Taylor & Francis
Taylor & Francis Group

International Journal of Human–Computer Interaction

ISSN: 1044-7318 (Print) 1532-7590 (Online) Journal homepage: www.tandfonline.com/journals/hihc20

CrossKeys: Text Entry for Virtual Reality Using a Single Controller via Wrist Rotation

Zhaoxun Liu, Xiaolong Liu & Lili Wang

To cite this article: Zhaoxun Liu, Xiaolong Liu & Lili Wang (2025) CrossKeys: Text Entry for Virtual Reality Using a Single Controller via Wrist Rotation, International Journal of Human–Computer Interaction, 41:9, 5150-5162, DOI: 10.1080/10447318.2024.2358456

To link to this article: https://doi.org/10.1080/10447318.2024.2358456

Published online: 05 Jun 2024.

Submit your article to this journal

Article views: 270

View related articles

View Crossmark data

Citing articles: 2 View citing articles

Full Terms & Conditions of access and use can be found at
https://www.tandfonline.com/action/journalInformation?journalCode=hihc20

INTERNATIONAL JOURNAL OF HUMAN–COMPUTER INTERACTION
2025, VOL. 41, NO. 9, 5150–5162
https://doi.org/10.1080/10447318.2024.2358456

Taylor & Francis
Taylor & Francis Group

Check for updates

CrossKeys: Text Entry for Virtual Reality Using a Single Controller via Wrist Rotation

Zhaoxun Liu[a], Xiaolong Liu[a,b], and Lili Wang[a,b]

[a]School of Computer Science and Engineering, Beihang University, Beijing, China; [b]State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China

ABSTRACT

Text entry has long been an indispensable part of people's lives; notwithstanding, in virtual reality (VR), an efficient and handy text entry method for such an environment is still wanted. There are two negative factors in the very majority of existing text entry methods: 1) constrained by two-dimensional mapping; 2) must manipulate with both hands to ensure efficiency. Few are employing the three-dimensional space a virtual environment provides, with which text entry could perform much better; also, even with several methods enabling one-hand manipulation, the trade-off is off-balance when sacrificing performance. Therefore, we propose an innovative text entry method to achieve a faster speed, higher accuracy, and better user experience. We design a cross-like layout to reduce the average distance of spatial displacement when selecting characters. In selecting and entering characters, a user simply rotates the wrist and points the embodied controller to one of the seven character blocks in the virtual environment; afterward, within the selected block, the user enters the target character or auto-completion candidate word via two basic trackpad interactions, touching and clicking. We evaluate our CrossKeys mainly based on two criteria: efficiency, task load, and three tasks: learnability and performance test, fatigue test, and evaluation of in-motion performance. To evaluate efficiency, we analyze words per minute (WPM) and error rate (NCER, TER); to evaluate task load, we analyze NASA-TLX and Simulator Sickness Questionnaire (SSQ) results. According to the data from our participants after only 2 hours of first-time training, our CrossKeys performs well with an average WPM of 17.73, a peak WPM of 24.73, and an error rate (NCER) of 0.30% along with a low task load.

KEYWORDS

Virtual reality; text entry; wrist-based interaction; single controller input; 3D keyboard layout

1. Introduction

Text entry has been widely applied to traditional electronic devices, such as the most omnipresent QWERTY keyboard for personal computers and the T9 keyboard for mobile smartphones. Texting has long been an indispensable part of people's lives; countless scenarios, like online chatting and searching, cry for an advanced text entry method. Notwithstanding, in VR, even with the fast development itself, an efficient and handy text entry method for such an environment is still wanted. Although there are already several text entry methods introduced, few are employing the three-dimensional space a virtual environment provides, indicating that there is still room for improvement in terms of entry efficiency and user experience.

Many researchers before us have already explored various possibilities of techniques we discuss in this paper. For instance, methods like PizzaText (Yu et al., 2018), FingerT9 (Wong et al., 2018), and DepthText (Lu et al., & 2019) all aim at text entry in virtual environments. Then, about 3D user interaction in VR, we have DeepSketch2Face (Han et al., 2017) and many other methods to decode and translate spatial movements into other types of information. Last but not least, about wrist-based interaction, there are also many human-computer interaction research groups that have examined the possibility of such an interaction, such as WristFlicker (Strohmeier et al., 2012) and WRIST (Yeo et al., 2019).

In this paper, we introduce CrossKeys, a novel, efficient, and handy text entry technique for VR using a single controller via wrist rotation (Figure 1). By rotating the wrist along with simple clicking, a user can easily and accurately enter text using one single controller. In the process of text entering, a user completes all the tasks with wrist-controller interaction, including character inputting, auto-completion selection, and deleting.

To further explore its learning curve and evaluate its performance, we then designed and conducted three user studies, concentrating on learnability, fatigue, and in-motion performance, respectively. In the study of learnability, a novice user could be expected to achieve a WPM of 11.76 and an NCER of 0.64% after the first 45 minutes of training; after the third 45 minutes of training, the same user could achieve a WPM of 17.73 and an NCER of 0.30%, which has already outperformed the current state-of-the-art method,

CONTACT Lili Wang wanglily@buaa.edu.cn School of Computer Science and Engineering, Beihang University, Beijing, China; State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China

© 2024 Taylor & Francis Group, LLC

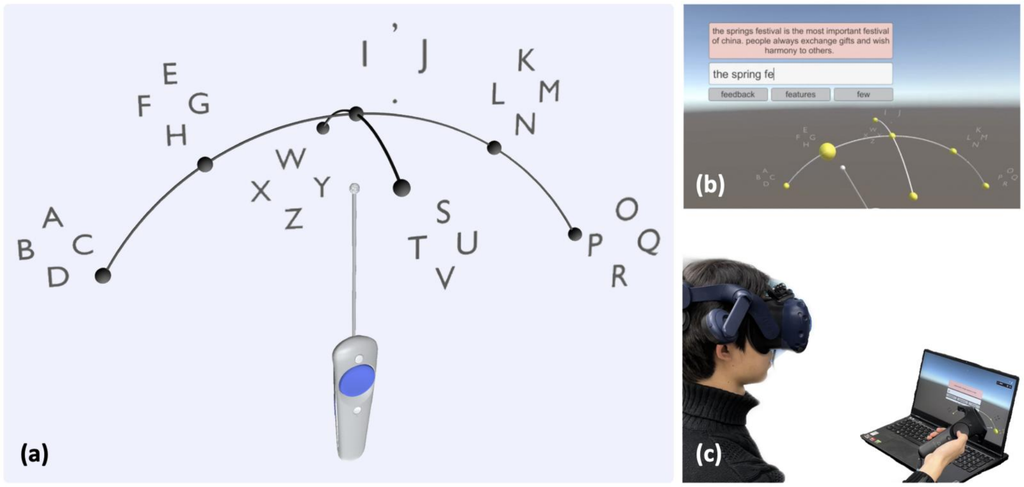


Figure 1. The overview of CrossKeys. (a) is the core of CrossKeys, consisting of the three-dimensional structure of character layout and the embodied controller in the virtual environment; (b) is the user perspective in most of our pilot and user studies, consisting of a text indicator, an input field, a set of predictive auto-complete candidates, and the CrossKeys in use; (c) is the third-person perspective of a CrossKeys user proceeding a text entry task during a user study, which clearly exhibits that CrossKeys is based on single controller manipulation and wrist rotation. As it can be seen by comparing the controller directions in (a) and (c) that the embodied controller in the virtual environment is based on the real controller but rotated perpendicularly to make a more natural wrist rotation around a horizontal axis.

PizzaText. In fatigue study, we evaluated the workload of our CrossKeys. In in-motion study, we unprecedentedly enable VR text entry in moderate movement, and compared performance with static studies.

Our contributions in this work include: 1) CrossKeys, a text entry method that employs the three-dimensional space a virtual environment provides; 2) a light-weighted text entry method using only one controller with simple wrist rotating and clicking; 3) outperforming the similar state-of-the-art method with a average WPM of 17.73; 4) a wider application scenario where users can enter text in virtual environments during moderate motion like walking.

2. Related work

Text entry in VR is a fundamental technique; notwithstanding, it has been a considerable challenge to optimize the entering speed while lowering the error rate. To address this problem, researchers in the field have proposed a variety of methods, including entering via interactions based on tracking of the user's head, hands, or eyes. These work helped to finalize our CrossKeys.

2.1. Text entry in virtual environments

First of all, the majority of text entry methods in VR are based on the tracking of hands (Dudley et al., 2023; Shen et al., 2023). One category of them is to map characters to the spatial coordinate derived from the user's hands. One of the most innovative among them is the ATK system (Yi et al., 2015) introduced by X. Yi and his colleagues; the system enables users to type in VR via tracking fingers' movement in the air and provide visual response on a desktop display. Also, (Xu et al., 2019) and (Xu et al., 2020) proposed smartwatch text entry methods based on finger tipping, and Gupta et al. (2019) introduced typing with an additional ring-shaped hardware to track hand movement.

Text entry based on controllers is also very common. PizzaText (Yu et al., 2018) divides the round trackpad of a controller into seven pieces and enables users to select characters from different pieces via touching on the corresponding one. Besides PizzaText, there are many other similar methods based on finger-touching on a controller, such as FingerT9 (Wong et al., 2018) and QwertyRing (Gu et al., 2020). Finger-touching, such as Liang et al. (2023); Shen et al. (2023); Yi et al. (2023), is a common idea for text entry in VR; nonetheless, if a method is solely based on the tracking of finger-touching, it can be tiresome for users as constantly finger moving always causes fatigue.

Head-mounted display (HMD) is a basic wearable device in VR; besides providing virtual visual environments for users, it can also detect users' head movement and enable users to select with a ray pointer. Text entry based on HMD is also very popular in VR. For instance, DepthText (Lu et al., 2019) employs the acceleration-sensitive embedded IMU sensor to translate head movement into texts; RingText (Xu et al., 2019), another method based on HMD, realizes text entry by aiming the ray pointer at characters on a round virtual keyboard with the slight movement of a user's head. Yu et al., (2017) implements a virtual keyboard to simulate physical typing.

Besides, text entry based on eye movement is another innovative method, such as Filteryedping (Pedrosa et al., 2015), EyeSwipe (Kurauchi et al., 2016). Via eye interactions like gazing, blinking, and staring, texts can be selected and entered.

2.2. 3D Interaction in virtual reality

Three-dimensional interaction has already been realized and applied by text entry in traditional electronic devices, such as Vulture (Markussen et al., 2014). With the advent of VR, 3D interaction has also be emphasized as an important part of VR technology. To enable users to obtain the same spatial perception as reality through three-dimensional interaction in virtual environments, so as to realize text entry, is a goal pursued by researchers in the VR field.

Modelling and drawing in virtual environments are another two popular topics (Darabkh et al., 2018; Han et al., 2017; Jiang et al., 2021; Delanoy et al., 2018; Kwan & Fu 2019; Luo et al., 2021; Sormann et al., 2004), and nearly all of them involve 3D interaction, which is an indispensable part of our research.

Closely related to text entry in virtual environments is the 3D virtual keyboard; but due to the influence of traditional 2D keyboards, 3D keyboards are not common. The team that introduced Cubic Keyboard (Yanagihara & Shizuki 2018) has proposed a cube-like 3D keyboard that divides a cube into $3 \times 3 \times 3$ smaller cubes and embeds letters in them. However, as the letters overlap, it does not provide a preferable visual experience. In addition to 3D virtual keyboard, 3D gesture is also a research topic for text entry. In a paper proposed by S. Chen and his colleagues (Chen et al., 2019), the authors explore gesture-based text entry technology, using a VR controller of six degrees of freedom for gesture typing, making the keyboard no longer constrained to flat surfaces, but the whole space can be a "keyboard".

3D interaction in VR scenarios is multisensory, enabling users to interact well in a three-dimensional space with visual, auditory, and tactile senses. To provide better real-time stereoscopic 3D images, Y. Ikei and his colleagues propose a method called TwinCam (Ikei et al., 2019), which uses two 360° cameras to provide high-quality Visual Telepresence. In the auditory aspect, H. Kim and his colleagues propose an immersive audio spatial system (Kim et al., 2019) that enables spatial audio to be synchronized with visual information. In addition to visual and auditory satisfaction of user spatial interaction needs, haptic feedback is also essential. D. Valkov and his colleagues propose a stand-alone hardware device for alerting users in immersive virtual environments (IVEs) of possible collisions with real-world objects (Valkov and Linsen 2019), enhancing the risk predictability of VR systems. Of course, there are studies with greater ambitions that seek to develop a complete 3D interaction system, such as the research conducted by T. M. Takala and his colleagues (Takala et al., 2019), which introduced a stand-alone, wearable system for full-body and finger tracking that can fully enhance the user's 3D interaction perception.

In our research, three-dimensional interaction is also an important concept, which is more reflected in users' visual interaction with the 3D keyboard. Therefore, our CrossKeys also aims to enable users to have a better 3D interaction experience.

2.3. Interaction based on wrist movement

Spurred by single-handed text entry methods such as FingerText (Lee et al., 2021) and understood their shortcomings such as costly learning effort and high task load while using, we turned to finding another type of single-handed interaction approach in VR, wrist-based interaction. Wrist motion-based interaction is a less common interaction method. Although wrist rotation is influenced by the body's muscular and skeletal structure, wrist motion can utilize less load in exchange for better responsiveness as long as a reasonable range of rotation is controlled.

WrisText (Gong et al., 2018) proposed a wrist-turn-based text input on a smartwatch, which divides the keyboard on the display into 6 parts and selects letters on a region by turning the wrist to that region. This input method is a bit similar to our CrossKeys, however, it does not handle wrist rotation properly. 6 directions of rotation make the wrist load different, and certain rotation angles can bring some discomfort to the user. In 2019, Shirin Feiz et al. explored and studied the feasibility of wrist gestures for non-visual interactions with wearables (Feiz & Ramakrishnan 2019). They concluded that wrist gestures are a new type of input that users can use for a range of one-handed interactions with these devices. In addition, wrist gestures are particularly attractive to people with visual impairment (PVI) and can provide them with additional assistance.

To better detect wrist rotation, many studies have considered the use of sensors to listen to wrist motion, such as WristFlicker (Strohmeier et al., 2012) and WRIST (Yeo et al., 2019). They have mathematically separated wrist rotation from arm and hand motion, allowing wrist motion to be better detected. Wrist rotation is sensitive, and using it with equally sensitive sensors can better improve the recognition of wrist motion interactions.

We design CrossKeys to take advantage of this flexibility in wrist movement, using the controller as a "sensor" to detect wrist movement and making it better listened to.

3. Method

3.1. Design rationale

A hand along with its skeletal structure connecting to the wrist is capable to complete various interactive actions with a VR controller; however, most of the similar controller-based text entry methods apply only actions based on fingers, failing to notice the possibility of wrist-based interaction. Rotating a VR controller is a natural tendency when holding one. Therefore, we drew a conclusion that the panning of the hand in six dimensions would make it more painstaking for users than wrist rotation, where users would feel a lot less fatigued. Eventually, we adopted wrist rotation as the principle interaction approach.

The reason for introducing wrist rotation is the potential to increase the freedom of operation for users inputting text in 3D space. By incorporating wrist rotation, we hope to offer a more intuitive and natural method for users to flexibly manipulate text elements in a virtual environment. This technique allows users to position and edit text in three-dimensional space more quickly and intuitively. Such manipulation could positively impact user task performance in virtual environments, especially when dealing with complex spatial layouts or multidimensional text operations.

Additionally, introducing wrist rotation techniques aims to explore their potential advantages and limitations in practical application scenarios. Through this research, we have the opportunity to gain a deeper understanding of the actual impact of this technology on users, providing stronger support for future design decisions.

CrossKeys breaks the limitation of the two-dimensional mapping stereotype and harnesses the rotation of a wrist to select from characters; by mapping three-dimensional movement into a spatial keyboard, CrossKeys outperforms the state-of-the-art method for text entry in VR, in consideration of speed and accuracy.

3.2. Keyboard layout

CrossKeys, a three-dimensional keyboard, consists of two crossed arc segments. The arrangement of the seven character blocks is derived from a design concept mentioned before that ulnar-axis rotation is easier; Therefore, we put five blocks in a total of seven on the arc accessed by ulnar-axis rotation and the rest on the other; the reason behind this arrangement is that human wrists are more comfortable with rotating instead of leaning, according to Esmaeili et al. (2011). For each character block, we divide 28 characters (26 English alphabetic letters, aligned with comma (,) and period (.)) into 7 groups; each character block corresponds to one particular group, i.e. each character block contains 4 characters. Breaking down the complete set of 28 characters into smaller, manageable groups of 4 characters each simplifies the system and promotes uniformity, making it easier to work with and ensuring an equal distribution across all groups.

On deciding the distance between every two adjacent blocks, we had 5 members in the lab to trial our CrossKeys and rotate their wrists to the extreme of a subjective comfortable degree in four directions: left, right, front, and behind. We then averaged these degrees to equally distance the blocks.

3.3. Interaction and functionality

As shown in Figure 2, the user interface of CrossKeys consists of four parts: a text indicator, an input field, a set of predictive auto-complete candidates, and the CrossKeys itself. The text indicator displays the particular test text assigned to the current study participant; the input field provides an area to enter texts; the set of candidates provides three possible words with the same prefix as the current inputted character sequence that are likeliest to be entered. On deciding how many candidates to reveal to the user to achieve the most optimized efficiency, we referred to P. Quinn and S. Zhai's work (Quinn and Zhai 2016) to evaluate the trade-off between suggestion savings and interaction costs and make sure that the efficiency saving is improved while the cost of finding, selecting, and interacting is correspondingly reduced. Due to the concern of cognitive load and screen visibility, we settled the number of candidates revealed to 3; specifically, presenting more than three candidates at once can overwhelm users, leading to cognitive overload, while showing fewer than three significantly restricts the enhancement of text entry speed.

To lower the task load and make it easier for new users to learn and adapt, we decide to use as few buttons as possible. Therefore, on finalizing which buttons to use, we examined the most frequently-used programmable and interactive ones; finally, we programme three buttons: *TouchPad*, *Trigger*, and *GripButton* as shown in Figure 3. To elucidate our interaction method more clearly, we divide the operating space into 5 areas as shown in Figure 4: 4 *Deactivated* areas and 1 *Activated* area. When in the *Activated* area, *TouchPad* is to select, and enter normal characters; when in the four *Deactivated* areas, *TouchPad* is to

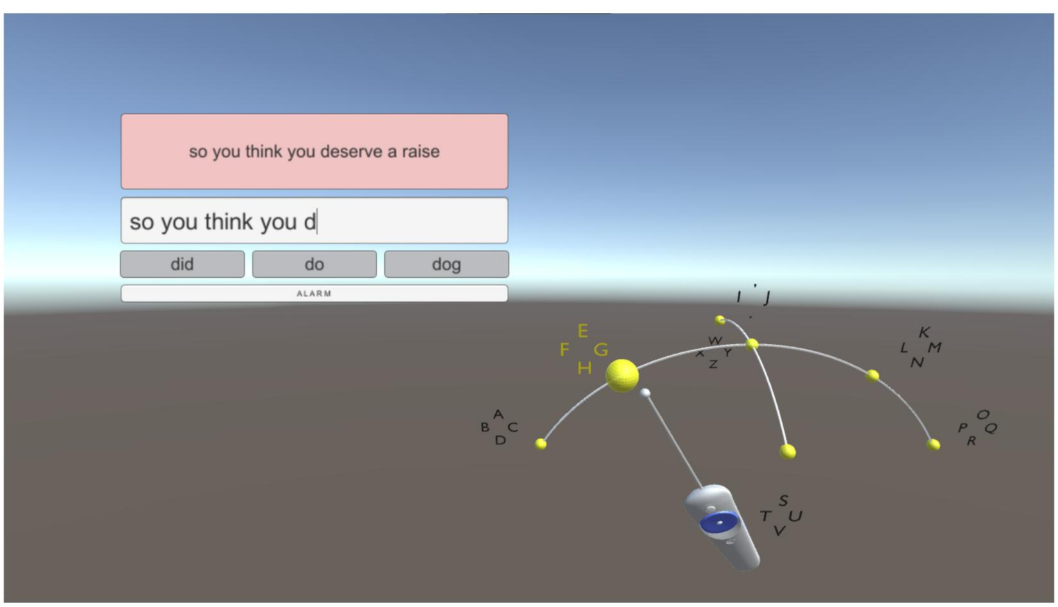


Figure 2. The user interface system of CrossKeys, in which currently the zoomed block, $E-F-G-H$ is selected.

Figure 3. *TouchPad*, *Trigger*, And *GripButton* on a HTC vive controller. These three interactive units enable all the interaction and functionality designed for our CrossKeys.

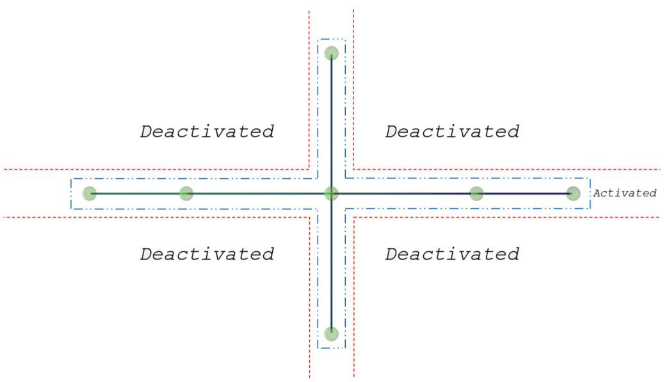


Figure 4. The projection of four *activated* areas and one *deactivated* area of CrossKeys' spatial interaction.

choose between three predictive auto-complete candidates. *Trigger* is to delete the normal character at the tail of the current entering sequence at any given time, while *GripButton* is to enter a blank space at any given time.

The CrossKeys system has four main functions:

- **Selecting and Entering a Normal Character (excluding blank space)**: After moving the tip to the *Activated* area, turn the wrist to point to the character block where the target character is located, making it highlighted. At this time, according to the position of the target character in the character block (up-down-left-right), gently touch the corresponding position on the *TouchPad* to select it from the highlighted character block.
- The location mapping complies with the following formula: $SelectedBlock = EuclideanDistance(Tip,\ Block) < 5cm$
- **Selecting and Entering an Auto-complete Candidate**: Once the target word occurs in the candidate list, move the tip out of the *Activated* area into any of the four *Deactivated* areas. At this time, according to the position of the target word in the candidate list (left-middle-right), click the *TouchPad* on the corresponding position to select the target word and enter.
- **Entering a Blank Space**: At any given time, click the *GripButton* to enter a blank space. The occurrence of blank space will interrupt and reset the current auto-complete prediction.
- **Delete a Character**: At any given time, pull and click the *Trigger* to delete a character at the tail of the current entering sequence.

3.4. Text entry assistance

To better assist users with text entry using CrossKeys, we have added text entry assistance to the base design.

3.4.1. Auto – completion

CrossKeys keyboard provides users with an auto-completion feature. We have designed and implemented an algorithm to automatically predict words to be entered based on matching the current inputted character sequence with words with the same sequence as prefixes. More specifically, we implemented a trie (prefix tree) that stores common words or phrases, where each node represents a character and paths from the root to leaf nodes represent complete words. As the user types, navigate through the trie based on the current input prefix to suggest words or phrases that share the same prefix, dynamically updating the suggestions with each new character typed.

3.4.2. Special Display of Controller

In the name of consistency, we embody the physical controller into the virtual environment and rotate it by a vertical angle to make the embodied controller in the virtual environment point upward while the real controller is horizontally pointing to the front so that the rotation of the wrist and the action of pointing can be conducted more naturally since the CrossKeys' character region is actually above the user's manipulating hand. Once accustomed to this simple displacement, users do not need to awkwardly twist their wrists to reach certain character blocks and subsequently achieve better efficiency.

4. Pilot Study

We evaluated different keyboard layouts and determined whether or not to highlight the selected character block in order to derive the most promising overall layout and interaction interface.

4.1. Participants and hardware setup

Six participants (half of them are men and the other half are women) with a total average age of 20 distributing equally from 18 to 22, all with normal or corrected-to-normal vision and no impaired wrists conducted this study. During our recruitment process, all participants demonstrated a high level of proficiency with Qwerty keyboards, being able to type at least 50 words per minute on a physical Qwerty keyboard. We assessed this capability to guarantee the effectiveness and reliability in the pilot study. We used an HTC Vive Pro head-mounted display, an Intel Core i7 processor PC with a dedicated NVIDIA GTX 1070 graphics card, and an AMD processor laptop with an NVIDIA GTX 3060 graphics card. Participants use the laptop to run the pilot studies and we monitor in real-time via PC. The experimental program is written in *C# .NET* programming language and runs on the Unity 3D platform.

4.2. Metrics

We employed WPM (words per minute), TER (total error rate) and NCER (not corrected error rate) to evaluate our entry method. The text entry speed was measured and evaluated with WPM using the following formula, where according to the universal standard, 5 consecutive letters, including spaces and symbols like commas and periods, make a word:

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

$|S|$ represents the length of the entered phrases. T represents the task completion time, which was recorded as the time elapsed from when the first letter or phrase is selected to the end of the trail.

The error rate is to evaluate our method's accuracy. The error rate is calculated based on standard typing metrics (Soukoreff & MacKenzie 2003). Total error rate (TER) takes both corrected error rate (CER) and not corrected error rate (NCER) into account. NCER refers to errors found during the final examination of the text participants entered, where $TER = NCER + CER$. We report the error rate based on TER and NCER.

4.3. Analysis method

For Pilot User Study I, where the factor (layout options) had three levels, we employed one-way repeated measures ANOVA. LSD Correction was employed for posthoc pairwise comparisons, and Greenhouse-Geisser adjustment was employed for degrees of freedom for violations to sphericity. For Pilot User Study II, we employed T-test with Form (highlighting and no highlighting) as the variable. Due to the within-subject design, the T-test is paired samples t-test. Shapiro-Wilk Test is employed for testing the normal distribution of the data.

4.4. Pilot Study I: Keyboard layout

The role of this pilot study was to decide on the most promising keyboard layout and the more comfortable interaction layout. For the keyboard layout, we considered three design dimensions to design three different options (Pilot experimental options, or PEC).

4.4.1. PEC i

The first layout is designed strictly in alphabetical order, dividing the 26 letters from A to Z into 7 groups and filling them into the template of the input layout in turn, as shown in Figure 5 (a).

4.4.2. PEC II

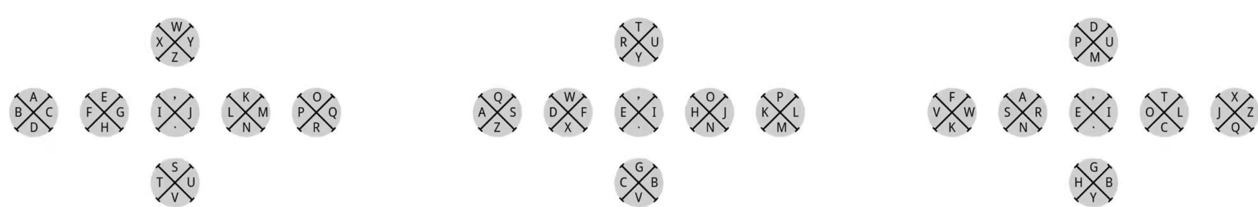
In designing the second layout, we imitated the layout of the QWERTY keyboard in daily use by arranging the letters as similar to the QWERTY layout as possible, as shown in Figure 5 (b).

4.4.3. PEC III

The third layout is based on the frequency of characters in MacKenzie and Soukoreff (2003). After sorting and organizing the frequency of use of all letters, we place the letters with a high frequency of use in or as close to the default original position as possible, and the resulting layout is shown in Figure 5 (c).

4.4.4. Procedure

We use a within-subject design, in which participants perform the tasks in all conditions. For each session, participants would enter randomly-selected phrases using the input method to which they have been assigned. All phrases would be randomly generated from the MacKenzie Phrase Set (MacKenzie and Soukoreff 2003).

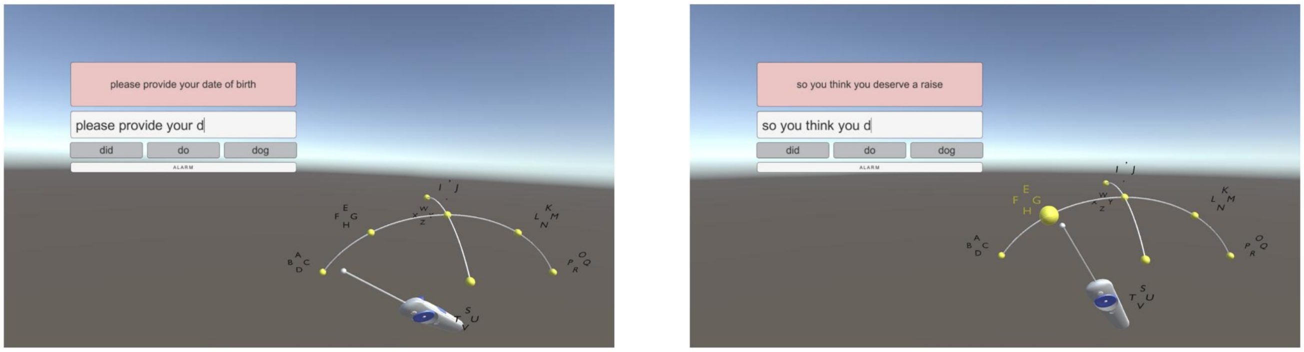


(a) The scheme for *PEC I : Alphabetical Order*. Every gray circle represents an actual WordBlock in CrossKeys, while the crosses help to illustrate the layout and are not shown in the real application.

(b) The scheme for *PEC II : QWERTYlike*. The meaning of the gray circles and the crosses is the same with Figure 1.

(c) The scheme for *PEC III : Frequency*. The meaning of the gray circles and the crosses is the same with Figure 1.

Figure 5. Three initial options for our CrossKeys' keyboard layout, from which we selected *PECI* after our fastidious pilot studies.



(a) The scheme for *PCC* : *No Highlight*. The character block does not highlight itself by changing its color when pointed at.

(b) The scheme for *PEC* : *Highlight*. The character block turns yellow when it is pointed at.

Figure 6. The pilot study to evaluate the effect of highlighting selected character block by changing its color to an accent one like yellow.

Table 1. Average WPM, TER, and NCER in Pilot Study I. Statistical significance is denoted with an asterisk.

Metrics and total *p*-value	Condition	Mean Value $\pm std.dev$	Pair	Pair *p*-value
WPM:0.004*	PEC I	12.31 ± 5.34	PEC I, PEC II	0.002*
	PEC II	9.42 ± 3.88	PEC I, PEC III	0.013*
	PEC III	9.66 ± 3.09	PEC II, PEC III	0.694
TER:0.815	PEC I	$3.1\% \pm 10.5\%$	PEC I, PEC II	0.559
	PEC II	$5.5\% \pm 15.4\%$	PEC I, PEC III	0.814
	PEC III	$4.0\% \pm 12.5\%$	PEC II, PEC III	0.687
NCER:0.245	PEC I	$0.5\% \pm 1.3\%$	PEC I, PEC II	0.183
	PEC II	$1.6\% \pm 3.6\%$	PEC I, PEC III	0.190
	PEC III	$0.9\% \pm 2.5\%$	PEC II, PEC III	0.338

4.4.5. Results and discussion

The ANOVA yields a significant effect of the layout on WPM ($F_{1.59, 36.56} = 7.474$, $p = .002^*$). Table 1 shows the average entry speed for the three options. Post-hoc comparisons revealed that PEC I ($M = 12.31$, $SD = 5.34$) was significantly faster than ($p = .002^*$) PEC II ($M = 9.42$, $SD = 3.88$) and significantly faster than ($p = .013^*$) PEC III ($M = 9.66$, $SD = 3.09$).

The ANOVA doesn't show any significant effect of layout on TER ($F_{2,46} = .205$, $p = .815$) or on NCER ($F_{1.155, 26.563} = 1.443$, $p = .247$). The results of the analysis provide evidence for the speculation that the layout does not generate a significant negative impact on the error rate of text entry. Table 1 shows TER and NCER for the three letter layouts.

According to the experimental results, the first layout designed strictly in alphabetical order was chosen as the best keyboard letter layout, with fast entry speed and similarly low error rates.

4.2. Pilot Study II: Highlighting the selected character block

The study setup is shown in Figure 6. The participants in this study were the same with Pilot Study I, and this study was conducted immediately after Pilot Study I. This pilot study was to figure out whether highlighting the selected character block could make any difference. We took the data from the previous experiment condition which possessed the best performance (the alphabetical order) as the PCC (short for pilot control condition) in the comparison study for the interaction interface.

Table 2. T-test results including average WPM, TER and NCER in Pilot Study II. Statistical significance is denoted with an asterisk.

Metrics	Condition	Mean Value $\pm std.dev$	*p*-value	Cohen's d
WPM	PCC	12.31 ± 5.34	0.001*	0.32
	PEC	13.78 ± 5.33		
TER	PCC	$3.1\% \pm 10.5\%$	0.205	0.28
	PEC	$5.6\% \pm 6.8\%$		
NCER	PCC	$0.5\% \pm 1.3\%$	0.447	0.28
	PEC	$1.0\% \pm 2.1\%$		

4.2.1. PCC

The data are copied from those of Pilot Study I, where the layout is in alphabetical order, the PEC I.

4.2.2. PEC

This experiment condition is to highlight the letter blocks users point to by changing the colors of the certain block, while in all the previous experiment conditions in Pilot Study, I did not highlight any of the blocks at any time.

4.2.3. Procedure

The participants only needed to perform the experiment with PEC. Before the experiment, participants had 10 minutes to get familiar again with the equipment and the key layout. The participants would enter four sessions, of which the phrases were randomly generated from the MacKenzie Phrase Set.

4.2.4. Results and discussion

Based on $T - tests$, PEC has a significant effect on WPM ($p = .001^*, Cohens'\ d = 0.32$). Also, PEC ($M = 13.78$, $SD = 3.83$) is greater in WPM than PCC ($M = 12.31, SD = 5.34$). No significant effect was shown on TER ($p = 0.205$) and NCER ($p = 0.447$). The result is reported in Table 2.

According to the data obtained on the subject, the use of highlighting the selected character block significantly improves WPM, while having no adverse effect on TER or NCER. So we decide to use the text entry display interface with highlighted high-frequency letters in the following user study.

5. User study

We evaluate learnability and usability in our user study. The hardware setup is the same with our pilot study.

5.1. Study design

5.1.1. Participants and hardware setup

32 participants in total (half are men and half are women) of an average age of 20 distributing equally from 18 to 30, all with normal or corrected-to-normal vision and no impaired wrists conducted this study. Half of the participants have experience in using VR devices. The cohort was required to perform all the tasks in our user study. The hardware used was the same as in Pilot Study, and a treadmill was used as well.

5.1.2. Task 1

This task is to evaluate CrossKeys' performance and learnability. Participants were assigned to enter fifteen phrases, which would be randomly generated from the MacKenzie Phrase Set, as fast as possible.

5.1.3. Task 2

This task is to evaluate the performance of CrossKeys under long-term input, which we define as a fatigue test. The participants would need to enter phrases generated from the Mackenzie Phrase Set for 2 minutes, 6 minutes, and 10 minutes.

5.1.4. Task 3

Since our CrossKeys is totally capable to be manipulated easily with a single controller, it is necessary to evaluate its performance when being used while moderate motions, among which the most common is walking. Therefore this task is designed to evaluate the performance of CrossKeys in non-stationary situations. Participants from Task 2 would try to enter words in a moving condition. They would each need to respectively enter ten phrases in stationary condition and walking on a treadmill at a fixed speed.

5.1.5. Metrics

The objective metrics are the same as those in the pilot studies, including WPM, TER and NCER. Moreover, we import several subjective metrics to evaluate users' feelings when using CrossKeys. We use SSQ (Simulator Sickness Questionnaire) (Kennedy et al., 1993) testing rates of Oculomotor, Nausea, Disorientation and Total Severity, using NASA-TLX (Hart 2006) testing workload.

5.2. Task 1: Learnability and performance test

The role of task 1 is to evaluate the learnability of CrossKeys and also to make an evaluation of the general performance of CrossKeys from the objective perspective.

5.2.1. Procedure

Every participant would undergo a three-day experiment. The whole experiment contains a series of sessions, with one session to be finished in each day (Session 1, Session 2, and Session 3). In each day, the participants would have 45 minutes to train to enter phrases as fast and accurately as possible. After the training, they would enter 15 phrases generated from the MacKenzie Phrase Set as a session.

5.2.2. Analytical method

We employed a one-way repeated measures ANOVA with Sessions as the within-subject variable. The Greenhouse-Geisser adjustment was employed for degrees of freedom for violations of sphericity.

5.2.3. Results and discussion

Table 3 shows the mean WPM, TER and NCER in task 1. The ANOVA reveals a significant effect of the Session on WPM ($F_{2,30} = 207.197$, $p = .000^*$). Compared to Session 1, the WPM improvement in Session 2 and Session 3 are 19.41% and 50.82%. This was make possible by the cumulative learning time and the participants' growing expertise in utilizing CrossKeys for text entry. Additionally, the observed improvement suggests that our text entry method possesses untapped potential for achieving even higher words per minute (WPM). By Session 3, the average WPM had climbed to 17.73, with one participant notably reaching 24.73 WPM after typing a single set of phrases. This performance indicates that CrossKeys typically offers faster WPM rates compared to other selection-based entry methods, such as HiPad introcued by Jiang and Weng (2020).

The ANOVA doesn't show any significant effects on TER ($F_{2,30} = 0.011$, $p = .989$) and NCER ($F_{1.171,17.571} = 4.100$, $p = .053$). In general, NCER had decreasing values, which also resulted from the increasing proficiency of the participants, like WPM. In Session 2, TER shows a slight increase. This may be due to the fact that, during the actual input, there may be candidate words that are very close to

Table 3. Average WPM, TER, NCER in task 1. Statistical significance is denoted with an asterisk.

Metrics	Session	Mean Value ±*std.dev*	*p*-value	Comparison to Session 1
WPM	1	11.76 ± 1.4	0.000*	–
	2	14.04 ± 1.19		+19.41%
	3	17.73 ± 1.5		+50.82%
TER	1	7.52% ± 5.16%	0.970	–
	2	7.90% ± 9.91%		+4.98%
	3	7.49% ± 12.11%		−0.49%
NCER	1	0.64% ± 0.46%	0.053	–
	2	0.44% ± 0.31%		−31.24%
	3	0.30% ± 0.36%		−52.90%

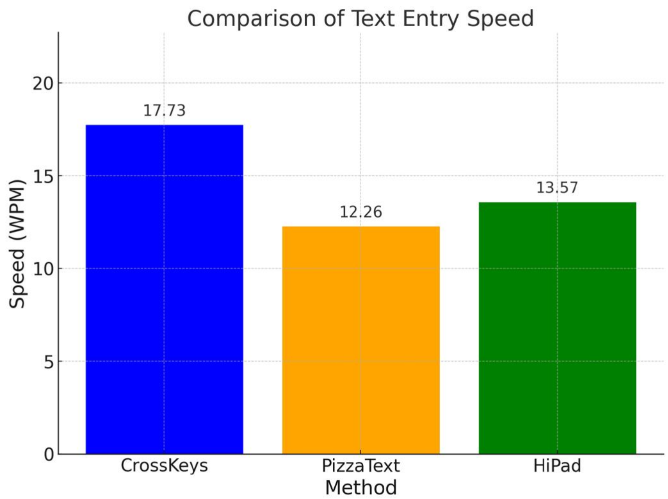


Figure 7. Text entry speed in words per minute (WPM) for CrossKeys (our method), PizzaText (Yu et al., 2018), and HiPad (Jiang and Weng 2020). as depicted, CrossKeys demonstrates a significantly higher speed compared to the other two methods.

the target word, but the candidate word list does not show the target word itself. At this point, participants are more likely to choose the candidate word that is close to the target word first, and then make changes to that candidate word. In other words, when entering texts, users under certain circumstances would sacrifice TER for speed (WPM) and NCER by selecting a partly identical candidate word and deleting the different part on its tail. This leads to an increase in the CER, thus resulting in an increase in TER, and yet no increase in NCER.

We chose to use HiPad (Jiang & Weng 2020) and PizzaText (Yu et al., 2018), which adopt the analogous interaction mode (input for virtual reality using a single handle or controller), as benchmarks, and the comparison result on WPM is presented in Figure 7.

On the basis of the WPM and error rates, we can conclude that CrossKeys is highly efficient while having a relatively low not corrected error rate. At the same time, according to the trend of the data, our method also has good learnability, and if it takes a few more days of learning, it is likely to achieve higher input speed.

5.3. Task 2: Fatigue test

We designed three conditions as follows to fully evaluate the user experience when intensely using our CrossKeys for a prolonged time by recording and analyzing the participants' continuous task loads.

5.3.1. CC

The participants would enter phrases randomly generated from the MacKenzie Phrase Set (MacKenzie & Soukoreff 2003) for two consecutive minutes.

5.3.2. EC1

The participants would enter phrases generated the same way as CC, for six consecutive minutes.

Table 4. Average WPM, TER, NCER in task 2. Statistical significance is denoted with an asterisk.

Metrics	Condition	Mean Value $\pm std.dev$	*p*-value
WPM	CC	10.48 ± 3.00	0.313
	EC1	9.75 ± 2.21	
	EC2	9.72 ± 2.17	
TER	CC	$12.20\% \pm 7.53\%$	0.012*
	EC1	$15.90\% \pm 9.41\ \%$	
	EC2	$16.07\% \pm 6.60\%$	
NCER	CC	$2.90\% \pm 5.56\%$	0.339
	EC1	$5.22\% \pm 5.83\%$	
	EC2	$2.83. \pm 2.28\%$	

5.3.3. EC2

The participants would enter phrases generated the same way as CC, for ten consecutive minutes.

5.3.4. Procedure

The entire experiment was conducted using a within-subject design. Every time they finished one test of three, they would need to finish the SSQ and NASA-TLX tests for their current feelings. Participants have a 15-minute break between tests to ensure adequate rest periods, which helps in preventing the exacerbation of motion sickness from extended use of VR devices, potentially affecting the outcomes of the experiment. Typically, a 15-minute rest period is sufficient for individuals to recuperate from any unease; however, additional rest time will be granted upon request, at the discretion of the participant.

5.3.5. Analytical method

We employed a one-way repeated measures ANOVA with Duration (2 min, 6 min, 10 min) as the within-subject variable. The Greenhouse-Geisser adjustment was employed for degrees of freedom for violations of sphericity. LSD correction was employed for post hoc pairwise comparisons.

5.3.6. Results and discussion

Table 4 shows the mean WPM, TER and NCER in task 2. In terms of WPM, EC1's data is down 6.97% compared to CC's data, and EC2's data is down 0.3% compared to EC1's data. The ANOVA doesn't show any significant effect of Session on WPM ($F_{1.254, 18.808} = 1.146$, $p = .313$). The decrease in the data was probably caused by a certain degree of fatigue of the participants, but the ANOVA indicates that such a decrease was not severe. This may indicate that the present input method has relatively good fatigue resistance. In terms of error rates, the ANOVA reveals a significant effect of Duration on TER ($F_{2,30} = 3.608$, $p = .039$). The EC1 data (M = 15.90%, SD = 9.41%) are up 30.33% from the CC ($M = 12.20\%, SD = 7.53\%$) data, and the EC2 data ($M = 16.07\%, SD = 6.60\%$) are up 1.07% from the EC1 data. Post-hoc pairwise comparisons showed that the TER of EC1 was significantly ($p = .025$) higher than that of CC, and there is no significant difference ($p = .928$) between TERs of EC1 and EC2. The ANOVA didn't show any significant effect of Duration on NCER ($F_{2,30} = 1.677$, $p = .204$). There could be two reasons for the

Table 5. Statistics of Simulator Sickness ratings in terms of oculomotor (O), nausea (N), disorientation (D), and total severity (TS). Statistical significance is denoted with an asterisk.

Metrics	Conditon	Mean Value $\pm std.dev$	*p*-value	*p*-value compared to EC2
N	CC	7.75 ± 11.67	0.271	0.216
	EC1	7.75 ± 9.36		0.164
	EC2	10.14 ± 13.25		–
O	CC	17.06 ± 20.43	0.003*	0.009*
	EC1	18.00 ± 14.87		0.000*
	EC2	30.79 ± 20.98		–
D	CC	9.57 ± 20.78	0.023*	0.066
	EC1	7.83 ± 19.00		0.001*
	EC2	18.27 ± 22.57		–
TS	CC	5.63 ± 8.63	0.011*	0.024
	EC1	5.29 ± 7.35		0.000*
	EC2	10.03 ± 8.92		–

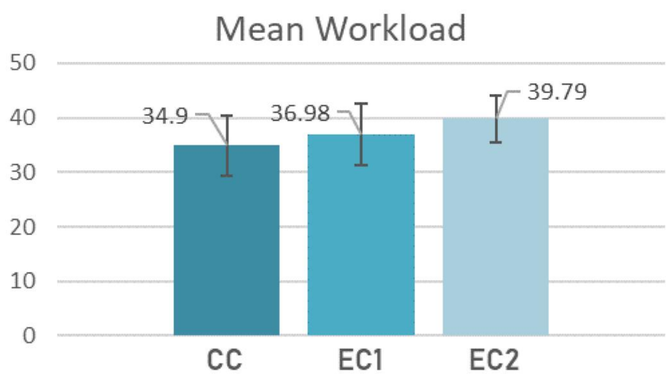


Figure 8. The average workload of three status (short, longer and prolonged). Error bars indicate ± 1 standard deviation.

increasing TER under the three conditions. The first one is similar to the explanation in Task 1, in which participants sacrificed TER for better WPM in some cases. Moreover, NCER remains relatively stable at a low level, indicating that the TER is indeed largely due to the increase in CER (corrected error rate) caused by the participants' modification of their input. Another reason was that participants were affected by fatigue, which resulted in a certain number of times when the wrong letter or candidate word was chosen.

Table 5 shows the SSQ ratings for Oculomotor (O), Nausea (N), Disorientation (D), and Total Severity (TS) and Figure 8 shows the mean workload scores under the NASA-TLX test. The ANOVA shows that there is no significant effect of Duration on N ($F_{2,30} = 1.364,\ p = .271$). The ANOVA reveals a significant effect on O ($F_{2,30} = 7.107,\ p = .003$), D ($F_{1.398,20.974} = 5.225,\ p = .023$) and TS ($F_{1.46,21.901} = 6.504,\ p = .011$). Post-hoc pairwise comparisons show that all data (including O, D, TS) of EC2 are respectively significantly higher than those of CC and EC1, and the LSD comparison *p*-values were shown in Table 6. Although the rise in EC2 may indicate that the participants didn't feel comfortable under EC2, the actual scores of these three items are still maintained at a very low level, with the highest one being O in EC2, reaching only 30.79. No significant effect was shown between CC and EC1 in four items, as is shown in Figure 8. and there was a decrease in O and TS from CC to EC1. This may show that the participants felt it more comfortable to enter phrases for around 6 minutes. As for the NASA-TLX test, the workload fractions of CC, EC1, EC2 showed a slowly increasing trend, but the *ANOVA* yielded no significant effect of Duration on workload ($F_{2,30} = 2.749,\ p = .080$). This indicates that the increase in workload caused by prolonged use of CrossKeys is not severe and is acceptable.

Table 6. *p*-Values of SSQ scores comparing CC with EC1 in Task 2.

Metrics	N	O	D	TS
p-value	1.000	0.837	0.609	0.831

According to objective metrics, CrossKeys has good usability in relatively long continuous input, i.e., it maintains stable input efficiency and error rates, indicating that it has a certain degree of fatigue resistance. According to subjective metrics, CrossKeys can maintain a low level of workload; when combined with SSQ feedback, the optimal time of our method under the condition of keeping people comfortable for longer input time is six minutes to ten minutes.

5.4. Task 3: Evaluation of in-motion performance

During the study of in-motion performance, we employed a treadmill to simulate daily walking; to ensure safety, as shown in Figure 9 we also embodied the walking path into the virtual environment and set it to the same speed so that the participants would feel more directed and natural while walking on a treadmill. Also, participants were required to hold one hand on a side of the treadmill's handles and try to make sure that their arms are on the same plane as their bodies so that they would prevent slipping out of the treadmill or falling down. We designed two conditions as follows.

5.4.1. CC

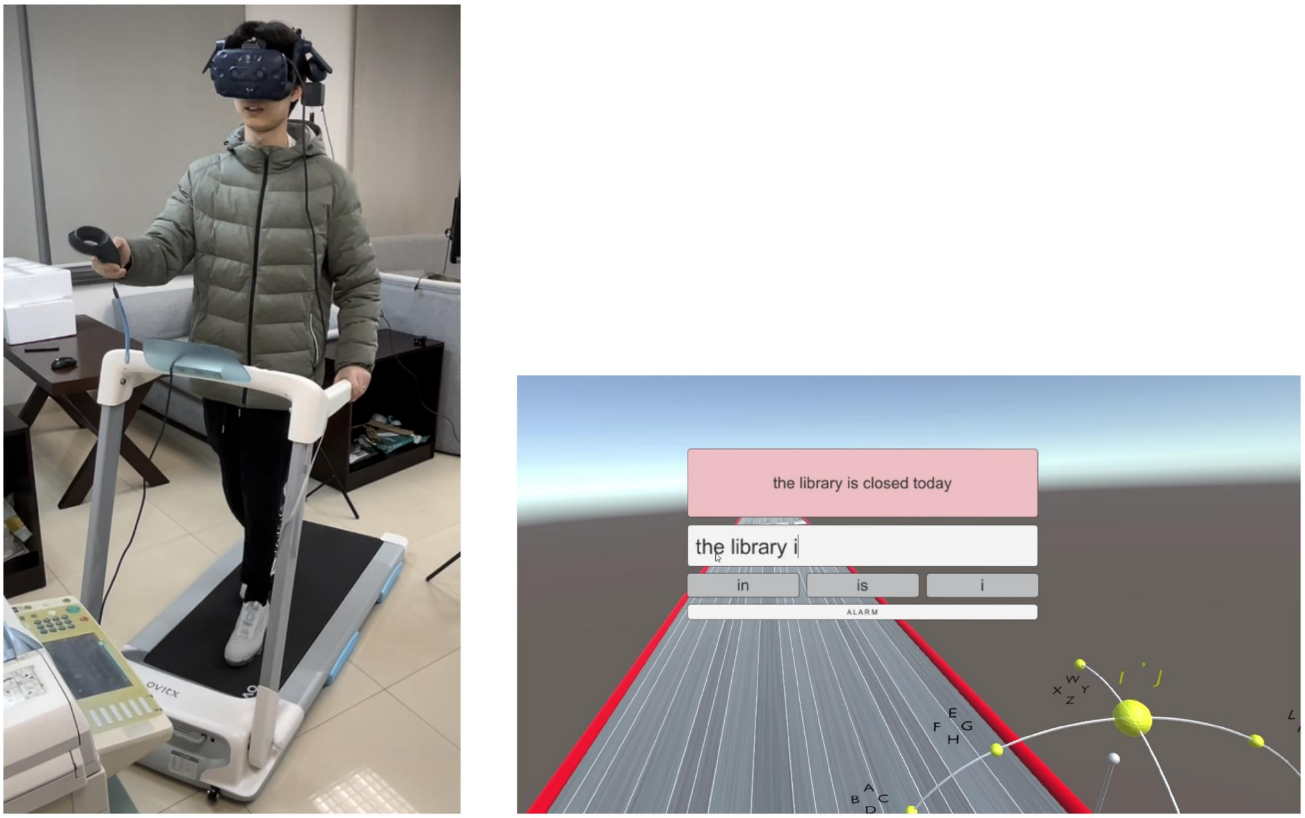
The participants stand still on a power-off treadmill and enter ten phrases randomly selected from the MacKenzie Phrase Set.

5.4.2. EC

The participants enter ten phrases generated the same way as CC, only under this condition they will be walking on a power-on treadmill of a daily walking speed (1.6 m/s).

5.4.3. Procedure

The entire experiment was conducted complying to a within-subject design. Participants were given 45 minutes to review the use of CrossKeys and practice it to be as proficient as possible. In addition, they would also use this time to familiarize themselves with the treadmill. Between the training period and sessions, participants would have enough time to rest. In Session 1 (CC), the participants need to stand on the stationary treadmill and enter ten sets of phrases which are randomly generated from the Mackenzie PhraseSet. In Session 2 (EC), they need to walk on the treadmill running at 1.6 meters per second and enter ten phrases the same way as they did in CC. The task starts with Session 1 and ends with Session 2. After each session, the participants would be asked to fill the NASA-TLX form to record their subjective evaluation of a myriad of task loads.



(a) The scene of one participant typing and walking on a treadmill.

(b) The virtual environment during the study of in-motion performance, where a walking path is embodied to ensure participants feel directed and natural while walking on a treadmill.

Figure 9. 3^{rd} (a) And 1^{st} (b) person view of typing using CrossKeys on a treadmill during task 3: Evaluation of in-motion performance.

Table 7. Average WPM, TER, NCER in task 3.

Metrics	Conditon	Mean Value $\pm std.dev$	*p*-value	Cohen's d
WPM	CC	17.42 ± 3.92	0.916	0.01
	EC	17.37±3.73		
TER	CC	7.80%±7.91%	0.894	0.02
	EC	7.64%±7.15%		
NCER	CC	1.03%±3.93%	0.546	0.14
	EC	1.29%±3.12%		

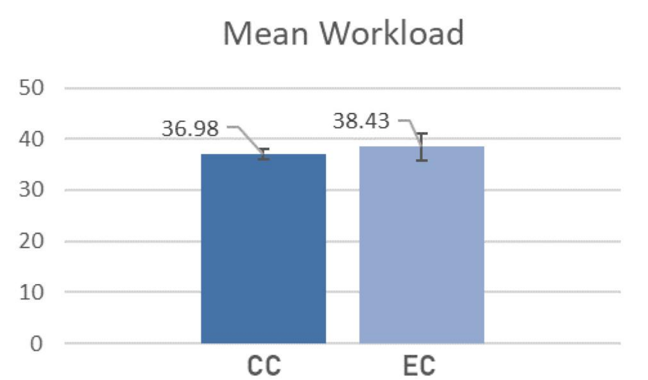


Figure 10. The average workload of two status (staitionary and in-motion). Error bars indicate ± 1 standard deviation.

5.4.4. Analytical method

We employed a Paired *T*-test with Status (Stationary, in-motion) as the within-subject variable. Shapiro-Wilk Test is employed for testing the normal distribution of the data.

5.4.5. Results and discussion

Table 7 shows the mean WPM, TER and NCER in task 3. The *T*-tests show that there is no big significant effect of Status on WPM ($t = 0.150$, $p = .916$, $Cohen's\ d = 0.01$), TER ($t = 0.190$, $p = .894$, $Cohen's\ d = 0.02$) or NCER ($t = -0.606$, $p = .546$, $Cohen's\ d = 0.14$). The WPM of EC decreased by 0.297% compared to that of CC. NCER in EC increased by 24.501% compared to CC. The decline in WPM was predictable, as it is likely that participants in the walking state would not be able to fully focus on entering phrases, and the decline in WPM of EC was not significant, suggesting that CrossKeys still retains good usability in the walking state. The increase in NCER may be due to the fact that in the walking state, subjects do not put the same degree of attention on whether their input is consistent with the requested phrase as in the stationary state. Also, although the rise in NCER was great seemingly, the mean NCER of EC was actually only 1.29%, which is still at a low level and not significantly different from NCER of CC. This means that CrossKeys also maintains a low NCER in the walking state.

Figure 10 shows the mean workload scores under the NASA-TLX test. The mean workload of EC is 3.92% higher than that of CC. The *T*-test shows that such difference is significant ($t = -2.178$, $p = .046$), and the *p*-value is relatively close to the threshold 0.05. By checking the scores for each option in the questionnaire, we found that the main reason for the high scores in EC was that participants generally scored higher for PHYSICAL DEMAND than CC. The main source of the high scores for this item may be the slight increase in physical burden that participants experienced while walking.

Based on subjective and objective metrics, we can conclude that CrossKeys has good usability in non-stationary situations, i.e., it guarantees input efficiency and low error rates in motion situations, while the slight increase in the workload is mainly from motion, not our input method.

6. Conclusion

The results of our fastidiously designed pilot and user studies outcomes well and demonstrates that our CrossKeys outperforms the state-of-the-art method with an average WPM of 17.73 and an error rate (NCER) of 0.30% along with a low task load according to the data collected from the participants after only about 2 hours of training. We also introduce in-motion text entry in VR, making there could be a wider application of our CrossKeys.

However, this research has a number of limitations and some room to be refined with greater expertise, suggesting new directions for future work:

1. The currently believed best layout is selected from a relatively small amount, which indicates that there might be other layouts of higher efficiency to be found.
2. From an ergonomics perspective, users whose dominant hand is on their right still feel quite unnatural when reaching for the blocks behind and on the right; likewise, users whose dominant hand is on their left, also feel the same discomfort when reaching for the blocks behind and on the most left. Maybe unequally distanced blocks would help.

In conclusion, CrossKeys is efficient and light-weighted enough to be utilized in various scenarios in virtual environments. Even with limitations, the promising future of CrossKeys can hardly be overshadowed.

Disclosure statement

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

ORCID

Zhaoxun Liu http://orcid.org/0009-0003-5209-2242

References

Chen, S., Wang, J., Guerra, S., Mittal, N., & Prakkamakul, S. (2019). Exploring word-gesture text entry techniques in virtual reality. In *Extended abstracts of the 2019 CHI conference on human factors in computing systems* (pp. 1–6). Association for Computing Machinery. https://doi.org/10.1145/3290607.3312762

Darabkh, K. A., Alturk, F. H., & Sweidan, S. Z. (2018). Vrcdea-tcs: 3d virtual reality cooperative drawing educational application with textual chatting system. *Computer Applications in Engineering Education*, *26*(5), 1677–1698. https://doi.org/10.1002/cae.22017

Delanoy, J., Aubry, M., Isola, P., Efros, A. A., & Bousseau, A. (2018, July). 3d sketching using multi-view deep volumetric prediction. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, *1*(1), 1–22. https://doi.org/10.1145/3203197

Dudley, J. J., Zheng, J., Gupta, A., Benko, H., Longest, M., Wang, R., & Kristensson, P. (2023). nov) Evaluating the performance of hand-based probabilistic text input methods on a mid-air virtual qwerty keyboard. *IEEE Transactions on Visualization and Computer Graphics*, *29*(11), 4567–4577. https://doi.org/10.1109/TVCG.2023.3320238

Esmaeili, M., Gamage, K., Tan, E., Campolo, D. (2011). Ergonomic considerations for anthropomorphic wrist exoskeletons: A simulation study on the effects of joint misalignment. In *2011 IEEE/RSJ international conference on intelligent robots and systems*, San Francisco, CA, USA (pp. 4905–4910). IEEE. https://doi.org/10.1109/IROS.2011.6095136

Feiz, S., & Ramakrishnan, I. V. (2019). Exploring feasibility of wrist gestures for non-visual interactions with wearables. In *Proceedings of the 16th international web for all conference*. Association for Computing Machinery. https://doi.org/10.1145/3315002.3317570

Gong, J., Xu, Z., Guo, Q., Seyed, T., Chen, X. A., Bi, X., & Yang, X.-D. (2018). 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). Association for Computing Machinery. https://doi.org/10.1145/3173574.3173755

Gu, Y., Yu, C., Li, Z., Li, Z., Wei, X., & Shi, Y. (2020). Qwertyring: Text entry on physical surfaces using a ring. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, *4*(4), 1–29. https://doi.org/10.1145/3432204

Gupta, A., Ji, C., Yeo, H.-S., Quigley, A., & Vogel, D. (2019). Rotoswype: Word-gesture typing using a ring. In *Proceedings of the 2019 CHI conference on human factors in computing systems* (pp. 1–12). Association for Computing Machinery. https://doi.org/10.1145/3290605.3300244

Han, X., Gao, C., & Yu, Y. (2017). Deepsketch2face: A deep learning based sketching system for 3d face and caricature modeling. *ACM Transactions on Graphics*, *36*(4), 1–12. https://doi.org/10.1145/3072959.3073629

Hart, S. G. (2006). *Nasa-Task Load Index (Nasa-Tlx); 20 Years Later. Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *50*, 904–908.

Ikei, Y., Yem, V., Tashiro, K., Fujie, T., Amemiya, T., & Kitazaki, M. (2019). Live stereoscopic 3d image with constant capture direction of 360° cameras for high-quality visual telepresence. In *2019 IEEE conference on virtual reality and 3d user interfaces (VR)* (pp. 431–439).

Jiang, H., & Weng, D. (2020). Hipad: Text entry for head-mounted displays using circular touchpad. In *2020 IEEE conference on virtual reality and 3d user interfaces (VR)* (pp. 692–703).

Jiang, Y., Zhang, C., Fu, H., Cannavò, A., Lamberti, F., Lau, H. Y. K., & Wang, W. (2021). Handpainter - 3d sketching in vr with hand-based physical proxy. In *Proceedings of the 2021 CHI conference on human factors in computing systems*. Association for Computing Machinery.

Kennedy, R. S., Lane, N. E., Berbaum, K. S., & Lilienthal, M. G. (1993). Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The International Journal of Aviation Psychology*, *3*(3), 203–220. https://doi.org/10.1207/s15327108ijap0303_3

Kim, H., Remaggi, L., Jackson, P. J., & Hilton, A. (2019). Immersive spatial audio reproduction for VR/AR using room acoustic modelling from 360° images. In *2019 IEEE conference on virtual reality and 3d user interfaces (VR)* Osaka, Japan (pp. 120–126). IEEE. https://doi.org/10.1109/VR.2019.8798247

Kurauchi, A., Feng, W., Joshi, A., Morimoto, C., & Betke, M. (2016). Eyeswipe: Dwell-free text entry using gaze paths. In *Proceedings of the 2016 CHI conference on human factors in computing systems* (pp. 1952–1956). Association for Computing Machinery. https://doi.org/10.1145/2858036.2858335

Kwan, K. C., & Fu, H. (2019). Mobi3dsketch: 3d sketching in mobile ar. In *Proceedings of the 2019 CHI conference on human factors in computing systems* (pp. 1–11). Association for Computing Machinery. https://doi.org/10.1145/3290605.3300406

Lee, D., Kim, J., & Oakley, I. (2021). Fingertext: Exploring and optimizing performance for wearable, mobile and one-handed typing. In *Proceedings of the 2021 CHI conference on human factors in computing systems*. Association for Computing Machinery. https://doi.org/10.1145/3411764.3445106

Liang, C., Hsia, C., Yu, C., Yan, Y., Wang, Y., & Shi, Y. (2023, January) Drg-keyboard: Enabling subtle gesture typing on the fingertip with dual IMU rings. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, *6*(4). https://doi.org/10.1145/3569463

Lu, X., Yu, D., Liang, H.-N., Feng, X., & Xu, W. (2019). Depthtext: Leveraging head movements towards the depth dimension for hands-free text entry in mobile virtual reality systems. In *2019 IEEE conference on virtual reality and 3d user interfaces (VR)* (pp. 1060–1061).

Luo, Z., Zhou, J., Zhu, H., Du, D., Han, X., & Fu, H. (2021). Simpmodeling: Sketching implicit field to guide mesh modeling for 3d animalmorphic head design. In *The 34th annual ACM symposium on user interface software and technology* (pp. 854–863). Association for Computing Machinery. https://doi.org/10.1145/3472749.3474791

MacKenzie, I. S., & Soukoreff, R. W. (2003). Phrase sets for evaluating text entry techniques. In *CHI '03 extended abstracts on human factors in computing systems* (pp. 754–755). Association for Computing Machinery. https://doi.org/10.1145/765891.765971

Markussen, A., Jakobsen, M. R., & Hornbæk, K. (2014). Vulture: A mid-air word-gesture keyboard. In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 1073–1082). Association for Computing Machinery. https://doi.org/10.1145/2556288.2556964

Pedrosa, D., Pimentel, M. D. G., Wright, A., & Truong, K. N. (2015). mar) Filteryedping: Design challenges and user performance of dwell-free eye typing. *ACM Transactions on Accessible Computing*, *6*(1), 1–37. https://doi.org/10.1145/2724728

Quinn, P., & Zhai, S. (2016). A cost-benefit study of text entry suggestion interaction. In *Proceedings of the 2016 CHI conference on human factors in computing systems* (pp. 83–88). Association for Computing Machinery. https://doi.org/10.1145/2858036.2858305

Shen, J., Dudley, J., & Kristensson, P. (2023). Fast and robust mid-air gesture typing for ar headsets using 3d trajectory decoding. *IEEE Transactions on Visualization and Computer Graphics*, *29*(11), 4622–4632. https://doi.org/10.1109/TVCG.2023.3320218

Sormann, M., Bauer, J., Zach, C., Klaus, A., & Karner, K. (2004). Vr modeler: From image sequences to 3d models. In *Proceedings of the 20th spring conference on computer graphics* (pp. 148–156). Association for Computing Machinery. https://doi.org/10.1145/1037210.1037233

Soukoreff, R. W., & MacKenzie, I. S. (2003). 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). Association for Computing Machinery.

Strohmeier, P., Vertegaal, R., & Girouard, A. (2012). With a flick of the wrist: Stretch sensors as lightweight input for mobile devices. In *Proceedings of the sixth international conference on tangible, embedded and embodied interaction* (pp. 307–308). Association for Computing Machinery. https://doi.org/10.1145/2148131.2148195

Takala, T. M., Hsin, C. C., & Kawai, T. (2019). Stand-alone, wearable system for full body vr avatars: Towards physics-based 3d interaction. In *2019 IEEE conference on virtual reality and 3d user interfaces (VR)* (pp. 1398).

Valkov, D., & Linsen, L. (2019). Vibro-tactile feedback for real-world awareness in immersive virtual environments. In *2019 IEEE conference on virtual reality and 3d user interfaces (VR)* (pp. 340–349).

Wong, P. C., Zhu, K., & Fu, H. (2018). Fingert9: Leveraging thumb-to-finger interaction for same-side-hand text entry on smartwatches. In *Proceedings of the 2018 CHI conference on human factors in computing systems* (pp. 1–10). Association for Computing Machinery. https://doi.org/10.1145/3173574.3173752

Xu, W., Liang, H.-N., Zhao, Y., Zhang, T., Yu, D., & Monteiro, D. (2019). 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. https://doi.org/10.1109/TVCG.2019.2898736

Xu, Z., Chen, W., Zhao, D., Luo, J., Wu, T.-Y., Gong, J., Yin, S., Zhai, J., & Yang, X.-D. (2020). Bitiptext: Bimanual eyes-free text entry on a fingertip keyboard. In *Proceedings of the 2020 CHI conference on human factors in computing systems* (pp. 1–13). Association for Computing Machinery. https://doi.org/10.1145/3313831.3376306

Xu, Z., Wong, P. C., Gong, J., Wu, T.-Y., Nittala, A. S., Bi, X., Steimle, J., Fu, H., Zhu, K., & Yang, X.-D. (2019). 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). Association for Computing Machinery. https://doi.org/10.1145/3332165.3347865

Yanagihara, N., & Shizuki, B. (2018). Cubic keyboard for virtual reality. In *Proceedings of the symposium on spatial user interaction* (pp. 170). Association for Computing Machinery. https://doi.org/10.1145/3267782.3274687

Yeo, H.-S., Lee, J., Kim, H-i., Gupta, A., Bianchi, A., Vogel, D., Koike, H., Woo, W., & Quigley, A. (2019). Wrist: Watch-ring interaction and sensing technique for wrist gestures and macro-micro pointing. In *Proceedings of the 21st international conference on human-computer interaction with mobile devices and services*. Association for Computing Machinery. https://doi.org/10.1145/3338286.3340130

Yi, X., Liang, C., Chen, H., Song, J., Yu, C., Li, H., & Shi, Y. (2023). From 2d to 3d: Facilitating single-finger mid-air typing on qwerty keyboards with probabilistic touch modeling. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, *7*(1), 1–25. https://doi.org/10.1145/3580829

Yi, X., Yu, C., Zhang, M., Gao, S., Sun, K., & Shi, Y. (2015). 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). Association for Computing Machinery. https://doi.org/10.1145/2807442.2807504

Yu, C., Gu, Y., Yang, Z., Yi, X., Luo, H., & Shi, Y. (2017). Tap, dwell or gesture? exploring head-based text entry techniques for HMDs. In *Proceedings of the 2017 CHI conference on human factors in computing systems* (pp. 4479–4488). Association for Computing Machinery. https://doi.org/10.1145/3025453.3025964

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

About the authors

Zhaoxun Liu received B.Eng. from the School of Computer Science and Engineering of Beihang University, China. His research focuses on HCI with specific interests in text entry and object manipulation in virtual reality.

Xiaolong Liu received Ph.D. degree from the School of Computer Science and Engineering of Beihang University, China. He is currently an Assistant Researcher with the National Innovation Institute, Beijing, China. His current research focuses on virtual reality, augmented reality, and HCI.

Lili Wang received her Ph.D. degree from Beihang University, Beijing, China. She is a professor with the School of Computer Science and Engineering of Beihang University and a researcher with the State Key Laboratory of Virtual Reality Technology and Systems. Her interests include virtual reality, real-time rendering and HCI.