PizzaText: Text Entry for Virtual Reality Systems Using Dual Thumbsticks

Difeng Yu, Kaixuan Fan, Heng Zhang, Diego Monteiro, Wenge Xu, and Hai-Ning Liang

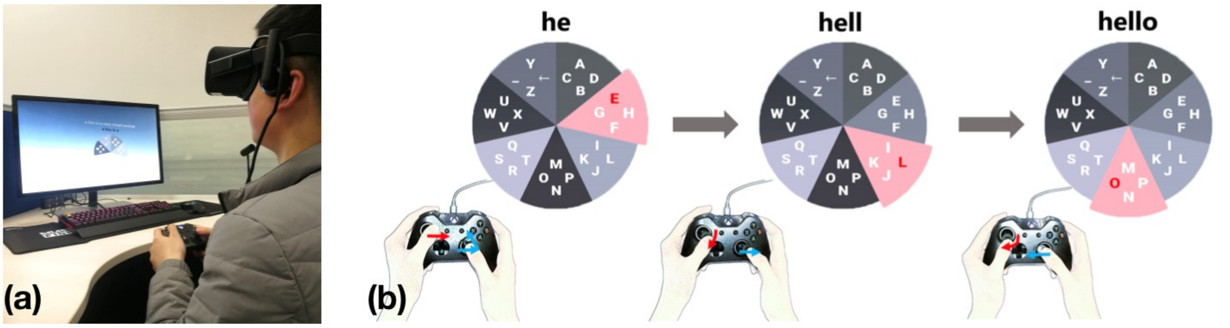

Fig. 1. (a) A user is entering some text using PizzaText in the virtual reality environment; (b) to enter "hello", the user performs a series of gestures using the two thumbsticks of the game controller.

Abstract—We present PizzaText, a circular keyboard layout technique for text entry in virtual reality (VR) environments that uses the dual thumbsticks of a hand-held game controller. Text entry is a common activity in VR environments but remains challenging with existing techniques and keyboard layouts that is largely based on QWERTY. Our technique makes text entry simple, easy, and efficient, even for novice users. The technique uses a hand-held controller because it is still an important input device for users to interact with VR environments. To allow rapid search of characters, PizzaText divides a circle into slices and each slice contains 4 characters. To enable fast selection, the user uses the right thumbstick for traversing the slices, and the left thumbstick for choosing the letters. The design of PizzaText is based on three criteria: efficiency, learnability, and ease-of-use. In our first study, six potential layouts are considered and evaluated. The results lead to a design with 7 slices and 4 letters per slice. The final design is evaluated in a five-day study with 10 participants. The results show that novice users can achieve an average of 8.59 Words per Minute (WPM), while expert users are able to reach 15.85 WPM, with just two hours of training.

Index Terms—Virtual reality, text entry, game controller, dual-joystick input, selection keyboard, circular keyboard layout

1 Introduction

Hand-held controllers are an important modality for text entry in existing consumer virtual reality (VR) systems. Users often have to enter personal information for service registration or to log in to a system, text short bursts of messages to other players during gameplay or make text annotations in virtual environments (VE). While many text entry techniques have been introduced using a game controller, there is still space for improvement in terms of efficiency, ease-of-use, and ease-of-learning, especially for novice users.

Most current text entry methods with controllers employ the aim-and-shoot technique [3, 37] but such methods can be tedious and cumbersome. Some other techniques, like the Google Drum Keys (by DayDream Lab), require a substantial training time for novice users; they are also prone to cause fatigue quickly because they involve large spatial movements to select the characters [12]. To avoid problems when using a game controller for text entry, researchers have investigated other approaches such as speech [4, 39] and mid-air typing [9, 41, 57]. However, speech techniques might be unsuitable in public situations [12, 56] and mid-air typing techniques might require extra devices (e.g. cameras or gloves) and often limit users to certain physical postures and locations. Therefore, it is still important to devise efficient and easy-to-learn text entry methods, including those based on hand-held controllers, for VR systems.

In this paper, we present PizzaText, a circular layout-based text entry technique for VR systems using the two thumbsticks of a game controller (Fig. 1). By rotating the two joysticks of the game controller, a user can easily enter text by using this circular keyboard layout. To overcome the limitations of the current techniques, we designed the keyboard layout iteratively focusing on improving efficiency, learnability, and usability. After designing potential layouts, we did an experiment with eighteen participants to determine the proper layout and number of letters per slice of PizzaText. The results indicated that participants could achieve an average of 8.59 WPM (s.e. = 0.58) using the best-performing technique (the 4 keys per slice layout) in the first study. To further explore its learning curve and evaluate its performance, we then conducted a 5-day study with ten participants to evaluate text entry speed and accuracy of the 4-key layout. We found that the expert users (five participants who performed best in the second experiment) achieved an average of 15.85 WPM (s.e. = 0.36) in the fifth day with 1.59% (s.e. = 0.31%) uncorrected errors. This performance can be considered high compared to other techniques using a game controller.

Our contributions in this work include (1) a circular text entry technique, PizzaText, for VR environments using dual joysticks; (2) a 5-day user study of the effectiveness of PizzaText; and (3) a set of design

• *Difeng Yu, Kaixuan Fan, Heng Zhang, Diego Monteiro, Wenge Xu, and Hai-Ning Liang are with the Computer Science and Software Engineering, Xi'an Jiaotong-Liverpool University, Suzhou, Jiangsu, China. E-mails: {Difeng.Yu14; Kaixuan.Fan16; Heng.Zhang14}@student.xjtlu.edu.cn; {D.Monteiro; Wenge.Xu; HaiNing.Liang }@xjtlu.edu.cn.*
• *Corresponding author: Hai-Ning Liang; E-mail: HaiNing.Liang@xjtlu.edu.cn*

Manuscript received 19 Mar. 2018; accepted 10 July. 2018.
Date of publication 14 Sept. 2018; date of current version 28 Sept. 2018.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TVCG.2018.2868581

rationales for text entry techniques in VE using dual thumbsticks.

2 Related Work

In this section, we present our review of the literature with respect to text entry techniques in VE, text entry methods with joysticks, and circular keyboard layout designs.

2.1 Text Entry in Virtual Environments

With the growing popularity of VR technologies, there has also been a growth in the number of techniques for text entry for VE. Virtual Notepad [40], one of the first works in the area, investigated the potential of virtual handwriting using a pressure-sensitive drawing tablet and pen. Although this work saved input as a series of pen strokes (and not actual text data), it suggested a possible usage of later 3D handwriting recognition approaches [26, 36] for entering text in immersive VE.

Speech techniques, due to their simplicity and efficiency, have played an important role in VE user interfaces. Bowman et. al [4] conducted an empirical comparison among four techniques (speech, a pen and tablet keyboard, a one-hand chord keyboard, and a typing emulation technique using pinch gloves) and indicated that the speech technique was the fastest medium for text entry at around 14 Words per Minute (WPM). A recent speech-based multimodal technique, SWIFTER [39], is claimed to be able to achieve an average input rate of 23.6 WPM. Despite this performance, speech recognition has severe limitations of ambient noise sensitivity, privacy, and being obtrusive in a shared environment [12]. It might also interfere with other cognitive tasks [45] and can be difficult to correct errors [49].

Touchscreen-based techniques [12, 15, 24, 28], on the one hand, enable mobile VR text entry and have a fairly good input speed (17-23 WPM [28]). On the other hand, because it is not easy for users to sense the location of their hands before the first press [15], this type of techniques might require extra movements for selecting the goal target. Other approaches such as mid-air typing [9, 41, 57] and physical keyboard-based techniques [12, 34, 50, 51] have also been proposed for text entry in VR systems. One challenge is that these techniques might require extra sensors or devices and may pose extra difficulty when the user is interacting with the VE using other input devices, for example, game controllers. They also confine users to restrictive postures and locations where sensors are installed.

Head-based techniques have also been explored recently for VE. Yu et. al [58] designed and compared three head-based text entry techniques for VR Head-Mounted Displays (HMD) and reported a performance of around 24.73 WPM with their best technique GestureType after 60 minutes of training. However, as stated by Yu et al., GestureType does not support inputting out of dictionary (OOD) words—i.e., passwords and chatspeak words will not work. In addition, the technique uses frequent head movements (to make swiping motions) and with them the possibility of motion sickness is likely to increase.

For existing consumer VR systems, such as the Oculus Rift, HTC Vive, and Sony PlayStation VR, hand-held controllers are an important modality for interaction in general and for text entry in particular instead of head-based techniques. Although a number of text entry techniques have been introduced for existing consumer VR systems (like Google Drum Keys and aim-and-shoot techniques [3, 37]) using a game controller, empirical evaluations are limited. As claimed by Grubert et al. [12], some current text entry methods with controllers might cause the user to get fatigued quickly and require a substantial learning curve. Indeed, efficient text entry methods in VR with hand-held game controllers have still remained underexplored. Therefore, in this work, we want to investigate potential VR text entry techniques for a hand-held controller which is able to support a fast entry speed, requires minimal learning time, and involves relatively small spatial movements to minimize fatigue and tiredness. Due to the diversity of the hand-held controllers, in this research, we will focus on the Xbox One controller, as it is one of the most commonly used devices and has the same features as other brands of controllers.

2.2 Text Entry with Joysticks

Some techniques have been proposed for joystick-operated text entry and can be categorized into two main streams: selection- and gesture-based.

By using selection-based joystick input techniques, the user has to move the highlighted keystroke to the target location and then select the target key. The speed of single joystick QWERTY selection keyboard has been shown to be 6.2 WPM [54]. Wilson and Agrawala [52] later employed a bimanual text entry technique for QWERTY keyboards and reported a 7.1 WPM performance for novice users. Sandnes and Aubert [44] proposed a similar approach by simulating the two-finger typing mechanism on a QWERTY keyboard and claimed an average speed of 6.75 WPM with less than one hour of practice. While the user may be more familiar with QWERTY layouts, the rectangle design might not be suitable for joystick input and might cause problems such as under- or over-shooting the goal targets [35]. The TwoStick technique [25] employed a 9×9 grid keystrokes with two levels of control which are mapped to two sticks respectively. This technique was slower than QWERTY layout during the early parts of their experiment (at 4.3 WPM) but can become faster (at 14.9 WPM) after around five hours of training.

Gesture-based joystick input techniques require the user to draw unistroke alphabet letters using the stick. The gesture-based technique called MDITIM [20] was proposed earlier and the users were shown to be able to reach around 5.6 WPM using a joystick. However, the authors found that, according to their results, MDITIM was not fast enough during the first five hours of practice. Another technique, EdgeWrite [54], allows the character to be input by straight movement of the joystick and has been reported to reach 6.4 WPM on average. A more recent technique, Feature Stroke [13], used letter-group based gestures and the text entry speed was showed to be 3.88 WPM for a novice user and 7.83 WPM for an expert user in the non-predictive mode. The error rate for novices was shown to be extremely high (33.62%). The writing-with-joystick [14] allows users to write letters using a free-form approach and by using a handwriting recognition system it can achieve 4.55 WPM without word prediction.

Another text entry method called Quikwriting [21] allows the text to be entered by moving the cursor over text zones. Participants were shown to improve from 4 WPM to 13 WPM after 5 hours of training. Apart from inputting Roman characters, joystick text entry has also been proposed for other languages [10, 23].

From this review, we can see that current non-QWERTY layouts or gesture-based techniques using joystick might require long periods of training and might not be able to get a suitable text entry speed for novice users. This has motivated us to design a new and easy-to-learn layout which allows novice users to achieve a relatively fast typing speed with very low error rates.

2.3 Circular Keyboard Layout

Although less common, several circular keyboard techniques have been proposed for different scenarios. The early work T-Cube [48] has been proposed for pen-based text entry and showed that a circular layout requires a small screen area and might be fast for experts. Another stylus-based technique Cirrin [33] and its enhanced version [5] applied world-level gesture input to a circular layout. The order of letters along the circumference was optimized for efficiency, but it might not be easy still for users to learn how to use it quickly. TUP [42] was designed and evaluated on the touchpad with the text entry speed of 6-7 WPM for novice users. Huckauf and Urbina designed pEYEwrite [19] which employed a hierarchical circular interface with gaze-based input and reported a speed of 7.85 WPM for novice users and 12.33 WPM maximum for an expert user. SliceType [2] was designed to use the screen area more efficiently by applying a language prediction model to merge keys of their inner-outer circle layout. The authors reported a 5.42 WPM text entry speed for mouse input and 3.45 WPM for gaze input. Apart from these works, circular techniques have been designed for walk-up tabletop installations [17] and very large wall displays [46]. Moreover, a circular keyboard layout can provide the potential extra benefit in certain scenarios such as interacting with circular interfaces

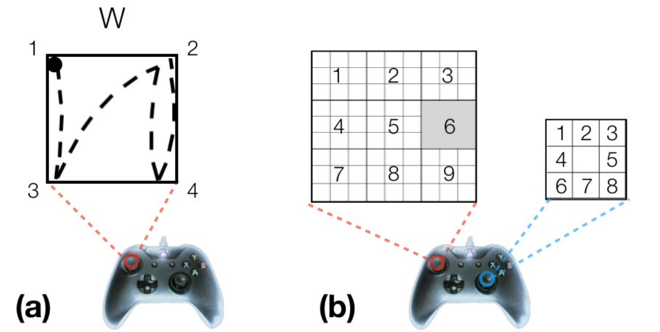

Fig. 2. (a) EdgeWrite is a base-4 technique and is shown using a continuous gesture to draw the character 'w'. (b) TwoStick is a base-9 technique for the left stick and a base-8 technique for the right stick.

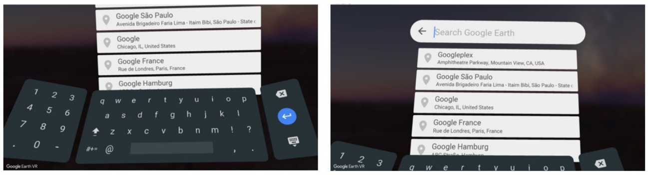

Fig. 3. The example case of Google Earth VR. When the user is entering the text using the keyboard, he or she might not be able to check the text entered. Also, when checking the entered text, the user would not be able to type on the aim-and-shoot keyboard.

(e.g. smartwatches [11, 56]).

According to our review, some existing circular keyboard layout techniques suffer from low text entry speed, especially for novice users. Most joysticks equipped on the game controller nowadays are bounded by a circular physical edge, which can provide constrained movements, and when taking advantage of this, it has been shown that sliding the joystick along the plastic edge can be easy and accurate [53]—in a similar way that EdgeWrite takes advantage of the edge boundaries to give feedback and constrain users' movements [54, 55]. From this review, we can also see that circular layout may provide benefits for joystick text entry, especially using a bimanual approach.

3 Design Rationale

We followed the design rationale below for designing an efficient joystick text entry method we call PizzaText for VR HMD.

3.1 Efficiency

Most joystick-based input methods suffer from low text entry speed. One important reason is that some techniques require a long series of finger movements to enter a character. For example, with the EdgeWrite alphabet [54], the user has to perform four end-to-end movements to enter the characters 'k', 'm', or 'w'. Keystrokes per character (KSPC) analysis is a simple method to quantify this movement. KSPC represents the number of keystrokes, on average, to generate each character of text in a given language using a given text entry technique [30]. As shown by MacKenzie et al. [32], if each of the four corners (base-4) of EdgeWrite is considered as a key press (see Fig. 2a), the KSPC of this technique is equal to 4.356. Although EdgeWrite is primarily designed for gesture input, relatively high KSPC causes the technique to potentially suffer from low text entry speed. The TwoStick technique [25] approach, using this analysis, can be considered as a base-9 text entry method for the left stick and base-8 text entry method for the right stick (see Fig. 2b). Its KSPC is nearly equal to 2. This seems to be beneficial for fast entry speed because intuitively it requires the joystick to travel shorter distances to input characters. However, there is a trade-off between the number of bases (number of keys) and KSPC for joystick input. As indicated by existing literature [18, 27], the thumb control of the joystick might not be that accurate when the number of base increases (and this is also proved by our current work). Increasing the number of keys controlled by a joystick with one movement might cause the users to input text with a higher error rate or spend more time on selection. In order to increase text entry speed using the joystick, one needs to design a technique with low KSPC while not sacrificing accuracy—this is one key motivation of our work.

3.2 Learnability

How well users can learn a new keyboard layout is a well-recognized problem [7, 22] and a large number of keyboard layouts that depart from the traditional QWERTY have been rejected because of the high learning effort involved [6, 8]. As such, it is crucial to consider how to reduce learning load for novice users so that they can quickly transition into expert users. Learning is a complex activity involving exploratory, sense-making, and trial-and-error practices. It is still currently not easy to model the user learning process accurately for a new keyboard layout of any shape [22]. Gong et al. [11] suggested that trade-offs might exist between efficiency and learnability. For example, gesture-based input techniques might be faster than some QWERTY selection-based techniques, but it will usually require a longer time for training to achieve a good level of performance. Some other works (e.g. [38]) encourage using users' past experience to bridge the gap between what they are used to and new layouts—which is useful especially for novice users. In our current work, since we are trying to use a circular layout, which would help to lower KSPC to strive for high-speed typing, it might be difficult to link the users' understanding of the QWERTY layout to the new circular layout in a physical sense. As a result, we decided to rely on the users' familiarity with the order of characters in the alphabet to transition them to the new layout.

3.3 Usability

One challenge for the current commercial VR HMD is the limited vertical field of view. Typically, it is 35° downward from the center of the display to the bottom; the human visual system is typically 75° downward from the nose [12]. Because of this, keyboards positioned at the lower part of the user's view may not be visible to the user using an HMD when he or she is looking horizontally straight ahead (e.g. to check the text entered, see Fig. 3). This may cause the user to do cross-checking movements regularly by rotating his or her head which could quickly result in tiredness, dizziness, and fatigue [58]. This might also increase the visual search time for the characters according to Fitt's law [29]. One solution to solve this is by using eyes-free text entry techniques such as mid-air handwriting [1, 26, 36]. Gesture-based techniques with a joystick might work well for eyes-free text entry. However, as discussed earlier, these techniques might require long series of movements which could impact negatively their efficiency. Because we want to create a text entry technique that is both efficient and useful, it should enable fast text entry speed regardless of the keyboard size. In other words, we want a technique whose keyboard can be scaled down to a relatively small size so that it can be placed near the text display location and within the users' field of view. As shown in [43], on touch-enabled devices, larger size keyboards will suffer from input speed, while small size keyboards will significantly increase the error rate. This might also be the case for the current common aim-and-shoot techniques or Google Drum Key like approaches. In short, we want to devise a technique which can be placed anywhere on the head-mounted display and can be scaled down into a relatively small size—both of these two features will help improve its usability.

We also thought to apply word-prediction algorithms to our technique to further increase its speed. However, current controller text entry methods in VE are often used for entering a username, password, billing information, or webspeak phrases. To enter this kind of text, a prediction algorithm might not be very helpful. In addition, as reported later, our technique can achieve a high WPM even for novice users without the use of any prediction algorithms.

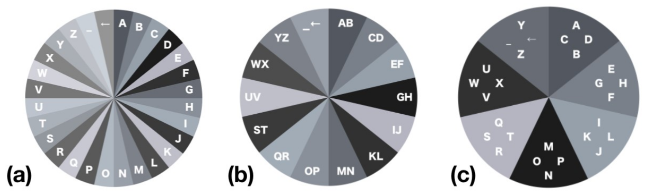

Fig. 4. The possible keyboard layouts of PizzaText: (a) 1-key PizzaText: one key per slice; (b) 2-key PizzaText: two keys per slice; (c) 4-key PizzaText: four keys per slice.

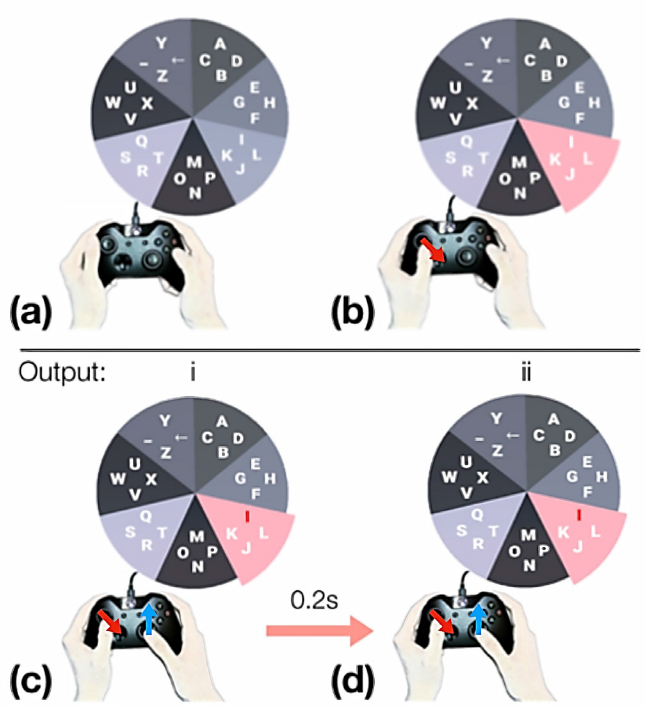

Fig. 5. The procedure for inputting the character ‘i’ using the 4-key PizzaText keyboard layout. The user first tilts the left stick to select the desired slice and then pushes the right stick to the location of the target key within the slice.

4 PizzaText

Following the three design rationales, we designed PizzaText, a circular keyboard layout that enables text entry using the two dual thumbsticks of a controller. In this keyboard, one character or several characters are arranged in one slice piece of the circle. In our first design stage, we derived three PizzaText keyboard layouts with one key, two keys, and four keys per slice (see Fig. 4). To select the characters, the user will have to use one joystick to tilt to the direction of the slice which contains the target character first. Once the desired slice is highlighted, the user needs to tilt the other joystick to the direction of the target key within the slice (or press the trigger in the case of the one key per slice design). Fig. 5 shows the procedure of entering the character ’i’ in the four keys per slice (4-key) keyboard. The user first tilts the left joystick towards the direction of the slice that contains the character ’i’ (in this case, it is from 12.86 degrees to 64.29 degrees with respect to the horizontal line). The user then pushes the right joystick to the upward direction (from 45 degrees to 135 degrees with respect to the horizontal line). The keyboard will output an ’i’ as a result. PizzaText supports continuous one-character text entry by holding the gesture for a short time. After a threshold (for this study, it is pre-defined to be 0.2 seconds), ’ii’ will be shown as an output. After inputting a character, a short audio feedback (typing sound) will be provided.

With KSPC = 2, the effort for users to enter text using PizzaText is relatively low; thus, they may be able to increase their entry speed quickly. In the 2-key and 4-key conditions, we decreased the required accuracy for the left thumbstick from the 1-key layout (which is 12.9 degree per keystroke). The slices are arranged in alphabetic order and the character(s) within each slice are designed to be intuitive for a native Latin alphabet user (one of the designer). Moreover, this keyboard can be scaled to different sizes since it only requires the user to push the joysticks to different directions and this property might be helpful for the limited vertical field of view in VR HMD.

5 User Study 1: Layout Comparison

As mentioned earlier, increasing the number of keys per slice in PizzaText might decrease the physical demand of the left thumb and transfer this workload to the right thumb. In this case, the right thumb can be used as a selection mechanism for the target characters instead of using a trigger like in the case of 1-key PizzaText (for example, in Fig. 5, the user tilts upward the right joystick to select the ‘i’ character in 4-key PizzaText). To make this selection accurate and fast, we set the maximum direction of the right joystick to be only four (up, down, left, right), since the previous work in [25] indicated that dividing the piece into more key dimensions might lead to the high error rate. Apart from this layout design, we also thought that there might be a difference between clockwise and counter-clockwise placement of the slices. In a preliminary study, we found that some users were good at clockwise PizzaText, while others were better in counter-clockwise PizzaText. Because of this, we wanted to find a general trend for this property to enhance its usability for most users. Thus, the goal of this first study was to investigate the performance of different keyboard layouts (1-key, 2-key, 4-key per slice) and the potential effect of the keyboard direction (clockwise or counter-clockwise).

5.1 Hypotheses

We formulated three hypotheses for the experiment:

H.1. *The 4-key layout will have higher text entry speed than 1-key and 2-key layouts.* Since 1-key and 2-key layouts require more motions of the left joystick to locate the correct slice, users may need more time to enter the text sentences.

H.2. *The clockwise layout will induce faster text entry speed than counter-clockwise layout.* We made this hypothesis based on our observation in the preliminary study.

H.3. *The 4-key and 2-key layouts will have lower NASA-TLX workload than 1-key layout.* This was because the 4-key and 2-key layouts would reduce the workload of the left thumb but would not increase the workload of the right thumb by much.

5.2 Participants

Eighteen participants (4 females; 14 males) between the ages of 19-28 (M=21) were recruited from a local university campus to take part in this study. According to our pre-experiment questionnaire, 12 participants had some limited experience with VR; and of these participants, 3 of them were native Latin alphabet users.

5.3 Apparatus and Materials

The experiment was conducted on an Intel Core i7 processor PC with a dedicated NVIDIA GTX 1080 Ti graphics card. The program was developed using C#.NET and was run in the Unity3D platform.

Fig. 6 shows the devices we used in this experiment. Fig. 6a shows the Oculus RIFT CV1, an HMD VR device that completely immerses the user into the 3D virtual world and allows the user to look at any direction. Fig. 6b shows the Xbox One controller. It contains joysticks, triggers, bumpers, buttons, and a direction pad. In this study, we used the two joysticks and the right trigger for selection in the 1-key layout.

5.4 Experiment Design and Procedure

The whole experiment lasted approximately 80 minutes for each participant. Before the trials started the participants were asked to fill in a pre-experiment questionnaire to gather their demographic information

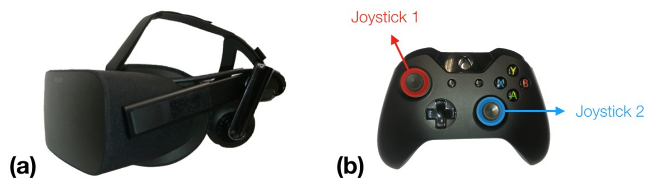

Fig. 6. The devices used in the experiment: (a) the Oculus RIFT virtual reality head-mounted display; and (b) the Xbox One controller.

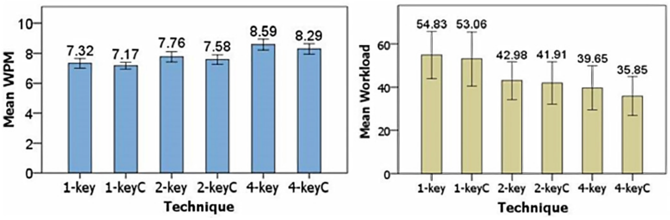

Fig. 7. Mean text entry speed and NASA-TLX workload (lower is better) across six designs of PizzaText. Error bars indicate ±2 standard errors.

and were then given time to get familiar with the VE and the Xbox controller. After this initial stage, they would proceed to carry out the experiment sessions. In each session, participants would first complete a practice trial with the current layout being evaluated and then type 10 phrases, which were randomly generated from the MacKenzie phrase set [31]. After each session, participants were asked to complete the NASA-TLX questionnaire [16] for the current layout. In all, the participants had to finish 6 experimental sessions in total (3 Techniques × 2 Directions). We instructed the participants to enter the texts "as quickly and accurately as possible". After the experiment, we asked the participants to give some comments on the designs.

The study used a 3 × 2 within-subjects design with two factors: Layouts (1-key, 2-key, and 4-key per slice) and Direction (clockwise and counter-clockwise). We fully counterbalanced the order of the techniques and directions. For the whole experiment, we gathered 3 (layout) × 2 (direction) × 10 (phrase) × 18 (participant) = 1080 timed trials.

5.5 Results

In this work, the text entry rate is measured by Words per Minute (WPM) using the following formula

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

where $|S|$ is the length of the transcribed string and T is the task completion time in seconds. The task completion time was recorded as the time elapsed from when the first letter is selected using the joystick to the end of the trail. The error rate is reported based on total error rate (TER) and not corrected error rate (NCER) [47]. Uncorrected errors were the errors found left in the final transcribed text and total errors include both uncorrected errors and corrected errors.

We used a two-way repeated measures ANOVA and Bonferroni corrections for pair-wise comparisons. We also used a Greenhouse-Geisser adjustment to correct for violations of the sphericity assumption.

5.5.1 Text Entry Speed

There were significant effects of Layout ($F_{1.99,33.76} = 56.586, p = 6.059 \times 10^{-22}$) and Direction ($F_{1,17} = 5.887, p = .016$) on WPM. No significant interaction effect on Layout × Direction ($F_{1.90,32.23} = .214, p = .796$) was found. The pairwise comparisons indicated that 4-key layout was significantly faster than the other two layouts (1-key: $p = 7.780 \times 10^{-19}$ and 2-key: $p = 7.045 \times 10^{-4}$) and also that the 2-key layout was significantly faster than 1-key layout ($p = 1.491 \times 10^{-10}$).

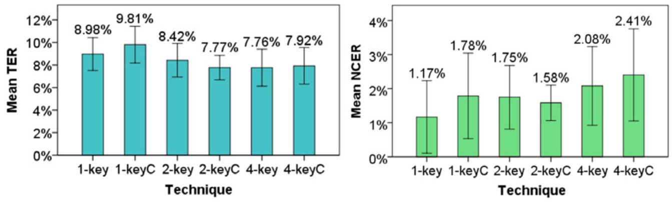

Fig. 8. Mean total error rate (TER) and not corrected error rate (NCER) across six types of PizzaText designs. Error bars indicate ±2 standard errors.

Fig. 7 shows the mean text entry speed (left figure) for the six layouts. Overall, without considering the direction, the average text entry speed for 1-key, 2-key, and 4-key layouts was 7.25 WPM (s.e. = 0.12), 7.67 WPM (s.e. = 0.15), and 8.44 WPM (s.e. = 0.16) respectively for novice users.

5.5.2 Total and Uncorrected Error Rate

Fig. 8 shows the TER (left figure) and NCER (right figure) across six types of PizzaText layouts.

For TER, ANOVA yielded no significant effects of Layout ($F_{1.91,32.40} = 2.804, p = .065$) and Direction ($F_{1,17} = .049, p = .825$). Also, no significant interaction effect on Layout × Direction ($F_{1.96,33.30} = .493, p = .607$) were found.

For NCER, we found no significant effects of Layout ($F_{1.80,30.52} = 1.114, p = .325$) and Direction ($F_{1,17} = .364, p = .547$) on NCER. No significant interaction effect was found on Layout × Direction ($F_{1.89,32.18} = .259, p = .760$) either.

5.5.3 Work Load

Fig. 7 shows the NASA-TLX workload scores (right figure) across the six designs (lower is better).

ANOVA results indicated that Layout ($F_{1.48,25.14} = 8.033, p = .001$) exhibited a significant main effect on workload, but Direction ($F_{1,17} = .836, p = .373$) did not. No significant interaction effect between Layout and Direction was observed ($F_{1.86,31.59} = 19.282, p = .843$). The pairwise comparisons revealed that 4-key and 2-key layouts require much less workload than 1-key layout ($p = .020$). No significant difference was found between 4-key and 2-key layouts ($p = .503$).

5.5.4 Native vs. Non-Native Latin alphabet users

After collecting data for the first few participants, we noticed that there could be a difference between the participants who were native Latin alphabet users and those who were not. The average text entry speed for native users was 9.71 WPM (s.e. = 0.17) while for and non-native users 7.40 WPM (s.e. = 0.07). Native users got 10.68% in TER (s.e. = 0.98%) and 3.25% in NCER (s.e. = 0.88%) on average, while non-native users got 8.00% in TER (s.e. = 0.31%) and 1.50% in NCER (s.e. = 0.20%).

We thought this might due to the two groups having different familiarity with the alphabet and this could lead to different text entry speed. This may also indicate that PizzaText could support higher performance if the users are Native Latin alphabet users.

5.6 Discussion

The study results offer strong evidence to support our hypotheses **H.1**, **H.2**, and **H.3**.

By decreasing the base number of the left joystick (1-key: 28-base, 2-key: 14-base, and 4-key: 7-base) and transferring some workload to the right joystick (1-key: 0-base; 2-key 2-base; 4-key: 4-base), the text entry speed increased significantly (**H.1**). Moreover, this also helped to reduce the workload for users (**H.2**). We also found that most users were more accustomed to the clockwise placement of the slices (**H.3**) and this kind of familiarity might have helped the participants to transition to the circular layout faster and learn how to use it quickly.

In all, according to our experimental results, the 4-key design of

PizzaText with the clockwise placement of the slices might have the largest potential to be taken into real usage and might require a further investigation to evaluate its performance.

6 User Study 2: Performance Evaluation

We conducted a five-day user study to evaluate the performance of the 4-key PizzaText (the best performing design found in the first study). Participants were recruited from the first experiment using the following method. We ordered the participants based on the average text entry speed they got in the first study and then divided them into two groups using the median speed as the threshold. We picked five participants from each group and asked them whether they would like to continue for a 5-day experiment to form two new groups for this second study. As a result, one group had relatively low text entry speed in the first experiment (we called this the 'novice' group), and the other group had relatively high text entry speed (we called this the potential 'expert' group). The goal of this study was to evaluate how well users could perform text entry using 4-key PizzaText. We were also interested in knowing how the two groups' performance would improve throughout the 5 days with short periods of practice.

6.1 Participants, Apparatus, and Materials

Ten participants (all males) from the first experiment are involved in this study and were aged between 19-28 (M=22) years old. We used the same apparatus and devices as in the first experiment.

6.2 Experiment Design and Procedure

The whole experiment contains a series of sessions, with one session occurring in each day. For each session, the participants would type 10 phrases, which were randomly generated from [31]. Prior to the experiment, participants were allowed to practice for as long as they wanted. Also, just like in the last experiment, we instructed the participants to enter the texts "as quickly and accurately as possible". The whole procedure took participants around 20 minutes to complete, depending on the speed they entered the text phrases and the time they spent practicing. In all, we collected 5 (participant) × 2 (group) × 5 (session) × 10 (phrase) = 500 phrases.

6.3 Results

We employed a mix-design ANOVA with Sessions (from day one to day five) as the within-subject variable and Group (novice group and potential expert group) as the between-subjects variable. Bonferroni correction was used for pair-wise comparisons and Greenhouse-Geisser adjustment was used for degrees of freedom for violations to sphericity.

6.3.1 Text Entry Speed

ANOVA tests yielded a significant effect of Session ($F_{3.67,33.06} = 69.374, p = 5.801 \times 10^{-41}$), but not Session × Group ($F_{3.67,33.06} = 0.978, p = .419$) on text entry speed. There was a significant effect of Group ($F_{1,9} = 119.368, p = 1.200 \times 10^{-18}$) on text entry speed. This meant that although participants in the two groups had a significant difference in text entry speed, their learning process was somewhat similar.

Post-hoc pair-wise comparisons revealed significant differences between Session 1 and 2 ($p = 7.429 \times 10^{-4}$), Session 1 and 3 ($p = 1.626 \times 10^{-13}$), and Session 2 and 3 ($p = 4.922 \times 10^{-7}$) . No significant differences were found between Session 3 and Session 4 ($p = .121$), nor between Session 4 and Session 5 ($p = .530$). However, significant differences were found between Session 3 and Session 5 ($p = 5.376 \times 10^{-4}$). This trend shows that the learning curve was getting more and more stable after each session.

Overall, the average text speed across all tested conditions was 12.26 WPM (s.e. = 0.15). In particular, the novice group achieved 10.20 WPM (s.e. = 0.27), while the potential expert group achieved 14.32 WPM (s.e. = 0.27). Fig. 9 shows the WPM by session/day for each participant and the two groups. The average speed for the first session was 10.40 WPM (s.e. = 0.22); it bumped up to 13.77 WPM (s.e. = 0.25) in the last session, with an increase of 32.4%.

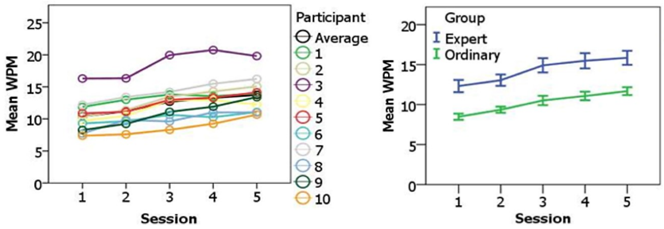

Fig. 9. Text entry speed over 5 days for each participant (left) and the average for each group (right). Error bars indicate ±2 standard errors.

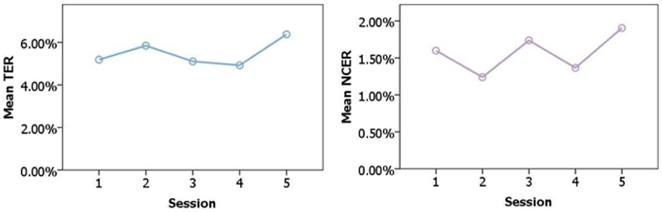

Fig. 10. Mean total error rate (TER) and not corrected error rate (NCER) over 5 days.

In the last session, the potential expert group improved their performance to 15.85 WPM (s.e. = 0.36) from the first session of 12.32 WPM (s.e. = 0.30); while the novice group improved to 11.68 WPM (s.e. = 0.36) from the first session of 8.48 WPM (s.e. = 0.30). The highest speed was 20.76 WPM and was achieved in Session 4 by one participant in the potential expert group. This participant (#3 in Fig. 9) also achieved a very high speed (averaging 18.63 WPM) using PizzaText over 5 days.

6.3.2 Total and Uncorrected Error Rate

For TER, ANOVA tests yielded no significant effects of Session ($F_{3.50,31.52} = .962, p = .420$), Group ($F_{1,9} = 1.404, p = .239$), or Session × Group ($F_{3.50,31.52} = .864, p = .474$). For NCER, ANOVA yielded no significant effects of Session ($F_{3.10,27.94} = .364, p = .786$), Group ($F_{1,9} = .005, p = .943$), or Session × Group ($F_{3.10,27.94} = 1.995, p = .113$).

Overall, the average TER and NCER across all study conditions were 5.49% (s.e. = 0.30%) and 1.57% (s.e. = 0.21%) respectively. In particular, the average TER and NCER for the potential expert group were 5.08% (s.e. = 0.50%) and 1.59% (s.e. = 0.31%), whereas for the novice group they were 5.91% (s.e. = 0.50%) and 1.56% (s.e. = 0.31%). Fig. 10 shows the TER and NCER over 5 days. As expected, over the 5-day training period, the increased in text entry speed did not significantly influence the error rate.

6.3.3 Native vs. Non-Native Latin alphabet users

As reported in the first study section, we noticed that there was a difference between native and non-native alphabet users in terms of text entry speed. We further explored if this gap still existed with further training. In this study, two native Latin alphabet users were both in the expert group.

Overall, the native and non-native users achieved 13.78 WPM (s.e. = 0.22) and 11.89 WPM (s.e. = 0.18) on average across five sessions, respectively. In the last session, the native users reached 15.10 WPM (s.e. = 0.58), while the non-native users reached 13.43 WPM (s.e. = 0.37). This would seem to suggest that there still remained a gap between native and non-native Latin alphabet users even after the training period. This would also indicate that users native to the Latin alphabet might be able to achieve better performance overall using PizzaText.

Fig. 11. (a) The relative size of PizzaText scaled to 80%, 60%, and 40%; (b) an envisioned scenario of using a reduced size version of PizzaText that would take little space and could be placed in an inconspicuous region of the display.

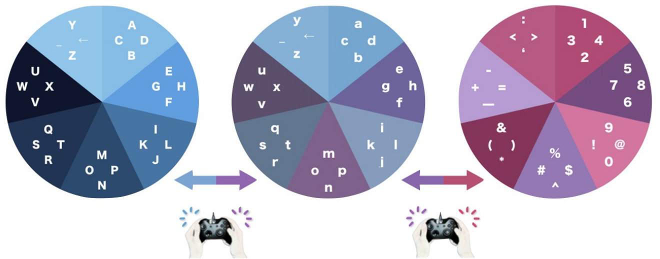

Fig. 12. A potential extension of PizzaText to support numbers, lower-case/uppercase characters, and special symbols.

6.4 Is PizzaText sensitive to size?

We conducted a small, follow-up experiment (with 3 participants from the second study) to test if the size of PizzaText would significantly influence performance. We scaled the size to 80%, 60%, and 40% of the original size and counterbalanced the order of its presentation (see Fig. 11a). Note that scaling the size to 25% and below would make the characters on the keyboard become too small to see clearly. We collected 3 participants $\times$ 3 scales $\times$ 10 phrases = 90 timed trials.

According to our result, the average text entry speed of 40%, 60%, and 80% were 12.94, 13.03, 13.37 WPM, respectively. This would suggest that scaling PizzaText to a smaller size might not cause a significant drop in performance. It seems that the size of PizzaText could probably be reduced to 40% of the original size. Fig. 11b shows an envisioned scenario of using PizzaText for text entry in a VE.

6.5 Discussion and Future Work

In terms of efficiency, the average speed of 4-key PizzaText for novice users is 8.59 WPM and 15.85 WPM for expert users after around only two more hours of training (taking the unlimited training time into account). This result indicates that PizzaText outperforms the current joystick text entry techniques such as EdgeWrite [54], Dual QWERTY [52], TwoStick [25] with 6.4 WPM, 7.1 WPM, 5.1 WPM for novice users, respectively. Long-term practice with Quikwriting [21], EdgeWrite [54], and TwoStick [25] led to 13 WPM, 11-14.7 WPM, 14.9 WPM. The test conditions are not exactly the same for direct comparison, however. For one, we have tested our technique in a VR environment, which was new for some participants, and this might have affected their text entry speed. Also, most of our participants were non-native Latin alphabet users; this might have influenced negatively their text entry speed. Results from both studies indicate that the native Latin alphabet users using PizzaText have had better performance than those who were not native.

With respect to learnability, we believe PizzaText was easy to learn. Without much practice, novice users could achieve a relatively high text entry speed (8.59 WPM) in a VE using the joystick as input. After around two more hours of training only, participants in the potential expert group were able to achieve 15.85 WPM. This training time seemed much shorter than other new layouts such as Quikwriting [21].

As stated, PizzaText only requires users to tilt the joysticks to a certain direction to select characters. Because of this, the performance should not be affected significantly by the size of the keyboard. Smaller sizes of PizzaText were tested in a small follow-up experiment and the results indicated PizzaText still worked well in relatively smaller sizes (even with only 40% of the original size). This suggests that PizzaText is more usable in current virtual/augmented reality HMD which have a relatively small field of view.

This research only considers lowercase characters text entry in order to compare with some other baseline techniques. However, since passwords can contain uppercase characters and other special symbols, future research could explore how PizzaText would scale to support those characters. For example, it is possible to have a mode switch mechanism (like using either the right or left trigger) to act in the same manner as the Shift key to allow typing uppercase characters. Similarly, pressing the other trigger that is not linked to an action could switch and activate another auxiliary pizza to allow selecting special characters and numbers (see Fig. 12).

In addition, we believed that PizzaText could not only be well suited for the Xbox controller used in this study but could also work with other controllers with dual joysticks such as the Oculus Touch controller and the Lenovo Explorer controller—though some further evaluation might be needed to verify this.

7 Conclusion

In this paper, we have presented PizzaText, a text entry technique for virtual reality environments using the dual thumbsticks of a game controller. The technique allows the user to control the two joysticks to select target characters on a circular keyboard layout. Its design is guided by three design criteria: efficient performance, ease-of-learning, and ease-of-use. Two user studies are conducted to identify the best design features of the technique first and then to evaluate its performance and usability. The first study compares several potential layouts of PizzaText to inform its final design. The second study is a 5-day experiment with 10 participants to further evaluate the final layout of PizzaText with 7 slices and 4 characters per slice. The results of the two studies show that novice users can achieve 8.59 Words per Minute (WPM) while the expert users can achieve 15.85 WPM after having only around 2 hours more of training using the 4-key PizzaText. We believe that the PizzaText is an efficient, easy to learn, and easy to use text entry technique that can be used for a wide range of virtual reality head-mounted displays and applications

Acknowledgments

The authors wish to thank the participants for their time and the reviewers for their comments and feedback that have helped us improve our paper. This work was supported in part by XJTLU Key Program Special Fund and XJTLU Research Development Fund.

References

[1] C. Amma, M. Georgi, and T. Schultz. Airwriting: Hands-free mobile text input by spotting and continuous recognition of 3d-space handwriting with inertial sensors. In *Wearable Computers (ISWC), 2012 16th International Symposium on*, pp. 52–59. IEEE, 2012.

[2] B. Benligiray, C. Topal, and C. Akinlar. Slicetype: Fast gaze typing with a merging keyboard. *arXiv preprint arXiv:1706.02499*, 2017.

[3] D. Bowman, E. Kruijff, J. J. LaViola Jr, and I. P. Poupyrev. *3D User interfaces: theory and practice*. Addison-Wesley, 2004.

[4] D. A. Bowman, C. J. Rhoton, and M. S. Pinho. Text input techniques for immersive virtual environments: An empirical comparison. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 46, pp. 2154–2158. SAGE Publications Sage CA: Los Angeles, CA, 2002.

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

[6] P. A. David. Clio and the economics of qwerty. *The American economic review*, 75(2):332–337, 1985.

[7] M. Dunlop and J. Levine. Multidimensional pareto optimization of touch-screen keyboards for speed, familiarity and improved spell checking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2669–2678. ACM, 2012.

[8] A. Dvorak. There is a better typewriter keyboard. *National Business Education Quarterly*, 12(2):51–58, 1943.

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

[10] K. Go, H. Konishi, and Y. Matsuura. Itone: a japanese text input method for a dual joystick game controller. In *CHI'08 Extended Abstracts on Human Factors in Computing Systems*, pp. 3141–3146. ACM, 2008.

[11] J. Gong, Z. Xu, Q. Guo, T. Seyed, X. Chen, X. Bi, and X.-D. Yang. Wristext: One-handed text entry on smartwatch using wrist gestures. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, p. 181. ACM, 2018.

[12] 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. *arXiv preprint arXiv:1802.00626*, 2018.

[13] Z. Gu, C. Chu, X. Xu, and Z. Dong. Feature stroke: A text entry method using joystick. 2015.

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

[15] J. Gugenheimer, D. Dobbelstein, C. Winkler, G. Haas, and E. Rukzio. Facetouch: Enabling touch interaction in display fixed uis for mobile virtual reality. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pp. 49–60. ACM, 2016.

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

[17] U. Hinrichs, H. Schmidt, T. Isenberg, M. S. Hancock, and S. Carpendale. Bubbletype: Enabling text entry within a walk-up tabletop installation. 2008.

[18] N. Hirotaka. Reassessing current cell phone designs: using thumb input effectively. In *CHI'03 Extended Abstracts on Human Factors in Computing Systems*, pp. 938–939. ACM, 2003.

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

[20] P. Isokoski and R. Raisamo. Device independent text input: A rationale and an example. In *Proceedings of the working conference on Advanced visual interfaces*, pp. 76–83. ACM, 2000.

[21] P. Isokoski and R. Raisamo. Quikwriting as a multi-device text entry method. In *Proceedings of the third Nordic conference on Human-computer interaction*, pp. 105–108. ACM, 2004.

[22] J. P. Jokinen, S. Sarcar, A. Oulasvirta, C. Silpasuwanchai, Z. Wang, and X. Ren. Modelling learning of new keyboard layouts. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 4203–4215. ACM, 2017.

[23] H. Kim and G. Lee. Korean edgewrite: A korean text entry method for a joystick. *Proc. TriSAI*, 2008.

[24] Y. R. Kim and G. J. Kim. Hovr-type: Smartphone as a typing interface in VR using hovering. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*, pp. 333–334. ACM, 2016.

[25] T. Költringer, P. Isokoski, and T. Grechenig. Twostick: writing with a game controller. In *Proceedings of Graphics Interface 2007*, pp. 103–110. ACM, 2007.

[26] P. O. Kristensson, T. Nicholson, and A. Quigley. Continuous recognition of one-handed and two-handed gestures using 3d full-body motion tracking sensors. In *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*, pp. 89–92. ACM, 2012.

[27] S. Kurniawan, A. King, D. G. Evans, and P. Blenkhorn. Design and user evaluation of a joystick-operated full-screen magnifier. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 25–32. ACM, 2003.

[28] Y. Lu, C. Yu, X. Yi, Y. Shi, and S. Zhao. Blindtype: Eyes-free text entry on handheld touchpad by leveraging thumb's muscle memory. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(2):18, 2017.

[29] I. S. MacKenzie. Fitts' law as a research and design tool in human-computer interaction. *Human-computer interaction*, 7(1):91–139, 1992.

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

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

[32] I. S. MacKenzie, R. W. Soukoreff, and J. Helga. 1 thumb, 4 buttons, 20 words per minute: Design and evaluation of h4-writer. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pp. 471–480. ACM, 2011.

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

[34] M. McGill, D. Boland, R. Murray-Smith, and S. Brewster. A dose of reality: Overcoming usability challenges in vr head-mounted displays. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pp. 2143–2152. ACM, 2015.

[35] A. K. Mithal and S. A. Douglas. Differences in movement microstructure of the mouse and the finger-controlled isometric joystick. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 300–307. ACM, 1996.

[36] T. Ni, D. Bowman, and C. North. Airstroke: bringing unistroke text entry to freehand gesture interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2473–2476. ACM, 2011.

[37] J. Olofsson. Input and display of text for virtual reality head-mounted displays and hand-held positionally tracked controllers, 2017.

[38] D. N. Perkins and G. Salomon. Transfer of learning. *International encyclopedia of education*, 2:6452–6457, 1992.

[39] 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 *3D User Interfaces (3DUI), 2016 IEEE Symposium on*, pp. 109–112. IEEE, 2016.

[40] I. Poupyrev, N. Tomokazu, and S. Weghorst. Virtual notepad: handwriting in immersive vr. In *Virtual Reality Annual International Symposium, 1998. Proceedings., IEEE 1998*, pp. 126–132. IEEE, 1998.

[41] M. Pratorius, U. Burgbacher, D. Valkov, and K. Hinrichs. Sensing thumb-to-finger taps for symbolic input in vr/ar environments. *IEEE computer graphics and applications*, 2015.

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

[43] A. Rodrigues, H. Nicolau, K. Montague, L. Carriço, and T. Guerreiro. Effect of target size on non-visual text-entry. In *Proceedings of the 18th International conference on human-computer interaction with mobile devices and services*, pp. 47–52. ACM, 2016.

[44] F. E. Sandnes and A. Aubert. Bimanual text entry using game controllers: Relying on users' spatial familiarity with qwerty. *Interacting with Computers*, 19(2):140–150, 2006.

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

[46] G. Shoemaker, L. Findlater, J. Q. Dawson, and K. S. Booth. Mid-air text input techniques for very large wall displays. In *Proceedings of Graphics interface 2009*, pp. 231–238. Canadian Information Processing Society, 2009.

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

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

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

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

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

[52] A. D. Wilson and M. Agrawala. Text entry using a dual joystick game controller. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pp. 475–478. ACM, 2006.

[53] J. Wobbrock. The benefits of physical edges in gesture-making: Empirical support for an edge-based unistroke alphabet. In *CHI'03 Extended Abstracts on Human Factors in Computing Systems*, pp. 942–943. ACM, 2003.

[54] J. O. Wobbrock, B. A. Myers, and H. H. Aung. Writing with a joystick: a comparison of date stamp, selection keyboard, and edgewrite. In *Proceedings of Graphics Interface 2004*, pp. 1–8. Canadian Human-Computer Communications Society, 2004.

[55] J. O. Wobbrock, B. A. Myers, and S. E. Hudson. Exploring edge-based input techniques for handheld text entry. In *Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on*, pp. 280–282. IEEE, 2003.

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

[57] X. Yi, C. Yu, M. Zhang, S. Gao, K. Sun, and Y. Shi. Atk: Enabling ten-finger freehand typing in air based on 3d hand tracking data. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pp. 539–548. ACM, 2015.

[58] C. Yu, Y. Gu, Z. Yang, X. Yi, H. Luo, and Y. Shi. Tap, dwell or gesture?: Exploring head-based text entry techniques for hmds. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pp. 4479–4488. ACM, 2017.