including the second and third rows of normal QWERTY keyboard. Due to the high using frequency of the space bar, it is designed to be twice the size of a regular key. The buttons on the right are arranged in the same way. The eye-shaped keyboard can be switched to special character mode by changing the letters on the keyboard to numbers and other characters.

3.3 Entry interaction on eye-shaped keyboard

Entry Process

User operates two handles to move the rays in VR environment to select the letter with our eye-shaped keyboard. The eye-shaped keyboard can be divided into two kinds of regions: key regions, where the letter keys are located, and inner eye regions, which refer the empty space between the upper and lower eyelids of the same left or right eye.

In Figure 3, when a user wants to enter a letter, firstly he/she needs to move the ray to the inner eye region, then he/she moves the ray from the inner eye region to the region of the key corresponding to target letter and then go back to the inner eye region before inputting the next letter. If the rays have not returned to the inner eye region, subsequent input operations will be ignored. If the user moves the joystick of the left handle up and performs the above-mentioned letter input operation, the uppercase form of the corresponding letter will be input. Only the ray controlled by the left hand can trigger the keys in the left eye region, and vice versa.

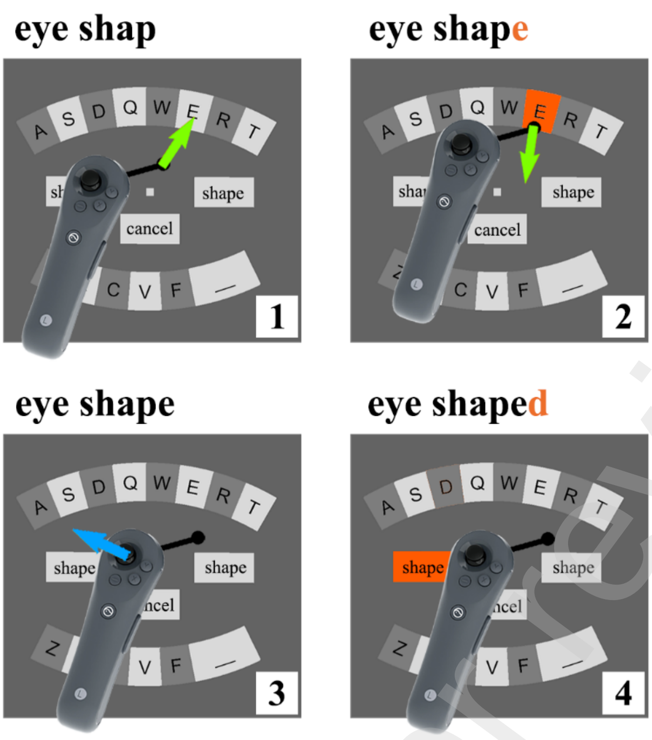

Figure 3 Eye-shaped keyboard text entry with auto-complete (left-handed example). The user has entered 'eye shap' and the target is 'eye shaped'. (1) The user first moves the ray up along the green arrow and into the region of the letter 'e'. This letter will then be entered. (2) The user then moves the ray down along the green arrow back to the inner eye region. (3 and 4) The auto-complete shows "shaped" in the first position so the user moves the left-hand joystick along the blue arrow to the left, completing the full word "shaped".

Switch to Special Character Keyboards

When moving the right joystick up, the keyboard will switch among the normal keyboard and the special character keyboards.

Ray Direction

We set the direction of the pointing ray to the handle at a 65-degree angle, instead of the conventional direction along the handle. We tested multiple angles and chose the angle, which is more comfortable for the user to turn the wrist and control the ray direction.

Text Entry Feedback

Some feedback has been added to our interactions to help users select keys easily. One is the highlighting of the selected key. Another is the visualization of the endpoints of the rays when they intersect the eye-shaped keyboard. One more is that the grip vibrates and the HMD beeps when a key is triggered.

Text Entry Assistance

The eye-shaped keyboard has an integrated auto-complete that will infer words based on the letters the user has entered and list the four words the user is most likely to want in the center of the two eyes. As is shown in Figure 3, the user pushes the left stick to the left to select the first candidate and to the right to select the second candidate. Similarly, the right-hand joystick corresponds to the third and fourth candidate words. When the joystick is moved but not released, moving it down cancels the selection of the candidate.

4 Keyboard layout variants

In this section, we first give the basic layout of the eye-shaped keyboard. Then several variants are provided, which will be discussed in pilot study.

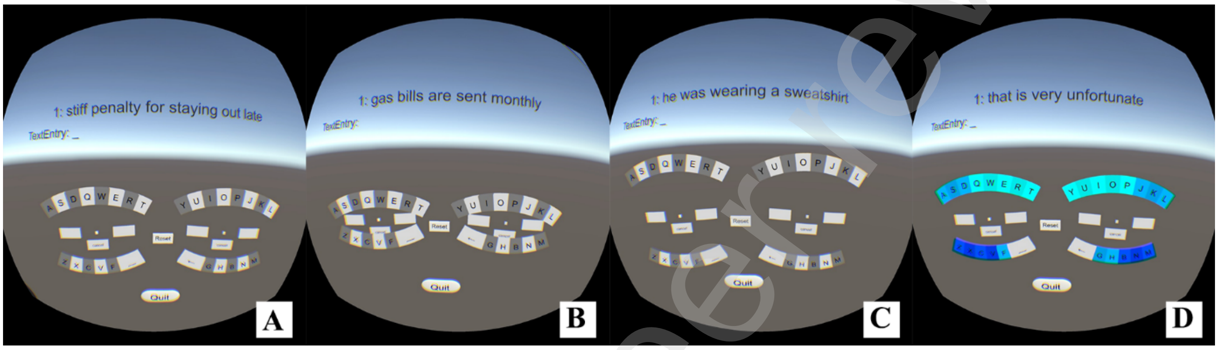

Figure 4 The basic layout and the variants of the eye-shaped keyboard in pilot user study. (A) The basic layout of eye-shaped keyboard. (B) The variant that has smaller (0.1m) upper and lower eyelid spacing compared to the basic layout. (C) The variant that has larger (0.3m) upper and lower eyelid spacing. (D) The variant that uses color scale marking for keys. Similar depths of blue mark the same sequence of keys in QWERTY keyboard. Apart from the differences mentioned above, all other parameters are identical.

Basic Layout of Keyboard

Basic layout of keyboard is shown in Figure 4 (A). The position of the keyboard in front of the user is 0.65m. The total height and width of each eye part are 0.35m and 0.3m. The distance between the centers of the two eye parts is 0.4m. The distance between the centers of the medial arcs of the upper and lower eyelids, referred as the upper and lower monocular spacing, is 0.2m. The left eye and right eye have an equal number of 14 keys, including a key that takes up the size of two keys. Each key has a curvature of $2\pi/48$, an inner diameter size of 0.3m and a thickness of 0.05m. The two planes in which the eye parts are located are perpendicular to the user's line of sight, and the angle between the two eye planes is calculated to be 18.43 degrees.

With these parameters, the field of view (FoV) of eye-shaped keyboard is about 101°, which is promising and let users see each letter clearly with HMDs[26]. We also ran some tests with these parameters and got good feedback.

Variant1: Upper and Lower Eyelid Spacing

Differences in upper and lower eyelid spacing will alter the vertical travel distance of the ray, which may have an effect on the speed and accuracy of key selection. There are three levels for this variant: 0.1m (Figure 4 B, denoted as level1), 0.2m (Figure 4 A, denoted as level2, adopted by basic layout) and 0.3m (Figure 4 C, denoted as level3).

Variant2: Color Scale Marking for Keys

Eye-shaped keyboard references QWERTY key sequence. In eye-shaped keyboard, clearly identifying the keys from different rows of the QWERTY keyboard may help novice users quickly find the position of the keys of eye-shaped keyboard. D in Figure 4 shows the keyboard using color scale marking, namely using different colors to mark the keys on the keyboard. Three progressively darker blue are used to mark the letters from different rows of the QWERTY keyboard. There are two levels for this variant: enabling color scale marking for keys (denoted as level1) and disabling (denoted as level2, adopted by basic layout).

Variant3: Recoloring of Rays when Triggerable

Once the user has entered a character, they must return to the inner area of the eye before the next entry can be made. Helping users determine when they can re-enter characters may help to improve the typing performance. The variant will change the ray color as the ray returns to the inner eye region, thus indicating to the user that the characters can be entered. There are also two levels for this variant: enabling recoloring of rays (denoted as level1) and disabling (denoted as level2, adopted by basic layout).

Variant4: Alphabetical Key Sequence

Most users have experience with QWERTY key sequences. They may benefit from the experience when using the eye-shaped keyboard with QWERTY-like key sequences. The variant uses the alphabetic key sequences. The letters A through P are sequentially arranged in the upper eyelid region of both eyes, and letters Q through Z are sequentially arranged in the lower eyelid region of both eyes. We will compare QWERTY-like sequences with alphabetic key sequences for the eye-shaped keyboard, which can show the enhanced effect of QWERTY-like sequences.

5 Pilot study

We have evaluated our eye-shaped keyboard text entry method in a 10-people pilot study, which helped us choose the most promising parameter configuration.

5.1 Study design

Experiment condition designation

The experiment is designed using a within-subject design. Keyboards in different experimental conditions employed different levels of variants discussed in Section 4. The first experimental condition was the layout of the basic keyboard layout (PEC1), which is mentioned in section 4. The eye-shaped keyboard in PEC2 adopted the level1 of Variant1, while keyboard in PEC3 adopted the level3 of Variant1. Keyboard in PEC4 used level1 of Variant2. Keyboard of PEC5 used level1 of Variant3. Besides, in PEC6, the keyboard was set in alphabetical sequence to find out how different sequence affect typing.

Each experiment differed from PEC1 in only one independent variant, while the unmentioned conditions were the same. Detailed designation of all experiment conditions is shown in Table 1.

Table 1 Detailed experiment conditions designation. The meaning of variants is the same as Section 4.

Cond.	Spacing	Color Scale	Recoloring	Key Sequence
PEC1 (Basic)	0.2m	Disabled	Disabled	QWERTY
PEC2	0.1m	Disabled	Disabled	QWERTY
PEC3	0.3m	Disabled	Disabled	QWERTY
PEC4	0.2m	Enabled	Disabled	QWERTY
PEC5	0.2m	Disabled	Enabled	QWERTY
PEC6	0.2m	Disabled	Disabled	Alphabetical

Participants and Apparatus

We recruited N = 10 participants, consisting of NM = 8 male participants and NF = 2 female participants. Their age range is from 19 to 23. Among these subjects, 2 participants have experience using VR devices, and 4 participants are able to touch-type on a physical QWERTY keyboard. All participants are familiar with English but are not native speakers. Eye-shaped keyboard program is built with Unity3D. The VR device used in the test was the Pico Neo 2, and the grips used are the ones that came with the Pico Neo 2.

Task

In each experimental condition, each participant was asked to spend 8 minutes typing to become familiar with the eye-shaped keyboard in the corresponding experimental condition, and then transcribe 5 phrases as a test in that experimental condition. All experimental conditions are traversed in random order. All phrases were randomly extracted from MacKenzie's phrasebook with no duplicates[27]. Each phrase would be displayed in the upper part of the user view with current transcribed text.

Metrics

We evaluated the performance of our eye-shaped keyboard from two aspects: text entry speed and error rate. The specific measurement method is as follows.

Text entry speed. Text entry speed is measured in WPM, with a word defined as five consecutive letters, including spaces.

Error rate. The error rate is calculated based on the standard typing metrics[28], where the total error rate (TER) = not corrected error rate (NCER) + corrected error rate (CER).

Statistical analysis method

In order to highlight the impact of single parameter, we compared PEC2-6 with PEC1. The normal distribution of the data was confirmed using the Kolmogorov-Smirnov test and Shapiro-Wilk test. The means of normally distributed data were compared using Paired Sample T test. For data that did not fit the normal distribution, we used the Wilcoxon signed rank test for related samples.

5.2 Results

Table 2 Text entry speed of pilot study, measured by WPM. ROI shows the rate of increase of text entry speed in this experimental condition relative to PEC1. Significant differences are indicated with an asterisk

Condition	Avg ± std. dev.	ROI	P
PEC1	15.57 ± 2.90	0.00%	
PEC2	13.56 ± 2.79	-12.91%	0.002*
PEC3	13.55 ± 2.81	-12.97%	0.005*
PEC4	14.95 ± 2.99	-3.98%	0.388
PEC5	14.71 ± 3.14	-5.52%	0.365
PEC6	10.89 ± 2.53	-30.06%	0.001*

Text entry speed. Table 2 illustrates mean text entry speed for all conditions. PEC2-6 were compared to PEC1. Text entry speed of PEC1 is highest. Text entry speed of PEC2 and PEC3 is significantly lower ($p < 0.05$) than PEC1, and the decrease ratio reached nearly 13%. Text entry speed is higher in PEC1 compared to PEC4 and PEC5, but the difference is not significant. The decrease ratio is nearly 4% and 5.5% respectively. Furthermore, compared to QWERTY sequence, alphabetic sequence (PEC6) significantly reduces typing efficiency. The reduction ratio reaches 30%.

Error rate. Table 3 and Table 4 shows TER and NCER for experimental conditions respectively. PEC2-6 were compared to PEC1. In terms of both TER and NCER, average performance of text entry with PEC1 is the best among all conditions. But the difference in TER was not significant and ROI of TER between other PEC and PEC1 is from 13.46% to 36.54%. For NCER, the error rate of PEC1 is significantly lower than PEC2, PEC5 and PEC6. And the ROI of NCER is higher than 100%. Compared to PEC3 and PEC4, the effect is not significant. But the ROI of PEC3 relative to PEC1 reaches 477.78%.

Table 3 Error rate of pilot study, measured by TER. ROI shows the rate of increase of TER in this experimental condition relative to PEC1. Significant differences are indicated with an asterisk

Condition	Avg ± std. dev.	ROI	P
PEC1	0.104 ± 0.077	0.00%	
PEC2	0.142 ± 0.074	36.54%	0.323
PEC3	0.123 ± 0.638	18.27%	0.619
PEC4	0.118 ± 0.616	13.46%	0.647
PEC5	0.136 ± 0.068	30.77%	0.438
PEC6	0.120 ± 0.069	15.38%	0.673

Table 4 Error rate of NCER in pilot study. ROI shows the rate of increase of NCER in this experimental condition relative to PEC1

Condition	Avg ± std. dev.	ROI	P
PEC1	0.009 ± 0.012	0.00%	
PEC2	0.018 ± 0.016	100.0%	0.040*
PEC3	0.052 ± 0.072	477.78%	0.096
PEC4	0.015 ± 0.017	66.67%	0.449
PEC5	0.033 ± 0.020	266.67%	0.011*
PEC6	0.027 ± 0.029	276.80%	0.039*

The pilot study identified that our most promising layout is the configuration of PEC1, with 0.2m of upper and lower eyelid spacing, enabling user to type in a higher speed with low error rate, compared to 0.1m (PEC2) and 0.3m (PEC3). The biggest drop of NCER is as high as 477.78% (PEC3 compared to PEC1). According to the result of the study, the color scale marking for keys (PEC4) and recoloring of rays did not promote the performance of text entry, but even has a negative effect (i.e. NCER of PEC5 was significantly higher than PEC1, and the drop reaches 266.67%). Based on interviews with the participants after the experiments, we infer that the reason may be the two techniques are not very useful for users who have received a certain amount of training, but may cause distraction or visual obstruction. So neither is applied. Additionally, comparison between PEC1 and PEC6 shows that, QWERTY sequence significantly increases typing efficiency while keeping a relatively low error rate. So the keyboard still maintains QWERTY sequence.

6 User study 1: comparison study

In this study, we compared the eye-shaped keyboard with two other keyboards - circular keyboard and QWERTY keyboard. The users were beginners, who were new to them. At the same time, the improvement effect of auto-complete on the performance of eye-shaped keyboard was also evaluated.

6.1 Study design

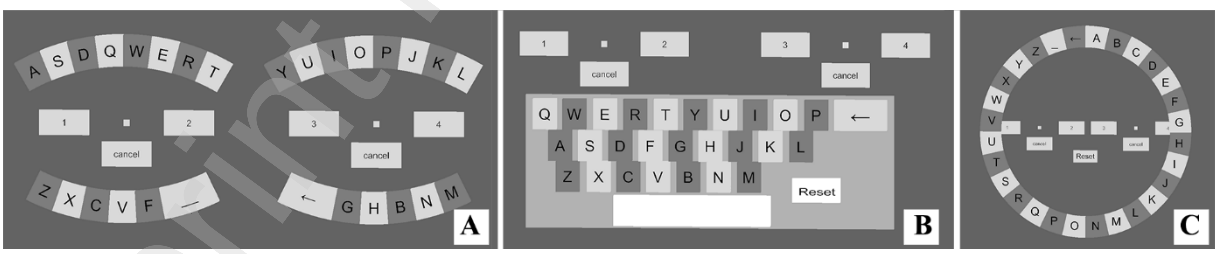

Figure 5 Eye-shaped keyboard (A) and QWERTY keyboard (B) and circular keyboard (C). These keyboards shown in the figure are equipped with auto-complete.

Experiment Condition Designation

We have compared our eye-shaped keyboard with (experimental condition 1, or EC1, Figure 5 A) and without auto-complete (EC2) to standard virtual QWERTY keyboard (control condition 1, or CC1, Figure 5 B) and keyboard with circular layout (CC2, Figure 5 C). For both EC1 and EC2, we had used the best parameter configuration from our pilot study. For each type of keyboard, we kept the graphical aspects the same and we also unified the input method into a handle ray. In order to make them perform their best, two CCs were both equipped with auto-complete.

Participants and Apparatus

We recruited N = 10 participants, consisting of NM = 6 male participants and NF = 4 female participants. Participants are between 19 and 28 with good or corrected vision. According to our pre-experiment questionnaire, none participants have experience using VR applications. All participants had experience using the physical QWERTY keyboard, but none of them could touch type. None of them are native English speakers, either. Condition of apparatus was the same as pilot study.

Task

We used a within-subject design, with only one independent variable: text entry method. The order of the four conditions was counterbalanced. For each condition, participants got an 8-phrase exercise of transcribing to warm up with corresponding text entry technique. After that, they were asked to take part in the text entry performance test, which required them to finish transcribing another 8 phrases. All phrases in the task are guaranteed to be randomly sampled from the MacKenzie's phrase set without repetition[27].

Metrics

Besides the metric mentioned in pilot study: WPM for typing speed, TER and NCER for error rate, we also designed and applied some more objective metrics to evaluate the text entry methods and compare their performance. Some subjective feedbacks were also used to measure the performance of different typing methods.

Motion of user's hands. In the user study, all text entry techniques use the handle ray as the input method. Less motion of hands means less effort to finally select the target letter, leading to less workload. In this experiment, we chose the average hand translation velocity (AHTV) and average angular velocity of hand rotation (AAVHR) as the metrics for motion of hands. Noticed that CC2 requires to use single handle to control the ray to select letters, but ECs and CC1 requires double hands, so we calculated the mean of double hands' translation velocity and average rotation angular velocity as the AHTV and AAVHR for ECs and CC1.

Subjective feedback. As for subjective feedback, we conducted NASA-TLX and SSQ to evaluate the workload and severity of Simulator Sickness caused by typing in VR environment with different methods. And we also did personal interviews to get some information of users' preference.

Statistical Analysis Method

Since the number of independent variable levels reached to 4, we employed a repeated measures ANOVA with Bonferroni corrections for pair-wise comparisons. We also used Mauchly's test to verify the sphericity assumption and Greenhouse-Geisser adjustment to correct for violations. We quantified effect magnitude using Cohen's D. Friedman test would be used if any of the above assumptions were not true. Other methods were the same as pilot study.

6.2 Results

Table 5 Average typing speed in each condition of Comparison Study, in WPM, and statistical comparison of the three conditions. Statistical significance is denoted with an asterisk

Condition	Avg ± std. dev.	Comparison	p	Cohen's D	Effect size
EC1	10.66 ± 1.47				
EC2	9.537 ± 1.43	EC1-EC2	0.044*	0.741	Medium
CC1	11.77 ± 2.00	EC1-CC1	0.030*	0.814	Large
CC2	8.65 ± 2.44	EC1-CC2	0.012*	0.985	Large

Text entry speed. The average typing speed is given in Table 5. Typing speed of EC1 is significantly higher ($p < 0.05$) than EC2 and CC2. The corresponding effects are of medium and large size, respectively. But the speed of EC1 is significantly lower than CC1, with a large size of effect. According to the result, auto-complete plays a significantly important role in enhancing the text entry speed. And eye-shaped layout is better than ring layout in promoting the typing speed. But the QWERTY layout seems to be better than our eye-shaped layout. We believe the key cause is the familiarity--8-phrase training seems not enough to make participants' familiarity with eye layouts comparable to QWERTY layouts. In User Study 2, or Growth Study, we will specifically train participants on eye-shaped layout to verify this conjecture.

Table 6 Average TER in each condition of Comparison Study

Condition	Avg ± std. dev.	Comparison	p	Cohen's D	Effect size
EC1	0.116 ± 0.05				
EC2	0.069 ± 0.03	EC1-EC2	0.021*	0.883	Large
CC1	0.177 ± 0.08	EC1-CC1	0.057	0.689	Medium
CC2	0.125 ± 0.06	EC1-CC2	0.616	0.164	V.small

Table 7 Average NCER of Comparison Study

Condition	Avg ± std. dev.	Comparison	p	Cohen's D	Effect size
EC1	0.036 ± 0.02				
EC2	0.022 ± 0.01	EC1-EC2	0.119	0.545	Medium
CC1	0.021 ± 0.01	EC1-CC1	0.084	0.614	Medium
CC2	0.042 ± 0.05	EC1-CC2	0.627	0.159	V.small

Error rate. The average error rate of TER and NCER are given in Table 6 and Table 7. ANOVA shows significant difference in TER between EC1 and EC2. TER of EC1 is higher than EC2, and Cohen's D revealed large effect size. For other comparisons in TER and NCER, ANOVA found no significant differences. Judging from experience, auto-complete may reduce the user's manual selection actions, and thus reduce the wrong selections they produce. But the result of experiment shows that auto-complete actually increases the total error rate. Based on observations of the participants' behavior, we speculate that this is because many participants preferred to choose some close but wrong option of word from auto-complete and correct it to type the target word more quickly (as shown in Figure 6). The reason for the huge difference between TER and NCER could also be the same. Based on the typing results (i.e. only focusing on NCER), the auto-complete improves the input speed, and does not significantly increase the error rate, so it is retained.

Sample: questioning the wisdom of the courts
Input Stream: question_←ing the_wisdom_of_the_court_←s

Figure 6 One example for the typing skill of auto-complete. Sample is the target phrase. Black letters are inputted without auto-complete, and blue ones are inputted by auto-complete. To input "questioning", the user entered "question_" with the help of the auto-complete, then deleted "_" and added "ing", which reduces the number of entries, but increases the TER.

Table 8 Average motion of user's hand in AHTV of Comparison Study. Measured by meters per second

Condition	Avg ± std. dev.	Comparison	p	Cohen's D	Effect size
EC1	0.158 ± 0.025				
EC2	0.183 ± 0.029	EC1-EC2	< 0.001*	1.667	V.large
CC1	0.214 ± 0.025	EC1-CC1	< 0.001*	2.252	Huge
CC2	0.202 ± 0.055	EC1-CC2	0.032*	0.800	Large

Table 9 Average motion of user's hand in AAVHR of Comparison Study. Measured by degrees per second

Condition	Avg ± std. dev.	Comparison	p	Cohen's D	Effect size
EC1	35.97 ± 6.44				
EC2	41.58 ± 5.34	EC1-EC2	0.013*	0.973	Large
CC1	31.71 ± 6.09	EC1-CC1	0.204	0.433	Small
CC2	41.39 ± 7.42	EC1-CC2	0.064*	0.668	Medium

Motion of user's hands. In Table 8 and Table 9, mean motion of user's hands is shown in AHTV and AAVHR. According to ANOVA analysis, users' AHTV in EC1 is significantly less than in EC2, CC1 and CC2. All comparisons show large size of effect (Cohen's D > 0.80). The size of effect of comparison between EC1 and EC2 is even very large (Cohen's D > 1.20) and between EC1 and CC1 is huge (Cohen's D > 2.0). And AAVHR of EC1 is also significantly less than EC2 and CC2, but insignificantly larger than CC1.The result reveals that motion of hands (translation and rotation) of EC1 is significantly less than EC2 and CC2. In other words, eye-shaped layout is significantly better that of circular layout in decreasing work load of user's hands. And auto-complete also has the same effect. Compared with CC1, the user's hand translation speed is significantly lower in EC1, but the hand rotation speed is insignificantly higher. So eye-shape keyboard significantly reduces the translation workload of hands, and keeps a similar rotation workload compared to QWERTY keyboard.

NASA-TLX and SSQ. ANOVA shows that different entry methods have no significant effect on the degree of simulated sickness. But in terms of task load, ANOVA indicates that task load of CC2 (keyboard with circular layout) is significantly heavier than EC1 (eye-shaped keyboard with auto-complete). Other differences are not significant. According to feedback of participants, the circular layout forces the user's focus to move around the central ring over and over to find the target key, which creates great frustration and high level of task load. However, this is not the case for the QWERTY layout that users are more familiar with or the similar eye-shaped layout. This might be the cause of high task load of CC2.

7 User study 2: growth study

In order to balance the amount of training and discover the potential of our method in increasing entry speed and reducing error rates, we conducted another user study, or Growth Study, where we monitored the development of user's text entry performance with incremental amount of training.

7.1 Study design

Experiment Condition Designation

In the experiment, we adopted a mixed design. For the independent variable of different proficiency, the without subject design was adopted. The participants were divided into two groups, novice group (as EC) and adept group (as CC). The adepts were selected from participants who typed relatively faster in first user study. The novice group consisted of people we engaged who had never participated in previous experiments. For another variable of amount of training, we used within subject design.

Participants and Apparatus

We engaged N = 8 participants in total, NM = 6 males and MF = 2 females. Among them, 4 male participants formed the adept group, who had participated in first user study. The other 4 participants (2 males and 2 females) who had never taken part in previous studies formed the novice group. Participants are between 19 and 28 with good or corrected vision. For participants in adept group, they had experience using the physical QWERTY keyboard, but none of them could touch type, and only had experience of VR in first user study. Participants in novice group also had experience using the physical QWERTY keyboard and couldn't touch type. They had no experience of VR. None of participants are native English speakers. Apparatus is controlled to be the same as first user study.

Task

The experiment lasted for 3 days, with 2 sessions per day. During each session, participants of both groups needed to type with eye-shaped keyboard for 20 minutes as training. Then, they would be asked to enter 8 phrases as a test. Phrase set and other arrangement were the same as first user study.

Metrics

We only used text entry speed and error rate to evaluate the performance of our eye-shaped keyboard.

Statistical Analysis Method

We used a 3 × 2 mix-design ANOVA with sessions of training (from 1 to 6) as the within subject variable and group class (novice and adept) as the without subject variable. Other analysis methods are the same as previous user study.

7.2 Results

Mixed ANOVA indicates a significant influence of Session ($F = 36.668$, $p < 0.001$) and a significant effect of Group Class ($F = 13.603$, $p = 0.010$) on text entry speed. But effect of both factors (Session × Group Class) is not significant. Pairwise comparison demonstrates significant difference between session 1-2, 1-3, 1-4, 1-5, 1-6, 2-4, 2-5, 2-6, 3-4, 3-5, 3-6, 4-6, 5-6 (all $p < 0.05$).

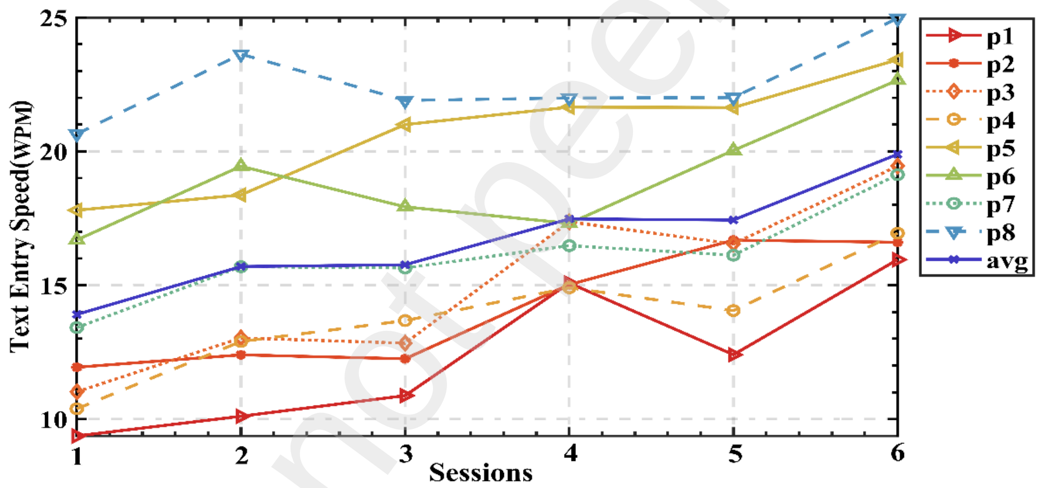

Figure 7 Mean text entry speed of 6 sessions (3 days) for each participant.

Figure 7 shows the mean text entry speed by sessions for each participant and the two groups. Average text entry speed of all sessions was 16.70 WPM. Mean speed of the novice group reached 13.83 WPM, while for adept group, the mean text entry speed reached 19.57 WPM. Mean speed of Session 1 was 13.91 WPM. After 6 sessions of exercise, it reached 19.89 WPM, with an increase of 42.99%. Among all participants' performance, the maximum increase achieved 76.48% (11.02 WPM to 19.45 WPM), and the peak speed achieved 24.97 WPM (in Session 6). So training does help promote user's typing efficiency, and this result is not the peak. In other words, user's typing efficiency is still trending upwards.

For TER, according to ANOVA, neither Session ($F = 2.172$, $p = 0.084$) nor Group Class ($F = 0.121$, $p = 0.740$) has significant effect. The combined effect is also not significant ($F = 1.498$, $p = 0.22$). For NCER, ANOVA reveals no significant effect caused by Session, Group Class ($F = 2.958$, $p = 0.136$) or combination of both ($F = 0.435$, $p = 0.821$).

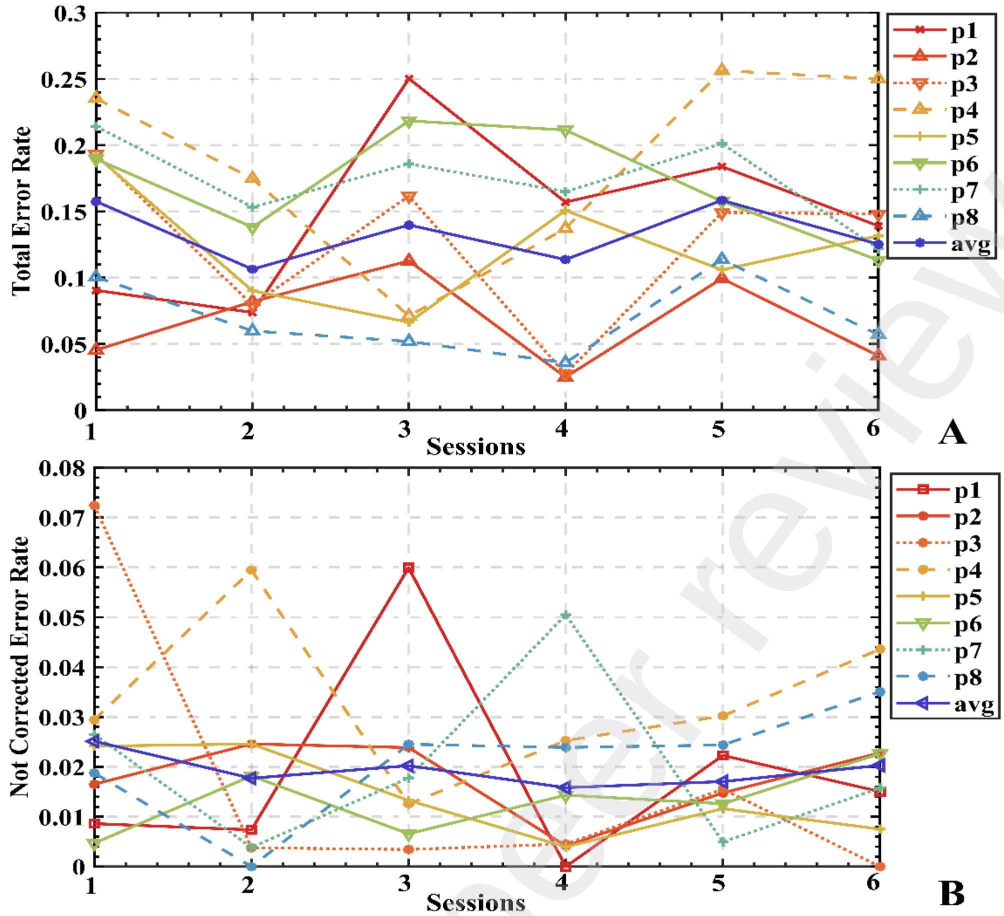

Figure 8 Mean error rate of TER (A) and NCER (B) of 6 sessions.

Figure 8 shows the mean TER and NCER over six sessions. The average of TER and NCER for all sessions were 13.36% and 1.94% respectively. Average TER and NCER of the novice group were 13.27% and 2.17%. For adept group, they were 13.44% and 1.71%.

8 Conclusions, limitations, and future work

We have already described the eye-shaped keyboard, an effective method for text entry in VR environments, with a keyboard in the shape of dual eyes. The user interacts with the keyboard by typing with both hands, each independent of the other. Our key sequence retains the main part of the QWERTY key sequence. Once the ray enters the key area, a character can be entered, i.e. one character per action. In the first user study, or Comparison Study, our keyboard proved to performed significantly more efficiently, more accurately with less task load and less hand movement than circular keyboard for beginners. Compared with QWERTY keyboard, the eye-shaped keyboard is more accurate, significantly reduces hand translation while keeping similar efficiency for beginners. In the second user study, or Growth Study, the eye-shaped keyboard proved to be easy to learn, efficient to type on and full of potential.

Our approach has several limitations. Firstly, our keyboard with the shape of dual eyes occupies part of user's field of view. Our method makes each letter correspond to a key in order to be able to enter a letter in one step, which occupies a larger area than the keyboard with one key for multiple letters. Although users often focus on text entry rather than on the objects behind the keyboard when typing, there is value in providing more area for visualization applications in specific situations. Future work will contain various methods of reducing the obstruction of the field of view and test them through experiments. For example, attempts could be made to reduce the size of the letter area or even reduce the key area to a curved line. Different parts of the line could be marked in different colors to differentiate between the keys, and letters could be marked around. In this way, the keyboard is largely unobscured by the objects behind it. Alternatively, a high transparency of the keys may help with the obscuration problem.

Another limitation of this work is that the eye-shaped keyboard imposes certain requirements on the controller and the input environment. In our approach, the user's two hands control the eye-shaped keyboards on each side. In some extreme cases, the users may not be able to use both hands for text entry at the same time, such as device failure, user disability, etc. In addition, if there is not enough space around the users to move their hands, it can also be extremely disruptive to our method. In future work, we may try to provide multiple text entry methods to adapt to different input environments or to improve the environmental adaptability of the current eye-shaped keyboard. For example, one-hand controlled eye-shaped keyboard or an HMD controlled eye-shaped keyboard could be tried.

Another limitation of this work is that the performance of the eye-shaped keyboard in complex situations was not tested. Although our eye-shaped keyboard provides number input as well as special character input, in the experiments the text did not contain numbers or special characters. The special character keyboard of the eye-shaped keyboard has been simply designed and it may not achieve the desired input speed in practice. The users need to switch frequently between the normal and special character keyboards if there are frequent alternations between letters and special characters, which consumes much time. Future work could be devoted to testing and improving text entry performance in complex situations. Possible solutions include adding more frequently used characters to the normal keyboard, well designing the layout of the symbol keyboard, etc.

Finally, our approach is not limited to VR. In future research we will try to apply eye-shaped keyboard to augmented reality, mixed reality, 3D human-computer interaction and other areas, such as using fingers for manipulation and other means.

References

1 Venolia D, Neiberg F. T-Cube: A fast, self-disclosing pen-based alphabet. In: Adelson B, Dumais S, Olson J, eds. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Boston, Massachusetts USA, 1994. 265-270
DOI:10.1145/191666.191761

2 Mankoff J, Abowd G D. Cirrin: A word-level unistroke keyboard for pen input. In: Mynatt E, Jacob R J K, eds. Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology, New York, USA, 1998. 213-214
DOI:10.1145/288392.288611

3 Katsuragawa K, Wallace J R, Lank E. Gestural text input using a smartwatch. In: Buono P, Lanzilotti R, eds. Proceedings of the International Working Conference on Advanced Visual Interfaces, New York, USA, 2016. 220-223
DOI: 10.1145/2909132.2909273

4 Yi X, Yu C, Xu W, et al. COMPASS: Rotational keyboard on non-touch smartwatches. In: Mark G, Fussell S, eds. Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, New York, USA, 2017. 705-715
DOI: 10.1145/3025453.3025454

5 Gong J, Xu Z, Guo Q, et al. Wristext: One-handed text entry on smartwatch using wrist gestures. In: Mandryk R, Hancock M, eds. Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, New York, USA, 2018. 1-14
DOI: 10.1145/3173574.3173755

6 Yu D, Fan K, Zhang H, et al. PizzaText: text entry for virtual reality systems using dual thumbsticks. IEEE Transactions on Visualization and Computer Graphics, 2018, 24(11): 2927-2935
DOI: 10.1109/TVCG.2018.2868581

7 Xu W, Liang H N, Zhao Y, et al. Ringtext: Dwell-free and hands-free text entry for mobile head-mounted displays using head motions. IEEE Transactions on Visualization and Computer Graphics, 2019, 25(5): 1991-2001
DOI: 10.1109/TVCG.2019.2898736

8 Kim S, Kim G J. Using keyboards with head mounted displays. In: Brown J, Cai Y, eds. Proceedings of the 2004 ACM SIGGRAPH International Conference on Virtual Reality Continuum and Its Applications in Industry, New York, USA, 2004. 336-343
DOI: 10.1145/1044588.1044662

9 Walker J, Li B, Vertanen K, et al. Efficient typing on a visually occluded physical keyboard. In: Mark G, Fussell S, eds. Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, New York, USA, 2017. 5457-5461
DOI: 10.1145/3025453.3025783

10 Bovet S, Kehoe A, Crowley K, et al. Using traditional keyboards in vr: Steamvr developer kit and pilot game user study. In: Corcoran P, Bertozzi E G, eds. 2018 IEEE Games, Entertainment, Media Conference (GEM), Galway, Ireland, 2018. 1-9
DOI: 10.1109/GEM.2018.8516449

11 Hoppe A H, Otto L, van de Camp F, et al. qvrty: Virtual keyboard with a haptic, real-world representation. In: Stephanidis C, eds. HCI International 2018–Posters' Extended Abstracts: 20th International Conference, HCI International 2018, Las Vegas, NV, USA, 2018. 266-272

12 Knierim P, Schwind V, Feit A M, et al. Physical keyboards in virtual reality: Analysis of typing performance and effects of avatar hands. In: Mandryk R, Hancock M, eds. Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, New York, USA, 2018. 1-9
DOI: 10.1145/3173574.3173919

13 Grubert J, Witzani L, Ofek E, et al. Effects of hand representations for typing in virtual reality. In: Kiyokawa K, Steinicke F, eds. 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Tuebingen/Reutlingen, Germany, 2018. 151-158
DOI: 10.1109/VR.2018.8446250

14 McGill M, Boland D, Murray-Smith R, et al. A dose of reality: Overcoming usability challenges in vr head-mounted displays. In: Begole B, Kim J, eds. Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, New York, USA, 2015. 2143-2152
DOI: 10.1145/2702123.2702382

15 Yu C, Gu Y, Yang Z, et al. Tap, dwell or gesture? exploring head-based text entry techniques for hmds. In: Mark G, Fussell S, eds. Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, New York, USA, 2017. 4479-4488
DOI: 10.1145/3025453.3025964

16 Speicher M, Feit A M, Ziegler P, et al. Selection-based text entry in virtual reality. In: Mandryk R, Hancock M, eds. Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, New York, USA, 2018. 1-13
DOI: 10.1145/3173574.3174221
17 Mehring C, Kuester F, Singh K D, et al. KITTY: Keyboard independent touch typing in VR. In: Watson B, Rizzo S, eds. IEEE Virtual Reality 2004, Chicago, IL, USA, IEEE, 2004. 243-244
DOI: 10.1109/VR.2004.1310090
18 Xu W, Liang H N, He A, et al. Pointing and selection methods for text entry in augmented reality head mounted displays. In: Zhao Q, Wang Y, eds. 2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Beijing, China, 2019. 279-288
DOI: 10.1109/ISMAR.2019.00026
19 Boletsis C, Kongsvik S. Text input in virtual reality: A preliminary evaluation of the drum-like vr keyboard. Technologies, 2019, 7(2): 31
DOI: 10.3390/technologies7020031
20 Benligiray B, Topal C, Akinlar C. Slicetype: fast gaze typing with a merging keyboard. Journal on Multimodal User Interfaces, 2019, 13: 321-334
DOI: 10.1007/s12193-018-0285-z
21 Zhang Z, Sun M, Gao B Y, et al. 2-thumbs typing: A novel bimanual text entry method in virtual reality environments. In: 2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW), Lisbon, Portugal, 2021. 530-531
DOI: 10.1109/VRW52623.2021.00147
22 Huckauf A, Urbina M H. Gazing with pEYEs: towards a universal input for various applications. In: Raiha K J, Duchowski A T, eds. Proceedings of the 2008 Symposium on Eye Tracking Research & Applications, New York, USA, 2008. 51-54
DOI: 10.1145/1344471.1344483
23 Ma X, Yao Z, Wang Y, et al. Combining brain-computer interface and eye tracking for high-speed text entry in virtual reality. In: Berkovsky S, Hijikata Y, eds. 23rd International Conference on Intelligent User Interfaces, New York, USA, 2018. 263-267
DOI: 10.1145/3172944.3172988
24 Majaranta P, Ahola U K, Špakov O. Fast gaze typing with an adjustable dwell time. In: Grinter R, Rodden T, eds. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, New York, USA, 2009. 357-360
DOI: 10.1145/1518701.1518758
25 Rajanna V, Hansen J P. Gaze typing in virtual reality: impact of keyboard design, selection method, and motion. In: Sharif B, Krejtz K, eds. Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications, New York, USA, 2018. 1-10
DOI: 10.1145/3204493.3204541
26 Howard I P, Rogers B J. Binocular vision and stereopsis. Oxford University Press, USA, 1995.
27 MacKenzie I S, Soukoreff R W. Phrase sets for evaluating text entry techniques. In: Cockton G, Korhonen P, eds. CHI'03 Extended Abstracts on Human Factors in Computing Systems, New York, USA, 2003. 754-755
DOI: 10.1145/765891.765971
28 Soukoreff R W, MacKenzie I S. Metrics for text entry research: An evaluation of MSD and KSPC, and a new unified error metric. In: Grinter R, Rodden T, eds. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, New York, USA, 2003. 113-120
DOI: 10.1145/642611.642632