Frameless

Volume 3 | Issue 1 Article 21

2020

RotateEntry: Controller-rolling-style Text Entry for Three Degrees of Freedom Virtual Reality Devices

Ziming Li
Rochester Institute of Technology, zl1398@rit.edu

Roshan Peiris
Rochester Institute of Technology, Roshan.Peiris@rit.edu

Recommended Citation
Li, Ziming and Peiris, Roshan (2020) "RotateEntry: Controller-rolling-style Text Entry for Three Degrees of Freedom Virtual Reality Devices," *Frameless*: Vol. 3 : Iss. 1 , Article 21.
Available at: https://scholarworks.rit.edu/frameless/vol3/iss1/21

RotateEntry: Controller-rolling-style Text Entry for Three Degrees of Freedom Virtual Reality Devices

Cover Page Footnote

We would like to thank all the subjects in this study for their voluntary participation and feedback, which help us identify the usability issues on the evaluated text input methods.

FRAMELESS

RotateEntry: Controller-rolling-style Text Entry for Three Degrees of Freedom Virtual Reality Devices

Ziming Li*
Rochester Institute of Technology

Roshan Peiris
Rochester Institute of Technology

Abstract: *In this work, we propose RotateEntry, a controller-rolling-style method for text entry on three degrees of freedom virtual reality devices. To move the key-selecting cursor in two dimensions on a QWERTY layout virtual keyboard, we developed three variants of RotateEntry: Rotate Column Rotate, Rotate Key, and Rotate Column Point. We conducted a comparative empirical evaluation of the four text input methods, including three proposed controller-rolling-style text input methods and the standard raycasting-style one. Text entry performance, accuracy, workload, usability, and user experience were tested and evaluated. Due to the COVID-19 situation, our study was conducted remotely. The impact of using online formats on VR research had also been assessed. After evaluating with 5 participants, we identified that Rotate Key had a higher text entry rate, outstanding overall user experience, and excellent overall workload performance among the three variants of RotateEntry. However, no evidence had been investigated to support the hypothesis that RotateEntry had better performance and experience compared to Raycasting.*

Keywords: *virtual reality, text entry method, three degrees of freedom, controller, QWERTY keyboard layout*

* Ziming Li (zl1398@rit.edu)
Submitted November 29, 2020
Accepted December 1, 2020
Published online July 15, 2021

I. INTRODUCTION

Three degrees of freedom (3DoF) virtual reality (VR) devices are VR controllers which can detect rotational movements to interact with virtual content presented in a head-mounted display. Many recent commercial VR devices such as the Oculus Go, Google Daydream have featured 3DoF controllers with additional inputs such as buttons. Compared to a 6DoF VR device, which can track positional and rotational movements in a 3-dimensional space, a 3DoF VR device has a lower demand for spatial movements. A 3DoF device can primarily support activities with less movement, such as web browsing, watching videos, and VR socialization. Among these scenarios, the text input technique plays an essential role.

The controller-based raycasting keyboard is a popular built-in text entry solution for 3DoF VR devices. While using this type of text input method, the user can aim the virtual ray, which is cast by the controller, at a particular key on the virtual keyboard, and enter the focused character by pressing a button on the VR controller (Boletsis and Kongsvik 2019). The raycasting-style text input method provides an intuitive and precise way for text entry on VR devices. However, while using the raycasting input, the user has to hold a controller, keep raising his/her arm in use, and frequently move the arm in space to aim at a key. After long term use, this using posture can cause arm muscle fatigue (Grubert et al. 2018), resulting in reducing text input performance, accuracy, and user experience. Also, the aim-and-shoot style interaction implemented by the raycasting input would be challenging for those who cannot keep their arms in mid-air for a long time.

This paper proposes RotateEntry, a "controller-rolling-style" text entry method for 3DoF VR devices. RotateEntry moves a key-selecting cursor through the virtual keyboard using the controller's rolling angle and relative pitching angle. In this way, it frees the VR user from the need to enter text with a fixed posture, and instead, the user can put his/her hand holding the controller in any spatial position. Hence, it could be a potentially more efficient and effort-saving way for VR text entry than the raycasting solution.

This study focused on the interaction comparison between RotateEntry and the standard raycasting method. Since the typical raycasting-style text input technique uses a QWERTY keyboard layout (Dube and Arif 2019), RotateEntry implemented the same keyboard layout, aiming to eliminate the potential effect of the keyboard layout. We developed three interaction methods named Rotate Column Rotate, Rotate Key, and Rotate Column Point. They were using the RotateEntry concepts that look at how to move the cursor across the standard virtual QWERTY keyboard. Next, we evaluated these three methods and the traditional raycasting method in a comparative empirical study. The knowledge obtained from this study might help us identify a proper interaction for RotateEntry and provide us insights to improve it further.

II. RELATED WORK

Existing VR text entry techniques can mainly be classified into three types: (a) game-controller-based text input methods, which entering text by using two thumbsticks on the gamepad; (b) VR-HMD-based (Head Mounted Display) text input methods, which entering text based on user's head motions; and (c) VR-controller-based text input methods, which entering text by using VR controllers.

In Pizzatext, a game-controller-based text input method, a customized keyboard layout was implemented (Yu et al. 2018). This text entry technique divided a circular layout into several slices, while each slice includes a certain number of letters. Yu's team have also offered three different keyboard layouts for their method. The words per minute (WPM) and total error rate (total ER) of each proposed layout were tested. Therefore, the best performance keyboard layout for their proposed text entry method could be identified by analyzing the data.

RingText was a dwell-free VR-HMD-based text entry method (Xu et al. 2019). In RingText, a circular keyboard layout with a "go-and-hit" character selection interaction was implemented. Xu's team have conducted three studies for the RingText evaluation. The first study was a comparative evaluation of 12 types of RingText keyboard layout designs, which helped them determine a proper layout design for RingText. The second one was a within-subject comparative experiment, which helped identify the text-entry rate and accuracy of RingText compared with those of the other four hands-free text input techniques. The last one was a 4-day study, which measured the trend of the novice and expert users' typing performance after long-term practice.

A potential solution for VR-HMD-based text input methods is SliceType (Benligiray, Topal, and Akinlar 2018), a gaze typing method involving a customized circular keyboard layout. The work from the Benligiray's team applied the eye-tracking technique to text entry. Their text input method's virtual keyboard can dynamically allocate a larger space for the target character key, which can be more comfortable for an eye-tracking cursor to focus on and assist in typing text more efficiently.

In terms of the VR-controller-based text entry techniques, Boletsis and Kongsvik's work evaluated four text entry techniques built for dual-controllers VR devices. The four evaluated techniques all used a QWERTY keyboard layout but with different keystroke methods. Boletsis and Kongsvik have conducted a within-subject comparative empirical evaluation among those four controller-based input methods. The text entry rate and accuracy were tested using the scale of words-per-minutes and total error rate. In addition to the text entry performance, the usability and the user experience of those text-entry techniques were also investigated using the System Usability Scale (SUS) questionnaire (Brooke 2013) and the Game Experience Questionnaire (GEQ) (IJsselsteijn, De Kort, and Poels 2013). In their evaluation, both the auto-completion and auto-correction functionalities had been disabled to eliminate potential effects.

The prior studies discussed above have proposed a series of VR text input methods based on various text entry interactions and virtual keyboard layouts. They also presented a few excellent research methods and metrics for a VR text input technique evaluation. However, while a user's perception of spatial presence in the VR display is minimal (Seibert and Shafer 2017), there is still a lack of study on controller-based VR text input methods less reliant on the perception of spatial presence.

III. EVALUATED TEXT INPUT METHODS FOR 3DOF VR DEVICES

Four different controller-based text input methods for 3DoF-single-controller VR devices were developed for the evaluation in this study, which was Rotate Key (RK), Rotate Column Point (RCP), Rotate Column Rotate (RCR), and Raycasting. The first three text input methods were implemented using the proposed RotateEntry interaction, and the last one was implemented using the standard raycasting-style interaction. Since this study focused on evaluating and comparing the text entry interactions, all of the aforementioned text entry methods shared the same QWERTY keyboard layout.

A. RotateEntry Concept

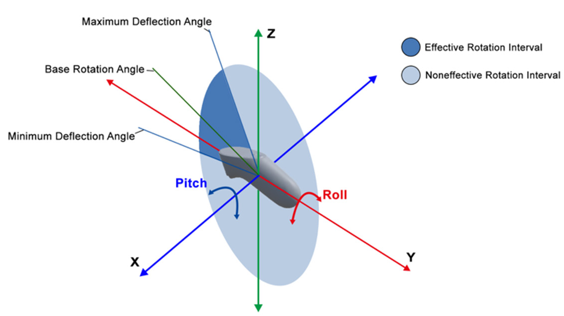

Fig. 1. The concepts of "roll", "pitch", "effective rotation (pitching) interval", "base rotation (pitching) angle", "maximum deflection angle", and "minimum deflection angle". (The controller's base rotation [rolling] angle and effective rotation [rolling] interval is not shown.)

As stated above, the implementation of RotateEntry was to capture the controller's rotational input of a particular axis. The orientation of RotateEntry's key-selecting cursor's movement was detected based on the offset between the controller's current effective rotational input and the base rotation (pitching/rolling) angle value. The way the base pitching angle captured was various base on each interaction method. In contrast, the base rolling angle was always equal to the initial rolling angle of the controller. The controller's effective rotation (pitching/rolling) interval was specified based on the base rotation (pitching/rolling) angle (See Fig. 1).

By dividing the effective rotation interval by the number of rows or the number of columns of the QWERTY keyboard layout, a specific rotation interval was allocated to each key. The key whose allocated rotation interval included the controller's current effective rotation angle would be highlighted. For the current effective rotation angle's value, if the controller's current rotation angle were greater than the maximum deflection angle of the pre-defined effective rotation interval, the maximum deflection angle would be accepted. Similarly, if the controller's current rotation angle were less than the minimum deflection angle, the latter would be accepted.

B. RK: "Rotate Key" Interaction Method

Fig. 2. RK: rolling the controller to move the cursor (light blue outline) left and right while pitching the controller to move the cursor up and down

RK (see Fig. 2) used a cursor for key selection on the virtual keyboard. The user rolled the VR controller to move the cursor horizontally while pitching the controller was to move the

Fig. 3. RCP: (left) rolling the controller to switch the highlighted column (light blue outline), press the trigger button on the controller to lock the highlighted column; (right) Once a column is locked, pitching the controller to move the cursor (light blue outline) up and down within the locked column (dark blue outline)

Fig. 4. RCR: (left) rolling the controller to switch the highlighted column (light blue outline), press the trigger button on the controller to lock the highlighted column; (right) Once a column is locked, rolling the controller to move the cursor (light blue outline) up and down within the locked column (dark blue outline)

cursor vertically. The horizontal movement of the cursor was based on the controller's effective rolling angle. The cursor's vertical movement was detected based on the controller's current effective pitching angle. Its base pitching angle was captured on the input method's activation.

Once the user's posture changed, he/she could press a particular key on the controller to reset the base pitching angle to the controller's current pitching angle. This allowed the user to use the interaction method in any posture.

C. RCP: "Rotate Column Point" Interaction Method

RCP (see Fig. 3) separated the interaction in RK into two steps. The first step was column selection. In this step, a column of keys on the virtual keyboard would be highlighted at a time. The user could switch the highlighted column by rolling the controller. Like Rotate Key, the column-selecting cursor's movement was based on the controller's effective rolling angle. The user could press the trigger button on the controller to lock the highlighted column, which would lead the user to the second step – row selection. In this step, the user could move a key-selecting cursor vertically within the locked column by pitching the controller. The cursor's movement was based on the controller's current effective pitching angle. Its base pitching angle was captured on the row selection mode's activation. The user could press a particular key on the controller to quit the row selection mode.

D. RCR: "Rotate Column Rotate" Interaction Method

RCR (see Fig. 4) also used a two-step interaction. Its interaction was very similar to the RCP. The significant difference between them was that the row selection of the RCR was roll-based, instead of pitch-based. In RCR's row selection mode, the user moved the key-selecting cursor within the locked column by rolling the controller. In RCR, both the horizontal movement and vertical movement of its keyboard cursor was based on the controller's effective rolling angle.

E. "Raycasting" Interaction Method

Raycasting (see Fig. 5) was a mainstream VR text entry solution that implemented an aim-and-shoot style for the virtual keyboard interaction. For a keystroke, the user could cast the virtual ray, emitted from the top end of the controller, to a particular key on the keyboard.

When a particular key was highlighted, the final confirmation in these four text input methods to select the highlighted character would be pressing the controller's trigger button.

IV. EVALUATION STUDY

A comparative empirical evaluation was conducted to evaluate the text-entry rate, text-entry accuracy, system usability, user workload, and user experience of the four aforementioned VR text input methods.

Fig. 5. Raycasting: moving the controller in space and casting the virtual ray to a particular key to highlight it

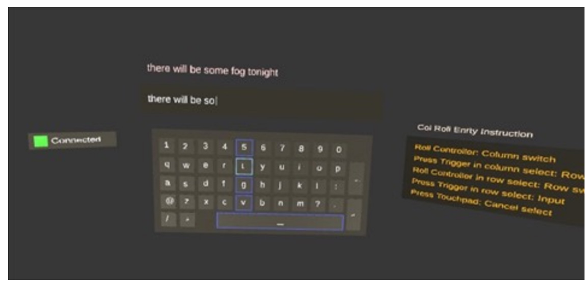

Fig. 6. A screenshot of the evaluated demo's user interface

A. Technical Detail

All four text input methods evaluated in this study were developed on the Unity game engine. The evaluation was a demo application running on an Oculus Go VR headset with an Oculus Go 3DoF controller. A QWERTY layout virtual keyboard, whose layout was similar to the Oculus Go built-in keyboard one, was implemented (see Fig. 6). All four text input techniques in the demo shared the same keyboard layout.

To eliminate potential deviation caused by other functionalities, neither auto-completion nor auto-correction functionalities were used for the VR text input techniques tested in this study. Moreover, no audible or haptic feedback was implemented for the keystrokes.

Due to the COVID-19 situation, the studies were conducted remotely on the Zoom.us video conferencing platform. Therefore, for the remote testing control and data monitoring, a Node.js server was developed. A MongoDB database was implemented on the server-side to record the testing data. Socket.io was used to provide real-time network communication between the HMD side and the server side. Furthermore, a dashboard web page was developed to set up the current tested text input method and the current phrase on the server side, and to visualize the HMD side's input data. These additional developments helped the experimenter monitor the study progress efficiently and intervene in case of potential issues.

B. Study Design

In this study, the within-subjects design was used with one independent variable (text entry method) consisting of four levels (experimental conditions): the three RotateEntry text input methods (RK, RCP, RCR) and the "ray-casting-style" text input method (Raycasting). As the dependent variables, text-entry rate, text-entry accuracy, workload, usability, and user experience of the four VR text input methods were tested.

The text-entry rate and text-entry accuracy metrics of the study were measured based on the words-per-minute (WPM) metric presented by Wobbrock (Wobbrock 2007), and the total error rate (total ER) metric presented by Soukoreff and MacKenzie (Soukoreff and MacKenzie 2003).

Besides, we used the following to test the user's subject experiences: a 10-item System Usability Scale (SUS) questionnaire was used to evaluate the subjective system usability of the text input techniques; the Game Experience Questionnaire (GEQ) was used to measure the user experience of the text input techniques; the NASA task load index (NASA TLX) was used to measure a participant's task workload.

As the main task, the participant was required to enter a phrase in a VR environment using the text entry method determined by the condition. Each condition presented five phrases selected from the phrase set used in Boletsis and Kongsvik's study. The selected phrases are shown in Table 1. Each participant was presented with a randomized order of the conditions, and the phrases were presented in a randomized order for each condition. Therefore, each participant completed a total of 20 tasks (4x5).

Table.1. Phrases used in the study

Phrase	*Length (Character)*
my preferred treat is chocolate	31
question that must be answered	30
there will be some fog tonight	30
physics and chemistry are hard	30
we are subjects and must obey	29

C. Participants

Due to the COVID-19 situation, the study was conducted remotely, and the size of the participant group was limited. The study procedures and protocols were approved by the university's Internal Review Board. A group of 5 people was recruited. The participants were selected through word-of-mouth based on their accessibility to an Oculus Go device. Among the 5 participants in the study, 2 were males, and 3 were females. Their ages ranged from 21 to 28 (Mean = 25.2, SD = 2.77). One of them used VR devices frequently, three of them rarely used VR devices, and one of them never used a VR device before. Those four participants who had used VR devices had only used 3DoF ones (Oculus Go or Gear VR), and they mainly used the devices for entertainment proposes (playing games, watching movies).

D. Procedure

Once recruited, the participants were provided with the Informed Consent form via email. Once a participant provided their consent, the study software was emailed to the participants with detailed installation instructions. The participant was free to install the software before or during the experimental session.

After filling out a demographic questionnaire during the study session, each participant was asked to complete four experimental conditions. Before each task in the condition, a participant had 5 minutes of practice time to get familiar with the tested text input technique for the upcoming task. Then, in each condition, the participant was asked to enter five preselected phrases as fast and accurate as possible using one of the four VR text input methods. Each phrase was shown to the participant at a time and kept displaying on the user interface until the participant completed it. After completing each condition, the participant was told to fill in the SUS questionnaire, the GEQ questionnaire, and the NASA TLX questionnaire. There was a 5-minute break after each condition. The same procedure was used within the remaining VR text input methods. The participant was told to use the dominant hand to hold the controller and not switch the hand in use during the whole testing session. Both the order of the VR text input methods and the phrases' order were randomly organized for each participant. The character input from the VR controller, the WPM data, and the total ER data was monitored and recorded from the server-side during the test. After the participant completes all the tasks, a semi-structured interview was conducted to collect the participant's comments for each of the four evaluated VR text input methods. The experiment took approximately 90 minutes per participant to complete.

V. RESULT

A. Text Entry Rate

Table 2 shows the words-per-minutes data of the four evaluated text input methods. For each task in the study, the data of the first attempt was discarded. The results further analyzed the data using repeated measures ANOVA. A significant main effect was found on the Text Entry Method ($F[3, 12] = 73.769$, $p < 0.001$). Post-hoc comparisons showed significant differences between RCP and RK ($p = 0.005$), RCP and Raycasting ($p < 0.001$), RK and RCR ($p = 0.006$), RK and Raycasting ($p < 0.001$), and RCR and Raycasting ($p < 0.001$). No significant difference was revealed between RCP and RCR ($p = 0.772$).

Table.2. Words-per-minutes (WPM) performance of the evaluated text input methods

VR Text Input Methods	*WPM Mean (SD)*	*WPM Range*
RCP	2.89 (0.74)	2.30 – 4.15
RK	5.51 (1.48)	3.06 – 6.88
RCR	3.08 (1.10)	1.83 – 4.76
Raycasting	11.35 (0.56)	10.70 – 12.03

B. Text Entry Accuracy

Table 3 shows the total error rate data of the evaluated text input methods. For each task in the study, the data of the first attempt was discarded. The results further analyzed the data using repeated measures ANOVA, which revealed a significant main effect (F[3, 12] = 3.844, p = 0.039).

C. System Usability

Table 4 shows the SUS scores and the SUS ratings of the evaluated text input methods. The SUS rating is obtained based on a 7-point adjective scale from Bangor and his team's work (Bangor, Kortum, and Miller 2009).

D. User Experience

Table 5 shows the GEQ scores of the four evaluated text input methods across the nine dimensions. Each item value ranges from 0 ("not at all") to 4 ("extremely"). For the scores from the "Competence", "Immersion", "Flow", "Positive Affect" dimensions, the higher is better. For the scores from the "Tension", "Challenge", "Negative Affect", "Returning to Readlity", and "Tiredness" dimensions, the lower is better.

The Friedman Test indicates that there are no statistically significant differences in all the GEQ dimensions, except Tension (X2[3] = 8.333, p = 0.040) and Negative Affect (X2[3] = 8.455, p = 0.037).

E. Workload

Table 6 shows the scores of the four evaluated text input methods across the six NASA TLX dimensions. The value of each item ranges from 1 ("very low") to 7 ("very high"). In this case, for the scores from all dimensions, except for "Performance", the lower is

Table.3. Total Error Rate (total ER) of the evaluated text input methods

VR Text Input Methods	*Total ER Mean (SD)*	*Total ER Range*
RCP	8.35% (4.48%)	4.23% - 16.01%
RK	8.99% (4.66%)	4.09% - 14.15%
RCR	7.76% (4.68%)	0.81% - 13.45%
Raycasting	2.69% (2.19%)	0.00% - 5.12%

Table.4. The SUS scores and ratings of the evaluated text input methods

VR Text Input Methods	*SUS Mean (SD)*	*SUS Range*	*SUS Rating*
RCP	60 (30.10)	10 – 87.5	OK
RK	61 (10.55)	45 – 72.5	OK
RCR	64.5 (25.46)	32.5 – 100	OK
Raycasting	87.5 (16.01)	70 – 100	Excellent

Table.5. GEQ scores across the 9 GEQ dimensions of the evaluated text input methods

GEQ Dimensions	*GEQ Mean (SD)*				*X^2 (p-value)*
	RCP	*RK*	*RCR*	*Raycasting*	
Competence	2.5 (1.27)	2.9 (0.96)	3.2 (0.76)	3.4 (0.65)	7.000 (0.072)
Immersion	2.4 (1.67)	3 (0.71)	2.2 (1.30)	3.2 (1.10)	1.923 (0.589)
Flow	2.6 (0.89)	3 (0.94)	2.9 (1.14)	2.1 (0.89)	4.674 (0.197)
Tension	2 (1.27)	1.7 (1.40)	2 (1.58)	0.5 (1.12)	8.333 (0.040)
Challenge	2.6 (0.96)	2.5 (1.06)	2.4 (0.65)	1 (1.00)	7.625 (0.054)
Negative Affect	2.1 (1.43)	1.5 (1.37)	2 (1.37)	0.4 (0.89)	8.455 (0.037)
Positive Affect	2.6 (1.56)	3.1 (0.42)	2.7 (0.67)	3.3 (0.76)	3.000 (0.392)
Returning to Reality	1.1 (0.74)	0.8 (1.10)	0.7 (0.76)	0.4 (0.55)	6.818 (0.078)
Tiredness	2 (1.58)	1.2 (0.84)	1.7 (1.48)	0.4 (0.89)	7.769 (0.051)

Table.6. NASA TLX scores across the 6 NASA TLX dimensions of the evaluated text input methods

NASA TLX dimensions	*NASA TLX Mean (SD)*				*X^2 (p-value)*
	RCP	*RK*	*RCR*	*Raycasting*	
Mental Demand	5 (1.87)	3.6 (2.19)	4.8 (1.64)	2.4 (1.52)	6.391 (0.094)
Physical Demand	5.6 (2.07)	5 (1.87)	5.4 (1.95)	2.4 (1.95)	8.500 (0.037)
Temporal Demand	3.8 (2.17)	4.6 (1.52)	3.8 (1.79)	3 (1.87)	1.884 (0.597)
Performance	5.6 (1.14)	6.4 (0.89)	6 (1.41)	6.6 (0.89)	5.625 (0.131)
Effort	5 (1.87)	4.8 (1.64)	5.4 (1.52)	2.8 (2.17)	4.256 (0.235)
Frustration	4.6 (1.95)	3.8 (2.39)	4 (2.00)	1.8 (1.79)	7.786 (0.051)

better. For the scores from the "Performance" dimension, the higher is better.

The Friedman Test indicates that there are no statistically significant differences in all NASA TLX dimensions, except Physical Demand, which X2(3) = 8.500, p = 0.037.

F. Interview Comment

Table 7 shows the comments collected from the semi-structured interview sessions with the 5 participants. The comments under each condition are ranked according to the number of participants who had mentioned it. Each comment has been labeled as positive ("P"), negative ("N"), or neutral ("-"), based on its outcome.

VI. DISCUSSION

The study result indicates that Rotate Key (RK) has the highest text entry rate among the three variants of RotateEntry, with 5.51 WPM. There are no significant differences in the text entry rate between Rotate Column Point (RCP) and Rotate Column Rotate (RCR), with 2.89 WPM and 3.08 WPM respectively. The three variants' differences might be attributed to RK using a one-step control while the other two use two-step ones. For the two using the two-step control, there would be extra time consumption in switching between the two modes of moving the cursor horizontally and vertically, which may result in a decrease in input rate. However, in terms of text entry performance, Raycasting outperforms the other three evaluated text input methods, with 11.35 WPM. 80% of the participants indicates that selecting a character using Raycasting is intuitive, which just needs to aim at a particular key on the virtual keyboard. While 40% of participants indicate the "column locking" mechanism on RCP and RCR is complex, and 40% of participants point out that moving the cursor on RK is difficult. Only 20% of the participants suggest that the interaction on RK is intuitive.

As for the accuracy of text entry, the total error rate of RCR is 7.76%, which is the lowest compared to RCP (8.35%) and RK (8.99%). Raycasting yields the lowest total error rate (2.69%) among the evaluated text input methods. As indicated by some of the participants (RCP: 60%, RK: 20%, RCR: 40%), the key-selecting cursors on all of the RotateEntry text input methods are sensitive, which may result in the high total error rate of RCR, RCP, and RK. This issue may be due to the cursor control algorithm of RotateEntry lacked the noise-reducing process for the input signal.

In terms of the subjective system usability in the evaluation, Raycasting yields the best SUS rating ("Excellent"), while the ratings of the three RotateEntry text input methods are all "OK". Among the three RotateEntry text input methods, RCR has a slightly higher mean SUS score (64.5), compared to RCP

Table.7. Comments collected from the 5 participants in the interview sessions

	Interview Comment	*Count*
	RCP	
N	The cursor was sensitive.	3
N	Column locking was complex.	2
N	It was complex. Because it had two movements involving the rotation and then you had to do up and down also.	1
P	It had a good pace.	1
N	It was a bit tiring.	1
P	The typing was getting very easier because I could select the columns and then rows, instead of just going to select all the alphabet and searching for everything.	1
P	It has two distinct interaction patterns.	1
N	it was a little difficult for me to start. When I did not know it was rotating the wrist left and right.	1
	RK	
N	When I have to move right left, it is fine, but if I had to select the rows by going up and down, that is difficult.	2
N	It was a bit tiring.	2
N	It is kind of confusing. It was like moving on its own. I felt like I did not have control, but actually, I did.	2
P	It was easy to use. It needs little effort. I just have to tweak and move up and down.	2
N	The cursor was too sensitive for me.	1
P	It was intuitive.	1
N	When my cursor was moved beyond the delete button, I was lost because I could not see where my cursor was.	1
	RCR	
N	I had to rotate around to go down to the bottom row. It was difficult.	2
N	The column locking makes it complex to use.	2
N	The cursor is sensitive.	2
P	Since its control was rotation only, I felt confident to do it.	1
-	Some letters (e, t) were very easy to locate, while some were hard (space, delete).	1
P	I could just get my wrist on the table stiff and then move.	1
	Raycasting	
P	It was intuitive. I did not have to use multiple clicks. I just had to point directly to what I want.	4
N	It was effortful to use. I had to point at the keyboard, and then to the exact key.	1
P	It was comfortable to use.	1
P	I knew where exactly a key was at the keyboard, so I was effortless to type quickly.	1

(60) and RK (61). It is worth mentioning that RCP and RCR have significantly higher standard deviations in their SUS scores (RCP: 30.10, RCR: 25.46), compared to RK (10.55) and Raycasting (16.01). It indicates that, compared to RK and Raycasting, the subjective usability of RCP and RCR varies significantly among the 5 participants.

When it comes to the user experience, Raycasting shows the best GEQ performance among the four text input methods on all dimensions except Flow, which supports its high SUS rating ("Excellent") and the high proportion (80%) of user comments which consider it is intuitive. Among the three RotateEntry text input methods, RK has the highest GEQ performance on 6 GEQ dimensions, including Immersion, Flow, Tension, Negative Affect, Positive Affective, and Tiredness. What is more, on the Flow dimension, RK's performance surpasses all the other three evaluated text input methods, which may explain the participant comments that indicate it is easy to use (40%) and intuitive (20%). RCR and RCP have similar GEQ performance. However, RCR mildly outperforms all of the GEQ dimensions except Immersion and Tension, compared to the latter. It may support the higher mean SUS score of RCR (64.5) than RCP (60).

Raycasting has superior workload performance. Its NASA TLX scores are the best on all six dimensions among the evaluated text input methods. Moreover, it has outstanding performance on Physical Demand (2.4), Effort (2.8), and Frustration (1.8) compared to the other three techniques. The above evidence may support its remarkable performance on the GEQ dimension of Tension, Challenge, and Tiredness. Among the three RotateEntry text input methods, RK marks the best NASA TLX scores on all dimensions, except Temporal Demand (4.6). On the Temporal Demand dimension, RK is also the worst performer among the evaluated text input methods. The workload performance of RCP and RCR is similar. Nevertheless, RCR has slightly higher performance on all dimensions compared to RCP, except Performance and Frustration.

Overall, in this evaluation study, Raycasting has achieved the better performance. It surpassed the three RotateEntry text input methods on all dimensions, except GEQ Flow. However, it should be noted that four out of our five participants had frequent experience with VR and using the standard Raycasting method. In this first study, we compared Raycasting only as a baseline. Nevertheless, 1 of the 5 participants (20%) indicates that "Raycasting was effortful to use. I had to point at the keyboard, and then to the exact key", while in terms of RK, one of the variants of RotateEntry, the participant indicates that "RK needs little effort. I just have to tweak and move up and down". It may provide evidence that RotateEntry has potential advantages on the accessibility aspect.

Among the three RotateEntry text input methods, RK yields a high text entry rate, outstanding overall performance on the GEQ dimensions, and the NASA TLX dimensions. However, meanwhile, it has the lowest text entry accuracy and a relatively low SUS score. Some participants (40%) argue that its cursor control is confusing, and some of the participants (40%) also find the cursor control is difficult when it comes to row selecting. Moreover, it may have a usability issue that the cursor would become invisible when it moves to a blank space as stated by one participant.

RCP and RCR have similar performance. However, compared to the former one, RCR exhibits higher text entry rate and accuracy, a higher SUS score, and better overall performance on both of the GEQ dimensions and the NASA TLX dimensions. 40% of the participants indicate that the column locking mechanism on these text input techniques is problematic. Other than that, the comments on their controls are various. Some participants may prefer RCP's control ("it has two distinct interaction patterns"), while some may prefer RCR ones ("since its control was rotation only, I felt confident to do it").

A. Study Limitations

Due to the COVID-19 situation, the evaluation only had 5 participants recruited. Although the general difference among the evaluated text input techniques has been investigated, the small sample group may result in reducing the potential to reflect the statistically significant effects on some of the dimensions (Button et al. 2013).

As stated above, the experiment duration for each participant was around 90 minutes. As a result of the tight schedule, each participant only had approximately 5-minute practice time to get familiar with the text input method evaluated in the following condition. Since the tested device's built-in text input technique was raycasting-style, the participant may be potentially an expert user on Raycasting. At the same time, he/she may be a novice user on the three proposed RotateEntry text input techniques. It may bias the performance and experience result of the evaluated text input techniques.

B. Future Work

Firstly, as indicated from the discussion above, a noise reduction algorithm for the input signal would be embedded for the three proposed RotateEntry text input techniques to improve their text-entry accuracy, usability, and user experience performance.

Secondly, unlike Raycasting, the three proposed RotateEntry text input techniques make no demand on users to aim at a key so that future study may focus on the effect of the keyboard size and the keyboard position of RotateEntry. Furthermore, instead of the standard QWERTY keyboard layout implemented in this study, a customized keyboard layout potentially more suitable for RotateEntry would also be investigated.

Finally, for the future experiment design, a larger sample size group and a long-term experiment session would be implemented.

VII. CONCLUSION

In this study, we proposed RotateEntry, a text-entry interaction that moved the key-selecting cursor on a virtual keyboard by rolling a 3DoF VR controller. Based on the concept of RotateEntry, three controller-rolling-style text input methods were developed: Rotate Key, Rotate Column Point and Rotate Column Rotate. A comparative empirical evaluation with 5 participants was conducted to test and evaluate the text-entry rate, text-entry accuracy, system usability, user experience, and workload of the three RotateEntry text input methods and a standard Raycasting text input method. By analyzing the evaluation results, we identified that Rotate Key had an outstanding text input speed, higher overall user experience scores, and excellent overall workload performance, compared to the other two RotateEntry text input techniques. However, no evidence was identified to support that RotateEntry had better performance and experience compared to Raycasting.

VIII. REFERENCES

Bangor, Aaron, Philip T. Kortum, and J. T. Miller. 2009. "Determining what individual SUS scores mean: adding an adjective rating scale." *Journal of Usability Studies archive* 4 (2009): 114-123.

Benligiray, Burak, Cihan Topal, and Cuneyt Akinlar. 2018. "SliceType: Fast Gaze Typing with a Merging Keyboard." *Journal on Multimodal User Interfaces* 13, no. 4 (2018): 321–34. https://doi.org/10.1007/s12193-018-0285-z.

Boletsis, Costas, and Stian Kongsvik. 2019. "Controller-Based Text-Input Techniques for Virtual Reality: An Empirical Comparison." *International Journal of Virtual Reality* 19, no. 3 (2019). https://doi.org/10.20870/ijvr.2019.19.3.2917.

Boletsis, Costas, and Stian Kongsvik. 2019. "Text Input in Virtual Reality: A Preliminary Evaluation of the Drum-Like VR Keyboard." *Technologies* 7, no. 2 (2019): 31. https://doi.org/10.3390/technologies7020031.

Brooke, J. 2013. "SUS: a retrospective." *Journal of Usability Studies archive* 8 (2013): 29-40.

Button, Katherine S., John P. A. Ioannidis, Claire Mokrysz, Brian A. Nosek, Jonathan Flint, Emma S. J. Robinson, and Marcus R. Munafò. 2013. "Power Failure: Why Small Sample Size Undermines The Reliability Of Neuroscience". *Nature Reviews Neuroscience* 14 (5): 365-376. doi:10.1038/nrn3475.

Dube, Tafadzwa Joseph, and Ahmed Sabbir Arif. 2019. "Text Entry in Virtual Reality: A Comprehensive Review of the Literature." *Human-Computer Interaction. Recognition and Interaction Technologies Lecture Notes in Computer Science*, 2019, 419–37. https://doi.org/10.1007/978-3-030-22643-5_33.

Grubert, Jens, Lukas Witzani, Eyal Ofek, Michel Pahud, Matthias Kranz, and Per Ola Kristensson. 2018. "Text Entry in Immersive Head-Mounted Display-Based Virtual Reality Using Standard Keyboards." *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, 2018. https://doi.org/10.1109/vr.2018.8446059.

IJsselsteijn, W., de Yaw Yvonne Kort and K. Poels. 2013. "The Game Experience Questionnaire." (2013).

Seibert, Jonmichael, and Daniel M. Shafer. 2017. "Control Mapping in Virtual Reality: Effects on Spatial Presence and Controller Naturalness." *Virtual Reality* 22, no. 1 (2017): 79–88. https://doi.org/10.1007/s10055-017-0316-1.

Soukoreff, R. William, and I. Scott Mackenzie. 2003. "Metrics for Text Entry Research." *Proceedings of the conference on Human factors in computing systems - CHI '03*, 2003. https://doi.org/10.1145/642611.642632.

Wobbrock, Jacob O. 2007. "Measures of Text Entry Performance." *Text Entry Systems*, 2007, 47–74. https://doi.org/10.1016/b978-012373591-1/50003-6.

Xu, Wenge, Hai-Ning Liang, Yuxuan Zhao, Tianyu Zhang, Difeng Yu, and Diego Monteiro. 2019. "RingText: Dwell-Free and Hands-Free Text Entry for Mobile Head-Mounted Displays Using Head Motions." *IEEE Transactions on Visualization and Computer Graphics* 25, no. 5 (2019): 1991–2001. https://doi.org/10.1109/tvcg.2019.2898736.

Yu, Difeng, 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, no. 11 (2018): 2927–35. https://doi.org/10.1109/tvcg.2018.2868581.