H4VR: One-handed Gesture-based Text Entry in Virtual Reality Using a Four-key Keyboard

Saba Fallah
sabaflh@my.yorku.ca
Dept. of Electrical Engineering and Computer Science, York University
Toronto, Ontario, Canada

I. Scott MacKenzie
mack@yorku.ca
Dept. of Electrical Engineering and Computer Science, York University
Toronto, Ontario, Canada

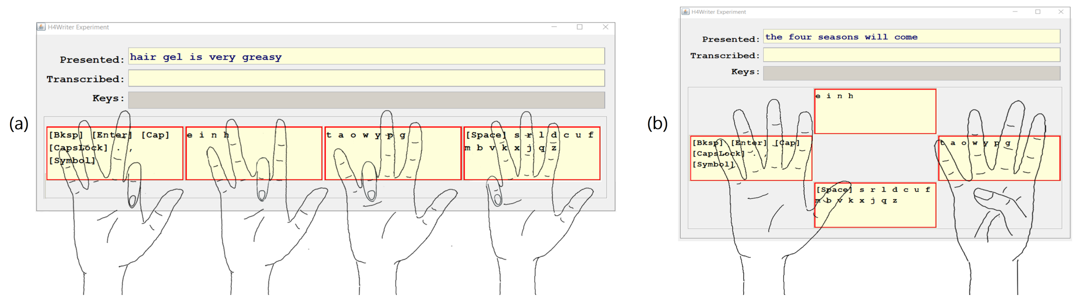


Figure 1: H4VR keyboards and selection methods. (a) Flat layout and code-based selection. (b) Cross layout and virtual mouse selection.

ABSTRACT

H4VR is a text entry method offering two layouts – flat and cross – of a four-key keyboard in VR. Hand gestures select the keys with a different selection method for each layout. The entry speed and error rate were measured and analyzed in a five-session longitudinal user study. Five participants entered ten phrases on each keyboard in each session. They reached an average entry speed of 4.57 and 3.63 words per minute on the flat and cross keyboards, respectively. The average error rate was less than 2% on both keyboards.

CCS CONCEPTS

• **Human-centered computing** → **Empirical studies in HCI**; **User studies**.

KEYWORDS

Gesture-based text entry, Virtual reality, Huffman codes, H4-Writer

ACM Reference Format:
Saba Fallah and I. Scott MacKenzie. 2023. H4VR: One-handed Gesture-based Text Entry in Virtual Reality Using a Four-key Keyboard. In *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems (CHI EA '23), April 23–28, 2023, Hamburg, Germany*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3544549.3585876

CHI EA '23, April 23–28, 2023, Hamburg, Germany

ACM ISBN 978-1-4503-9422-2/23/04.
https://doi.org/10.1145/3544549.3585876

1 INTRODUCTION

Since its emergence in the 1970s [9], virtual reality (VR) has been the focus of considerable research. Rapid and continuous improvements in VR have led to extensive commercial use, with applications in gaming, learning, design, marketing, group meetings, etc.. With these applications, an inevitable task in VR is text entry.

Text entry is one of the main interaction methods in computing. Various methods have been developed and evaluated on desktop environments, but less research has occurred in a virtual setting. Typically in VR, text entry uses a Qwerty soft keyboard and hand-held controllers [16, 20]. Using hand-held devices such as controllers might be difficult for some users, especially the ones with motor disabilities [17]. An alternative is hand wearables, such as gloves [15]. However, hand wearables can cause discomfort and are costly. Another possibility is hand gesture recognition using a camera [15]. This is the focus of the present research.

While most text entry methods in VR rely on both hands, one-handed entry is also possible. Some current one-handed text entry methods in VR rely on point-select procedures. The user moves a pointer to the desired key and performs a select operation [18, 19]. Another method is to swipe a finger or a controller across a soft keyboard to make a unistroke gesture [3]. These methods involve more extensive hand motions, which require more space and can cause hand fatigue [10]. One-handed text entry in VR is also done by hand-gesture recognition, where gestures are assigned to characters [2, 8]. Remembering hand gestures and their assigned characters can affect learning time and entry speed.

A Qwerty keyboard has 26 keys just for the English alphabet. Each key must be sized to allow for easy and accurate selection. A keyboard that offers keys with more than one letter per key, also known as an ambiguous keyboard, presents fewer keys and has the

Space 44	e 22	t 33	a 34	o 31	i 24	n 23	s 42	h 21	r 433
l 411	d 432	c 414	u 413	f 412	m 434	w 324	y 323	p 322	g 321
b 4314	v 4313	k 4312	x 43114	j 43113	q 43112	z 43111			

Figure 2: H4VR character codes generated from base-4 Huffman codes.

benefit of occluding less space. An ambiguous keyboard can also rely on fewer hand gestures since there are fewer keys to navigate.

Our proposed method is for gesture-based one-handed text entry in VR. H4VR is a four-key soft keyboard based on Huffman codes [5]. Although not yet studied in VR, earlier research using this keyboard achieved entry speeds of 20.4 wpm using a game controller [14], 14.0 wpm using finger switches in wearable computing [1], and 15.0 wpm using finger taps on a smartphone [12].

With H4VR, base-4 Huffman encoding generates non-redundant codes for the English alphabet using the character frequency distribution of a corpus. The codes consist of four digits (nominally, '1', '2', '3', and '4'). See Figure 2. Frequent characters (e.g., 'e' and 't') are assigned shorter codes, whereas less frequent characters (e.g., 'q' and 'z') are given longer codes. Note in Figure 2 that the codes for the nine most-common characters (top row) have length two. By using Huffman coding, the layout of keys on an ambiguous keyboard can be optimized based on the frequency of use of the character. Huffman encoding helps to reduce the number of keys on the ambiguous keyboard while minimizing the number of selections required to choose a character.

An analytic metric to capture the efficiency of a text entry method is KSPC for keystrokes per character [11]. The equivalent here is GPC for gestures per character. GPC is the average number of gestures to enter a character of text in a given language using a given input method. It is a weighted average computed using a letter-frequency list from a corpus. For English and the 27 symbols in Figure 2, GPC = 2.32 for H4VR. No code is a prefix to another code, meaning delimiting actions between codes is unnecessary. The code set is easily extended to include digits, punctuation, symbols, and commands. This paper evaluates the entry speed and error rate of the keyboard layouts offered by H4VR through a longitudinal user study.

2 RELATED WORK

This section reviews related research on text entry techniques in VR and gesture-based text entry. Table 1 outlines the evaluated techniques based on their reliance on VR controllers, the number of hands engaged, keyboard type, the design of the experiment, and the observed results of text entry speed and error rate.

2.1 Text Entry in VR using Qwerty Keyboard

QWERTY keyboards have been adopted as a primary keyboard layout in VR. They leverage the user's familiarity with this layout to deliver an efficient text entry method. The following studies have examined the performance of QWERTY keyboards in VR using both controllers and hand gestures.

2.1.1 Controller-based Text Entry. Speicher et al. evaluated six text entry methods in VR on a Qwerty soft keyboard [16]: head pointing, controller pointing, controller tapping, freehand, discrete, and continuous cursor. They report that the controller pointing method using two handheld controllers to select keys had the best performance with an entry speed of 15.4 wpm. The freehand method using fingers on both hands to type on a virtual keyboard showed an entry speed of 9.77 wpm.

Xu et al. evaluated four one-handed entry methods on a head-mounted display (HMD) using a Qwerty soft keyboard [18]. The methods used a controller, hand, head, and a hybrid of hand and head to select keys. The controller method used ray-casting with a hand-held VR controller, whereby the user cast the ray onto the desired key. A button on the controller was pressed to select the key. This technique resulted in an entry speed of nearly 15 wpm.

2.1.2 Gesture-based Text Entry. In the same study done by Xu et al., the hand method used gesture-based text entry [18]. The user moved their hand to a key, using a closed-palm gesture to select the key. The hand method resulted in an entry speed of about 7 wpm. An error rate of less than 2% was reported for the controller and the hand techniques.

RotoSwype by Gupta et al. [4] allows for swipe typing in VR on a Qwerty soft keyboard with word prediction. A ring equipped with sensors is worn on the index finger to capture the movements and rotation of the user's hand. The ring contains a button used for word selection from the word prediction list. Sixteen participants were examined in a five-day study, where they typed 20 phrases each day. The study shows an entry speed of about 14 wpm with an average error rate of 1% on the last day of the study. Swipe typing leans on hand and wrist mobility which can cause hand fatigue.

2.2 Text Entry in VR using Ambiguous Keyboard

Ambiguous keyboards have not been extensively explored in VR. They occlude less space on the display while presenting fewer selection targets which can make them a favourable keyboard in VR. The following studies have examined the performance of ambiguous keyboards with various designs in VR using both controllers and hand gestures.

2.2.1 Controller-based Text Entry. PizzaText by Yu et al. [20] is a round ambiguous keyboard with seven slices appeared on the VR display, with four characters on each. Through thumbsticks on a game controller, the user selects the slice bearing the desired character. Ten participants in a five-day study typed ten phrases each day. The results showed an average speed of 8.59 wpm to 15.9 wpm depending on the users' expertise. The authors reported an error rate of 1.8%.

2.2.2 Gesture-based Text Entry. Jiang et al.'s HiFinger [7] employs a six-key ambiguous keyboard in VR and allows for one-handed wearable text entry. The characters are assigned a two-digit code from the digits 1 to 6. The user moves their thumb toward sensors placed on their fingers, selecting the code corresponding to a character. A three-day study with nine participants resulted in an average text entry speed of 9.82 wpm after 25 minutes of training.

Table 1: Comparison of Text Entry Research in Virtual Reality (VR)

Input Method	Controller-based	Hands	Entry Speed (wpm)	Error Rate (%)	Keyboard	Longitudinal
Controller pointing [16]	✓	2	15.44	0.97	Qwerty	✗
Freehand [16]	✗	2	9.77	7.57	Qwerty	✗
Hand [18]	✗	1	7	2	Qwerty	✗
Controller pointing [18]	✓	1	15	1.8	Qwerty	✗
PizzaText [20]	✓	2	12.26	1.8	Ambiguous - 7 keys	5 × 10 phrases
HiFinger [7]	✗	1	9.82	6.03	Ambiguous - 6 keys	3 × 25 min
PinchText [6]	✗	1	12.71	2	Ambiguous - 12 keys	6 × 10 phrases
RotoSwype [4]	✗	1	14	1	Qwerty	5 × 20 phrases

Jiang et al.'s PinchText [6] offers one-handed text entry in VR using pinch gestures and hand positions. It uses an ambiguous keyboard with 12 keys. The position of the user's hand determines the key set (i.e., one of three rows of keys) and different pinch gestures choose specific keys in the selected key set. A tracking system and a sensor detect the hand position. Conductive tape is used on five fingers to detect the pinch gesture. A six-block experiment shows an average entry speed of 12.7 wpm and 11.1 wpm for hand-up vertical (UpV) and hand-down vertical (DownV), respectively. PinchText requires moving the arm to position the hand to select a key set. This rapid arm movement can cause arm fatigue and needs space to perform the gestures.

Since limited research has been conducted on gesture-based text entry in VR, one of the goals of this study is to address this significant opportunity for further exploration and examination.

3 H4VR DESIGN

H4VR is a four-key soft keyboard based on Huffman codes in VR that allows for one-handed gesture-based text entry. It employs two layouts, *flat* and *cross*, each associated with its unique set of hand gestures for key selection. The hand gestures for both keyboard layouts are detected by a computer camera.

In the flat layout, the keys are arranged next to each other; see Figure 1. The cross layout is the original H4-Writer layout with the key positions forming a cross. Three fields are displayed at the top of the keyboards: presented text, transcribed text, and keys. The presented field shows the randomly selected phrase to be entered. The transcribed field displays the phrase that is being entered. The keys field shows the code being entered. The displayed characters on the keys change according to the selected key.

3.1 Flat keyboard (with code-based selection)

The flat keyboard uses a selection method called *code-based selection*. Code-based selection allows the user to select characters based on their associated code. The codes assigned to the characters are made up of digits ('1', '2', '3', '4'). Refer to Figure 2. With the palm facing the camera, the user can make hand gestures to select the desired key. The index, middle, ring, and pinkie finger should be closed to enter digits '1', '2', '3', or '4', or in other words, select the first, second, third, or fourth key, respectively. An open hand is an idle state where the user is not entering a code.

Figure 3a shows the gestures for the code-based selection. This method does not require positioning a pointer on the key, eliminating the need for hand motions. The flat design of the keyboard makes remembering gestures easier since the gestures were selected to imitate how a person presses a physical key by bringing down the corresponding finger.

The code-based selection can not be used on the H4VR cross keyboard; similarly, the virtual mouse selection (discussed next) can not be used on the H4VR flat keyboard.

3.2 Cross keyboard (with virtual mouse selection)

The cross keyboard uses a different gesture-based key selection method called *virtual mouse selection*. With the palm facing the camera, the movement of the hand is tracked, and a cursor moves accordingly on display. The user's hand is treated like a virtual mouse. With up/down and left/right movements of their dominant hand, the user places the cursor on the desired key. A selection happens when the user closes their thumb. See Figure 3b. The cross layout of the keyboard reduces the hand movement required to position the cursor on the desired key as opposed to the layout used in the flat keyboard as it reduces the distance between the keys.

4 METHODOLOGY

The evaluation of H4VR consisted of a longitudinal study with five sessions. The decision to use a longitudinal design was based on the learnability of the keyboards. During each session, participants entered 20 phrases selected randomly from the MacKenzie and Soukoreff [13] phrase set. Although the four-key keyboard used for this study allows for entering symbols, uppercase letters, and numbers, the sessions consisted of entering only lowercase letters and a space between words. Due to the limited number of sessions, the primary goal was to familiarize the participants with the position of each character on the keyboard and the codes assigned to them. The number of sessions and phrases for each session was selected based on the similar work reviewed in this paper.

4.1 Participants

Three male and two female participants were recruited from the local university (M = 25 years, SD = 0.89). This is fewer participants than in many studies, but such is common in longitudinal studies. Two participants had prior experience in text entry on a Qwerty keyboard in VR. None, however, had used ambiguous keyboards in

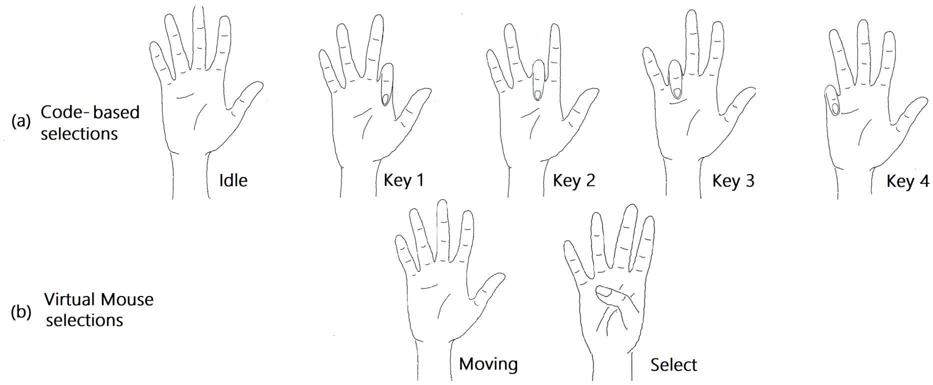


Figure 3: Selection methods for (a) H4VR flat keyboard, and (b) H4VR cross keyboard.

VR and none had worked with keyboards with similar character arrangements as H4VR in desktop or virtual environments. After completing each session, the participants received $20.

4.2 Apparatus

The H4VR software implements two keyboard layouts, *flat* and *cross*. See section 3 for details. The software was run on Microsoft *Windows 10* on a desktop environment.

Audio feedback as "click" is given when the participant selects a key. Before starting the first session, some participants asked for a view of their hand in the virtual environment. To address this, the image captured by the camera was displayed next to the keyboards during all sessions for all the participants. A Meta *Quest 2* head-mounted display (HMD) mirrored the desktop display, showing the H4VR interface. A 1080p *NexiGo* HD webcam with a resolution of 1920 × 1080 with 30 fps was used to detect hand gestures. *MediaPipe v0.7*, a Python library for hand and finger tracking, was employed to identify hand gestures.

4.3 Procedure

Participants engaged in a five-day study over two weeks. The table of character codes (Figure 2) was provided to participants one day before the first session to familiarize them with the codes. Participants were tested individually in a lab. Upon arrival at the lab, participants were instructed to sit comfortably in front of a camera connected to a PC. See Figure 4. The keyboard layouts and key selection methods were explained at the beginning of the first session. They put on and adjusted the HMD to have a clear display and a comfortable VR experience. The HMD mirrored the PC's display and showed the H4VR layout. The participant raises their hand with their palm facing the camera to start typing. The trials began when the participants selected the first character. They were allowed one trial before starting the experiment in each session.

During each session, participants entered ten phrases using the flat keyboard with the code-based selection method and ten phrases using the cross keyboard using the virtual mouse selection method. During each session, the participants could take breaks for up to ten minutes between the trials.

In the first session, all the participants first entered ten phrases using the code-based method and moved on to entering ten phrases using the virtual mouse method. In the second session, to offset order effects, they used the keyboard in reverse order. This swap happened for the rest of the sessions too. The first session took about an hour to complete. The time was less during the following sessions as the entry speed increased. The participant's entry speed and the error rate were recorded for each phrase. Upon completing the last session, they were asked to complete a questionnaire regarding their experience with the keyboards and key selection methods.

4.4 Design

The user study employed a 2 × 5 within-subjects design using the following independent variables and levels:

- Keyboard (flat + code-based selection, cross + virtual mouse selection)
- Session (1, 2, 3, 4, 5)

The dependent variables were text entry speed (wpm) and error rate (%). In addition, data were collected via the questionnaire administered in the last session. Each participant completed five sessions of about one hour each. To offset order effects, participants started each session using the opposite H4VR keyboard from the previous session. The total phrases entered was 5 participants × 2 text entry methods × 10 phrases per method × 5 sessions = 500.

5 RESULTS AND DISCUSSION

5.1 Entry Speed

The average text entry speed on the cross and flat keyboards was 2.87 wpm and 3.59 wpm, respectively. See Figure 5a. Using an ANOVA, the effect of the H4VR keyboard on entry speed was statistically significant ($F_{1,4} = 23.58$, $p < .01$). The higher average entry speed on the flat keyboard was expected since it eliminates the time required for moving and placing a pointer on a key.

Text entry speed on the cross keyboard started with a mean of 2.19 wpm in session 1 and finished at 3.63 wpm in session 5 (65.7% increase). Text entry speed on the flat keyboard started with a mean of 2.36 wpm in session 1 and finished at 4.57 wpm in session 5 (93.6% increase). Figure 5b shows the text entry speed increase over the five sessions for both keyboards. The average text entry speed on

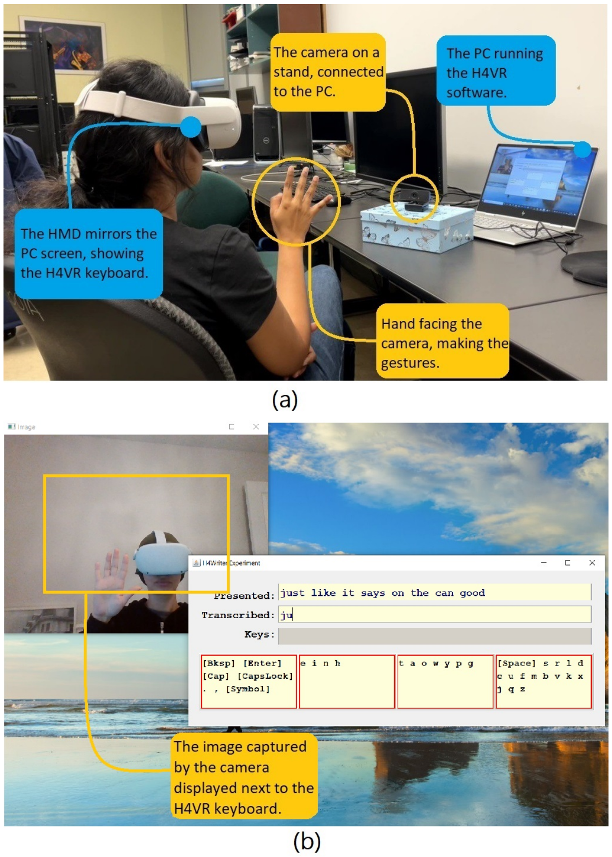


Figure 4: (a) The lab setting. (b) Desktop display of a participant using the flat keyboard.

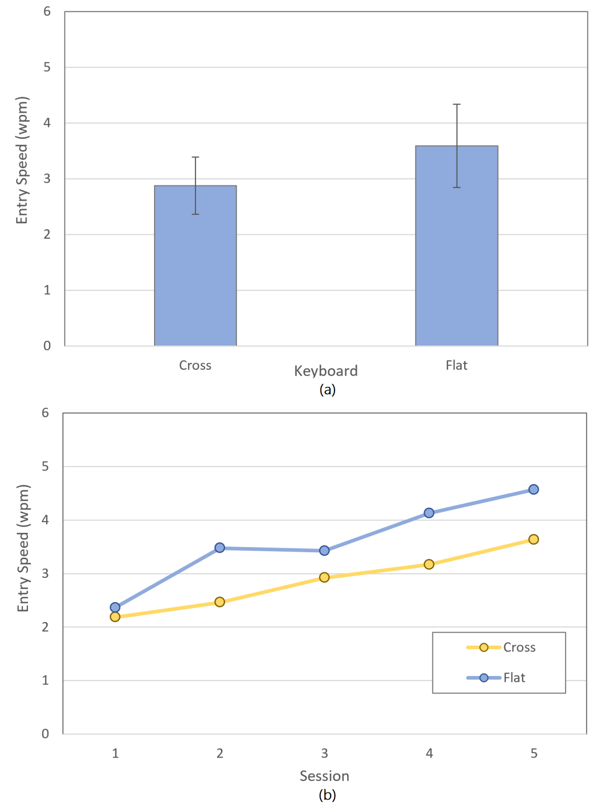


Figure 5: Entry speed (wpm) results: (a) Average entry speed (wpm) of H4VR keyboards in five sessions. Error bars show ±1 *SD*. (b) Entry speed (wpm) by H4VR keyboard and session.

the flat keyboard was higher than the cross keyboard for all the sessions. The effect of the session on entry speed was statistically significant ($F_{4,16}$ = 27.71, p < .0001), thus confirming the expected learning effect with practice. The average improvement in entry speed over the five sessions was also 28% higher in the flat keyboard than the cross keyboard.

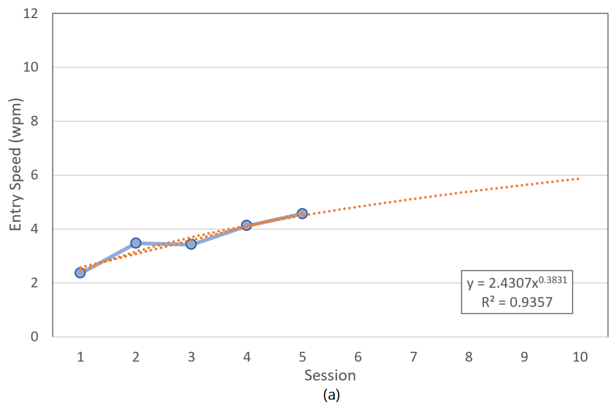
Since the keyboards rely on learning the location of each character, it is assumed that more practice can improve the text entry speed as remembering the location of the characters become easier. Figure 6a shows the power law of learning for the flat keyboard with entry speed = $2.43n^{0.38}$, where n is the session number. The model suggests an entry speed of about 6 wpm at session 10. As expected, the session × keyboard interaction effect was not statistically significant on the entry speed ($F_{4,16}$ = 2.23, p > .05). This indicates that the learning pattern was about the same for both keyboards.

The H4VR keyboards have fewer keys than the other reviewed studies with ambiguous keyboards. Considering the one-hand entry method used in H4VR, HiFinger [7] and PinchText [6] are the closest to H4VR. HiFinger, with six keys, reported an entry speed of 9.82 wpm after 75 minutes of use. This entry speed is about 73% faster than the H4VR flat keyboard and 92% faster than the H4VR cross keyboard. PinchText with 12 keys reported an entry speed of 12.71 wpm after six sessions of entering ten phrases. The prediction trajectory for the flat keyboard suggests that the expected entry speed on the sixth session is 4.8 wpm, showing a 90% lower entry speed than the PinchText. Although offering a lower entry speed, the flat keyboard does not rely on any attachments to the hand, such as sensors and conductive tape, unlike HiFinger and PinchText.

5.2 Error Rate

The average error rate was generally low for both H4VR keyboards. The average error rate for the cross keyboard was 0.79% (*SD* = 0.75), while the average error rate for the flat keyboard was 1.82% (*SD* = 1.42). The difference was statistically significant ($F_{1,4}$ = 7.72, p < .05). Figure 6b shows that the flat keyboard had a higher error rate for four sessions. Although the error rate on the last session was the lowest for both H4VR keyboards among other sessions, the effect of the session on error rate was not statistically significant ($F_{4,16}$ = 1.44, p > .05). The session × keyboard interaction effect also was not statistically significant on the error rate ($F_{4,16}$ = 0.9, ns).

H4VR utilizes an ambiguous keyboard with fewer keys than any similar work done in VR. Fewer keys make gesture-based selection easier since the user has to remember only four hand gestures. The gestures are easy to remember as there are only four selection



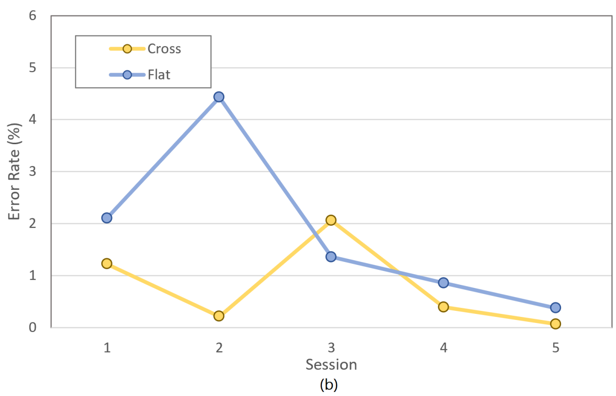


Figure 6: (a) Power model of text entry speed (wpm) on the flat layout over ten sessions. (b) error rate (%) on cross and flat layouts.

gestures in the flat keyboard and only one selection gesture in the horizontal keyboard. The small number of gestures relies less on users to recall the gestures and can prevent overwhelming the user. The gestures are also easy to make since they rely on minimum hand movement. A smaller part of the screen is occluded, which is one of the main advantages of using an ambiguous keyboard. It also relies on only one hand for text entry, which makes it more valuable to people who might have control of only one hand. While the condition of this study was to use the H4VR keyboards while receiving visual feedback on the keyboards and the text being entered, the flat keyboard can be utilized for eye-free text entry. By practice, the user can learn the codes assigned to the characters and the need for constant eye contact with the keyboard can be eliminated or reduced. However, this can increase the error rate, as seen in the HiFinger [7], which is an eye-free text entry method. HiFinger utilizes two more keys than H4VR, which can introduce a higher error rate than H4VR. The error rate of HiFinger was 107.2% higher than the H4VR flat keyboard at 6.03%. The error rate of PinchText [6] was similar to the H4VR keyboards at 2%. With 12 keys, PinchText did not report a lower error rate than H4VR.

5.3 Hand Fatigue and Preference

In the last session, participants were asked to rank their overall level of hand fatigue for both keyboards. Responses were on a 7-point Likert scale from 1 (*no fatigue*) to 7 (*very fatiguing*). The means were 3.2 and 4.4 for the flat and cross keyboards, respectively, indicating less hand fatigue for the flat keyboard. However, the difference was not statistically significant in a Wilcoxon Signed Ranks test (z = -1.069, p > .05). All participants reported that hand fatigue reduced as the sessions progressed. This could be a result of reduced trial duration as the participants got more familiar with the keyboards and the location of the characters. In the cross keyboard, getting familiar with the location of the characters can also reduce unnecessary hand movements during visual search, hence reducing hand fatigue. Four participants reported that they preferred using the flat keyboard, while one preferred cross.

6 CONCLUSION

In a five-session longitudinal evaluation of H4VR, participants reached an average entry speed of 3.63 and 4.57 words per minute on the cross and flat keyboards, respectively. As the sessions progressed, participants reported a significantly higher entry speed with an improvement of 65.7% and 93.6% on the cross and flat keyboards, respectively. The flat keyboard had a statistically significant higher entry speed than the cross keyboard.

The average error rate for both keyboards was low; however, the flat keyboard had a significantly higher error rate than the cross. More sessions of using the H4VR keyboards can lead to higher entry speed.

Hand-tracking, a feature that enables VR users to use their hands as input devices rather than controllers, tracks the users' hands and detects hand gestures. Some HMDs, such as HoloLens 2, the Leap Motion, and the Oculus Quest, offer hand-tracking using built-in cameras and sensors on the HMD. The user is also given a representation of their hand in the virtual environment.

For future work, the hand-tracking ability of the HMD will replace the external camera. Using the HMD hand-tracking can reduce hand fatigue since the hand does not need to be up and facing a camera. The gestures can be made anywhere with any hand orientation as long as it is recognizable by the HMD.

During the experiment, it was observed that if the hand was not fully facing the camera, gesture recognition could be incorrect. This issue also can be resolved with HMD hand-tracking, leading to lower mistakes and higher entry speed.

Having a hand representation in a virtual environment can also improve the user experience, which again can be achieved using the HMD hand-tracking ability.

Some participants also noted that fast double selection on the same key only gets recognized as one selection. Hardware that allows more immediate gesture recognition, another future improvement, can address this issue and improve entry speed.

REFERENCES

[1] Bartosz Bajer, I. Scott MacKenzie, and Melanie Baljko. 2012. Huffman base-4 text entry glove (H4 TEG). In *Proceedings of the 16th International Symposium on Wearable Computers*. IEEE, New York, 41–47. https://doi.org/10.1109/ISWC.2012.28

[2] Manas Kamal Bhuyan, D Ajay Kumar, Karl F MacDorman, and Yuji Iwahori. 2014. A novel set of features for continuous hand gesture recognition. *Journal*

on Multimodal User Interfaces 8, 4 (2014), 333–343.

[3] Sibo Chen, Junce Wang, Santiago Guerra, Neha Mittal, and Soravis Prakkamakul. 2019. Exploring word-gesture text entry techniques in virtual reality. In *Extended Abstracts of the ACM SIGCHI Conference on Human Factors in Computing Systems - CHi '19*. ACM, New York, 1–6. https://doi.org/10.1145/3290607.3312762

[4] Aakar Gupta, Cheng Ji, Hui-Shyong Yeo, Aaron Quigley, and Daniel Vogel. 2019. Rotoswype: Word-gesture typing using a ring. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems - CHI '19*. ACM, New York, 1–12. https://doi.org/10.1145/3290605.3300244

[5] David A Huffman. 1952. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE* 40, 9 (1952), 1098–1101.

[6] Haiyan Jiang, Dongdong Weng, Xiaonuo Dongye, and Yue Liu. 2022. PinchText: One-handed text entry technique combining pinch gestures and hand positions for head-mounted displays. *International Journal of Human–Computer Interaction* 0, 0 (2022), 1–17. https://doi.org/10.1080/10447318.2022.2115333

[7] Haiyan Jiang, Dongdong Weng, Zhenliang Zhang, and Feng Chen. 2019. HiFinger: One-handed text entry technique for virtual environments based on touches between fingers. *Sensors* 19, 14 (2019), 3063. https://doi.org/10.3390/s19143063

[8] Jung In Koh, Josh Cherian, Paul Taele, and Tracy Hammond. 2019. Developing a hand gesture recognition system for mapping symbolic hand gestures to analogous emojis in computer-mediated communication. *ACM Transactions on Interactive Intelligent Systems* 9, 1 (2019), 1–35. https://doi.org/10.1145/3297277

[9] Myron W Krueger, Thomas Gionfriddo, and Katrin Hinrichsen. 1985. VIDEOPLACE—an artificial reality. In *Proceedings of the ACM SIGCHI conference on Human Factors in Computing Systems - CHI '85*. ACM, New York, 35–40.

[10] Changsung Lim, Jina Kim, and Myung Jin Kim. 2022. Thumble: One-handed 3D object manipulation using a thimble-shaped wearable device in virtual reality. In *Adjunct Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology - UIST '22*. ACM, New York, 17.1–17.3. https://doi.org/10.1145/3526114.3558703

[11] I. Scott MacKenzie. 2002. KSPC (keystrokes per character) as a characteristic of text entry techniques. In *Proceedings of the Fourth International Symposium on Human-Computer Interaction with Mobile Devices - MobileHCI '02*. Springer, Berlin, 195–210. https://doi.org/10.1007/3-540-45756-9_16

[12] I. Scott MacKenzie and Steven J. Castellucci. 2013. Eye on the message: Reducing attention demand for touch-based text entry. *International Journal of Virtual Worlds and Human-Computer Interaction (VWHCI)* 1, 1 (2013), 1–9. https://doi.org/10.11159/vwhci.2013.001

[13] I. Scott MacKenzie and R. William Soukoreff. 2003. Phrase sets for evaluating text entry techniques. In *Extended Abstracts of the ACM SIGCHI Conference on Human Factors in Computing Systems - CHI '03*. ACM, New York, 754–755. https://doi.org/10.1145/765891.765971

[14] I Scott MacKenzie, R William Soukoreff, and Joanna Helga. 2011. 1 thumb, 4 buttons, 20 words per minute: Design and evaluation of H4-Writer. In *Proceedings of the 2011 ACM Symposium on User Interface Software and Technology - UIST '11*. ACM, New York, 471–480. https://doi.org/10.1145/2047196.2047258

[15] Munir Oudah, Ali Al-Naji, and Javaan Chahl. 2020. Hand gesture recognition based on computer vision: A review of techniques. *Journal of Imaging* 6, 8 (2020), 29 pages. https://doi.org/10.3390/jimaging6080073

[16] Marco Speicher, Anna Maria Feit, Pascal Ziegler, and Antonio Krüger. 2018. Selection-based text entry in virtual reality. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems - CHI '18*. ACM, New York, 1–13. https://doi.org/10.1145/3173574.3174221

[17] Ker-Jiun Wang, Quanbo Liu, Yifan Zhao, Caroline Yan Zheng, Soumya Vhasure, Quanfeng Liu, Prakash Thakur, Mingui Sun, and Zhi-Hong Mao. 2018. Intelligent wearable virtual reality (VR) gaming controller for people with motor disabilities. In *IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*. 161–164. https://doi.org/10.1109/AIVR.2018.00034

[18] Wenge Xu, Hai-Ning Liang, Anqi He, and Zifan Wang. 2019. Pointing and selection methods for text entry in augmented reality head mounted displays. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, New York, 279–288. https://doi.org/10.1109/ISMAR.2019.00026

[19] Caglar Yildirim. 2022. Point and select: Effects of multimodal feedback on text entry performance in virtual reality. *International Journal of Human-Computer Interaction* 0, 0 (2022), 1–15. https://doi.org/10.1080/10447318.2022.2107330

[20] Difeng Yu, Kaixuan Fan, Heng Zhang, Diego Monteiro, Wenge Xu, and Hai-Ning Liang. 2018. PizzaText: Text entry for virtual reality systems using dual thumbsticks. *IEEE Transactions on Visualization and Computer Graphics* 24, 11 (2018), 2927–2935. https://doi.org/10.1109/TVCG.2018.2868581