Design and Evaluation of Controller-Based Raycasting Methods for Efficient Alphanumeric and Special Character Entry in Virtual Reality

Tingjie Wan, Yushi Wei, *Student Member, IEEE*, Rongkai Shi, *Student Member, IEEE*, Junxiao Shen,
Per Ola Kristensson, Katie Atkinson, and Hai-Ning Liang, *Member, IEEE*

***Abstract*—Alphanumeric and special characters are essential during text entry. Text entry in virtual reality (VR) is usually performed on a virtual Qwerty keyboard to minimize the need to learn new layouts. As such, entering capitals, symbols, and numbers in VR is often a direct migration from a physical/touchscreen Qwerty keyboard—that is, using the mode-switching keys to switch between different types of characters and symbols. However, there are inherent differences between a keyboard in VR and a physical/touchscreen keyboard, and as such, a direct adaptation of mode-switching via switch keys may not be suitable for VR. The high flexibility afforded by VR opens up more possibilities for entering alphanumeric and special characters using the Qwerty layout. In this work, we designed two controller-based raycasting text entry methods for alphanumeric and special characters input (*Layer-ButtonSwitch* and *Key-ButtonSwitch*) and compared them with two other methods (*Standard Qwerty Keyboard* and *Layer-PointSwitch*) that were derived from physical and soft Qwerty keyboards. We explored the performance and user preference of these four methods via two user studies (one short-term and one prolonged use), where participants were instructed to input text containing alphanumeric and special characters. Our results show that *Layer-ButtonSwitch* led to the highest statistically significant performance, followed by *Key-ButtonSwitch* and *Standard Qwerty Keyboard*, while *Layer-PointSwitch* had the slowest speed. With continuous practice, participants' performance using *Key-ButtonSwitch* reached that of *Layer-ButtonSwitch*. Further, the results show that the key-level layout used in *Key-ButtonSwitch* led users to parallel mode switching and character input operations because this layout showed all characters on one layer. We distill three recommendations from the results that can help guide the design of text entry techniques for alphanumeric and special characters in VR.**

***Index Terms*—Virtual reality, text entry, keyboard layout, alphanumeric and special character entry, mode-switching, user study.**

I. INTRODUCTION

TEXT entry is indispensable in all interactive systems, including desktops, mobile devices, and virtual/augmented reality head-mounted displays (VR/AR HMDs). Daily text entry tasks, such as document editing, composing emails, and sending instant messages, commonly involve alphanumeric and special characters, including lowercase letters, uppercase letters, numbers, and symbols [14], [31], [51]. The use of different types of characters can significantly impact the meanings of words and improve the readability of the content. For instance, 'march' means to go forward as a verb, while 'March' means the third month of the year, which is also replaceable with the number '3' in the date formatting. In addition, using alphanumeric and special characters can also enable users to enter emojis [9] and passwords with different character combinations [32]. To allow access to different character types, a typical keyboard integrates two character types in the same key or has separate layers for different types of characters. Users need to use the *mode-switch keys* to switch to the target mode (or layer) and then input the desired character. For example, 'Shift' and 'Caps Lock' are two mode-switch keys in a standard Qwerty keyboard for a desktop setup, while '⇑' and '123' keys can switch the layers in a soft Qwerty keyboard for mobile devices (see Fig. 2(a) and (b)). Just like tapping on character keys, users tap on these mode-switch keys on a physical keyboard or a touchscreen.

Current commercial VR HMDs typically do not have a physical keyboard or a touchscreen for text entry tasks; instead, they normally provide a virtual keyboard with a Qwerty layout and ask users to interact with it via handheld controllers (see Fig. 2(c) and (d)). In this setup, users control a ray emitted from the controller (i.e., raycasting) to point to the target key and press a trigger button to confirm the selection [1]. A growing number of studies have designed and developed new virtual keyboard layouts to improve text entry performance and experience in

---

Manuscript received 14 August 2023; revised 8 December 2023; accepted 19 December 2023. Date of publication 3 January 2024; date of current version 31 July 2024. This work was supported in part by Suzhou Municipal Key Laboratory for Intelligent Virtual Engineering under Grant SZS2022004, in part by the National Natural Science Foundation of China under Grant 62272396, and in part by Xi'an Jiaotong-Liverpool University under Grant RDF-17-01-54. Recommended for acceptance by M. Marchal. *(Corresponding author: Hai-Ning Liang.)*

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by University Ethics Committee at Xi'an Jiaotong-Liverpool University under Application No. ER-SAT-12889407202211221 60449.

Tingjie Wan, Yushi Wei, and Rongkai Shi are with the School of Advanced Technology, Xi'an Jiaotong-Liverpool University, Suzhou 215000, China, and also with the Department of Computer Science, University of Liverpool, L69 3BX Liverpool, U.K. (e-mail: tingjie.wan20@student.xjtlu.edu.cn; yushi.wei21@student.xjtlu.edu.cn; rongkai.shi19@student.xjtlu.edu.cn).

Junxiao Shen and Per Ola Kristensson are with the Department of Engineering, University of Cambridge, CB2 1TN Cambridge, U.K. (e-mail: js2283@cam.ac.uk; pok21@cam.ac.uk).

Katie Atkinson is with the Department of Computer Science, University of Liverpool, L69 3BX Liverpool, U.K. (e-mail: katie@liverpool.ac.uk).

Hai-Ning Liang is with the Department of Computing, School of Advanced Technology, Xi'an Jiaotong-Liverpool University, Suzhou 215000, China (e-mail: haining.liang@xjtlu.edu.cn).

Digital Object Identifier 10.1109/TVCG.2024.3349428

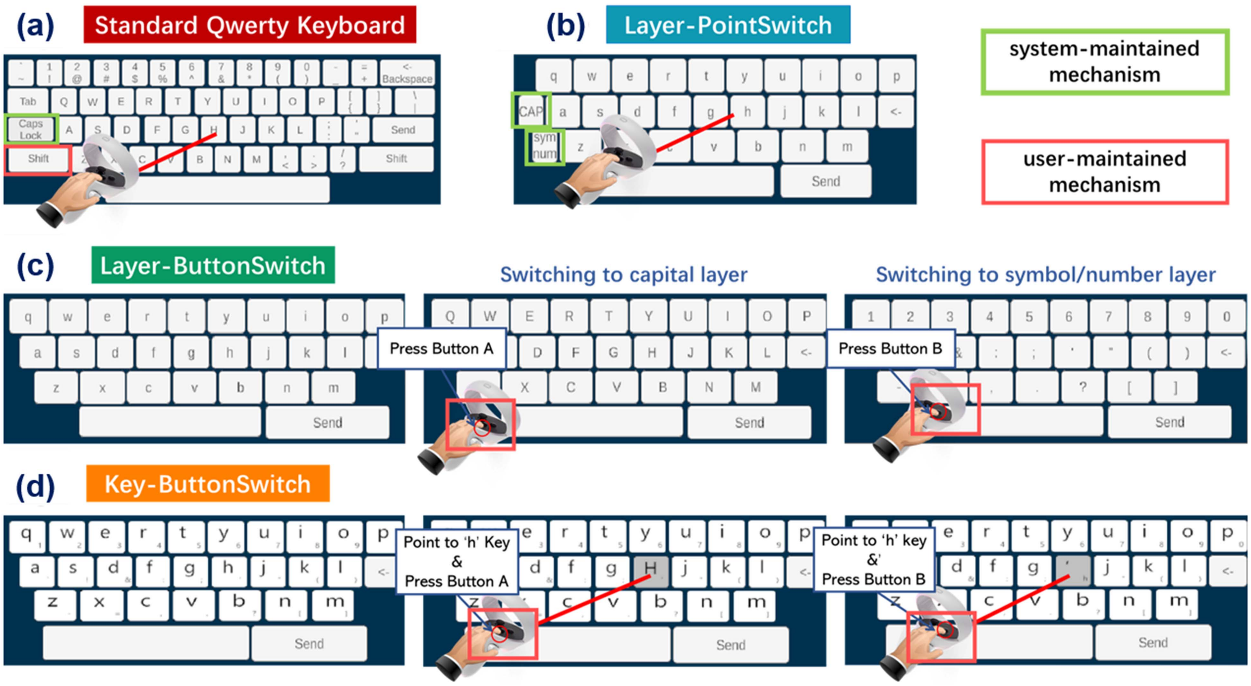


Fig. 1. Four controller-based raycasting text entry methods for alphanumeric and special characters. (a) Standard Qwerty Keyboard (STD, red), (b) Layer-PointSwitch (LPS, blue), (c) Layer-ButtonSwitch (LBS, green), and (d) Key-ButtonSwitch (KBS, orange).

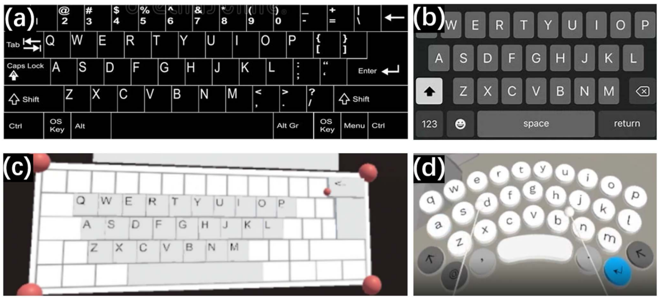


Fig. 2. (a) A standard Qwerty layout used in physical keyboards. (b) A soft keyboard layout used in touchscreen devices. (c) A virtual keyboard with a standard Qwerty layout in VR [50]. (d) A virtual keyboard with a soft keyboard layout in VR [4]. In (c) and (d), the controllers are used to point to the target key and make a confirmation.

VR HMDs. Examples include a circular layout [22], [59], [64], but such techniques are not widely adopted because they require users to learn the new layout and typing approach. Researchers have also investigated other interaction methods for text entry in VR, such as tapping on the virtual keys via controllers [4], [50], typing with freehand mid-air gestures [25], [50], and head pointing [29], [50], [59], [63]. However, most of these explorations only focus on text entry with lowercase letters. In general, text entry involving alphanumeric and special characters in VR HMDs is still understudied. Unlike the relatively constrained behavior (i.e., tapping) and fixed visual/interactive elements of physical and touchscreen keyboards, VR affords more design opportunities to support efficient text entry for alphanumeric and special characters.

In this work, we aim to explore efficient controller-based raycasting text entry methods for alphanumeric and special characters in VR HMDs. We focused on text entry methods based on the most widely adopted Qwerty virtual keyboard with controller-based raycasting input to ensure fast learnability and wider applicability. We first identified three design considerations for inputting different types of characters in VR HMDs quickly based on seamless transitions across character types. We then designed two text entry methods (*Layer-ButtonSwitch* and *Key-ButtonSwitch*) that utilized controller buttons for switching between modes. In our first study, we compared our two techniques with two other methods (*Standard Qwerty Keyboard* and *Layer-PointSwitch*), which were derived from common pointing-based approaches for both character selection and mode switching. While Study 1 focused on the short-term use of these four methods, Study 2 explored the longer-term, more prolonged use of our two proposed methods. Results from Study 1 indicated that our proposed methods (Layer-ButtonSwitch and Key-ButtonSwitch) led to statistically better user performance than the other two more common methods. Layer-ButtonSwtich had the best performance, and participants favored its use with both hands. On the other hand, Key-ButtonSwitch was preferred to be operated with one hand. Results from Study 2 showed that both Layer-ButtonSwitch and Key-ButtonSwitch were easy to learn, with users only needing about 1.25 hours of practice to become proficient.

In summary, this paper presents three main contributions:

1. We introduce two text entry methods for efficient alphanumeric and special character entry in VR, allowing for seamless transitions between the different character types.

2. We perform a formal evaluation of two generations of Qwerty keyboards (standard and soft Qwerty keyboards) and two new proposed text entry methods to examine their performance and user preference.
3. We provide three recommendations for alphanumeric and special character text entry using controller-based raycasting in VR.

II. RELATED WORK

Text entry in VR systems, an essential input task, has been receiving increasing attention as VR is becoming more widespread and aims to be integrated into people's daily activities. Text entry can be divided into two actions: character selection and mode switching. Mode switching is important when users need to enter different types of characters.

Current research on text entry in VR has primarily focused on character selection, particularly exploring and examining input approaches to improve efficiency and accuracy. One main direction is to connect an external input device to a VR HMD, such as using a physical keyboard or a touchscreen. In particular, this line of research has studied keyboard visualization [16], [37], [41], hand tracking [26], hand representation [15], and typing methods [7], [20]. However, having an additional device that is not part of the ecosystem not only adds additional cost and forces users to carry this device with them but also poses potential compatibility issues and security risks. Setting up and configuring devices may also require additional time and effort and may not seamlessly integrate with existing VR systems, resulting in potential workflow disruptions and reduced efficiency. A growing body of research has explored typing on a virtual keyboard in VR with different input modalities, including handheld controllers [4], [7], [22], [50], [61], [62], users' hands [10], [16], [17], [35], voice [38], head [30], [59], eyes [40], and electroencephalogram [33]. Some of these studies have explored the use of non-conventional layouts, for example, a circular layout with distinctive key mappings [22], [59], [64]. However, the Qwerty keyboard layout is still the most frequently used layout. This is because (1) users have familiarity with the Qwerty layout; (2) users tend not to prefer to learn a new layout; and (3) typing speeds using the Qwerty layout are typically satisfactory [27], [28]. Currently, researchers have adapted two typical Qwerty layouts from the physical world to virtual environments—a standard Qwerty keyboard layout [25], [50] and a soft keyboard layout [4], [5], [7], [58], [61], [62], as shown in Fig. 2(c) and (d), respectively.

Mode-switching is necessary for typing alphabetic, numerical, and special characters as mode-switching accommodates more characters than the number of keys available on a keyboard [56]. In general, there are two mode-switching mechanisms: *system-maintained* and *user-maintained* [24], [43], [45]. A system-maintained mechanism would establish a mode where users transition to another mode for the intended action, such as using the 'Caps Lock' key to type uppercase letters in a standard physical keyboard. In contrast, a user-maintained (also known as quasi- or spring-loaded) mechanism requires users to activate and maintain the mode kinesthetically while performing the intended action, such as pressing and holding the 'Shift' or 'Ctrl' keys for different functions. A standard physical keyboard uses a hybrid of both. A soft keyboard in touchscreen devices, which typically have a limited screen size, moves the numbers and symbols to other layers; thus, a system-maintained mechanism is commonly adopted when switching from one type to another.

VR environments afford more possibilities for new modalities and flexible interfaces. Prior research has explored different approaches for mode switching in virtual environments. When hand gestures are the main input approach, mode-switching using non-dominant hand gestures has been studied (e.g., [36], [46], [52]). Mode-switching by the non-dominant hand typically uses a user-maintained mechanism. It is an asymmetrical two-handed task in which each hand has a different role—the non-dominant hand controls and maintains the mode of the interface, while the dominant hand performs the intended actions, which often require more precision or are more demanding. This approach is limited to scenarios where both hands are performing the task, such as typing with two hands or where either hand is occupied. To cope with this problem, Shi et al. [45] investigated using user-maintained head movements for mode-switching. However, 3D interaction is already physically demanding, and introducing an extra control would make using the tool more complex [1]. Smith et al. [47] evaluated five mode-switching techniques for AR headsets: hardware button, virtual button, non-preferred hand, reach depth, and voice. Their results showed that utilizing the hardware button from the headset, though it might be limited in the number of supported modes, was precise and fast. The buttons on the controller are enough and suitable to support the mode-switching requirements, so we considered this approach in our work. As the system-maintained mode-switching mechanism [43] requires the user to repeatedly click the mode-switching key(s), porting it to the hardware buttons out of VR view may force the user to be distracted by the buttons instead of concentrating on typing in VR. As such, this mechanism is not so suitable for binding it to hardware buttons. The hardware buttons on the controller allow the transposition of user-maintained mode-switching from the keyboard, as users could return to the default mode by simply releasing the button rather than repetitive clicks, which is both fast and simple.

Currently, most VR text entry studies have primarily involved case-insensitive letters because they are simple and easy to use when exploring newly proposed typing methods. For instance, Speicher et al. [50] adapted the standard Qwerty keyboard layout in VR, but only enabled the input for letters for experimental purposes (see Fig. 2(c)). To input different types of characters, including lowercase/uppercase letters, symbols, and numbers, most virtual keyboards use system-maintained mode-switching (e.g., [4], [56], [61]), as shown in Fig. 2(d). Song et al. [48] proposed a gesture-based mode-switching method for freehand text entry on a virtual keyboard with a soft keyboard layout. Users tapped on the keyboard using their index finger for character selection while maintaining a gesture with wrist rotation or multiple fingers to access different keyboard layers. This method significantly improved the speed for typing alphanumeric and special characters for freehand text entry scenarios.

Our review of the literature has pointed to limited research on the design and evaluation of methods that can support efficient and seamless text entry of alphanumeric and special characters in VR. Instead of focusing on new, non-conventional layouts and input devices, our work is based on the Qwerty keyboard and controller-based pointing because we aim to leverage users' existing typing practices and use the most widely adopted interaction mechanism to support wider applicability of the text entry methods derived from our work.

III. TEXT ENTRY METHODS FOR ALPHANUMERICAL AND SPECIAL CHARACTERS

A. Design Considerations

In this section, we first describe three design considerations that an alphanumerical and special character entry method should meet to be usable and practical.

- *Applicability:* Most current commercial VR devices use handheld controllers for text input [4] and other interactions—examples include the HTC VIVE Pro 2/Pro Eye, Meta Quest 2/3, or PICO 4. Thus, we have focused on controller-based approaches. The text entry methods should be applicable to most commercial VR HMDs, which predominantly come with some type of handheld controller.
- *Learnability:* The keyboard layout that users are most familiar with is the Qwerty layout. Many layouts deviating from the Qwerty layout are not suggested due to their high learning costs and relatively low performance [21]. Therefore, we have only included the standard and soft keyboard layouts from PC and post-PC eras that most users are familiar with. In addition, we only made minor, yet meaningful, changes to the layout to enable fast and seamless transitions between different keyboard layers (or character types).
- *Efficiency and Accuracy:* Raycasting is the most commonly used pointing-based selection technique in VR, and it meets the de-facto standard for text entry in VR with an acceptable entry rate that benefits from a low error rate (over 15 words per minute and around 1% error rate according to prior work [50], [61], [62]). Pointing-based requires a smaller motor space than direct touch on the keys via virtual hands. Thus, we have chosen controller-based raycasting as the character selection approach—users can control the ray emitted from the controller to point to a target key and press the trigger button to confirm the selection.

B. Text Entry Methods

Based on the above considerations, we included four controller-based raycasting text entry methods for alphanumerical and special characters in VR HMDs. They include two pointing-switching methods *Standard Qwerty Keyboard (STD)* and *Layer-PointSwitch (LPS)*, and two button-switch methods *Layer-ButtonSwitch (LBS)* and *Key-ButtonSwitch (KBS)*, as shown in Fig. 1.

1) Standard Qwerty Keyboard (STD): STD is a virtual replica in VR HMD of a standard physical keyboard. It consists of five rows, including three rows for letters, a bottom row for the space key, and a top row for numbers and symbols. Similar to a physical keyboard, users can input with both hands—both controllers emit rays (in red), and the rays are always present simultaneously. By default, users can input lowercase letters and numbers. Users can press and hold the 'Shift' key to switch to another layer for uppercase letters and symbols. Besides, users can also toggle the 'Caps Lock' key to switch in or out the input mode for capital letters.

2) Layer-PointSwitch (LPS): LPS is adapted from soft keyboards and maintains the same arrangement of the character keys to reduce any learning needed. It has three layers—lowercase alphabetical characters, uppercase alphabetical characters, and numbers and symbols. All use a 4-row layout, with the top three rows for the character keys and the bottom row for the space key. The layout for the number and symbol layers is also adapted from a common soft keyboard. The transition between the layers was toggled via mode-switching keys ('CAP' and 'sym/num' keys). The mode-switching mechanism is system-maintained, which is consistent with the soft keyboard.

3) Layer-ButtonSwitch (LBS): Overall, LBS is similar to LPS except for its mode-switching method. In LBS, users no longer select the switch keys on the virtual keyboard; instead, they can press and hold the buttons on the controller for the target layer. The buttons on both controllers can be used for mode-switching. Pressing and holding Button 'A' on the right controller or Button 'X' on the left controller can transition to the uppercase layer. Similarly, Button 'B' on the right controller and Button 'Y' on the left controller are used for the symbol/number layer. Releasing the button switches the keyboard back to the default lowercase layer.

This design was chosen for two reasons. First, mode-switching in a soft keyboard layout requires only a limited number of toggles (usually two), which is available and supported on most commercial VR devices' controllers. Second, it may lead to a smaller motor space because it saves hand movement for locating the switch keys on the virtual keyboard. One potential drawback is that users may press the wrong button for different transitions before they become familiar with LBS. To mitigate this negative effect, we use the user-maintained mechanism for mode-switching. When users make a mistake, they can quickly release the current button and move to another one.

4) Key-ButtonSwitch (KBS): We further modified LBS and proposed KBS. KBS also leverages the user-maintained mode-switching mechanism via buttons. However, it only switches the mode for a single key rather than the entire keyboard layer. Each key on the keyboard contains three potential characters: a lowercase letter, its capital form, and a number or a symbol. When a key is being pointed at, Button 'A' or 'X' switches between the letter case, while Button 'B' or 'Y' switches between the letter and number/symbol. Visual cues are added to make KBS easy to learn and use. The character under the current mode would be displayed at the center of the key in regular size. Meanwhile, a candidate character is placed at the bottom right of the key. A candidate numerical character is displayed when the key is in alphabetical character mode, i.e., either a lowercase or uppercase letter is shown at the center of the key. In contrast, a candidate lowercase character is displayed when the current mode is for a numerical character. Uppercase letters are not shown as candidate characters because users can recall the capital letters from their lowercase counterparts [23], [53].

The rationale of its design is that users may only seek one character from the switched mode, and after selecting this character, they have to return back to the previous mode for the next input. Users may temporarily lose the location information of the key in another mode and require effort and time to search for that key, especially in VR environments where users no longer rely on typing muscle memory from their prior experience with physical keyboards. With KBS, all characters can be seen, regardless of the type of character the user is typing. This helps omit the step where the user may need to switch modes before locating characters.

IV. STUDY 1: COMPARATIVE EVALUATION

The goal of this study is to compare and evaluate the user performance and experience of the four text entry methods for alphabetical, numerical, and special characters.

A. *Participants and Apparatus*

A total of 24 participants were recruited (11 females, 13 males; aged 20 to 40, $M = 23.65, SD = 4.10$) from a local university. The number of participants was determined by power analysis using G $*$ Power [11] to ensure a power of 0.95 and an α of 0.05 for statistical analyses. All participants were non-native English speakers but were familiar with English as it was the language of instruction at the university. All participants had normal or corrected-to-normal vision and were right-handed. All reported being familiar with the Qwerty keyboard (either the standard physical keyboard layout or the soft keyboard layout) and would use it every day. Six participants were frequent VR users who used a VR HMD more than once per month, while the remaining only had little or no experience with VR HMDs.

We used a Meta Quest 2 to provide the experimental environment. It has a fast-switch LCD display, a resolution of 1832 × 1920 px per eye, and a 90 Hz refresh rate. It was connected to a Windows 10 PC with an Intel i7-7700 k CPU and an Nvidia GeForce GTX 1080 GPU. The techniques and virtual environment were implemented using Unity3D (v2021.3.1f1) with the XR interaction Toolkit (1.0.0-pre.8) and Oculus XR Plugin (1.8.1) packages. Participants completed all tasks in a sitting position.

B. *Materials*

As shown in Fig. 3, all four keyboards were positioned 2.19 m in front of users. The size of the STD was 2.78 m × 0.9 m, while the sizes for the remaining three text entry methods were 2.08 m × 0.7 m. All character keys had the same size (0.185 m × 0.148 m) regardless of the keyboards. All keyboards were tilted 15° back to minimize workload [54].

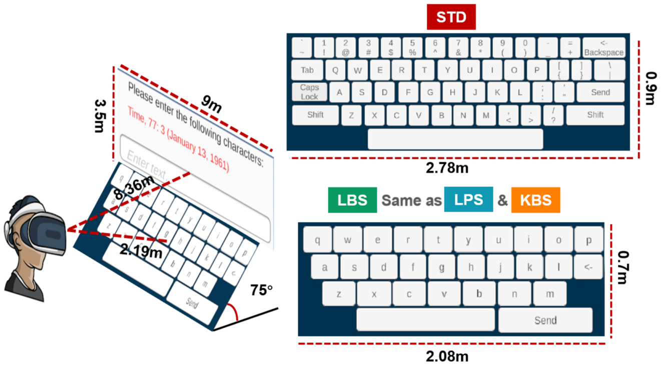


Fig. 3. Keyboard setting of the four text entry methods in the VR environment.

To involve alphabetical, numerical, and special characters in the text entry tasks, we randomly selected sentences from the Brown Corpus [12]. These sentences contained a large number of uppercase letters, numbers, and symbols (e.g., “Time, 77: 3 (January 13, 1961)”). As such, entering these sentences requires frequent mode switches. The sentences and the input box were displayed on a 9 m × 3.5 m area on the top of the keyboard, which is 8.36 m in front of the user (see Fig. 3). Once participants completed the current sentence, they needed to press the ‘Send’ key to continue to the next sentence.

C. *Experiment Design and Procedure*

This study used a within-subjects design with TEXT ENTRY METHOD as the independent variable comprising four conditions (STD, LPS, LBS, and KBS). The order of TEXT ENTRY METHOD conditions was counterbalanced via a Latin-Square approach. For each condition, 15 sentences needed to be transcribed. The sentences in each condition were randomly selected from the Brown corpus with no duplicates. The first five sentences were for training, and participants’ performance for these five was not logged. The following ten sentences were formal trials and were recorded. This led to a total of 960 trials used for further analyses (= 24 participants × 4 text entry methods × 10 sentences).

At the beginning of the experiment, participants first filled out a consent form and a demographics questionnaire. Next, they were introduced to the VR device, task, techniques, and controls. We then asked participants to wear the headset and start the experiment. The experiment consisted of four sessions corresponding to the four text entry methods. Participants were instructed to complete the text entry task as fast and as accurately as possible. After each session, we gave participants post-task questionnaires to gather their subjective feelings about the just-used method. A five-minute break was given between two sessions, but more time would be given if requested. After completing all sessions, participants were asked to rank the four methods according to their overall preference and provide the reasons and further verbal comments, if any. The whole experiment lasted for approximately 60 minutes.

D. Evaluation Metrics

We measured task performance using the objective data recorded during the experiments. Additionally, we collected different aspects of subjective feedback using three questionnaires.

- *Entry Rate* was measured in Words Per Minute (WPM) [60], which was the number of words transcribed divided by the time taken to transcribe the text (in minutes). In this work, a word was defined as any type of five characters, including lowercase and uppercase letters, numbers, symbols, and spaces.
- *Layer Switching Interkey Interval* (Layer Switching IKI) [48] was the time between successive key selections to switch from one layer to another.
- *Error Rate* [49] was calculated based on the standard character-level typing metrics, where the total error rate (TER) = not corrected error rate (NCER) + corrected error rate (CER).
- *Workload* for completing the task with the given text entry method was measured via a NASA-TLX workload questionnaire [19]. It comprises six subscales for six clusters: mental, physical, temporal demand, frustration, effort, and performance. The subscales were assessed on a 0 to 100 scale with interval increments of 5 (the lower, the better).
- *Usability* of each text entry method was measured via a System Usability Scale (SUS) questionnaire [6]. It consists of 10 questions rated on a 5-point scale. The weighted overall score was used for analysis (ranging from 0 to 100; the higher, the better).
- *Ranking* was a rank of all text entry methods based on participants' overall preferences. It was completed after participants had experienced all four methods. We also interviewed participants about the reasons for their rankings.

E. Hypotheses

We tested four hypotheses in this user study:

- *H1:* The entry rate and layer switching IKI of the two button-switch methods (LBS and KBS) would be faster than that of the two pointing-switch methods (STD and LPS) for inputting alphanumerical and special characters since a button-switch method did not need a pointing action.
- *H2.* No significant difference in entry accuracy (i.e., NCER) would be found among the four text entry methods because they had the same pointing-based character selection mechanism.
- *H3:* The two button-switch methods would lead to a lower workload than the pointing-switch methods because pressing the controller button would only require a small, simple, and fast finger operation without arm movement.
- *H4:* The two button-switch methods would have higher usability than the other two methods due to the improved speed and reduced workload.

F. Results

We used IBM SPSS 26 [13] for data analysis. Before the statistical analysis, we identified 13 sentences (1.35% of the 960 sentences) that participants could not complete and removed them from the analysis. Shapiro-Wilk test showed that entry rate and NASA-TLX data were normally distributed ($p > .05$), which was also confirmed by the Q-Q plots. On the other hand, the TER, NCER, layer switching IKI, and SUS data were not normally distributed ($p < .05$). For the entry rate data, we applied repeated-measures (RM-) ANOVAs. If the assumption of sphericity was violated in RM-ANOVAs, we reported the degrees of freedom with Greenhouse-Geisser correction (when $\epsilon < .75$) or Huynh-Feldt correction ($\epsilon > .75$). As there are six dimensions in NASA-TLX data, we used Multivariate ANOVA (MANOVA) to compare the differences. Besides, we used Friedman tests for layer switching IKI, TER, NCER, SUS, and ranking data. Effect sizes were reported using partial eta squared (η_p^2) for ANOVA tests and Kendall's W for Friedman tests. If significant differences were found, Bonferroni-adjusted pairwise comparisons were conducted.

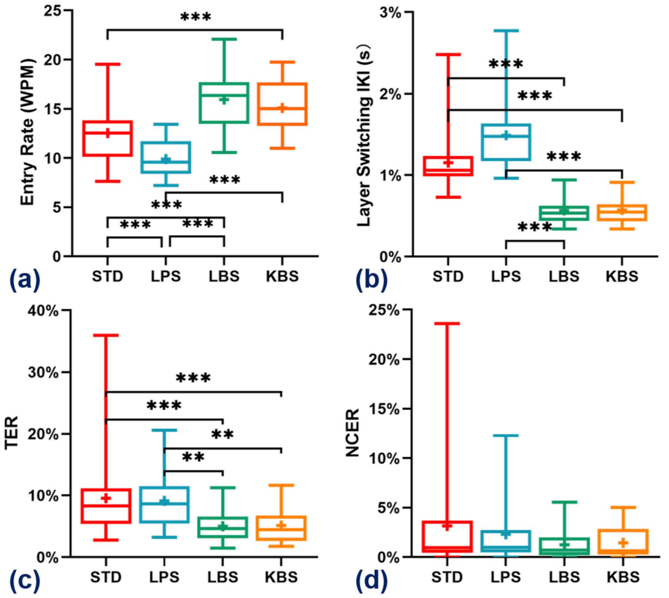


Fig. 4. Boxplots of (a) entry rate, (b) layer switching IKI, (c) TER and (d) NCER. The plus icon (+) represents the mean of each text entry method. ***, **, and * represent a. 001, .01, and .05 significance level (Bonferroni-adjusted), respectively. The same marking scheme is used in Fig. 5.

1) Entry Rate: An RM-ANOVA revealed that TEXT ENTRY METHOD had a significant main effect on entry rate ($F_{3,69} = 84.16, p < .001, \eta_p^2 = .785$). Post-hoc tests showed significant differences for each pair of the methods, as shown in Fig. 4(a). When comparing pointing-switch methods and button-switch methods, we found both LBS and KBS had significantly higher entry rates than STD and LPS (all $p < .001$). Of the two conventional pointing-switch methods, LPS was faster than STD ($p < .001$).

2) Layer Switching IKI: A Friedman test revealed that there was a statistically significant effect of the TEXT ENTRY METHOD on layer switching IKI ($\chi_3^2 = 62.370, p < .001, W = .866$). In post-hoc tests, we found both KBS and LBS had significantly shorter layer switching IKI than STD and LPS. Fig. 4(b) illustrates these results.

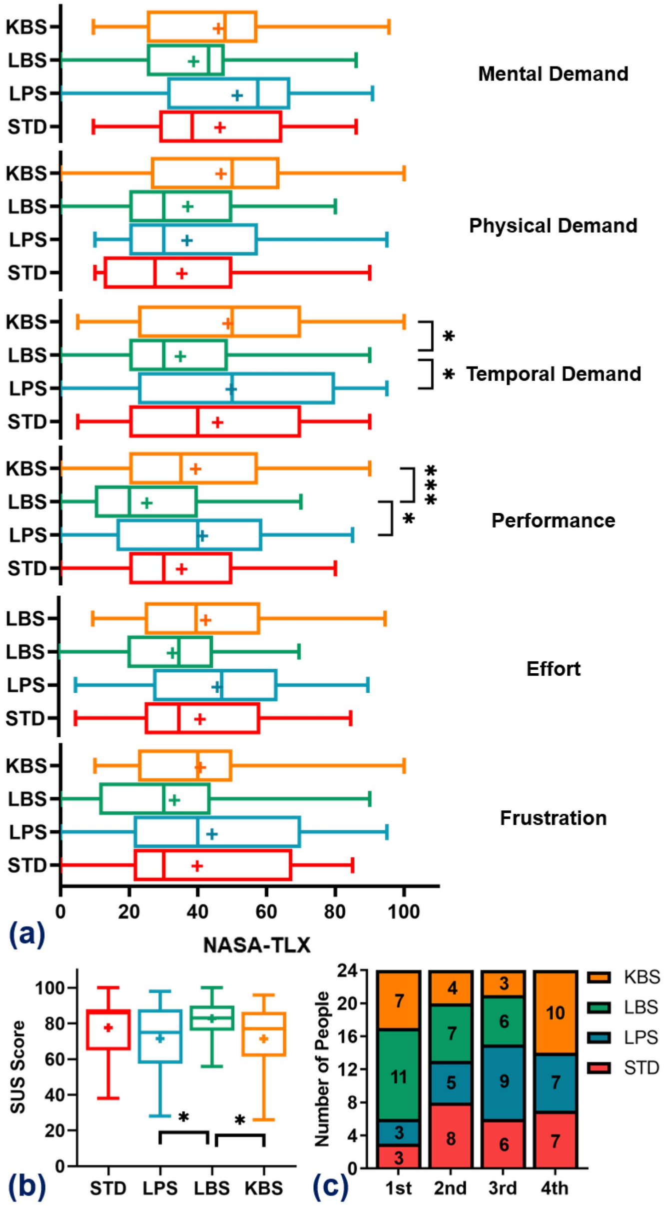


Fig. 5. Boxplots of (a) NASA-TLX scores (the lower, the better) and (b) median SUS scores (the higher, the better). (c) The ranking of each technique.

3) Error Rate: Friedman tests indicated that TEXT ENTRY METHOD had significant main effects on TER ($\chi^2_3 = 29.150, p < .001, W = .405$), but no significant differences on NCER ($p > .05$). Post-hoc tests showed that LBS ($Mdn = 4.65\%$) and KBS ($Mdn = 4.46\%$) had significantly lower TER than STD ($Mdn = 8.27\%$) (both $p < .001$) and LPS ($Mdn = 8.64\%$) (LBS versus LPS: $p = .001$, KBS versus LPS: $p = .003$). Fig. 4(c) and (d) summarize the TER and NCER results, respectively.

4) Perceived Workload: Fig. 5(a) shows the NASA-TLX scores for the four text entry methods. MANOVA revealed there was no significant difference in perceived workload ($F = 2.864, p = .099, Wilks'\Lambda = .104, \eta_p^2 = .896$). RM-ANOVAs showed a significant main effect of TEXT ENTRY METHOD on temporal demand ($F_{3,69} = 3.130, p = .031, \eta_p^2 = .120$) and performance ($F_{1.969,45.287} = 3.513, p = .039, \eta_p^2 = .132$). Post-hoc tests indicated that LBS ($M = 34.58, SD = 24.58$) required less temporal demand than LPS ($M = 49.38, SD = 29.13$) ($p = .033$) and KBS ($M = 48.54, SD = 26.23$) ($p = .027$). On the contrary, participants were more satisfied with the performance of using LBS ($M = 24.58, SD = 19.78$) than LPS ($M = 40.21, SD = 26.02$) ($p = .027$) and KBS ($M = 39.38, SD = 23.6$) ($p = .001$).

5) Perceived Usability: STD received the highest median SUS score ($Mdn = 86.00$), while LBS was the second ($Mdn = 83.00$), LPS was the third ($Mdn = 75.00$), and KBS was the fourth ($Mdn = 77.00$). The result of the Friedman test revealed a significant difference in SUS scores among the four text entry methods ($\chi^2_3 = 12.489, p = .006, W = .173$), and pairwise comparisons found the perceived usability of LBS was higher than LPS ($p = .026$) and KBS ($p = .016$). The results are summarized in Fig. 5(b).

6) User Ranking: Friedman test revealed no significant difference in the ranking ($p > .05$). As shown in Fig. 5(c), there was no clear tendency towards favoring or disliking a text entry method for alphabetical, numerical, and special characters. For LPS and STD, both have 3 participants who liked the most, and 7 participants voted it as the least favorite. Seven participants preferred to use KBS, while 10 participants disliked it. We found that eleven (45.83%) participants ranked LBS first (the most favored), and no one ranked it fourth (the least favored).

G. Discussion

The results offer evidence to support our hypotheses *H1* regarding entry rate and layer switching IKI and *H2* regarding error rate, while *H3* and *H4* regarding perceived workload and usability are not supported.

1) Task Performance: Our results show that the typing performance (text entry rate, layer switching IKI, and error rate) of LBS and KBS (i.e., the two button-switch methods) outperformed STD and LPS (i.e., the two pointing-switching methods). Although involving alphanumerical and special characters in a text entry task resulted in a significant reduction in typing speed [42], the average entry rates of LBS and KBS still reached 15.94 WPM and 15.08 WPM, respectively. These results are closely aligned with the results exhibited from some raycasting techniques even though the transcribed texts in their tasks are simpler (i.e., with fewer or even without capital letters and symbols/numbers), such as 16.65 WPM [4] using MacKenzie's phrase set [34], 15.44 WPM [50], 17.4 WPM [62] and 19.75 WPM [61] with the Enron mobile email dataset [55]. The layer switching IKI of LBS and KBS showed a significant decrease compared to the pointing-switch methods; their mean layer switching IKI (0.54 s and 0.56 s) was less than half of that of STD (1.06 s) and LPS (1.48 s). These results show that using the buttons on controllers for mode switching made switching between different layers more fluid than switching via targeting and triggering the mode-switching keys on the virtual keyboard. Effectively, button-based switching makes text entry with alphanumerical and special characters more efficient.

We did not observe a significant difference in NCER among the four text entry methods, while the TER of the two button-switch methods (LBS and KBS) was significantly lower than the two pointing-switch methods (STD and LPS), as shown in Fig. 4. Since the character selection method was the same for all conditions, we believe the difference originated from the mode-switching mechanisms. Our results imply that switching layers via controller buttons can reduce the occurrence of errors. One possible reason is that pointing to a character key from another layer with a button-switching method does not require activating a virtual mode-switching key first (usually located at the left bottom corner), which involves two segments of pointer movement. With button-switching methods, users can go straight to the desired key, thus reducing the chance of making errors during navigation or pointer movement.

Although users do not see the buttons on the controller while wearing the HMD, this did not affect their typing performance. This is because, in the user-maintained mode-switching mechanism, the keyboard only switches to the corresponding mode if the user keeps the keys pressed. Even if the user switches incorrectly, the keyboard returns to the default mode (lowercase) by simply releasing the key. Using user-maintained mode-switching could possibly make it easier and faster to correct switching errors.

It is interesting that LPS, as the most commonly used virtual keyboard layout capable of inputting various types of characters in current VR text entry techniques (see Section II), performed the worst. In the mode-switching process, a pointing-switching method involves 4 Degrees of Freedom (DoFs), three to determine the direction of the controller rays and one to press the trigger button. The button-switching methods involve only 2 DoFs, one for moving the finger up and down on the controller to locate the mode-switching button and one for pressing the button. STD is a replica of the standard physical Qwerty keyboard, and typical users are quite familiar with the position and usage of the mode-switching keys. But the soft Qwerty keyboard has some minor differences according to different types of touchscreens [8], such as the position of mode-switching keys, delete key, and send key, etc. As such, it might take some mental effort and time for users to become familiar with and adjust to these differences in LPS.

2) Subjective Feedback: All four text entry methods show acceptable usability (mean score over 70) [3]. Most subjective data from NASA-TLX and ranking questionnaires did not show significant differences (*H3*). Only the LBS' temporal demand and performance scores in the NASA-TLX questionnaire were significantly lower than KBS and LPS. The SUS score of LBS was higher than LPS and KBS, but not STD. The SUS score of KBS did not show any difference with STD and LPS (*H4*). This is because some users were influenced by their inherent typing habits on a physical keyboard or mobile device keyboard, resulting in a preference for STD or LPS, even if they had better text entry performance using LBS and KBS. Their non-preference for KBS mainly stems from unfamiliarity with the usage of KBS. However, we found that participants' preferences for STD and KBS were polarised, with those who liked STD the most often hating KBS and vice versa. Participants with VR text entry experience preferred LBS and KBS, while those who did not have VR text entry experience preferred STD. For STD, some participants expressed a preference for this keyboard and gave high scores in the SUS questionnaire for the learning requirements. However, many felt it was unnecessarily complex. Based on our interviews and observations, most participants would only use either 'Cap Lock' or 'Shift' to switch to the uppercase mode, even though they knew both were available. This feedback indicates that the presence of the two switching mechanisms simultaneously brought about some redundancy.

It is noteworthy that the objective workload of LBS did not significantly outperform KBS, even if it fared better on user preference and two dimensions of NASA-TLX (temporal demand and performance). With LBS, participants only needed to determine which layer the character belonged to before switching modes. But with KBS, they had to find and point exactly to the target key before pressing the mode-switching button, which would let them subjectively perceive time pressure. This is the major reason for KBS' poor ratings on users' feelings. Another reason is that the symbols are located on the lower left, and the font is smaller than the letters in the middle, making it difficult for users to distinguish between similar symbols such as ':' and ';' due to the fixed keyboard position.

In addition, we observed that participants preferred to point to the key with their dominant hand (i.e., right hand for all our participants) and press the switching button on the controller with the non-dominant hand (left hand) when using LBS, while with KBS, they liked to do the two actions only with the dominant hand. This preference may be because in LBS, switching and pointing are strictly serial actions—users first switch modes, locate the target, and then point to it. Such sequential operations are the same as STD and LPS; that is, the same as conventional keyboards. Users may map the controller buttons to the mode-switching keys that are generally located on the left-bottom corner and only use the left hand for switching modes.

On the other hand, KBS breaks this custom since the actions of switching modes and locating characters can be done in parallel. Furthermore, the transited characters are displayed on the key. Thus, participants may find it more natural to switch to a desired mode and input the characters using the same hand at the same time.

Some participants commented that KBS allowed for the association of three characters from a fixed position, whereas the layout in the separated layouts of LPS and LBS interrupted this association. Conversely, another group of participants said they did not like KBS because it was different from the usual keyboard layout they were familiar with, and as such, they were not familiar with its layout. In addition, due to participants' unfamiliarity, they might need more visual scanning time [2], [56] for character input and mode switching. However, some participants with VR text entry experience (e.g., P6, P7, P11) performed better with KBS than LBS, even though overall, KBS' entry rate was slower than LBS. Considering the polarization of KBS in the ranking of preferences, we speculate that some participants disliked the interaction and did not have better performance because of the limited practice and use during the short duration of the study. When they gained more experience and became more experts, their performance with LBS and KBS would likely show a difference with the one-day study. Thus, the effect of the *key*-level and *layer*-level keyboard approaches on text entry performance and subjective preference needs further exploration.

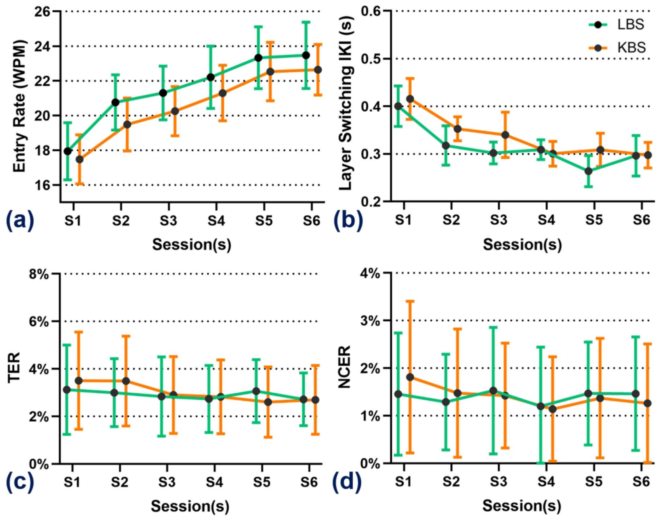


Fig. 6. (a) Mean entry rate; (b) mean layer switching IKI; (c) mean TER; and (d) mean NCER of the two text entry methods across the 6 sessions. Error bars represent 95% confidence intervals. The same labeling scheme is also used in Fig. 7 below.

V. Study 2: Evaluation of Prolonged Use

Given that the Layer-ButtonSwitch (LBS) and Key-ButtonSwitch (KBS) outperformed the other two text entry methods in terms of both objective performance and subjective preference, we wanted to investigate their performance when users have some consecutive practice sessions and thus more cumulative use. To this end, we conducted a three-day user study with two sessions per day.

A. *Participants, Apparatus, and Materials*

Ten participants (6 males and 4 females, aged from 21 to 31, $M = 23.71, SD = 3.55$) were recruited from the same university campus for this second study. All participants had normal or corrected-to-normal vision. They considered themselves familiar with the Qwerty keyboard. We used the same apparatus and materials as in Study 1. Eight of them participated in the first study.

B. *Procedure and Design*

For objective measurements, we used a within-subjects design with Text Entry Method (LBS and KBS) and Session as two independent variables. The experiment was conducted on three consecutive days, with two sessions per day. Participants were required to perform text entry using LBS and KBS across six sessions. We also used the Brown Corpus [12] in this study. Participants had to type 16 sentences (6 sentences as training trials and 10 sentences as formal trials) randomly selected from the corpus without duplication for each text entry method. The order of Text Entry Method was counterbalanced by a Latin square approach. Participants could take a break of at least three minutes between the two text entry methods. Each session lasted approximately 30 minutes. In total, we collected 1200 sentences (10 users × 2 techniques × 10 sentences × 3 days × 2 sessions).

Subjective measurements were on a daily basis. Each day, participants completed two sessions but filled out NASA-TLX and SUS questionnaires only after the second session. Thus, the independent variable Session was tuned to Day. We did this because participants would feel tired from repeatedly filling out the same questionnaires.

C. *Hypotheses*

We formulated four hypotheses for this user study:

- *H5:* The entry rate and layer switching IKI of the two text entry methods would gradually improve across sessions for typing alphanumerical and special characters.
- *H6:* The objective performance of KBS would outperform LBS with more practice time. As the key-level keyboard shows the numbers and symbols, it helps users to find the target character more quickly.
- *H7:* The workload of the two text entry methods would gradually reduce over the three days as users gradually increase their understanding of how to use them.
- *H8:* The usability of text entry methods would be improved as users gain proficiency and familiarity in three days.

D. *Results*

Before conducting statistical analyses, we identified and removed five sentences of 1200 sentences (or 0.41%) that participants failed to complete. The results of Shapiro-Wilk tests indicated that TER, NCER, and layer switching IKI, NASA-TLX, and SUS data were not normally distributed ($p < .05$). Thus, we applied Aligned Rank Transform [57] to these data before analyzing multi-dimensional data (NASA-TLX data) with MANOVA tests and one-dimensional data using RM-ANOVA tests. For normally distributed data (entry rate data), we applied RM-ANOVAs directly. Post-hoc pairwise comparisons with Bonferroni correction were used if significant differences were identified.

1) Entry Rate: The RM-ANOVAs yielded significant effects of Session ($F_{5,45} = 83.083, p < .001, \eta_p^2 = .902$) and Text Entry Method ($F_{1,9} = 17.152, p = .003, \eta_p^2 = .656$), and there was no interaction effect between Session × Text Entry Method ($p > .05$). Post-hoc pairwise comparisons indicated significant differences between sessions 1-2, 1-4. 1-5, 1-6, 2-5 2-6 and 3-6 ($p < .001$), 1-3, 2-4, 3-5 ($p = .001$), 4-5 ($p = .015$), 4-6 ($p = .022$) as well as 3-4 ($p = .040$). LBS (M=21.50 WPM, $SD = 2.05$) was significantly faster than KBS (M=20.62 WPM, $SD = 1.97$). Post-hoc pairwise comparisons also showed LBS was statically higher than KBS in session 2 ($p = .012$), session 3 ($p = .012$), and session 4 ($p = .024$). Fig. 6(a) shows the mean entry rate of the two

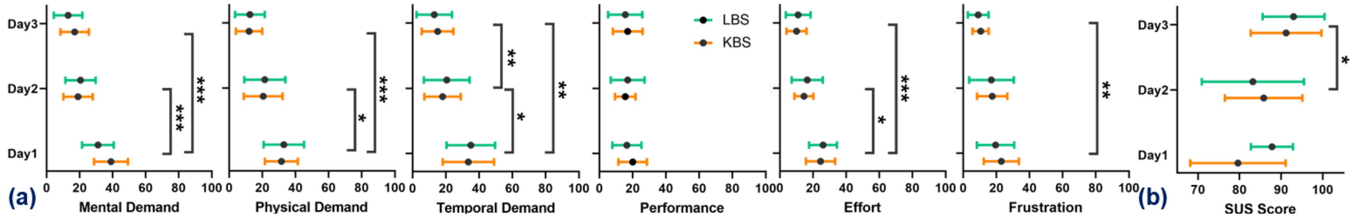


Fig. 7. (a) Mean NASA-TLX scores (the lower, the better); (b) mean SUS scores of LBS and KBS methods (the higher, the better).

TABLE I
SIGNIFICANT RM-ANOVA TEST AND POST-HOC RESULTS FOR NASA-TLX OVER THREE DAYS

		RM-ANOVA			Post-hoc		
	TEXT ENTRY METHOD	F	p	η_p^2	Day 1 vs. Day 2	Day 1 vs. Day 3	Day 2 vs. Day 3
Mental	Layer-ButtonSwitch	43.923	<.001	.830	.001	<.001	.013
	Key-ButtonSwitch	16.810	<.001	.651	.003	<.001	ns
Physical	Layer-ButtonSwitch	13.542	<.001	.601	.027	.001	ns
	Key-ButtonSwitch	16.991	<.001	.654	.030	<.001	ns
Temporal	Layer-ButtonSwitch	13.118	<.001	.593	.041	.005	ns
	Key-ButtonSwitch	14.050	<.001	.610	ns	.003	.044
Effort	Layer-ButtonSwitch	12.088	<.001	.573	.032	.005	ns
	Key-ButtonSwitch	19.724	<.001	.687	.043	<.001	ns
Frustration	Layer-ButtonSwitch	5.319	.015	.371	ns	.004	ns
	Key-ButtonSwitch	7.577	.004	.457	ns	.028	.025

'ns' means no significant difference.

text entry methods per day. The entry rate for both text entry methods seemed to have stabilized after the fifth session. The entry rate of LBS was improved to 23.47 WPM ($SD = 2.54$) in the last session from 17.94 WPM ($SD = 2.18$) in the first session. Similarly, the entry rate of KBS has improved from 17.48 WPM ($SD = 1.87$) to 22.64 WPM ($SD = 1.93$) over six sessions.

2) Layer Switching IKI: RM-ANOVAs revealed significant main effects of both SESSION ($F_{5,45} = 16.941, p < .001, \eta_p^2 = .653$) and TEXT ENTRY METHOD ($F_{1,9} = 34.048, p < .001, \eta_p^2 = .791$), and no significant interaction effect was found between SESSION × TEXT ENTRY METHOD ($p > .05$). Post-hoc pairwise comparisons found significant differences between sessions 1-2 ($p = .03$),1-3 ($p = .007$), 1-4 ($p = .001$), 1-5($p < .001$) and 1-6 ($p = .003$). Only in session 5, the layer switching IKI of LBS was statistically lower than of KBS ($p = .023$).

3) Error Rate: Fig. 6(c) and (d) summarize the TER and NCER results, respectively. RM-ANOVA did not reveal any significant differences for TER and NCER ($p > .05$). TER and NCER have been fluctuating at around 3% and 1.5%, respectively, over the 6 sessions.

4) Perceived Workload: Fig. 7(a) summarizes the mean NASA-TLX scores over three days. MANOVAs revealed DAY had a significant effect on workload with six dimensions ($F = 4.758, p < .001, Wilks'\Lambda = .098, \eta_p^2 = .687$). For each dimension of NASA-TLX, RM-ANOVAs showed there were significant effects across three days in mental ($F_{2,18} = 51.264, p < .001, \eta_p^2 = .851$), physical ($F_{2,18} = 17.204, p < .001, \eta_p^2 = .657$), temporal ($F_{1.163,10.467} = 15.202, p = .002, \eta_p^2 = .628$), effort ($F_{2,18} = 19.349, p < .001, \eta_p^2 = .683$) and frustration ($F_{2,18} = 8.461, p = .003, \eta_p^2 = .485$). Between LBS and KBS, there was not any significant difference in any dimension. Table I showed the significant results of RM-ANOVAs ($p < .05$) and post-hoc pairwise comparisons about NASA-TLX scores for each text entry method.

5) Perceived Usability: RM-ANOVAs found significant main effect of DAY in SUS scores ($F_{2,18} = 4.361, p = .029, \eta_p^2 = .326$). Fig. 7(b) summarizes the mean SUS scores over three days. Post-hoc pairwise comparisons showed the SUS scores on day 3 were significantly higher than on day 2 ($p = .027$).

E. Discussion

Our results support our hypotheses *H5*, *H7*, and *H8* but not *H6*. They show that both typing methods had similar learning curves, especially in entry rate (both over 22 WPM in the fifth session, which is efficient given that the tasks involved various types of characters) and layer switching IKI (around 0.30 s from the fourth session). TER stayed below 4% from the first session, which is acceptable (compared to a TER of 11.05% in [4]). For the two text entry methods, the increase in typing speed stabilized from the fifth to the sixth session (*H5*) and did not show a significant difference between the two sessions. Additionally, the decrease in layer switching IKI demonstrated a similar stable trend since the second session (*H5*). This indicates that users may have become proficient and comfortable with both methods after five sessions of practice (equivalent to a total of just about 1.25 hours). Similarly, as participants gradually became familiar with these methods, the mental, physical, and temporal demands significantly reduced (from somewhat high (30–49) to medium (10–29)) [39], and users feel they were paying less effort and gained less frustrated (*H7*). There was a significant drop in the three demands and effort on the second day; only temporal demand had a decrease on the second to the third days. Perhaps it is because participants have achieved the transition to a more 'expert' level after four to five sessions of practice. The average SUS scores for the two text entry methods were also higher than 70 across the three days and up to over 90 on the third day (*H8*). This further reinforces the fact that the benefits of the Qwerty layouts are that they are not only quick for users to learn how to use but also user-friendly, as Section II mentioned.

Our results confirm that the performance of KBS remains slower compared to LBS in the entry rate, even with continuous practice; hence, they do not support *H6*. The entry rate of KBS was statistically slower than LBS in sessions 2, 3, and 4 (the middle sessions). While in sessions 5 and 6 (on the last day), when the participants received more practice and became 'experts', there were not any significant differences between KBS and LBS. One possible reason is the learning curve for the two methods is different. In session 1, all participants were new to the techniques, and their text entry speeds on LBS and KBS did not vary significantly. However, the layout of LBS only had minor differences from the commonly used soft keyboard layout, which involved limited learning costs. Due to this, participants' performance was significantly improved after a single session. With more practice from sessions 2–5, participants' performance in using LBS improved gradually and steadily. In contrast, participants were less familiar with the layout of KBS. They need more time to grasp its usage. However, the key-level layout in KBS displays all characters, which is beneficial in locating the target character. This can explain why the improvement of entry rate for KBS was smaller than LBS in session 2, but greater from session 3 to session 5 (see the slopes of LBS and KBS in Fig. 6(a)). By session 5, participants had reached an 'expert' level in using both LBS and KBS, and their performance did not vary significantly between the two methods.

Both the objective measurements and the NASA-TLX ratings indicate that the key-level layout had the same level of performance as the layer-level layout when the participants got sufficient practice and were experienced. On the other hand, the average SUS score of KBS got close to LBS's gradually (Fig. 7(b)). We observed participants consistently using one hand for both mode switching and character input in KBS throughout the experiment. As mentioned in Study 1 (Section IV-G2), with KBS, the visibility of all characters allows users to switch modes while pointing to the desired character simultaneously, resembling parallel actions.

Having the characters visible also allows users to locate the target key before switching across modes, possibly aiding unintentional memorization of key positions [18], which helps improve text entry operations. Consequently, the input speed of KBS gradually approached that of LBS and showed no significant differences in sessions 5 and 6. Future research may be conducted to validate the capability and explore further the advantages of KBS in one-handed text entry scenarios.

In LBS, entering alphanumeric and special characters is done by layer-switching with a controller button, while in KBS, clear pointing actions are required. Thus, for text entry methods involving explicit pointing and selection actions, both methods can be applied and are not limited to the application of raycasting-based techniques. However, when the pointing action is not clearly distinguished, a method similar to LBS is more suitable as it does not involve target pointing before a layer-switching action. In addition, the keyboard layout of the two text entry methods is the cleanest version of the Qwerty layout. As such, when taking space into account, the two could also be applicable to devices with controller support to achieve more optimal use of screen space (e.g., in AR HMDs, which often have small field-of-views).

As the current commercial VR/AR HMD controllers have multiple keys, controller button switching can be easily adapted to other VR/AR devices and is not limited to Meta Quest 2 controllers used in this work. As switching via a controller button does not need interaction with visual objects, we can see that not only raycasting-based text entry methods are suitable for mode-switching with the controller button–simple and efficient, but other Qwerty layout-based text entry methods can benefit from this approach. Furthermore, except for controller buttons, other low DoF mechanisms also could be considered to do mode-switching, as the fewer the DoFs involved in the technique, the easier to use, especially in reducing the amount of muscle or hand movements required [1].

VI. Summary of Findings and Design Recommendations

In this section, we summarise the results of two user studies and distill three design recommendations for inputting alphanumeric and special characters in VR.

- *When handheld controllers are used, button-switching methods can be used for alphanumeric and special character entries:* From the first controlled experiment, we found that mode-switching using a controller button resulted in faster and more accurate alphanumeric and special characters entries compared to pointing-switching methods. The higher the DoFs involved, the more complex the operation becomes, which could negatively affect performance and usability [1]. Typically, mode-switching requires accuracy over expressiveness. Typing, especially long-term typing, takes physical (and often mental) effort. The lower the effort and the more accurate the selection mechanism, the more suitable it is for mode-switching.
- *User-maintained mechanism can be ideal for mode-switching in VR text entry:* In LBS and KBS, we used a user-maintained mechanism for mode-switching to enhance usability and simplicity; they did not need users to see the pointer position of the keys on the controller once they had some practice. According to the objective performance results of Study 1, a user-maintained mechanism not only solves this problem but also helps to reduce the error rate, which is in line with prior work [43]. It also provides an

effortless, simple, and fast way to return to the default mode and an easy way to correct wrong mode-switching.

- *Key-level layout can contribute to the parallelization of mode switching and character input actions:* In the Key-level layout, users could switch modes almost simultaneously while pointing at the desired character, given that all characters are visible throughout the process. Conversely, in the layer-level layout, participants must switch modes first before they can proceed to make character selections in the new mode since the user cannot see the target location or keys in other layers.

VII. LIMITATIONS AND FUTURE WORK

This research has the following four limitations, which could serve as directions for future work. First, given the identified design considerations, we only focused on controller-based raycasting character selection and leveraged controller buttons for mode-switching for inputting alphanumeric and special characters. Based on the findings derived from this work, we plan to explore more mode-switching methods with different interaction metaphors in the future, such as gesture-based or gaze-assisted interactions in VR/AR systems (e.g., [44], [65]).

Second, we used sentences from the Brown corpus because the sentences come with different types of alphabetical, numerical, and special characters, and are therefore reasonably representative of typical texts people type. Future work can explore other types of sentences or text fragments (e.g., complicated password combinations [56], or phrase sets used in some text entry evaluations [55]). Third, we wanted to evaluate our methods with more participants, though our sample size and tested trials passed the power analysis that ensured validity and reliability. In the future, it will be interesting to test our proposed methods and new ones with different population groups, such as older adult users. Fourth, as the first exploration of the topic, we used English as the default text entry language and investigated the text entry methods supporting efficient mode-switching between alphanumeric and special characters. However, it is also common to switch between languages other than English. In the future, we will extend our work to switch between multiple languages.

VIII. CONCLUSION

In this work, we designed two controller-based raycasting text entry methods (*Layer-ButtonSwitch* and *Key-ButtonSwitch*) and compared them with two other methods (*Standard Qwerty Keyboard* and *Layer-PointSwitch*) that were derived from physical and soft Qwerty keyboards. Results from two user studies showed that Layer-ButtonSwitch and Key-ButtonSwitch allowed participants to switch between layers smoothly. Layer-ButtonSwitch performed best in terms of efficiency and accuracy, while Key-ButtonSwitch facilitated parallel mode switching and character input operations. Our findings provide valuable guidance for the design of text entry methods for VR applications involving alphanumeric and special characters. Our work also underscores the importance of considering user preferences and designing text entry methods that are easy to learn and use. Overall, this work contributes to the development of effective and efficient alphanumeric and special characters text entry methods in VR and opens up new design possibilities.

ACKNOWLEDGMENTS

The authors thank the participants for their time and the reviewers for the insightful and detailed comments, which helped improve our article.

REFERENCES

[1] F. Argelaguet and C. Andujar, "A survey of 3D object selection techniques for virtual environments," *Comput. Graph.*, vol. 37, no. 3, pp. 121–136, 2013, doi: 10.1016/j.cag.2012.12.003.

[2] A. S. Arif and W. Stuerzlinger, "Analysis of text entry performance metrics," in *Proc. IEEE Toronto Int. Conf. Sci. Technol. Humanity*, 2009, pp. 100–105, doi: 10.1109/TIC-STH.2009.5444533.

[3] A. Bangor, P. T. Kortum, and J. T. Miller, "An empirical evaluation of the system usability scale," *Int. J. Hum. Comput. Interact.*, vol. 24, no. 6, pp. 574–594, 2008, doi: 10.1080/10447310802205776.

[4] C. Boletsis and S. Kongsvik, "Controller-based text-input techniques for virtual reality: An empirical comparison," *Int. J. Virtual Reality*, vol. 19, no. 3, pp. 2–15, 2019, doi: 10.20870/IJVR.2019.19.3.2917.

[5] C. Boletsis and S. Kongsvik, "Text input in virtual reality: A preliminary evaluation of the drum-like VR keyboard," *Technologies*, vol. 7, no. 2, 2019, Art. no. 31, doi: 10.3390/technologies7020031.

[6] J. Brooke et al., "SUS-A quick and dirty usability scale," *Usability Eval. Ind.*, vol. 189, no. 194, pp. 4–7, 1996.

[7] S. Chen, J. Wang, S. Guerra, N. Mittal, and S. Prakkamakul, "Exploring word-gesture text entry techniques in virtual reality," in *Proc. Extended Abstr. CHI Conf. Hum. Factors Comput. Syst.*, New York, NY, USA, 2019, pp. 1–6, doi: 10.1145/3290607.3312762.

[8] J. Cuaresma and I. S. MacKenzie, "A study of variations of qwerty soft keyboards for mobile phones," in *Proc. Int. Conf. Multimedia Hum.-Comput. Interact.*, 2013, pp. 126–1.

[9] E. Dresner and S. C. Herring, "Functions of the nonverbal in CMC: Emoticons and illocutionary force," *Commun. Theory*, vol. 20, no. 3, pp. 249–268, Jul. 2010, doi: 10.1111/j.1468–2885.2010.01362.x.

[10] J. Dudley, H. Benko, D. Wigdor, and P. O. Kristensson, "Performance envelopes of virtual keyboard text input strategies in virtual reality," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2019, pp. 289–300, doi: 10.1109/ISMAR.2019.00027.

[11] F. Faul, E. Erdfelder, A.-G. Lang, and A. Buchner, "G*Power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences," *Behav. Res. Methods*, vol. 39, no. 2, pp. 175–191, 2007, doi: 10.3758/BF03193146.

[12] W. N. Francis and H. Kucera, "Brown corpus manual," *Lett. Editor*, vol. 5, no. 2, 1979, Art. no. 7.

[13] D. George and P. Mallery, *IBM SPSS Statistics 26 Step by Step: A Simple Guide and Reference*. Evanston, IL, USA: Routledge, 2019, doi: 10.4324/9780429056765.

[14] J. Grubert, E. Ofek, M. Pahud, and P. O. Kristensson, "The office of the future: Virtual, portable, and global," *IEEE Comput. Graph. Appl.*, vol. 38, no. 6, pp. 125–133, Nov./Dec. 2018, doi: 10.1109/MCG.2018.2875609.

[15] J. Grubert, L. Witzani, E. Ofek, M. Pahud, M. Kranz, and P. O. Kristensson, "Effects of hand representations for typing in virtual reality," in *Proc. IEEE Conf. Virtual Reality 3D User Interfaces*, 2018, pp. 151–158, doi: 10.1109/VR.2018.8446250.

[16] J. Grubert, L. Witzani, E. Ofek, M. Pahud, M. Kranz, and P. O. Kristensson, "Text entry in immersive head-mounted display-based virtual reality using standard keyboards," in *Proc. IEEE Conf. Virtual Reality 3D User Interfaces*, 2018, pp. 159–166, doi: 10.1109/VR.2018.8446059.

[17] A. Gupta, C. Ji, H.-S. Yeo, A. Quigley, and D. Vogel, "RotoSwype: Word-gesture typing using a ring," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, New York, NY, USA, 2019, pp. 1–12, doi: 10.1145/3290605.3300244.

[18] C. L. Hart, J. A. Randell, and J. D. Griffith, "Ironic effects of attempting to remember," *North Amer. J. Psychol.*, vol. 9, no. 2, p. 201, 2007.

[19] S. G. Hart, "NASA-Task load index (NASA-TLX); 20 years later," in *Proc. Hum. Factors Ergonom. Soc. Annu. Meeting*, 2006, pp. 904–908, doi: 10.1177/154193120605000909.

[20] J. Henderson, J. Ceha, and E. Lank, “STAT: Subtle typing around the thigh for head-mounted displays,” in *Proc. 22nd Int. Conf. Hum.-Comput. Interact. Mobile Devices Serv.*, New York, NY, USA, 2020, Art. no. 27, doi: 10.1145/3379503.3403549.

[21] R. S. Hirsch, “Effects of standard versus alphabetical keyboard formats on typing performance,” *J. Appl. Psychol.*, vol. 54, no. 6, 1970, Art. no. 484, doi: 10.1037/h0030143.

[22] H. Jiang and D. Weng, “HiPad: Text entry for head-mounted displays using circular touchpad,” in *Proc. IEEE Conf. Virtual Reality 3D User Interfaces*, 2020, pp. 692–703, doi: 10.1109/VR46266.2020.00092.

[23] B. A. Johnson, “The effect of lower case letter forms on beginning reading and writing: An action research,” *Graduate Res. Papers*, 1995. [Online]. Available: https://scholarworks.uni.edu/grp/2616

[24] J. Johnson and G. Engelbeck, “Modes survey results,” *ACM SIGCHI Bull.*, vol. 20, no. 4, pp. 38–50, Apr. 1989, doi: 10.1145/67243.67248.

[25] W. Kim and S. Xiong, “Pseudo-haptics and self-haptics for freehand mid-air text entry in VR,” *Appl. Ergonom.*, vol. 104, 2022, Art. no. 103819, doi: 10.1016/j.apergo.2022.103819.

[26] P. Knierim, V. Schwind, A. M. Feit, F. Nieuwenhuizen, and N. Henze, “Physical keyboards in virtual reality: Analysis of typing performance and effects of avatar hands,” in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, New York, NY, USA, 2018, pp. 1–9, doi: 10.1145/3173574.3173919.

[27] P. O. Kristensson, “Five challenges for intelligent text entry methods,” *AI Mag.*, vol. 30, no. 4, pp. 85–85, 2009, doi: 10.1609/aimag.v30i4.2269.

[28] P. O. Kristensson, “Next-generation text entry,” *Computer*, vol. 48, no. 07, pp. 84–87, 2015.

[29] X. Lu, D. Yu, H.-N. Liang, and J. Goncalves, “iText: Hands-free text entry on an imaginary keyboard for augmented reality systems,” in *Proc. 34th Annu. ACM Symp. User Interface Softw. Technol.*, 2021, pp. 815–825, doi: 10.1145/3472749.3474788.

[30] X. Lu et al., “Exploration of hands-free text entry techniques for virtual reality,” in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2020, pp. 344–349, doi: 10.1109/ISMAR50242.2020.00061.

[31] F. Lyddy, F. Farina, J. Hanney, L. Farrell, and N. K. O’Neill, “An analysis of language in university students’ text messages,” *J. Comput.-Mediated Commun.*, vol. 19, no. 3, pp. 546–561, Apr. 2014, doi: 10.1111/jcc4.12045.

[32] J. Ma, W. Yang, M. Luo, and N. Li, “A study of probabilistic password models,” in *Proc. IEEE Symp. Secur. Privacy*, 2014, pp. 689–704, doi: 10.1109/SP.2014.50.

[33] X. Ma, Z. Yao, Y. Wang, W. Pei, and H. Chen, “Combining brain-computer interface and eye tracking for high-speed text entry in virtual reality,” in *Proc. 23rd Int. Conf. Intell. User Interfaces*, New York, NY, USA, 2018, pp. 263–267, doi: 10.1145/3172944.3172988.

[34] I. S. MacKenzie and S. X. Zhang, “An empirical investigation of the novice experience with soft keyboards,” *Behav. Inf. Technol.*, vol. 20, no. 6, pp. 411–418, 2001, doi: 10.1080/01449290110089561.

[35] T. Ogitani, Y. Arahori, Y. Shinyama, and K. Gondow, “Space saving text input method for head mounted display with virtual 12-key keyboard,” in *Proc. IEEE 32nd Int. Conf. Adv. Inf. Netw. Appl.*, 2018, pp. 342–349, doi: 10.1109/AINA.2018.00059.

[36] C. Park, H. Cho, S. Park, Y.-S. Yoon, and S.-U. Jung, “HandPoseMenu: Hand posture-based virtual menus for changing interaction mode in 3D space,” in *Proc. ACM Int. Conf. Interactive Surf. Spaces*, New York, NY, USA, 2019, pp. 361–366, doi: 10.1145/3343055.3360752.

[37] D.-M. Pham and W. Stuerzlinger, “HawKEY: Efficient and versatile text entry for virtual reality,” in *Proc. 25th ACM Symp. Virtual Reality Softw. Technol.*, New York, NY, USA, 2019, Art. no. 21, doi: 10.1145/3359996.3364265.

[38] S. Pick, A. S. Puika, and T. W. Kuhlen, “SWIFTER: Design and evaluation of a speech-based text input metaphor for immersive virtual environments,” in *Proc. IEEE Symp. 3D User Interfaces*, 2016, pp. 109–112, doi: 10.1109/3DUI.2016.7460039.

[39] A. D. Prabaswari, C. Basumerda, and B. W. Utomo, “The mental workload analysis of staff in study program of private educational organization,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 528, no. 1, May 2019, Art. no. 012018, doi: 10.1088/1757–899X/528/1/012018.

[40] V. Rajanna and J. P. Hansen, “Gaze typing in virtual reality: Impact of keyboard design, selection method, and motion,” in *Proc. ACM Symp. Eye Tracking Res. Appl.*, New York, NY, USA, 2018, Art. no. 15, doi: 10.1145/3204493.3204541.

[41] D. Schneider et al., “ReconViguRation: Reconfiguring physical keyboards in virtual reality,” *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 11, pp. 3190–3201, Nov. 2019.

[42] A. Sears and Y. Zha, “Data entry for mobile devices using soft keyboards: Understanding the effects of keyboard size and user tasks,” *Int. J. Hum.–Comput. Interact.*, vol. 16, no. 2, pp. 163–184, 2003, doi: 10.1207/S15327590IJHC1602_03.

[43] A. J. Sellen, G. P. Kurtenbach, and W. A. Buxton, “The prevention of mode errors through sensory feedback,” *Hum.–Comput. Interact.*, vol. 7, no. 2, pp. 141–164, 1992, doi: 10.1207/s15327051hci0702_1.

[44] R. Shi, Y. Wei, X. Qin, P. Hui, and H.-N. Liang, “Exploring gaze-assisted and hand-based region selection in augmented reality,” in *Proc. ACM Hum.-Comput. Interact.*, vol. 7, no. ETRA, May 2023, Art. no. 160, doi: 10.1145/3591129.

[45] R. Shi, N. Zhu, H.-N. Liang, and S. Zhao, “Exploring head-based mode-switching in virtual reality,” in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2021, pp. 118–127, doi: 10.1109/ISMAR52148.2021.00026.

[46] J. Smith, I. Wang, W. Wei, J. Woodward, and J. Ruiz, “Evaluating the scalability of non-preferred hand mode switching in augmented reality,” in *Proc. Int. Conf. Adv. Vis. Interfaces*, New York, NY, USA, 2020, Art. no. 19, doi: 10.1145/3399715.3399850.

[47] J. Smith, I. Wang, J. Woodward, and J. Ruiz, “Experimental analysis of single mode switching techniques in augmented reality,” in *Proc. 45th Graph. Interface Conf. Proc. Graph. Interface*, 2019, Art. no. 20, doi: 10.20380/GI2019.20.

[48] Z. Song, J. J. Dudley, and P. O. Kristensson, “Efficient special character entry on a virtual keyboard by hand gesture-based mode switching,” in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2022, pp. 864–871, doi: 10.1109/ISMAR55827.2022.00105.

[49] 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 *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, New York, NY, USA, 2003, pp. 113–120, doi: 10.1145/642611.642632.

[50] M. Speicher, A. M. Feit, P. Ziegler, and A. Krüger, “Selection-based text entry in virtual reality,” in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, New York, NY, USA, 2018, pp. 1–13, doi: 10.1145/3173574.3174221.

[51] L. Stark and K. Crawford, “The conservatism of emoji: Work, affect, and communication,” *Social Media + Soc.*, vol. 1, no. 2, pp. 1–11, 2015, doi: 10.1177/2056305115604853.

[52] H. B. Surale, F. Matulic, and D. Vogel, “Experimental analysis of barehand mid-air mode-switching techniques in virtual reality,” in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, New York, NY, USA, 2019, pp. 1–14, doi: 10.1145/3290605.3300426.

[53] G. B. Thompson, “The long learning route to abstract letter units,” *Cogn. Neuropsychol.*, vol. 26, no. 1, pp. 50–69, 2009, doi: 10.1080/02643290802200838.

[54] K. Tominaga, S. Fujita, R. Takakura, and B. Shizuki, “Investigating the effects of position and angle of virtual keyboard on text entry performance and workload,” in *Proc. Asian CHI Symp.*, New York, NY, USA, 2021, pp. 25–27, doi: 10.1145/3429360.3468174.

[55] K. Vertanen and P. O. Kristensson, “A versatile dataset for text entry evaluations based on genuine mobile emails,” in *Proc. 13th Int. Conf. Hum. Comput. Interact. Mobile Devices Serv.*, 2011, pp. 295–298.

[56] T. Wan et al., “Hands-free multi-type character text entry in virtual reality,” *Virtual Reality*, vol. 28, 2024, Art. no. 8, doi: 10.1007/s10055–023-00902-z.

[57] J. O. Wobbrock, L. Findlater, D. Gergle, and J. J. Higgins, “The aligned rank transform for nonparametric factorial analyses using only ANOVA procedures,” in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, 2011, pp. 143–146.

[58] W. Xu, H.-N. Liang, A. He, and Z. Wang, “Pointing and selection methods for text entry in augmented reality head mounted displays,” in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2019, pp. 279–288, doi: 10.1109/ISMAR.2019.00026.

[59] 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 Trans. Vis. Comput. Graph.*, vol. 25, no. 5, pp. 1991–2001, May 2019, doi: 10.1109/TVCG.2019.2898736.

[60] H. Yamada, “A historical study of typewriters and typing methods, from the position of planning Japanese parallels,” *J. Inf. Process.*, vol. 2, pp. 175–202, Feb. 1980.

[61] C. Yildirim, “Point and select: Effects of multimodal feedback on text entry performance in virtual reality,” *Int. J. Hum.–Comput. Interact.*, vol. 39, no. 19, pp. 3815–3829, 2023, doi: 10.1080/10447318.2022.2107330.

[62] C. Yildirim and E. Osborne, “Text entry in virtual reality: A comparison of 2D and 3D keyboard layouts,” in *Proc. Int. Conf. Hum.-Comput. Interact.*, C. Stephanidis, J. Y. C. Chen, and G. Fragomeni, Eds., Cham, Switzerland: Springer, 2020, pp. 450–460, doi: 10.1007/978–3-030-59990-4_33.

[63] C. Yu, Y. Gu, Z. Yang, X. Yi, H. Luo, and Y. Shi, "Tap, dwell or gesture? Exploring head-based text entry techniques for HMDs," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, New York, NY, USA, 2017, pp. 4479–4488, doi: 10.1145/3025453.3025964.

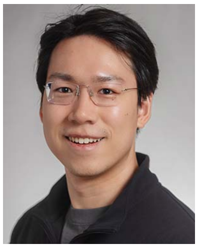
[64] D. Yu, K. Fan, H. Zhang, D. Monteiro, W. Xu, and H.-N. Liang, "PizzaText: Text entry for virtual reality systems using dual thumbsticks," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 11, pp. 2927–2935, Nov. 2018, doi: 10.1109/TVCG.2018.2868581.

[65] D. Yu et al., "Gaze-supported 3D object manipulation in virtual reality," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2021, Art. no. 734, doi: 10.1145/3411764.3445343.

Tingjie Wan received the BEng degree in Internet of Things engineering from Hunan University, China, and the MSc degree in human-computer interaction from the University of Nottingham, U.K. She is currently working toward the PhD degree with the School of Advanced Technology, Xi'an Jiaotong-Liverpool University, China, and with the Department of Computer Science, University of Liverpool, U.K. Her research interests include human-computer interaction, virtual/mixed reality, and text entry.

Yushi Wei (Student Member, IEEE) received the MRes degree in computer science from the University of Liverpool, U.K. He is currently working toward the PhD degree with the School of Advanced Technology, Xi'an Jiaotong-Liverpool University, China, and with the Department of Computer Science, University of Liverpool, U.K. His research interests include human-computer interaction, user behavior modeling, and target selection in extended reality.

Rongkai Shi (Student Member, IEEE) is currently working toward the PhD degree with the Department of Computing, Xi'an Jiaotong-Liverpool University, Suzhou, China, and with the Department of Computer Science, University of Liverpool, U.K. His research interests include virtual reality, augmented reality, and interaction design.



Junxiao Shen received the BA and MEng degrees from the University of Cambridge. He is currently working toward the PhD degree with the Department of Engineering, University of Cambridge and a member of Trinity College, Cambridge. His research interests include developing next-generation input methods and interactive AI agents in mixed reality.

Per Ola Kristensson is a professor of interactive systems engineering with the Department of Engineering, University of Cambridge and a fellow of Trinity College, Cambridge. He leads the Intelligent Interactive Systems Group, which belongs to the Engineering Design Centre. He is also a co-founder and co-director of the Centre for Human-Inspired Artificial Intelligence, University of Cambridge.

Katie Atkinson is a professor of computer science and dean with the School of Electrical Engineering, Electronics and Computer Science, University of Liverpool, U.K. Her research interests are focused on explainable AI and the interdisciplinary development of AI.

Hai-Ning Liang (Member, IEEE) is a professor with the Department of Computing, Xi'an Jiaotong-Liverpool University (XJTLU), Suzhou, China. He was the founding head of the department (2019–2023) and serves as the deputy director of the Suzhou Key Laboratory of Intelligent Virtual Engineering and the XJTLU Virtual Engineering Center. He does research in human-computer interaction, focusing on virtual/augmented/mixed reality, gaming, and visualization technologies.