CS 5678 MASK: Mixed-Reality Adaptive Swype Keyboard

Ethan Chang[1], Xy Fang[2], Richard Pan[3]
[1]Computer Science, Cornell Tech
[2]Connective Media, Cornell Tech
[3]Computer Science, Cornell Tech

Abstract – **Mid-air text entry has always been a core system control method and, therefore, an important research field for researchers in the virtual reality (VR) field. Due to limitations in physical haptic responses in VR, there is a need to research and develop innovative ways for users to input texts efficiently. This project proposes improvements to entering text in VR with a word-gesture keyboard using isomorphic inputs from handheld controllers by adjusting keyboard size and opacity. Based on these parameters, the results found significant changes in speed, accuracy, error rate, and user comfort.**

Keywords – **Gesture, Text, Input, Virtual, Reality, Controller**

I. INTRODUCTION

Text input has been a crucial part of human-computer interaction since the inception of consumer computing, at one point being the primary method for users to input instructions into their computers. While GUIs, mouse-based, and touchscreen interfaces have become increasingly popular, text-based input is still prevalent in specific workloads such as writing code, composing documents, and sending instant messages.

The conventional text-based input device for computers is the QWERTY keyboard, a holdover from the typewriter era that transferred into computing. Since then, a large user base has widely adopted the QWERTY keyboard and adapted it into touch-based, controller input-based, and VR-based forms. This broad adoption into different mediums preserved user familiarity above everything else to decrease learning times. While this approach has made text input approachable for users of all kinds, it has bottle-necked typing efficiency, especially in immersive VR scenarios.

In head-mounted VR, text input is a cumbersome and frustrating task that cannot be entirely avoided because many user experiences require inputs. Different forms of hardware paradigms have different optimal interaction methods. If VR is to mature into a widespread medium, then VR optimized methods must be developed to improve user experience and efficiency.

II. RELATED WORK

Previous work has shown the potential of gesture-based mid-air text entry on virtual keyboards [1]. Generally, there are two types of implementations for this – hand-tracking and controller tracking. Markussen et al. has shown that the input method could achieve typing speeds of 20 WPM for novice users, improving over traditional selection based input. Their implementation used hand tracking to map the user's movements onto a larger screen in front of them. Although mid-air gesture was faster than selection, it was still slower than touch-based gesture input by a large margin. The input methods may be similar, but some of gesture-based-input's advantages did not translate from a touch screen to mid-air. Markussen suspected that this could result from the decoupling of the input plane and display plane in mid-air text entry.

Chen et al. used VR controllers to implement a similar gesture-based text entry. However, they had a different control-display mapping utilizing ray-casting to point at a virtual keyboard plane in front of the user. This implementation achieved text entry speeds of up to 32 WPM. The faster typing speed could have been due to the visible raycasts, which shows that visual feedback is an essential factor even in gesture-based text entry. Chen et al. also compared this to using a separate touchscreen for gestures which resulted in a lower entry speed of 22.4 WPM further supporting the idea that a difference in the input plane and display plane would lead to higher mental demand and lower text entry rates [2].

Other methods have also been used to track user gestures, such as using a ring in Rotoswype [5], and head and gaze-based gestures [7]. However, these methods were slower than other selection and gesture-based methods and are mainly used when the need arises for hands-free text entry.

Another primary concern with gesture-based input is arm fatigue, even if interaction times are short. Juan et al. investigated a metric to quantify arm fatigue of mid-air interactions. They found a significant increase in exertion when the plane size was increased for a selection task thus requiring larger movements [6]. Wang et al. investigated the use of different postures that could influence the user's perception of fatigue [3]. They tested a hand up posture and a hand down posture and found that surprisingly the raised hand posture required less perceived exertion while being more accurate. This could be caused by increased visual guidance and hand-eye coordination in the raised hand posture allowing users to make faster movements with less cognition.

These papers have shown that increasing the cognitive load on users could potentially lower the efficiency of text entry. They have also suggested that disparities between visual cues and gestures would negatively affect performance. Our study aimed to design a gesture-based keyboard that

III. Hypotheses

This study tested the hypothesis that the size of the keyboard affects accuracy and speed, specifically that the

ec687@cornell.edu, xf48@cornell.edu, rp484@cornell.edu

smaller the keyboard is, the faster the user will be able to type at a similar accuracy. This hypothesis was assumed to be valid because more minor gestures lead to less distance traveled over the same time, thus increasing speed.

In addition, this study also hypothesized that gesture keyboards would generally increase the user's ability to type without constantly looking at the keyboard after an intermediate learning period. The expected results were to see a user's performance increase as the keyboard opacity decreases and time spent typing increases.

IV. METHOD

A. Approach

Ten participants were recruited for this study, all aged 21-26, with 50% female and 50% as male. Additionally, all participants were right-handed. In general, the participants regarded themselves as less experienced with VR typing but more experienced with some form of gesture typing, such as Swype for mobile phones.

The experiment's approach was to build a testing environment with VR HMD software with a parameter adjustable gesture keyboard on which our users performed a typing test. To accurately measure the effects of gesture typing in VR, the user was given free range of movement in 6 degrees of freedom. The keyboard was also constructed as 3D blocks instead of on a flat plane to account for the extra dimension, which is not relevant to mobile input.

The study was split into two parts, and the user was in the seated position during the duration of both tests. Each keyboard was designed to appear at around chest height 0.5m away from the user due to the results of this other study; then, the user was asked to adjust their seat position until they were in their most comfortable typing position. Users were given 5 minutes prior to the start of the tests to acclimate themselves to the virtual environment and the WGK. In both tests, our users were given the same set of 12 phrases split into four unique phrases per set and asked to type them using gestures as quickly and accurately as possible.

The first part of the study focused on testing the effect of keyboard size on speed and accuracy. The virtual screen where the user-typed text appeared remained at a consistent position around 2 meters away from the user. Three size configurations were tested: The first at a 'default' size of 0.4m long, the second 'small' at 0.2m, and a third 'large' keyboard with a length of 0.6m. These keyboards were set up for the user in random order to normalize for users becoming more familiar with VR gesture typing over time. Users were asked to hold down a delimiter on their primary controller (right-handed for all of the testers in this study) to start the input, then draw the shape of the word on the keyboard by tracing out the path to the letters, and finally releasing the delimiter to input the predicted word. Spaces were inputted automatically, and users were not allowed the option to delete characters. If the predictive algorithm could not find a related word to input, no word was inputted into the field, and users were given the option to try again or skip the word.

Users were given a ten-minute break between part 1 and part 2 of the study to recover from any physical exhaustion accrued while drawing the gestures with their arms in the first study.

The second study focused on the user's performance regarding visual stimuli. In theory, users should have developed some muscle memory over the words they have typed beforehand and should need fewer visual reminders of which keys they are inputting. Users were presented with the default keyboard size at opacity levels of 50%, 10%, and 0%, with the target text box and inputted text box remaining at 100% opacity. The 50% and 10% keyboards were presented randomly to normalize for learning. However, the 0% opacity keyboard was always presented last because it had the highest learning requirements. Due to the VR implementation not providing haptic feedback on touches as mobile input does, users were given an outline of the keyboard without keys drawn locations to guide their inputs.

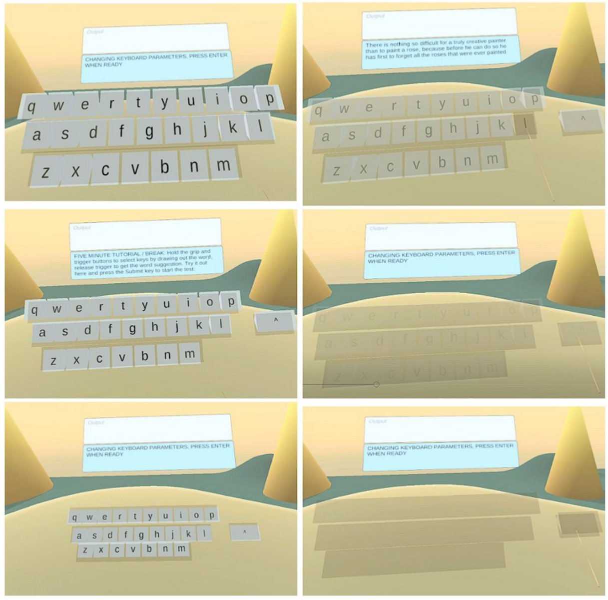

Fig. 1. Six Keyboard Conditions

B. Hardware and Software Requirements

This study used the Oculus Quest 2 headset with default controller inputs. The experiment software was developed in Unity using C# and the Unity XR Interaction toolkit. The assets from the gesture keyboard were derived from CutieKeys, an open-source VR mallet keyboard developed by NormalVR. The "gesture" interpreter used was the Damerau–Levenshtein string distance, a distance between words in a generated dictionary and actual raw characters the user inputted by dragging their controller across the keyboard. Damerau–Levenshtein is a recursive algorithm that takes substrings of every dictionary input and the raw input and adds a distance value for every string modification to those two strings. A list of distance values associated with dictionary words is generated for each stroke, and the shortest distance is inputted in the study implementation. This study uses an open-source implementation of this algorithm.

V. EVALUATION

MASK primarily focused on measuring typing speed, accuracy, and error rate, as optimizing these metrics is typically a sign of a more performant typing experience. Since Swype typing styles are inputted as sequences of shapes from word to word rather than individual characters, it is more representative to use metrics describing words rather than characters. The users' performance was measured using Words Per Minute (WPM), where WPM = words/mins. Accuracy (Acc). In this case, accuracy is the percentage of words that fully match the target words. Finally, to measure the error rate, the number of words typed that was incorrect was added to the number of target words missed and then divided by the length of the target sentence. After each independent variable segment, we presented the user with a qualitative survey asking them about their perceived performance, comfort levels, exhaustion, and frustrations with each implementation.

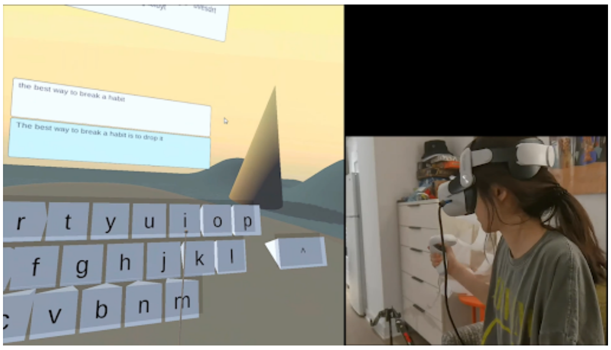

Fig. 2. Participant Interaction Example

VI. RESULTS

A. Text Entry Rate

The figure shows the mean text entry rate for each parameter tested. An RM-ANOVA test used the between treatments method to examine the significance between treatment groups. It found a significant difference between the text entry rates of the various keyboard sizes (F = 9.541, $p < 0.05$). The default keyboard performed 12% faster when compared to the large keyboard and 34% faster when compared to the small keyboard.

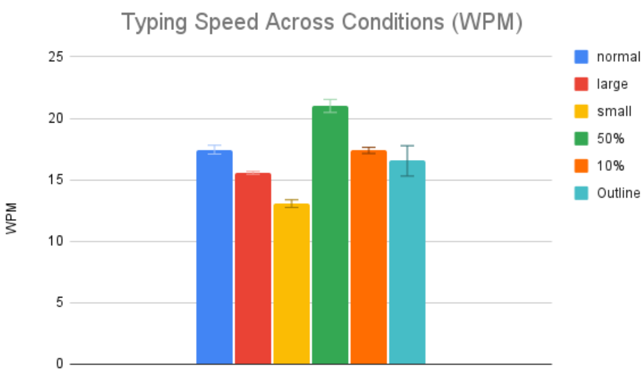

Fig. 3. Text Entry Rate across conditions

There was also a significant difference in the text entry rates of the keyboards of different opacities (F = 5.677, $p < 0.05$). The 50% opacity keyboard performed 21% better when compared to the 10% opacity keyboard and 27% better when compared to the outline-only keyboard. However, four out of ten participants did not finish the task with the outline only keyboard. Due to this, the outline-only keyboard was mainly left out of the rest of the data analysis.

B. Accuracy

An RM-ANOVA test used the between treatments method to examine the significance of the difference in accuracy between treatment groups. It found no significant difference in accuracy for the different keyboard sizes (F = 1.466, p = 0.26). There was also no significant difference in accuracy for the different opacities (F = 0.085, p = .78).

C. Error Rate

An RM-ANOVA test using the between treatments method was used to examine the significance of the difference in error rate between treatment groups. It found no significant difference in error rate for the different keyboard sizes (F = 1.502, p = 0.25). There was also no significant difference in error rate for the different opacities (F = 0.021, p = 0.89).

	ACC	ERR	WPM	STDACC	STDERR	STDWPM
normal	0.9296969697	0.1091043898	17.44703211	0.05660425496	0.1026286925	2.057233828
large	0.8915025253	0.3768566156	15.5697888	0.02357022604	0.0260956074	0.754166006
small	0.8477384135	0.5873692069	13.05573558	0.02371202553	0.02024378285	2.383478501
50%	0.8834848485	0.1884609599	21.00967168	0.03696239993	0.04919129283	2.548286783
10%	0.873989899	0.1969904401	17.38272958	0.03106984342	0.04285495644	1.439622792
Outline	0.5604110963	2.089658182	16.53808357	0.261265557	4.387748985	7.472167086

Fig. 4. Assorted Quantitative Data

D. Qualitative Results

After finishing the test for each keyboard parameter, users were presented with a qualitative survey in which they described their experience with the keyboard and rated attributes on a scale of 105.

In terms of keyboard size, the mental exertion spent by users was reported to be the same per size, however users reported that a great amount of physical exertion was needed for the Large keyboard. Users were generally uniformly frustrated with the typing experience for each size, with the small size being having slightly more reports as triggering the maximum amount of frustration. Perceived typing performance and accuracy tended to hover around the median option. There was slight lean in users regarding the default size as most performant and most accurate. While there was no significant motion required by this experiment, 3 users reported motion sickness severity of 4 or higher. Finally, the small size had the most users rank it the highest of the 3, and the large size had the most users rank it the least favorite iteration.

For the opacity settings, the physical exertion was regarded as the same for all keyboards, which is consistent with the size staying the same. However mental exertion and frustration were highest on the outline only keyboard and lowest on the most opaque option. Users perceived their typing performance and accuracy to be higher when the keyboard was most opaque and incredibly low when it was invisible. A similar amount of people reported extreme motion sickness, and there was a strong preference for the most opaque option and an overwhelming dislike of the outline only keyboard.

Some relevant comments users left were that the "Smallest was a little too small but still way more comfortable than large size", they "started giving up after a while for all keyboards, especially invisible", and that the process in general was fatiguing.

VII. DISCUSSION

A. Limitations

The most significant limitation of the MASK implementation was that the input was not a genuine gesture to string transformation but instead a string to string transformation, with the gesture being a side effect of the string input. As a result, the user's input movements are bound to where the keyboard is placed, and the character inputs must be accurate. Due to the user's pose having more freedom in a VR, the user may expect to have the ability to input a string from any starting pose.

Some other limitations lie in the technical implementation. Participants did not receive any feedback when they completed a task or inputted a work successfully. This can create dissociative feelings for participants when interacting with this keyboard. The mallet was directly attached to the controller's position, but no hand or controller model was used. Some participants reported an impaired feeling when they could only see a mallet representing their hand.

Due to time constraints and participants' willingness to participate, we limited our design to only six sections with four sentences in each section. Since our design has two different factors (size and opacity), each factor contains three different levels. This research can be more valid with a three-by-three factorial design.

The lack of user time spent on this typing method is also a limitation. The original hypothesis assumed the learning of gestures to input words, which requires a longer time for the participants to interact with this system. This study provides participants with only five minutes to practice and less than an hour to complete the whole study, which might not be sufficient for them to familiarize themselves with this specific input method fully.

This study only implemented one specific interaction technique (mallet touching), which was used to compare with the commonly used ray-casting inputting method on a standard QWERTY keyboard. It is possible to have a more accurate comparison between different keyboard input methods if both mallet-touching and ray-casting can be implemented into the study and compared directly.

Lastly, the participant sample of this study was limited due to the pool of available applicants. All participants are graduate students in technical programs, while most of the group has experience with VR and Swype Keyboard. This can potentially influence the learning effect of the study's result.

B. Future Works

This study can be improved in several ways in the future. A modern word-gesture prediction system like SHARK can be implemented to increase the prediction accuracy and lower the frustration for participants. The current method is based on calculating distances between letters to predict the possible word generated from a strength. At the same time, modern Swype keyboards generally used a shape prediction system of the gesture.

In addition, minor technical issues can be fixed to improve the typing experience for participants from aspects like auditory and haptic feedback, hand models, and trigger design to lower the required physical effort.

A future direction of this study is to implement the swype keyboard with ray-casting techniques, which can be better compared with traditional tapping keyboards with ray-casting in VR. This could also lead to future implementations on testing different distances between hand and keyboard. Faraway keyboard may require less hand movement, but might be at a cost of accuracy.

The study in the future can also expand to multiple days of practicing and measuring to track participant's learning curve over time. The current result suggests a 17 WPM during the first time participants ever interacted with this system. It is possible that participants can type faster after practices in another couple of sessions. The result has already been comparable with traditional VR ray-casting typing method in terms of speed.

C. Conclusion

This paper presents a exploration on a novel input technique in VR using word-gesture keyboard. A range of different conditions from different sizes of the keyboard to different opacity were tested across the study. Participants have reported a fastest typing speed when the keyboard is at its default size, and 50 percent opacity. The highest WPM across conditions is 21 while the lowest WPM across conditions is 13 (when the keyboard is small). This suggests that both keyboard size and keyboard opacity matters in certain range instead of what we hypothesized. Future works can continue test the best combinations of keyboard size and opacity for a VR swype keyboard to find the best keyboard design in promoting typing speed for users.

VIII. REFERENCES

[1] R. Qin, S. Zhu, yu-H. Lin, Y.-J. Ko, and X. Bi, "Optimal-T9: An Optimized T9-like Keyboard for Small Touchscreen Devices," Nov. 2018, pp. 137–146. doi: 10.1145/3279778.3279786.

[2] A. Nguyen, S. Bittman, and M. Zank, "Text Input Methods in Virtual Reality using Radial Layouts," in 26th ACM Symposium on Virtual Reality Software and Technology, New York, NY, USA, Nov. 2020, pp. 1–3. doi: 10.1145/3385956.3422114.

[3] H. Jiang and D. Weng, "HiPad: Text entry for Head-Mounted Displays Using Circular Touchpad," in 2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Mar. 2020, pp. 692–703. doi: 10.1109/VR46266.2020.00092.

[4] C. Boletsis and S. Kongsvik, "Controller-based Text-input Techniques for Virtual Reality: An Empirical Comparison," Int. J. Virtual Real., vol. 19, no. 3, Art. no. 3, Oct. 2019, doi: 10.20870/IJVR.2019.19.3.2917.

[5] A. Gupta, C. Ji, H. Yeo, A. Quigley, and D.l Vogel. "RotoSwype: Word-Gesture Typing using a Ring," in Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, New York, NY, USA, 2019, Paper 14, 1–12. doi: 10.1145/3290605.3300244

[6] J. Hincapié-Ramos, X. Guo, P. Moghadasian, and P. Irani. "Consumed endurance: a metric to quantify arm fatigue of mid-air interactions," in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, New York, NY, USA, 2014, 1063–1072. doi: 10.1145/2556288.2557130

[7] J. Hansen, K. Tørning, A. Johansen, K. Itoh, and H. Aoki. "Gaze typing compared with input by head and hand," in Proceedings of the 2004 symposium on Eye tracking research applications, New York, NY, USA, 2004, 131–138. doi: 10.1145/968363.968389