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

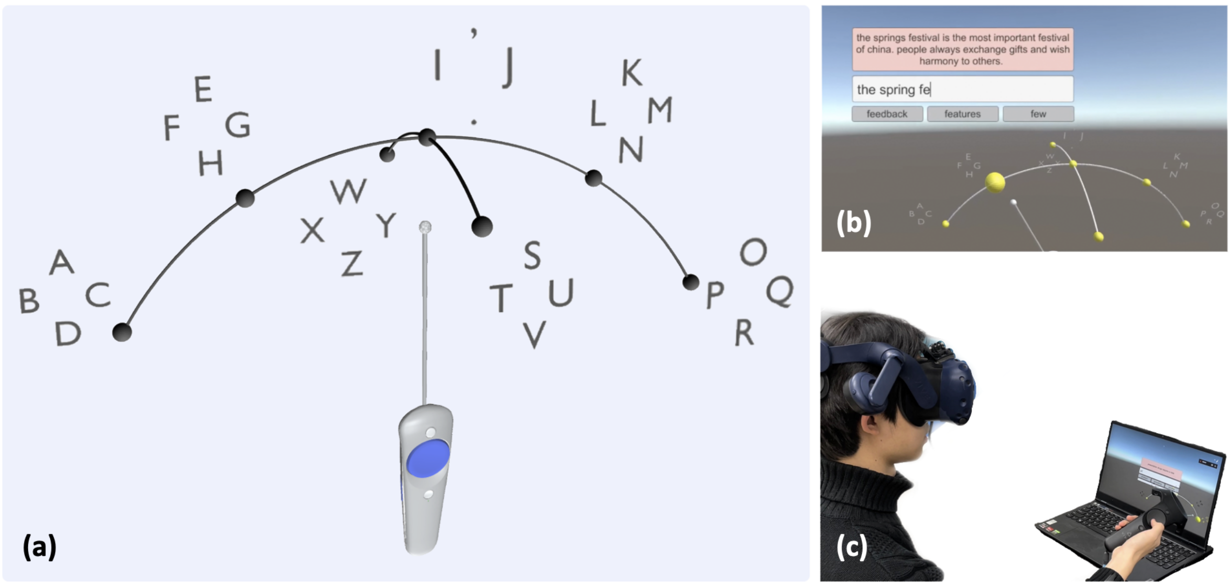


Fig. 1. The overview of CrossKeys. (a) is the core of CrossKeys, consisting of the three-dimensional structure of character layout and the embodied controller in the virtual environment; (b) is the user perspective in most of our pilot and user studies, consisting of a text indicator, an input field, 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.

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.

Index Terms—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. To start with, about text entry, there are many text entry methods, we can refer to, no matter they are devised for traditional electronic devices or newer technologies like VR. For instance, methods like PizzaText [33], FingerT9 [28], and DepthText [17] all aim at text entry in virtual environments. Then, about 3D user interaction in VR, we have DeepSketch2Face [6] 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 [25] and WRIST [31].

In this paper, we introduce CrossKeys, a novel, efficient, and handy text entry technique for VR using a single controller via wrist rotation (Fig. 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.

We initially designed three distinct layouts for our CrossKeys; to decide which to finalize, we conducted a set of pilot studies. The results indicated that participants could achieve an average of 12.31 WPM with the layout prioritized in alphabetical order after a very short period of training (10 min), which is significantly better than the other two layouts; therefore, we decided to make the layout strictly alphabetical. 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. In the study of fatigue, we asked the participants to enter phrases for 2 minutes, 6 minutes, and 10 minutes after training for 45 minutes. No significant difference was found under the analysis. There is also no significant difference in not corrected error rates among three conditions. The above illustrates that CrossKeys has relatively good fatigue resistance. In the study of in-motion performance, we employ a treadmill to simulate walking in real life. The results show that the WPM drops very little, from 17.42 to 17.36, indicating that our CrossKeys is light-weighted enough to manipulate while walking and, with a reasonable speculation, during moderate motion.

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 methods have achieved some success and also are a great inspiration to us. Moreover, in virtual environments, it is common that users need to perceive the space and interact with the system in a three-dimensional way. Many other pieces of research about three-dimensional interaction, even if they are not conducted for text entry in VR, greatly contribute to our ideas. Likewise, research about wrist-based interaction has also spurred our imagination and helped to finalize our CrossKeys.

2.1 Text Entry in Virtual Environments

Outside VR, daily, most people enter text via physical keyboards, voice recognition, and handwriting; these are the most omnipresent text entry methods for traditional electric devices. However, in VR, it would be unlikely and unsustainable to have a physical keyboard or to write on a pad; also, under countless circumstances, it would be improper to speak for voice recognition, such as when in a library where silence is needed. VR, as a technology at its burgeoning incipience, may not be so common for everyone to have access to it; so, text entry in VR could break the bound of traditional methods, hence the myriad of innovative methods that have been introduced.

First of all, the majority of text entry methods in VR are based on the tracking of hands. 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 [32] 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.

Text entry based on controllers is also very common. PizzaText [33] 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 [28] and QwertyRing [5]. Finger-touching 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 [17] employs the acceleration-sensitive embedded IMU sensor to translate head movement into texts; RingText [29], 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.

Besides, text entry based on eye movement is another innovative method, such as Filteryedping [21], EyeSwipe [14]. 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 [20]. 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.

In perceiving virtual space as reality, 3D-drawing is currently a popular research topic, where users move their fingers in a virtual environment to draw the desired graphics. In 2017, Delanoy Johanna et al. proposed a way to generate 3D sketches by predicting the depth volume based on several 2D sketches through machine learning [11] and DeepSketch2Face [6] was put forward as a deep learning based sketching system for 3D face and caricature modeling.. Hongbo Fu's team also implemented a way to perform 3D-drawing in mobile AR devices in a paper published in 2019 [15], but at this time the research was still dependent on the cell phone screen. In 2021, their team came up with HandPainter - a way to draw in virtual environments via hand-worn hardware [10]. This evolution makes 3D-drawing more realistic and allows users to have a better drawing experience.

Modeling in virtual environments is another important topic. Usually, designers model using professional 3D modeling software, such as MAYA, 3D Max, etc., but 3D-modeling born out of 3D input technology may overturn this traditional way. Users in virtual environments can quickly construct three-dimensional models directly through action of hands. The same is true in the field of augmented reality (AR). As early as the beginning of this century, the idea of automatically generating 3D models in VR through image sequences was already proposed [23]. In recent years, VR technology has gradually matured, the corresponding technology has been improved, and 3D-modeling has been a lot more common. In education, K. A. Darabkh develops a 3D drawing and modeling tool for schools [2]. In addition, the combination with 3D sketching technology makes the modeling accuracy of 3D-modeling improved, and H. Fu and his colleagues propose a method to generate details on the model by sketching a few strokes on the already built 3D model, making 3D-modeling more realistic [18].

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 [30] 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 [1], 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 [8], 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 [13] 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 [27], 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 [26], 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 [16] 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 [4] 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 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 feasibility of wrist gestures for non-visual interactions with wearables [3]. 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 [25] and WRIST [31]. 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

Our CrossKeys complies with the following design rationales.

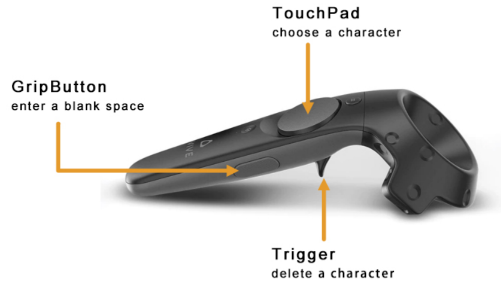


Fig. 2. *TouchPad*, *Trigger*, and *GripButton* on a HTC Vive controller. These three interactive units enable all the interaction and functionality designed for our CrossKeys.

The design of CrossKeys was partly inspired by sign language, most of which are expressed via hand movement. 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 in wrist-based interaction. Rotating a VR controller is a natural tendency when holding one. Many other applications in VR, such as video games, have already employed wrist-based actions to interact with the virtual environment, so this is not totally strange with VR users and practitioners.

Therefore, we thought that hand movement, particularly that of wrist in the context of VR, could be unprecedentedly applied to a text entry method. We initially devised two options to realize text entry based on wrist rotation: one is to perform character selection by panning the hand in six dimensions: front, back, left, right, up and down; the other is to perform selection by wrist rotation. After amounts of fastidious discussion and tests, we reasonably drew a conclusion that the panning of the hand in six dimensions would make it more painstaking for users than by wrist rotation, where users would feel a lot less fatigued. Eventually, we adopted wrist rotation as the principle interaction approach.

Traditional physical keyboards (such as qwerty keyboards, T9 keyboards) are two-dimensional; they are most accustomed to by ordinary electronic device users, making the majority of existing text entry methods inherit such a two-dimensional notion. But if we manage to break the limitation of this two-dimensional stereotype and create a three-dimensional keyboard that provides spatial interaction, a startling way to implement text entry can be introduced. CrossKeys harnesses the rotation of a wrist to select from characters; by mapping three-dimensional movement into a spatial keyboard, CrossKeys is, in the consideration of both speed and accuracy, a state-of-the-art method for text entry in VR.

About wrist rotation, on the one hand, there are thresholds to a combination of comfortable rotation angles; therefore, we manage to constrain users' wrist rotation within a ergonomics-friendly range, thanks to which users' wrist fatigue is successfully minimized. On the other hand, it is a fact that the rotation of the wrist with the ulnar radius as the axis is relatively easier, while the inward and outward movement will be much harder and is limited by hand muscles and bone structures. As a result, we assign five character blocks that can be accessed via ulnar radius rotation, and two fewer to be accessed via in-and-outward movement.

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. 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.

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 *Fig*.4, 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 with the current inputted character sequence that are likeliest to be entered. On deciding how many candidates to reveal to the user to achieve a most optimized efficiency, we referred to P. Quinn and S. Zhai's work [22] 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. After a thorough evaluation, we settled the number of candidates revealed to 3.

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 programm three buttons: *TouchPad*, *Trigger*, and *GripButton* as shown in *Fig*. 2. To elucidate our interaction method more clearly, we divide the operating space into 5 areas as shown in *Fig*.3: 4 *Deactivated* areas and 1 *Activated* area. When in the *Activated* area, *TouchPad* is to select, enter normal characters; when in the four *Deactivated* areas, *TouchPad* is to 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 a 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.

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 prefix. For example, when the user enters a sequence of "grad", the algorithm selects three candidate words ordered by using frequency from high to low, which are "graduation", "grade", and "graduate". When the dot is pointed to one of the four *Deactivated* areas, the left, middle, and right buttons of the trackpad can be clicked to select and enter the candidate on the corresponding place to improve entering speed and accuracy.

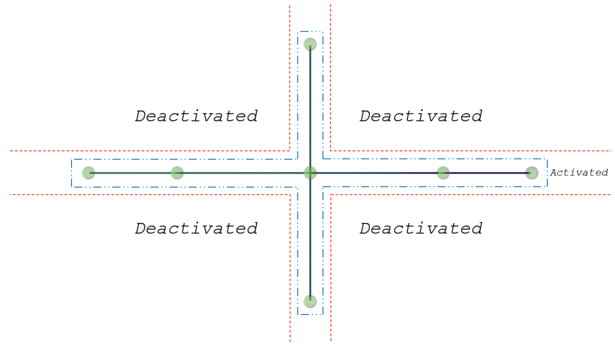


Fig. 3. The projection of four *Activated* areas and one *Deactivated* area of CrossKeys' spatial interaction.

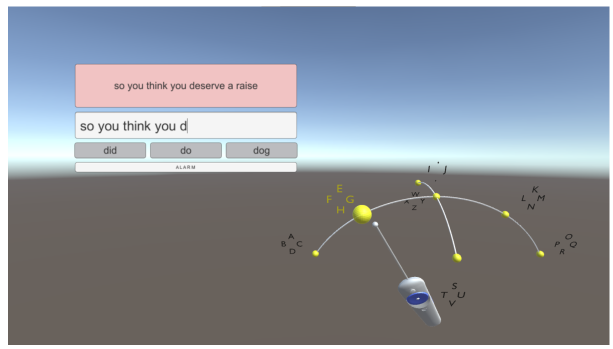


Fig. 4. The user interface system of CrossKeys, in which currently the zoomed block, $E-F-G-H$ is selected.

Special Display of Controller We concisely embody the 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 a 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.

Participants and Hardware Setup Six participants (half of them are men and the other half are women) in total of an 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.We used an Intel Core i7 processor PC with a dedicated NVIDIA GTX 1070 graphics card and an AMD processor laptop with a NVIDIA GTX 3060 graphics card. The experimental program is written in *C# .NET* programming language and runs on the Unity 3D platform.

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:

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

$$WPM = \frac{|S|}{T} \times \frac{60}{5} \quad (1)$$

$|S|$ 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.

Error rate is to evaluate our method's accuracy. The error rate is calculated based on standard typing metrics [24]. 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.

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 post-hoc 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 highlight) 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.

4.1 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).

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 Fig. 5 (a).

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 Fig. 5 (b).

PEC III The third layout is based on the frequency of letters in daily use. After sorting and organizing the frequency of use of all letters, we place the letters with high frequency of use in or as close to the default original position as possible, and the resulting layout is shown in Fig. 5 (c).

Procedure We use a within-subject design, in which participants perform the tasks in all conditions. The entire experiment consisted of 3 key layouts and 4 sessions. Before the experiment started, participants could practice and get familiar with the device as well as with the letter layout. During this time, they would try to enter 5 phrases to better familiarize themselves with the full operation and operating environment. For each session, participants would enter phrases using the input method to which they have been assigned. All phrases would be randomly generated from the MacKenzie Phrase Set [19]. After each session, participants would be given enough time to rest till the next test to avoid motion sickness, which could affect the results of the experiment.

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 significant negative impact on the error rate of text entry. $Table 1$ shows TER and NCER for the three letter layouts.

Eventually, 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.

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 Pilot Study II : Highlighting the Selected Character Block

The role of this pilot study was to figure whether the function of 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.

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

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

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.

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($p = 0.447$).

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.

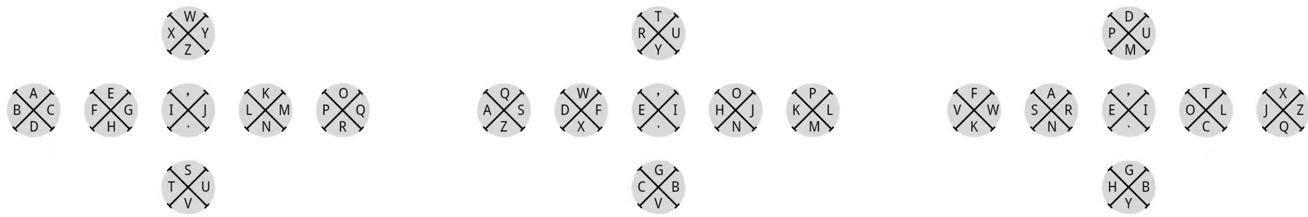
5 User Study

We design three tasks to evaluate our method from two perspectives: learnability and usability.

5.1 Study Design

Participants and Hardware Setup Thirty-two 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 hardware used were the same as in Pilot Study, and a treadmill was used as well.

Task 1 This task is to evaluate CrossKeys' performance and learnability. For 3 days, the participant would train for 45 minutes each day. Then, they would need to enter fifteen phrases, which would be randomly

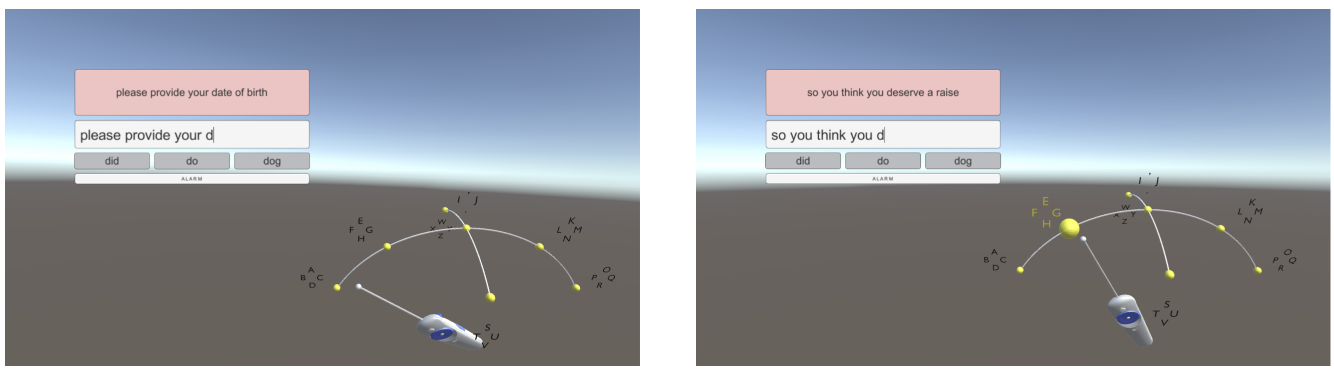


(a) The scheme for $PEC\ I:\ Alphabetical\ Order$. Every gray circle represents an actual WordBlock in CrossKeys, while the crosses help to illustrate the layout and are not shown in the real application.

(b) The scheme for $PEC\ II:\ QWERTY-like$. The meaning of the gray circles and the crosses is the same with $Fig.$ 1.

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

Fig. 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 into yellow when it is pointed at.

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

generated from the MacKenzie Phrase Set, as fast as possible.

Task 2 This task is to evaluate the performance of CrossKeys under long-term input, which we define as fatigue test. After training for 45 minutes, the participants would need to enter phrases generated from the Mackenzie Phrase Set for 2 minutes, 6 minutes, and 10 minutes.

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.

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) [12] testing rates of Oculomotor, Nausea, Disorientation and Total Severity, use NASA-TLX [7] 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.

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.

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

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 on Session 2 and Session 3 are 19.41% and 50.82%. This was made possible by the accumulation of learning time and the participants' increasing proficiency in using CrossKeys for text entry. Also, the increase implies that our text entry method still has potential to reach higher WPM. In Session 3, the mean WPM has reached 17.73, and one of the participants has reached 24.73 after typing one set of phrases. CrossKeys' WPM is generally faster than other selection based entry methods(for example, HiPad [9]).

The ANOVA doesn't show any significant effect of Session 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 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.

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		-
	2	14.04 ± 1.19	0.000*	+19.41%
	3	17.73 ± 1.5		+50.82%
TER	1	$7.52\% \pm 5.16\%$		-
	2	$7.90\% \pm 9.91\%$	0.970	+4.98%
	3	$7.49\% \pm 12.11\%$		−0.49%
NCER	1	$0.64\% \pm 0.46\%$		-
	2	$0.44\% \pm 0.31\%$	0.053	−31.24%
	3	$0.30\% \pm 0.36\%$		−52.90%

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

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

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.

CC The participants would enter phrases randomly generated from the MacKenzie Phrase Set for two consecutive minutes.

EC1 The participants would enter phrases generated the same way as CC, for six consecutive minutes.

EC2 The participants would enter phrases generated the same way as CC, for ten consecutive minutes.

Procedure The entire experiment was conducted using a within-subject design. The participants had 45 minutes to learn about the input method and try to enter phrases as practice. Every time they finished one test of three, they would need to finish the SSQ and NASA-TLX test for their current feelings. Participants may rest for any length of time between tests, so that the interval between each experiment was sufficient to assure that participants could avoid motion sickness from growing worse with the length of time using the VR device, which in turn would affect the results of the experiment.

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

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 of the data was probably caused by a certain degree of fatigue of the participants, but the ANOVA indicates that such 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 increasing TER under the three conditions. The first one is similar to the explanation in Task 1, 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 *Fig*. 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 on Table 4. 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 on *Table* 6. 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.

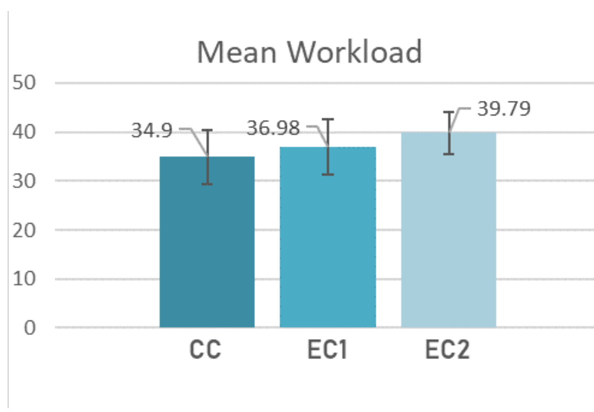


Fig. 7. The average workload of three status(short, longer and prolonged). Error bars indicate ± 1 standard deviation.

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 *Fig*.7 we also

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		-

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

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 with their bodies so that they would prevent slipping out of the treadmill or falling down. We designed two conditions as follows.

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

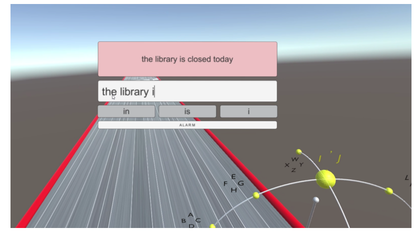
EC The participants enter ten phrases generated the same way as CC, only under this condition they will be walking on power-on treadmill of a daily walking speed (1.6 m/s).

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 until they are relaxed and ready. 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. 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.

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.

Results and Discussion *Table* 7 shows the mean WPM, TER and NCER in task 3. The T-tests shows 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.

(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.

Fig. 8. 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	$7.80\% \pm 7.91\%$	0.894	0.02
	EC	$7.64\% \pm 7.15\%$		
NCER	CC	$1.03\% \pm 3.93\%$	0.546	0.14
	EC	$1.29\% \pm 3.12\%$		

Fig. 9 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 the workload is mainly from motion, not our input method.

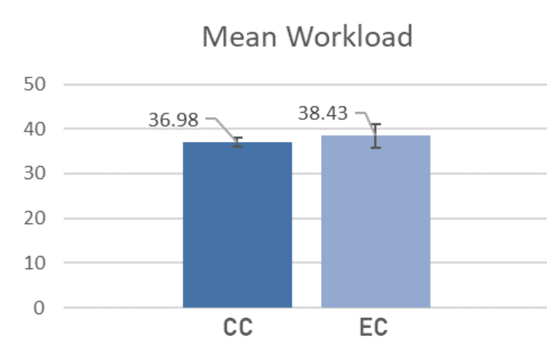


Fig. 9. The average workload of two status (Staitionary and In-motion). Error bars indicate ± 1 standard deviation.

6 Conclusion

We introduce CrossKeys, a novel, efficient, and handy text entry technique for VR using a single controller via wrist rotation. To perform better, apart from the support of auto-completion prediction, we design a cross-like layout to minimize the average distance of spatial displacement when selecting characters. CrossKeys breaks the boundary set by two-dimensional mapping of characters and innovatively employs the three-dimensional space a virtual environment provides, with which text entry could perform much better. We refine our design with two pilot studies focusing on keyboard layout and the function of highlighting. Afterward, during the user study, we evaluate our CrossKeys based mainly on two criteria: efficiency, task load, and three tasks: learnability and performance test, fatigue test, and evaluation of in-motion performance.

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 unprecedentedly make text entry in VR possible when in moderate motion like walking, 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, considering for a number of people, the current alphabetic layout is still not so easy to remember and learn.

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 most right; likewise, for users whose dominant hand in on their left, they also feel the same discomfort when reaching for the blocks behind and on the most left.

3. The current model of CrossKeys is based on a rationale that every two adjacent character blocks has the same distance. Honestly speaking, the notion is quite plain and simple and based only on five lab members' trial experience, so there could be a better solution for our CrossKeys to the arrange of character blocks that makes them not necessarily have to be equally distanced.

In conclusion, CrossKeys is a text entry method for VR based on wrist rotation and only a single controller is needed to fully manipulate, making interactions with a decent low-load user experience possible while typing. Even with limitations, the promising future of CrossKeys can hardly be overshadowed. CrossKeys is efficient and portable enough to be utilized in various scenarios in virtual environments.

References

[1] S. Chen, J. Wang, S. Guerra, N. Mittal, and S. Prakkamakul. Exploring word-gesture text entry techniques in virtual reality. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI EA '19, p. 1–6. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3290607.3312762

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

[3] S. Feiz and I. V. Ramakrishnan. Exploring feasibility of wrist gestures for non-visual interactions with wearables. In *Proceedings of the 16th International Web for All Conference*, W4A '19. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3315002.3317570

[4] J. Gong, Z. Xu, Q. Guo, T. Seyed, X. A. Chen, X. Bi, and X.-D. Yang. *WrisText: One-Handed Text Entry on Smartwatch Using Wrist Gestures*, p. 1–14. Association for Computing Machinery, New York, NY, USA, 2018.

[5] Y. Gu, C. Yu, Z. Li, Z. Li, X. Wei, and Y. Shi. Qwertyring: Text entry on physical surfaces using a ring. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 4(4), dec 2020. doi: 10.1145/3432204

[6] X. Han, C. Gao, and Y. Yu. Deepsketch2face: A deep learning based sketching system for 3d face and caricature modeling. *ACM Trans. Graph.*, 36(4), jul 2017.

[7] S. G. Hart. Nasa-task load index (nasa-tlx); 20 years later. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 50:904 – 908, 2006.

[8] Y. Ikei, V. Yem, K. Tashiro, T. Fujie, T. Amemiya, and M. Kitazaki. 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, 2019. doi: 10.1109/VR.2019.8797876

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

[10] Y. Jiang, C. Zhang, H. Fu, A. Cannavò, F. Lamberti, H. Y. K. Lau, and W. Wang. *HandPainter - 3D Sketching in VR with Hand-Based Physical Proxy*. Association for Computing Machinery, New York, NY, USA, 2021.

[11] D. Johanna, A. Mathieu, I. Phillip, A. A. Efros, and B. Adrien. 3d sketching using multi-view deep volumetric prediction. *Proceedings of the Acm on Computer Graphics Interactive Techniques*, 1(1):1–22, 2018.

[12] R. S. Kennedy, N. E. Lane, K. S. Berbaum, and M. G. Lilienthal. Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The International Journal of Aviation Psychology*, 3(3):203–220, 1993.

[13] H. Kim, L. Remaggi, P. J. Jackson, and A. Hilton. 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)*, pp. 120–126, 2019. doi: 10.1109/VR.2019.8798247

[14] A. Kurauchi, W. Feng, A. Joshi, C. Morimoto, and M. Betke. Eyeswipe: Dwell-free text entry using gaze paths. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, p. 1952–1956. Association for Computing Machinery, New York, NY, USA, 2016. doi: 10.1145/2858036.2858335

[15] K. C. Kwan and H. Fu. *Mobi3DSketch: 3D Sketching in Mobile AR*, p. 1–11. Association for Computing Machinery, New York, NY, USA, 2019.

[16] D. Lee, J. Kim, and I. Oakley. 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*, CHI '21. Association for Computing Machinery, New York, NY, USA, 2021. doi: 10.1145/3411764.3445106

[17] X. Lu, D. Yu, H.-N. Liang, X. Feng, and W. Xu. 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, 2019. doi: 10.1109/VR.2019.8797901

[18] Z. Luo, J. Zhou, H. Zhu, D. Du, X. Han, and H. Fu. 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*, UIST '21, p. 854–863. Association for Computing Machinery, New York, NY, USA, 2021. doi: 10.1145/3472749.3474791

[19] I. S. MacKenzie and R. W. Soukoreff. Phrase sets for evaluating text entry techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '03, p. 754–755. Association for Computing Machinery, New York, NY, USA, 2003. doi: 10.1145/765891.765971

[20] A. Markussen, M. R. Jakobsen, and K. Hornbæk. Vulture: A mid-air word-gesture keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, p. 1073–1082. Association for Computing Machinery, New York, NY, USA, 2014. doi: 10.1145/2556288.2556964

[21] D. Pedrosa, M. D. G. Pimentel, A. Wright, and K. N. Truong. Filteryedping: Design challenges and user performance of dwell-free eye typing. *ACM Trans. Access. Comput.*, 6(1), mar 2015. doi: 10.1145/2724728

[22] P. Quinn and S. Zhai. A cost-benefit study of text entry suggestion interaction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, p. 83–88. Association for Computing Machinery, New York, NY, USA, 2016. doi: 10.1145/2858036.2858305

[23] M. Sormann, J. Bauer, C. Zach, A. Klaus, and K. Karner. Vr modeler: From image sequences to 3d models. In *Proceedings of the 20th Spring Conference on Computer Graphics*, SCCG '04, p. 148–156. Association

for Computing Machinery, New York, NY, USA, 2004. doi: 10.1145/1037210.1037233

[24] R. W. Soukoreff and I. S. MacKenzie. Metrics for text entry research: An evaluation of msd and kspc, and a new unified error metric. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’03, p. 113–120. Association for Computing Machinery, New York, NY, USA, 2003.

[25] P. Strohmeier, R. Vertegaal, and A. Girouard. 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*, TEI ’12, p. 307–308. Association for Computing Machinery, New York, NY, USA, 2012. doi: 10.1145/2148131.2148195

[26] T. M. Takala, C. C. Hsin, and T. Kawai. 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. 1398–1398, 2019. doi: 10.1109/VR.2019.8798169

[27] D. Valkov and L. Linsen. 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, 2019. doi: 10.1109/VR.2019.8798036

[28] P. C. Wong, K. Zhu, and H. Fu. *FingerT9: Leveraging Thumb-to-Finger Interaction for Same-Side-Hand Text Entry on Smartwatches*, p. 1–10. Association for Computing Machinery, New York, NY, USA, 2018.

[29] W. Xu, H.-N. Liang, Y. Zhao, T. Zhang, D. Yu, and D. Monteiro. Ringtext: Dwell-free and hands-free text entry for mobile head-mounted displays using head motions. *IEEE Transactions on Visualization and Computer Graphics*, 25(5):1991–2001, 2019. doi: 10.1109/TVCG.2019.2898736

[30] N. Yanagihara and B. Shizuki. Cubic keyboard for virtual reality. In *Proceedings of the Symposium on Spatial User Interaction*, SUI ’18, p. 170. Association for Computing Machinery, New York, NY, USA, 2018. doi: 10.1145/3267782.3274687

[31] H.-S. Yeo, J. Lee, H.-i. Kim, A. Gupta, A. Bianchi, D. Vogel, H. Koike, W. Woo, and A. Quigley. 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*, MobileHCI ’19. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3338286.3340130

[32] X. Yi, C. Yu, M. Zhang, S. Gao, K. Sun, and Y. Shi. Atk: Enabling ten-finger freehand typing in air based on 3d hand tracking data. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software amp; Technology*, UIST ’15, p. 539–548. Association for Computing Machinery, New York, NY, USA, 2015. doi: 10.1145/2807442.2807504

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