HawKEY: Efficient and Versatile Text Entry for Virtual Reality

Duc-Minh Pham
Simon Fraser University
Surrey, British Columbia, Canada
ducminhp@sfu.ca

Wolfgang Stuerzlinger
Simon Fraser University
Surrey, British Columbia, Canada
w.s@sfu.ca

Figure 1: *First row - HawKEY*: RGB-D camera attached to VR headset (left), stabilizing straps (middle), and text entry task (*right*). *Second row - VR* Text Entry Interfaces (from left to right): *None, Frame, Model, Video,* and *Point Cloud.*

ABSTRACT

Text entry is still a challenging task in modern Virtual Reality (VR) systems. The lack of efficient text entry methods limits the applications that can be used productively in VR. Previous work has addressed this issue through virtual keyboards or showing the physical keyboard in VR. While physical keyboards afford faster text entry, they usually require a seated user and an instrumented environment. We introduce a new keyboard, worn on a hawker's tray in front of the user, which affords a compact, simple, flexible, and efficient text entry solution for VR, without restricting physical movement. In our new video condition, we also show the keyboard only when the user is looking down at it. To evaluate our novel solution and to identify good keyboard visualizations, we ran a user study where we asked participants to enter both lowercase sentences as well as complex text while standing. The results show that text entry rates are affected negatively by simplistic keyboard visualization conditions and that our solution affords desktop text entry rates, even when standing.

CCS CONCEPTS

• **Human-centered computing → Text Entry**; **Virtual Reality**; **Mixed / augmented Reality**.

KEYWORDS

Text Entry, Virtual Reality

ACM Reference Format:
Duc-Minh Pham and Wolfgang Stuerzlinger. 2019. HawKEY: Efficient and Versatile Text Entry for Virtual Reality. In *25th ACM Symposium on Virtual Reality Software and Technology (VRST '19), November 12–15, 2019, Parramatta, NSW, Australia.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3359996.3364265

1 INTRODUCTION

The recent introduction of consumer-grade head-mounted displays (HMDs), has made experiencing virtual environments (VEs) more affordable, even for average users. Thus, the applications for Virtual Reality (VR) have expanded and currently include many design, entertainment, training, and immersive analytics scenarios. Despite considerable advances in other VR technologies, text entry is still a challenge in VR, also because users are unable to see the physical keyboard. While solutions have been presented, VR typing performance is still well below non-immersive alternatives.

While controllers, touchpads, or joysticks are suitable for applications that require simple input, such as choosing an option or direction, they are insufficient for entering larger amounts of information, such as whole sentences. This lead to the integration of virtual keyboards in recent VR systems. Users can then select keys on a virtual keyboard with two handheld controllers or

VRST '19, November 12–15, 2019, Parramatta, NSW, Australia

ACM ISBN 978-1-4503-7001-1/19/11...$15.00
https://doi.org/10.1145/3359996.3364265

built-in touchpads to enter text. Such solutions afford typing in a "hunt-and-peck" style, but currently do not approach the text entry performance afforded by regular keyboards. We focus on physical keyboards and how users can efficiently use them in VR.

Most virtual keyboards require users to look at them to enter text. Physical keyboards provide not only haptic feedback but also afford eyes-free 10-finger touch typing for experienced typists. Yet, novices typically do not know the where every character or symbol is and even expert typists might need to look at the keyboard to locate uncommon keys like "{" or "~". Thus, users still need to see the keyboard and their hands to quickly find and activate such keys.

Visual feedback also helps users reach higher text entry performance than solely haptic feedback [Walker et al. 2017]. For this both the keyboard and the user's hands need to be visible in the VE [Grubert et al. 2018a; Knierim et al. 2018]. Yet, the VR representation of the physical keyboard should obstruct the VE as little as possible. Also, many current virtual keyboards occupy a significant part of the visual field, even if the keyboard is not being used. Here, we explore physical keyboard representations ranging from a very minimal form to fairly detailed visuals. We aim to identify the most appropriate representation that is not distracting, while still enabling eyes-free typing.

Previous work has proposed several methods to track and visualize the appearance of physical keyboards and user hands in VR. Most of them require modification of the user's environment, such as a green desk [McGill et al. 2015], markers on the hands [Grubert et al. 2018b; Knierim et al. 2018], or special keyboard covers [Jiang et al. 2018]. We aim for fewer modifications of the user's environment, while still affording efficient text entry. Thus, we only attach an RGB-D camera to the HMD and require only a keyboard tray to be worn.

There are several scenarios where users may have to mix physical movement (for VR navigation) and text entry at the same time. One example is an engineer annotating and commenting aspects inside a virtual 3D model. Another is in-game chatting in a collaborative VR game, or interacting with many aspects of the game environment through keyboard shortcuts. Also, an analyst may need to search for textual information in immersive analytics software to derive insights from a dataset shown in a virtual 3D space. With a portable keyboard, all these users would not be tied to any physical desk, and hence, be able to provide comments, messages, annotations, captions, or even small reports, as desired, directly from inside the VE. Therefore, and while previous work requires users to sit at a desk to enter text inside a VE [Jiang et al. 2018; Lin et al. 2017], we propose a new portable keyboard setup, *HawKEY*, that allows users to either sit or stand while still being able to enter text. HawKEY's design is light-weight and the tray is easy to put on and take off. Here, we also evaluate HawKEY in a user study.

Recent VR text entry studies involved only lowercase phrases [Knierim et al. 2018; Lin et al. 2017; McGill et al. 2015; Walker et al. 2017]. Yet, for many applications or keyboard shortcuts, the set of required letters and symbols is larger. Also, VR text editors or immersive analytics software may require the input of uppercase letters, numbers, or punctuation. Previous works has not addressed the effect of such unfamiliar characters. Thus, we evaluated the efficiency of HawKEY with tasks where the users had to either type *lowercase-only* or *more complex* phrases.

We present the following contributions:

- HawKEY, a portable keyboard, suitable for high-speed text entry while standing in VR, but also usable when sitting or walking.
- An investigation of which keyboard visualization methods are most beneficial.
- An investigation how unfamiliar characters affect text entry speed and error rates in VR.
- A simple method to make a video-based visualization of the physical keyboard only visible when the user is looking at it.

2 RELATED WORK

Previous work has investigated various approaches to text entry in VR, see a recent review [Dube and Arif 2019]. Some explored voice or gesture recognition to input words and phrases, but did not identify reasonably good performance [Bowman et al. 2002; Hsieh et al. 2016; Kuester et al. 2005; Prätorius et al. 2015]. Others showed not only the presence of the keyboard in the VE but also provided visual and/or haptic feedback [Grubert et al. 2018a; Gugenheimer et al. 2016; Knierim et al. 2018; Yi et al. 2015].

2.1 Virtual Keyboards

Virtual keyboards are the most common text entry interfaces for VR, as they are easy to implement. They vary in how the user selects a key on the virtual keyboard. Commercial systems, like the Oculus Rift and HTC Vive, use key selection mechanisms based on a virtual ray manipulated by VR controllers. With a Microsoft HoloLens key selection is controlled by head direction. Google presented a keyboard where users use two controllers to hit keys like drums, with vibration feedback [Doronichev 2016]. Boletsis et al. [Boletsis and Kongsvik 2019] also conducted a preliminary evaluation of such drum-like keyboard and their participants achieved 24.61 WPM. All these solutions typically achieve low text entry rates, even though an expert users was able to reach 50 WPM with the Google keyboard.

ATK [Yi et al. 2015], a 10-finger mid-air typing interface, tracked the fingers of the user's hands with a Leap Motion, affording up to 29.2 WPM. Vulture, a word-gesture virtual keyboard, uses optical tracking to determine the users' gestures for enable key selections [Markussen et al. 2014]. It allowed users to achieve up to 28.1 WPM after a training section. Dudley et al. introduced VISAR, a virtual keyboard, where users imitated the process of single-hand typing on physical touchscreens. Utilizing hand recognition and autocorrection, the system afforded 17.75 WPM [Dudley et al. 2018]. Yu et al. [Yu et al. 2017] investigated head-based text entry techniques, which enabled users to type as fast as 24.73 WPM with gestures.

HoVR [Kim and Kim 2017], a soft keyboard on a smartphone with hover capabilities, mirrored the keyboard into the VE to provide visual feedback, and achieved up to 9.2 WPM. With a touch cover attached to the front of the HMD, the FaceTouch system enabled typing with up to 10 WPM [Gugenheimer et al. 2016]. PizzaText [Yu et al. 2018] utilized a pair of thumbsticks and a pizza-like keyboard layout, which allowed novice typists and experts to achieve 8.59 WPM and 15.85 WPM, respectively.

Table 1: Typing performance (WPM) in baseline and the best VR conditions in previous work and HawKEY. Complex text involves numbers, punctuation, and uppercase. Green shows the best-performing VR options.

Study	Pose	Typist	Text	Baseline	VR
[Jiang et al. 2018]	seated	-	lowercase	32.5	23.1
[McGill et al. 2015]	seated	-	lowercase	58.9	38.5
[Grubert et al. 2018a]	seated	-	lowercase	-	38.7
[Lin et al. 2017]	seated	-	lowercase	39.8	28.1
[Doronichev 2016]	standing	expert	lowercase	-	50.0
[Knierim et al. 2018]	seated	expert	lowercase	67.2	69.2
	seated	novice	lowercase	45.4	39.8
HawKEY	standing	expert	lowercase	76.1	77.7
	standing	novice	lowercase	50.3	44.1
HawKEY	standing	expert	complex	45.7	41.5
	standing	novice	complex	27.9	21.6

2.2 Mixed Reality Solutions

The HiKeyb system [Jiang et al. 2018] segmented the user's hands with an RGB-D camera and showed them on the VR model of a tracked physical keyboard, which produced entry and error rates close to a real world baseline.

Recent work used optical tracking to track markers on the seated user's fingers/hands. Knierim et al. [Knierim et al. 2018] combined a virtual model and hand representations with different levels of detail in VR. Experienced typists benefited from the hand model conditions and were able to reach up to 69.2 WPM, comparable to their real world condition, and outperforming a no hand condition. Inexperienced typists also profited from the hand models, but still performed worse than the baseline. Grubert et al. [Grubert et al. 2018a] added a condition that shows video of the hands and physical keyboard. They found no significant difference between video hand, no hand, tracked hand model, and tracked fingertips.

McGill et al. [McGill et al. 2015] captured the physical keyboard and the user's hands with a RGB camera attached to the HMD's front. They compared four conditions: reality baseline; no keyboard in VR; partial blending where keyboard and hands were somewhat visible in the VE; and full blending which showed the real image. They found a significant effect of blending (partially or fully) over the no keyboard condition. However, the entry rates of blending conditions were still not comparable with the baseline. Follow-up work by Lin et al. [Lin et al. 2017] identified no difference between full blending, no keyboard, VR keyboard model with no hand, VR keyboard model with segmented real hands, and baseline reality. However, full blending and real hand conditions significantly reduced the error rate compared to no keyboard visualization.

Table 1 shows the typing performance of previous work and our study in words per minutes (WPM). It lists the real-world baseline and the VR conditions with the highest WPM.

3 STUDY DESIGN

We designed a two-factor within-subjects experiment to investigate text entry typing performance. The first factor, Text Entry Interface, has six levels including a real-world baseline and five different keyboard representations in VR. The second factor, Text Complexity, has two levels: one with only lowercase letters, and another that includes uppercase letters, punctuation, and numbers.

3.1 Text Entry Task

We evaluated the performance of different Text Entry Interfaces in a simple text entry task. Participants wore our HawKEY keyboard and copied a single target sentence shown on a virtual panel in each trial. The panel was positioned 1.5 meter away from the experimental area center, where the participant stood initially. When the text appeared, they typed it into a text box located just below that panel, see figure 1. With this setup, they could always see the presented text and the text box simultaneously.

We decided to enforce error correction, i.e., participants were not allowed to make any errors in their transcription. We chose this protocol, as research has shown that there is no significant difference in term of entry and error rates when correction is *recommended* or when it is *(en-)forced* [Arif and Stuerzlinger 2009].

Each trial automatically completed when the last character of the presented sentence was successfully transcribed by participants. Then they needed to wait 10 seconds until the next trial, when the next target sentence appeared on the panel. This gave participants a short break between tasks.

3.2 Text Entry Interfaces

The Text Entry Interface factor includes six conditions, with the first being in the real world, while the remaining ones are in VR, see figure 1.

- *Baseline:* This condition investigated the normal typing performance of participants. They wore HawKEY but no HMD and entered text in front of a large physical display.
- *None:* Participants enter text in VR without seeing the keyboard. This condition investigates the touch typing performance of our participants. To address the potential confound of hand visibility, we show the participants their hands as 3D point clouds, captured by the RGB-D camera on the headset.
- *Frame:* In this condition a rectangular frame represents the position of the physical keyboard in VR. With this, we examine how a minimalistic representation affects typing performance. Hands were visible as 3D point clouds.
- *Model:* We show a virtual keyboard model that matches the dimensions and appearance of the physical one. Participants again see their hands. By mixing virtual and physical content, we aim to discover if this combination is beneficial.
- *Video:* Here, we used the RGB-D camera to capture 2D video of the physical keyboard and display it on a 2D surface in VR, at a fixed position relative to the participant's body, corresponding to the tray's location. This video only appears when participants rotate their head (down) to look at the keyboard. As they can see their hands in the video, there is no 3D point cloud in this condition.
- *Point Cloud:* Here, both the participant's hands and keyboard are shown as a point cloud in VR. A simple depth clip ensures that only sufficiently close content is visible. Due to technical

limitations, the point cloud has less resolution and clarity than the the video condition.

Both *Video* and *PointCloud* do not require tracking the keyboard and are technically simpler. This enables us to discover potential differences between 2D and 3D mixed-reality visualizations for typing. We chose not to show the user's hand as a point cloud in the *Video* condition, as this creates "double-images". Even when segmenting the hands out of the video, there were still too many artifacts to make this a viable approach. We also investigated *Frame* and *Model* to inspect the benefit of minimalistic or more realistic virtual representations of the keyboard. Finally, *None* serves as a VR baseline condition to examine how typists perform when the keyboard is not visible.

3.3 Text Complexity

The complexity of the text in terms of familiarity with the involved characters may affect task difficulty. For some typists, unfamiliar characters or key combinations might take longer to enter correctly. Hence, we examine two types of sentences in our study.

Simple Sentence: In this condition, we present only sentences consisting of *lowercase* (English) alphabetical characters. These characters are very familiar to people who use a computer frequently and require only a single keystroke each. Through this condition, we also aim to evaluate touch typing performance with different TEXT ENTRY INTERFACES.

Complex Sentence: Typing becomes more challenging when uncommon characters, such as colons and brackets, appear or when modifier keys, such as *Shift*, are required. With this condition we investigate how typists deal with more challenging text entry tasks through different TEXT ENTRY INTERFACES. A complex sentence includes lowercase and uppercase alphabetical characters, digits, parentheses, spaces, punctuation marks, and other symbols. Only the *Shift* modifier key is required. An example of a complex sentence is: "Corporate income tax revenues increased by $5.6 billion, or 13.2%!"

Participants were restricted to keys with printable characters, *space bar*, *Shift*, and *Backspace*. All other keys, including *Caps lock*, *arrows*, *Tab*, and *Ctrl*, were disabled or ignored. The purpose of these restrictions was to avoid unwanted behaviors and increase accuracy. Only a single sentence with max. 20 words was presented at a time, which makes editing with cursor *arrows* mostly unnecessary for corrections. Words rarely contained more than a single uppercase character.

4 EXPERIMENTAL SETUP

4.1 Subjects

We recruited 16 participants from the local university for our study (7 female). Ages ranged from 19 to 30 years (M = 22.7, SD = 3.3). All of them had tried VR, but did not use it regularly. They earned 1% course credit through this study.

4.2 Apparatus

We used a Windows 10 PC with an Intel® Core™ i7-4790, 16GB, and a nVidia GTX 1080. As HMD, we used a HTC Vive Pro, with

Figure 2: *Top-left*: Baseline condition. *Bottom-left*: Using HawKEY while sitting. *Right*: A participant standing and wearing HawKEY and the VR headset during the study.

2880×1600 pixels at 90 Hz, see figure 2. The horizontal field of view is approximately $100°$.

For the baseline condition we used a 85" display on a stand, 1.5 m from the user, see figure 2. We showed the panel containing the presented text and the text box for transcription at the same (relative) locations as in the VE.

Similar to other work [Jiang et al. 2018], we used an Intel® RealSense™ D435, see figure 1, to capture RGB-D video, and used it to collect 2D or 3D video of the user's hands and optionally the keyboard. The horizontal field of view of its sensors is $\geq 70°$, wide enough to capture the whole keyboard tray.

We used a Logitech K480 Bluetooth keyboard, see figure 2, which is small enough to be fully captured by the RBD-D camera, enabling us also to show the user's hands at both sides of the keyboard. In our setup, the average end-to-end latency was 36 ms. To enhance key readability while still preserving the tactile cues on the "F" and "J" keys, we used customized stickers with larger symbols. The lack of a numeric keypad also makes input deterministic.

4.2.1 HawKEY Design. Inspired by a hawker's tray, HawKEY uses a tray-based design to provide versatile text entry, see figure 1. A hip strap attaches the tray to the user and braces a surface for the keyboard to rest on. Two adjustable shoulder straps further stabilize it roughly parallel to the ground. We attached a controller to the tray to track the keyboard's pose, which enables us to show its virtual representation. HawKEY can also be used while sitting in a chair away from a desk, see figure 2. We recognize that users will often also use controllers in VR. Thus, when typing, we recommend that users either dangle these on their wrist-straps or place them onto the top part of the keyboard tray.

4.3 Phrase Set

Inspired by the creation process for MacKenzie's phrase set [MacKenzie and Soukoreff 2003], we collected 67 *Simple* and 72 *Complex* sentences on various topics. *Simple* sentences had a mean of M = 65.7

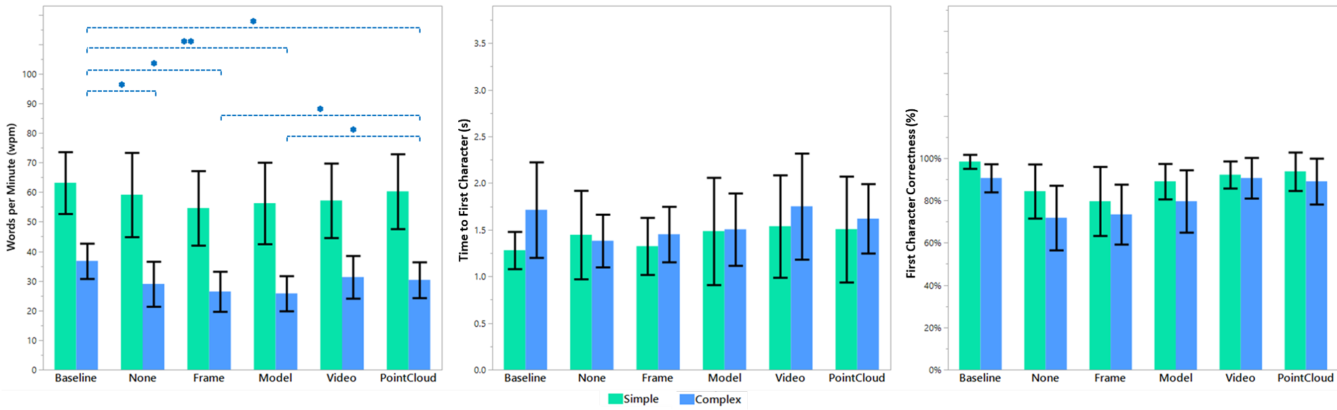

Figure 3: Average text entry measures (*left*: words per minute, *middle*: time to first character, *right*: first character correctness) with different keyboard representations in real and virtual environments ($^*p \leq .05$, $^{**}p < .01$, $^{***}p < .001$).

characters, $SD = 12.9$ (13.6 words, SD 2.6). *Complex* ones had $M = 59.0$ characters, $SD = 12.3$ (11.8 words, $SD = 2.5$). For each trial, we selected a *random* sentence from either set.

Our set of *Simple Sentences* is similar to MacKenzie's phrase set [MacKenzie and Soukoreff 2003], as used in previous work [Knierim et al. 2018; Lin et al. 2017; McGill et al. 2015]. To characterize our *Complex Sentence* set, we computed the ratio between the number of *"complex"* characters, i.e., characters that are not *lowercase alphabetical letters* or *spaces*, over the total characters. This ratio had a mean of $M = 19\%$, $SD = 8\%$. In addition, some of these *complex* characters required the *Shift* modifier key, $M = 9\%$, $SD = 5\%$ of the total.

4.4 Text Entry Conditions

There were six Text Entry Interface conditions. The first served as the baseline and took place in the real world. The remaining ones used VR. To ensure comparability, we set up the baseline to be as similar to the VR conditions as possible. There were eight repetitions for each Text Entry Interface. Four of them presented *Simple Sentences*, while the others four involved *Complex* ones. These two conditions of *Text Complexity* were counter-balanced among participants.

We chose to evaluate only standing conditions, as many VR scenarios assume that the user can move around freely, which is not possible in front of a desk.

4.4.1 Baseline condition. Here, participants stood at 1.5 meter distance to the large display, on which they saw the text panel in the same 3D environment. Then they performed the baseline text entry task while standing. This gave them a chance to get familiar with HawKEY and also served as an text entry performance measurement. As all recruited participants were very familiar with keyboards, we decided to run this baseline condition always at the beginning of the experiment and did not counter-balance this condition with the others.

4.4.2 VR conditions. Then, participants experienced all five VR conditions. In the VE, text was shown on a virtual panel located at a distance of 1.5 meters. They were asked to stand at the center of the experimental area and wore HawKEY and a Vive Pro headset to perform the task. The order of all five VR conditions was counterbalanced. We enforced a break of 10 seconds between sentences.

4.5 Procedure

We first asked each participant to read and sign the consent form and fill our pre-study survey. Then, we explained the purpose of the study, introduced HawKEY, and demonstrated the experimental text entry task. They were given 5 minutes to try HawKEY. The experimenter helped them to adjust the straps to ensure that they could type comfortably.

For each Text Entry Interface condition, participants were given at least two practice sentences to familiarize themselves with the keyboard appearance. After that, they proceeded to the experimental trials where we recorded data for analysis. At the end of each condition, they filled a NASA-TLX [Hart and Staveland 1988] and a questionnaire to record ratings and comments. Each condition took around 10 minutes and there was a final post-study survey at the end. Depending on the participant's typing performance, it lasted between 60 to 90 minutes.

5 RESULTS

We performed two-way repeated-measures ANOVA on all collected measures. Greenhouse-Geisser (if its epsilon was smaller than .75 [Ellen R. 1992]) or Huynh-Feldt correction was applied whenever Mauchly's test of sphericity was violated. A Shapiro-Wilk test could not reject that the data for word per minutes (WPM) was normally distributed across all combinations of the two factors. However, the data for other measures were non-normal. Hence, we applied Aligned Rank Transform [Wobbrock et al. 2011] on the data before performing ANOVA.

We were also interested in how different interfaces could help participants to overcome the challenge of *Complex Sentences*. Therefore, if the effect of Text Complexity was significant, we examined *Simple* and *Complex Sentences* separately with one-way repeated-measures ANOVA. In the following analysis, P and T denote the presented and transcribed text, respectively. To present all results compactly, we list statistical results in tables 2 and 3.

Table 2: Two-way Repeated-Measures ANOVA on TEXT ENTRY INTERFACE (TEI), TEXT COMPLEXITY (TC), and their interaction (TEI×TC) (*$p \leq .05$, **$p < .01$, ***$p < .001$, insig. $p > .05$).

Measure	Factor	ANOVA				
		df1	df2	F-value	p-value	η^2
WPM	TEI	5.000	75.000	5.453	***	.267
	TC	1.000	15.000	120.512	***	.889
	TEI×TC	5.000	75.000	2.102	insig.	.123
Time to First Character	TEI	5.000	75.000	.497	insig.	.032
	TC	1.000	15.000	10.882	**	.420
	TEI×TC	5.000	75.000	.535	insig.	.034
First Character Correct.	TEI	3.287	49.306	4.995	**	.250
	TC	1.000	15.000	8.498	*	.362
	TEI×TC	5.000	75.000	1.690	insig.	.101
KSPC	TEI	5.000	75.000	21.897	***	.593
	TC	1.000	15.000	192.797	***	.719
	TEI×TC	5.000	75.000	23.519	***	.611
EKS ER	TEI	2.737	41.062	18.522	***	.553
	TC	1.000	15.000	99.900	***	.869
	TEI×TC	5.000	75.000	19.900	***	.570
Total ER	TEI	2.606	39.095	16.642	***	.526
	TC	1.000	15.000	97.305	***	.866
	TEI×TC	5.000	75.000	13.121	***	.467

Table 3: Analysis of different TEXT COMPLEXITIES on TEXT ENTRY INTERFACES for entry rate and first character (*$p \leq .05$, **$p < .01$, ***$p < .001$, insig. $p > .05$). For averages across *Simple* and *Complex*, see table 2.

Measure	TEXT COMP.	ANOVA on TEXT ENTRY INTERFACES				
		df1	df2	F-value	p-value	η^2
WPM	Simple	3.210	48.155	2.044	insig.	.120
	Complex	5.000	75.000	9.918	***	.398
Time to First Character	Simple	5.000	75.000	.183	insig.	.012
	Complex	2.725	40.872	1.265	insig.	.078
First Char. Correctness	Simple	5.000	75.000	1.964	insig.	.116
	Complex	5.000	75.000	3.500	**	.189
KSPC	Simple	2.683	40.245	5.339	**	.262
	Complex	5.000	75.000	19.045	***	.559
EKS ER	Simple	2.658	39.876	5.261	**	.260
	Complex	5.000	75.000	18.079	***	.547
Total ER	Simple	2.680	40.193	5.094	**	.254
	Complex	2.879	43.191	19.619	***	.567

5.1 Entry Rates

Entry rate measures enable us to understand how efficiently users interact with different TEXT ENTRY INTERFACES. In this study, we use words per minute (WPM) to characterize how fast users transcribe the text.

5.1.1 Words per Minute (WPM). WPM is one of the most commonly used metrics for text entry tasks. A word is defined as five characters, including the space. Thus, *WPM* does not account for the number of keystrokes or how users performed corrections and only considers the number of characters in the transcribed text. Its is defined as $WPM = \frac{|T|-1}{S} \times \frac{1}{5} \times 60$, where $|T|$ is the length of the transcribed text, and S is the time in seconds between the first and last character entry.

We observed significant effects for both factors, but their interaction was not significant, see table 2. Investigating *Simple* and *Complex Sentences* separately, ANOVA revealed that the effect of TEXT ENTRY INTERFACE was only significant for *Complex* ones, see table 3. Holm-Bonferroni corrected post-hoc tests on the *Complex Sentences* showed that only *Video* was not different from *Baseline*, while *Model* and *Frame* were slower than the *PointCloud*, see figure 3.

5.2 First Character Statistics

We also recorded several metrics for the first character to help us to identify potential differences between interfaces in terms of finding and typing the first character.

5.2.1 Time to First Character: The time from the presentation of the target sentence until the first character entered by participants illustrates how quickly they can locate the first key and press it. This metric is measured in seconds. ANOVA revealed that the effect of TEXT COMPLEXITY was significant, see table 2.

5.2.2 First Character Correctness: Investigating if the first character matches that of the presented text helps to detect if participants had problems finding characters on the keyboard. We observed significant effects for both experimental factors. Yet, the interaction was not significant, see table 2. Investigating *Simple* and *Complex Sentences* separately, ANOVA revealed that the effect of TEXT ENTRY INTERFACE was only significant for *Complex* ones, see table 3. However, Holm-Bonferroni corrected post-hoc tests revealed no significant differences for *Complex Sentences*, see figure 3.

5.3 Error Rates

Error rates demonstrates how frequently users make and fix mistakes. As participants were forced to fix all incorrect characters in this study, we use keystrokes per character, the erroneous key stroke error rate, and the total error rate.

5.3.1 Keystrokes per Character (KSPC). This metric is the ratio of the number of keystrokes and characters in the transcribed text and increases when users makes more mistakes, because they have to delete and then re-enter any wrong characters. It is defined by $KSPC = \frac{|IS|}{|T|}$, where *IS* denotes the input stream of all keystrokes including printable keys, *Shift*, *Space*, and *Backspace*. $|IS|$ denotes the number of keystrokes.

We observed significant effects for both experimental factors and their interaction, see table 2. Investigating *Simple* and *Complex Sentences* separately, ANOVA revealed that the effect of TEXT ENTRY INTERFACE was significant for both *Simple* and *Complex Sentences*,

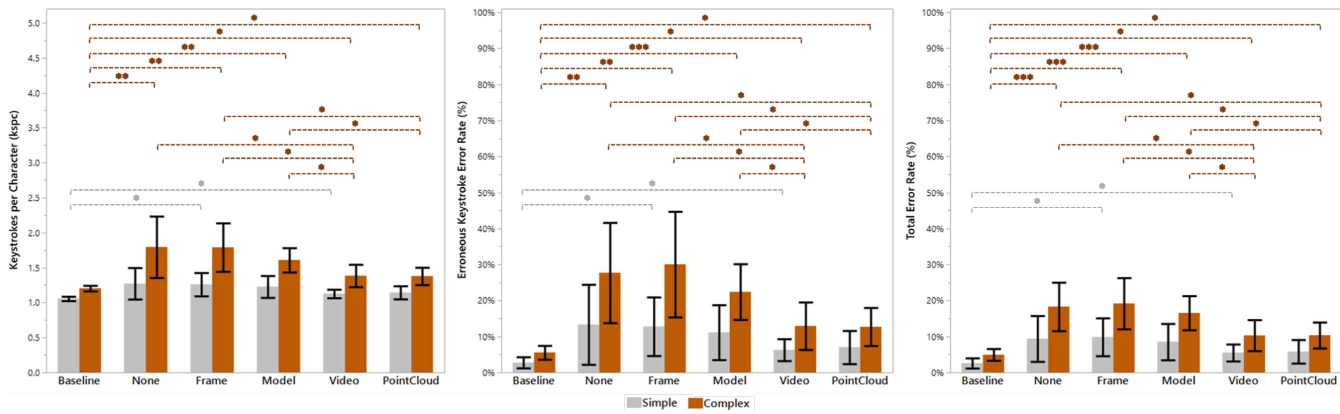

Figure 4: Error rates of users (*left*: keystrokes per character, *middle*: erroneous keystroke error rate, *right*: total error rate) with different keyboard visualizations in real and virtual environments ($^{*}p \leq .05$, $^{**}p < .01$, $^{***}p < .001$).

see table 3. Holm-Bonferroni corrected post-hoc tests for *Complex Sentences* showed that *KSPC* with *Baseline* was significantly smaller than the other conditions. Participants also produced significantly fewer keystrokes with *Video* compared to *None*, *Frame*, and *Model*. Also, *KSPC* with *Point Cloud* was significantly smaller than *Frame* and *Model*, see figure 4. On the other hand, post-hoc tests for *Simple Sentences* showed that *KSPC* with *Baseline* was significantly smaller than *Frame* and *Video*.

5.3.2 Erroneous Keystroke Error Rate (EKS ER). This metric investigates the rate of unnoticed errors or incorrect fixes in the input stream. As users must fix all mistakes in this study, there are no unnoticed errors. Thus, the erroneous keystroke error rate is described as $EKS\ ER = \frac{IF}{|P|} \times 100\% = \frac{IF}{|T|}$, where IF, incorrect fixes, denotes the number of keystrokes in the input stream which represent characters (excluding *Shift* and *Backspace*) that do not appear in the transcribed text.

We observed significant effects for both experimental factors and their interaction, see table 2. Investigating *Simple* and *Complex Sentences* separately, ANOVA revealed that the effect of Text Entry Interface was significant for both *Simple* and *Complex Sentences*, see table 3. Holm-Bonferroni corrected post-hoc tests for *Complex Sentences* showed that *EKS ER* with *Baseline* was significantly smaller than the other conditions. Participants also produced significantly less erroneous keystrokes with *Video* and *Point Cloud* when compared to *None*, *Frame*, and *Model*, see figure 4. Post-hoc tests for *Simple Sentences* indicated that *EKS ER* with *Baseline* was significantly smaller than *Frame* and *Video*.

5.3.3 Total Error Rate (Total ER). This metric is the ratio of the total of unnoticed errors and incorrect fixes and the total of corrected and incorrect characters [Arif and Stuerzlinger 2009]. In this study, this metric has the form $Total\ ER = \frac{IF}{|T|+IF} \times 100\%$.

We observed significant effects for both experimental factors and their interaction, see table 2. Investigating *Simple Sentence* and *Complex Sentence* separately, ANOVA revealed that the effect of Text Entry Interface was significant for both *Simple* and *Complex Sentences*, see table 3. Holm-Bonferroni corrected post-hoc tests for *Complex Sentences* showed that *Total ER* with *Baseline* was significantly smaller than the other conditions. Participants also made significantly fewer (total) errors with *Video* and *Point Cloud* compared to *None*, *Frame*, and *Model*, see figure 4. Post-hoc tests for *Simple Sentences* indicated that *Total ER* with *Baseline* was significantly smaller than *Frame* and *Video*.

5.4 Subjective Measures

To investigate in more detail how different Text Entry Interfaces support users in text entry task in VR we also recorded subjective measures for the five VR conditions. Participants were asked to rate and give comments for each of them. The *Baseline* condition was not investigated because it served as a typing performance test. Our subjective measures used a 0-100 Likert-scale, hence, we applied Aligned Rank Transform on the data before performing ANOVA.

5.4.1 Task Load Index. We asked participants to complete the NASA-TLX [Hart and Staveland 1988] to assess perceived workload during text entry. ANOVA revealed a significant effect of Text Entry Interface on TLX, $F(4, 60) = 4.966$, $p = .002$, $\eta^2 = .249$. Post-hoc tests indicated that *None*, *Frame*, and *Model* caused significantly higher workload than *Video*. The scores of *Frame* and *Model* were also significantly higher than *Point Cloud*, see figure 5.

5.4.2 Ease of Use. We asked participants how easily they could get familiar with each of our Text Entry Interfaces. The rating scale ranged from *0-very difficult* to *100-very easy*. ANOVA revealed that the effect of Text Entry Interface was significant, $F(4, 60) = 4.445$, $p = .003$, $\eta^2 = .229$. Post-hoc tests indicated that *None* and *Frame* were harder to use than *Video* and *Point Cloud*. *Model* also got significantly lower ratings than *Point Cloud*, see figure 5.

5.4.3 Comfortability. To investigate adoption potential, we asked participants to rate comfortability, with a scale from *0-very uncomfortable* to *100-very comfortable*. ANOVA revealed that the effect of Text Entry Interface was significant, $F(4, 60) = 2.809$, $p = .033$, $\eta^2 = .158$. Post-hoc tests indicated that *Video* was significantly more comfortable than *None*, *Frame*, and *Model*, see figure 5.

5.4.4 Perceived Typing Speed. To contrast the objective *WPM* metric, we also asked participants how they perceived their typing speed subjectively. The rating scale ranged from *0-very slow* to *100-very fast*. ANOVA revealed that the effect of TEXT ENTRY INTERFACE was not significant, $F(4, 60) = 1.637$, $p = .177$, $\eta^2 = .098$, see figure 5

5.4.5 Preference. We also recorded participant's preferences for how likely they would use each interface for VR text entry, using a rating scale from *0-very unlikely* to *100-very likely*. ANOVA revealed that the effect of TEXT ENTRY INTERFACE was significant, $F(4, 60) = 2.626$, $p = .043$, $\eta^2 = .149$. Post-hoc tests indicated that *Video* was significantly more preferred for VR text entry than *None* and *Frame*. *Point Cloud* also received significantly higher ratings than *Frame*. There was no significant difference between *Model* and the others, see figure 5.

6 DISCUSSION

The most noteworthy outcome is that HawKEY, our new VR text entry method affords text entry performance *when standing* that is comparable to seated usage, i.e., rates observed for seated users by Knierim [Knierim et al. 2018], Grubert [Grubert et al. 2018b], or McGill [McGill et al. 2015]. This means that our new text entry method enables users to efficiently and freely enter text while standing in a VE. We recognize that a subset of VR simulations is now being used while sitting on a swivel chair. Yet, such usage often makes only sense when the user is at least some distance away from a desk, which means that other text entry solutions that assume that the keyboard is placed onto a desk cannot be used. As HawKEY is also usable while seated, our new method affords text entry in almost all VR scenarios.

Overall, there was a significant difference between TEXT ENTRY INTERFACES in most measures except the *Time to First Character*, especially for *Complex Sentences*. Also, for *Simple Sentences*, there was a significant difference between TEXT ENTRY INTERFACES, as visible in the error rates.

Unsurprisingly, *Baseline* yielded generally better results than the other conditions. For the VR conditions, we can identify separable groups of TEXT ENTRY INTERFACES, with *Video* and *Point Cloud* emerging overall as the best VR solutions in *WPM*, *KSPC*, *EKS ER*, and *Total ER*. Thus, we can state that *Video* and *Point Cloud* have great potential for representing physical keyboards in VEs. Assuming (much) better depth camera technology becomes available, the results for *Point Cloud* could improve further. Yet, as the results for *Point Cloud* are already (mostly) within 5% of the *Baseline* in term of WPM, we see limited potential for improvements. In this context it is interesting to point out that our conditions are comparable to a standing baseline performance, similar to previous work which showed the same effect for seated performance [Knierim et al. 2018].

As the *Video* condition is technically (substantially) simpler to implement, we see this condition currently as the overall best choice for text entry in VR. Moreover, as the tray is usually worn in the same position, there may be no need to track it, which further simplifies this solution. Participants also appreciated that the keyboard "disappeared" in the *Video* condition when they looked straight ahead, i.e., when they just wanted to look at the VE. Whenever they looked down, the keyboard became again visible, which let them quickly resume typing.

6.1 Validity of Participant Group

We also analyzed if our participant group was biased towards experienced or inexperienced typists. For this, we compared the participant's *WPM* in the real-world *Baseline* condition, for *Simple* and *Complex Sentences*. Figure 6 shows the histograms of *WPM* for both conditions. A Shapiro-Wilk test could not reject that our participants came from a normally distributed group in term of typing experience with $p's > .9$. Overall, we conclude that our data were likely unbiased.

6.2 Familiarity with Lowercase Letters

Our participants did not seem to benefit from the representation of the physical keyboard while transcribing *Simple Sentences* in VR. There was no significant difference in term of typing speed (WPM) or the entry of first character between the *Baseline* condition and the others, (even) including *None*. A likely explanation is that the locations of lowercase alphabetical characters are very familiar to people used to computers. In other words, many participants could find letters without looking at the keyboard.

Still, participants tended to make more mistakes in VR even with *Simple Sentences*, as visible in the significant effect of TEXT ENTRY INTERFACES on *KSPC*, *EKS ER*, and *Total ER*, see table 3. Though the (conservative) Holm-Bonferroni corrected post-hoc tests only revealed some significant results, the average *Total Error Rates* of *None* (9.4%), *Frame* (9.9%), and *Model* (8.5%) were relatively larger than *Video* (5.5%), *Point Cloud* (5.8%), and *Baseline* (2.6%). Overall, we see that while participants made more errors in VR, they were able to fix them quickly enough to maintain text entry speeds that are still comparable to the *Baseline*.

6.3 The Challenge of Complex Sentences

In VR conditions with *Complex Sentences*, the typing speed significantly benefited from the *Point Cloud* condition (30.4 WPM) over *Frame* (26.4 WPM) and *Model* (25.8 WPM). This supports the superiority of more detailed keyboard representations over minimal forms. The pattern becomes clearer when looking at *Total Error Rate*, where three groups could be separated. Participants made less errors in the *Baseline* (5%) compared to all VR conditions. *Video* (10.3%) and *Point Cloud* (10.4%) had fewer errors relative to the *None* (18.3%), *Frame* (19.2%), and *Model* (16.5%) conditions.

6.4 Effect of Typing Experience

For *Complex Sentences* the appearance of uncommon symbols like punctuation and the need to use the *Shift* modifier significantly reduced typing performance in all metrics. *Complex Sentences* could thus be considered as a measure for (touch) typing experience. As the *WPM* distribution is normal, we divided our participant group by the mean of *WPM* in the *Baseline-Complex Sentence* condition, i.e., at 36.8 WPM, and designated the upper and lower parts as the *experienced* and *inexperienced* subgroups, respectively. Each subgroup consisted of 50% participants of the original group.

The *experienced* subgroup typed 76.1 WPM in the *Baseline* and reached up to 77.7 WPM in the *Video* condition with *Simple Sentences*. These values exceed the results of Knierim et al. [Knierim et al. 2018], with 67.2 WPM and 69.2 WPM, respectively. This is very notable, since our participants did this while standing, whereas

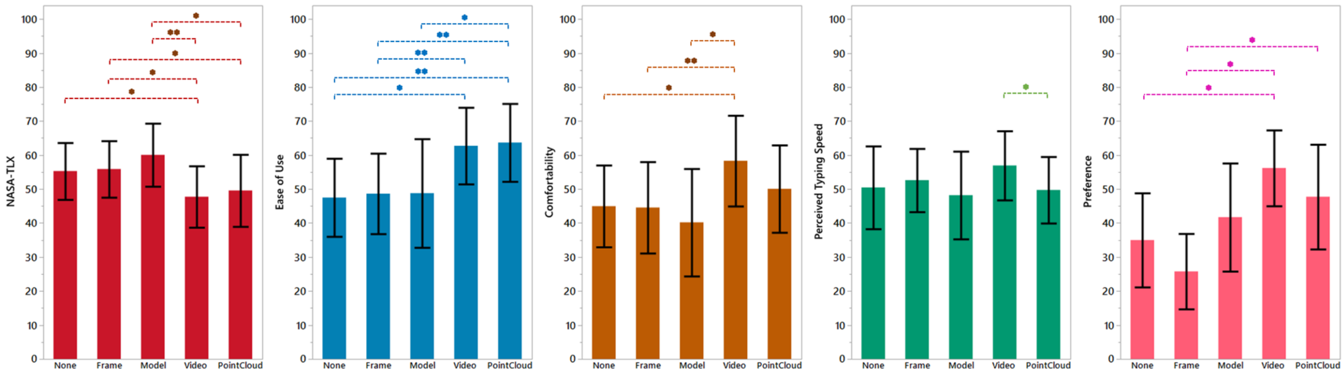

Figure 5: Subjective measures (*from left to right*: NASA-TLX, ease of use, comfortability, perceived typing speed, and preference) with different keyboard visualizations in VR (*$p \leq .05$, **$p < .01$, ***$p < .001$).

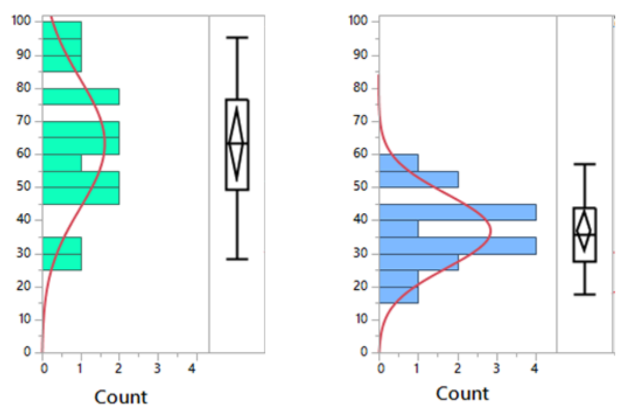

Figure 6: Histograms of *WPM* (vertical axis) in *Baseline* condition with *Simple* (left) and *Complex Sentences* (right).

they sat at a desk in Knierim et al.'s work, which could bias their results towards higher performance.

To gain a clearer picture of the difference between *inexperienced* and *experienced* typists, we performed between-subjects ANOVA on *WPM* and *Total ER* in the *Baseline* and *Video* conditions. The results indicated that *experienced* typists typed significantly faster in both baseline and the most preferred VR conditions, regardless how complicated the transcribed text was (all p's $< .01$). Yet, there was no significant difference between these subgroups in terms of *Total ER*, except *Video-Simple* ($p = .050$). Despite higher WPM, the *experienced* typists still made similar amounts of mistakes.

6.5 User Feedback

According to the NASA-TLX results, see figure 5, the workload with *Video* and *Point Cloud* was lower than *None*, *Frame*, and *Model*. Correlating the results with the feedback from users, the reduced workload in *Video* and *Point Cloud* could be explained by their relative ease of use. Participants mentioned that *Video* was "comfortable", "easier to use and to find keys". They pointed out that "seeing a video of [their] own hands was very helpful" and they "liked how [the keyboard] disappeared when looking straight ahead". Also, *Point Cloud* "looked very realistic" and was "pretty good" and a "more accurate keyboard". However, participants preferred *Video* to *Point Cloud* in terms of comfortability, as *Point Cloud* was "a bit more distracting" and "a bit difficult to see". While we improved the quality of the 3D point cloud display with software interpolation, it was still not comparable to *Video*. Participants preferred to use *Video* in VR as it "was easy to see, understand, and distinguish the keys".

6.5.1 HawKEY. Participants stated that "[HawKEY] is very good to type on" and that "the prototype is generally comfortable to wear". However, someone said that the "keyboard straps were a little uncomfortable" and they "wanted to move the keyboard [a bit] further away from the body".

6.6 Support for Multiple Input Methods

While HawKEY improves text entry in VR, it also introduces a conflict, as users will typically also interact with the VE through VR controllers. Our current solution for this issue is that we encourage users to use the provided controller wrist straps and to simply let the controllers dangle by side of the tray while the user is entering text with HawKEY. However, we also envision a revised version of HawKEY that includes controller "holders" to store them when not in use, see figure 7. This revised version can also be flipped up (towards the chest of the user) when not used for text entry, so that the user cannot inadvertently hit the keyboard with the controllers.

6.7 Limitations

For every Text Entry Interface, each participant transcribed 4 *Simple Sentences* and 4 *Complex Sentences*, which is a small number of sentences. While we intended to use a larger set of sentences, the whole study already took up 90 minutes with 6 interfaces, which taxed participants.

In the *Model* condition and although participants could see the virtual model of the keyboard and the point cloud of their own hands, they made more errors and had lower typing speed than with *Video* and *Point Cloud*. We speculate that this result may be a consequence of the difference in terms of latency. The RGB-D

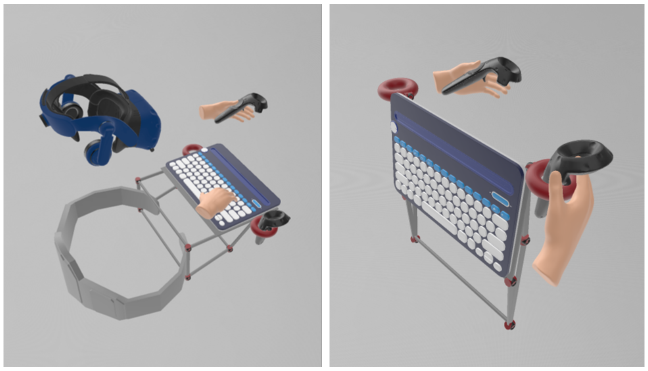

Figure 7: A revised, retractable design for HawKEY that supports multiple input methods in a seamless manner.

camera generating the point cloud has more latency than the Vive tracking system used to track and position the keyboard model. Any misalignment between these two systems could also have contributed to the lower performance of *Model*.

Most previous work investigated text entry solutions targeted at a sitting position. Google's drum-like keyboard [Doronichev 2016] was the only work investigating a standing position. Yet, we believe that standing, i.e., a posture that does not involve a chair and desk, might bias the typing performance towards a lower result, especially when large volumes of text need to be entered. Still, this leaves the question open how well typists perform with HawKEY while sitting.

7 CONCLUSION AND FUTURE WORK

We presented HawKEY, a new text entry method that is usable while standing in a VR system and which affords text entry rates that are comparable to those achievable while sitting at a desk. HawKEY was designed to be used while standing, but is equally usable while seated. We also examined different visual representations of physical keyboards in VR and found that, due to its technical simplicity, a see-through video condition is the overall best solution. It not only preserves real-world typing speeds for lowercase content, but also yields acceptable entry speeds and error rates for more complex characters, while receiving good ratings. Our participants also appreciated that the "video keyboard" automatically disappears when they looked straight ahead, which makes HawKEY unobtrusive during a VR experience. The point-cloud-based solution was also competitive, but is technically more challenging and was perceived as more distracting.

In the future, we plan to explore the addition of predictive text entry mechanisms, such as auto-correct and suggested word completions, to further increase text entry performance. In addition, we will also investigate how users take perform with HawKEY in a sitting posture.

REFERENCES

A. S. Arif and W. Stuerzlinger. 2009. Analysis of text entry performance metrics. In *2009 IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH)*. 100–105. https://doi.org/10.1109/TIC-STH.2009.5444533

Costas Boletsis and Stian Kongsvik. 2019. Text Input in Virtual Reality: A Preliminary Evaluation of the Drum-Like VR Keyboard. *Technologies* 7 (04 2019), 31. https://doi.org/10.3390/technologies7020031

Doug A. Bowman, Christopher J. Rhoton, and Marcio S. Pinho. 2002. Text Input Techniques for Immersive Virtual Environments: An Empirical Comparison. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 46, 26 (2002), 2154–2158. https://doi.org/10.1177/154193120204602611 arXiv:https://doi.org/10.1177/154193120204602611

A. Doronichev. 2016. Daydream Labs: exploring and sharing VR's possibilities. https://developers.googleblog.com/2016/05/daydream-labs-exploring-and-sharing-vrs.html. Accessed: 2019-04-05.

Tafadzwa Joseph Dube and Ahmed Sabbir Arif. 2019. Text Entry in Virtual Reality: A Comprehensive Review of the Literature. In *Human-Computer Interaction. Recognition and Interaction Technologies*, Masaaki Kurosu (Ed.). Springer International Publishing, Cham, 419–437.

John J. Dudley, Keith Vertanen, and Per Ola Kristensson. 2018. Fast and Precise Touch-Based Text Entry for Head-Mounted Augmented Reality with Variable Occlusion. *ACM Trans. Comput.-Hum. Interact.* 25, 6, Article 30 (Dec. 2018), 40 pages. https://doi.org/10.1145/3232163

Girden Ellen R. 1992. *ANOVA : repeated measures*. Sage Publications. https://doi.org/doi.org/10.4135/9781412983419

J. Grubert, L. Witzani, E. Ofek, M. Pahud, M. Kranz, and P. O. Kristensson. 2018a. Effects of Hand Representations for Typing in Virtual Reality. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 151–158. https://doi.org/10.1109/VR.2018.8446250

J. Grubert, L. Witzani, E. Ofek, M. Pahud, M. Kranz, and P. O. Kristensson. 2018b. Text Entry in Immersive Head-Mounted Display-Based Virtual Reality Using Standard Keyboards. In *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 159–166. https://doi.org/10.1109/VR.2018.8446059

Jan Gugenheimer, David Dobbelstein, Christian Winkler, Gabriel Haas, and Enrico Rukzio. 2016. FaceTouch: Enabling Touch Interaction in Display Fixed UIs for Mobile Virtual Reality. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 49–60. https://doi.org/10.1145/2984511.2984576

Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In *Advances in Psychology*, Peter A. Hancock and Najmedin Meshkati (Eds.). Human Mental Workload, Vol. 52. North-Holland, 139–183. https://doi.org/10.1016/S0166-4115(08)62386-9

Yi-Ta Hsieh, Antti Jylhä, Valeria Orso, Luciano Gamberini, and Giulio Jacucci. 2016. Designing a Willing-to-Use-in-Public Hand Gestural Interaction Technique for Smart Glasses. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 4203–4215. https://doi.org/10.1145/2858036.2858436

Haiyan Jiang, Dongdong Weng, Zhenliang Zhang, Yihua Bao, Yufei Jia, and Mengman Nie. 2018. HiKeyb: High-Efficiency Mixed Reality System for Text Entry.

Y. R. Kim and G. J. Kim. 2017. HoVR-Type: Smartphone as a typing interface in VR using hovering. In *2017 IEEE International Conference on Consumer Electronics (ICCE)*. 200–203. https://doi.org/10.1109/ICCE.2017.7889285

Pascal Knierim, Valentin Schwind, Anna Maria Feit, Florian Nieuwenhuizen, and Niels Henze. 2018. Physical Keyboards in Virtual Reality: Analysis of Typing Performance and Effects of Avatar Hands. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 345, 9 pages. https://doi.org/10.1145/3173574.3173919

Falko Kuester, Michelle Chen, Mark E. Phair, and Carsten Mehring. 2005. Towards Keyboard Independent Touch Typing in VR. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST '05)*. ACM, New York, NY, USA, 86–95. https://doi.org/10.1145/1101616.1101635

Jia-Wei Lin, Ping-Hsuan Han, Jiun-Yu Lee, Yang-Sheng Chen, Ting-Wei Chang, Kuan-Wen Chen, and Yi-Ping Hung. 2017. Visualizing the Keyboard in Virtual Reality for Enhancing Immersive Experience. In *ACM SIGGRAPH 2017 Posters (SIGGRAPH '17)*. ACM, New York, NY, USA, Article 35, 2 pages. https://doi.org/10.1145/3102163.3102175

I. Scott MacKenzie and R. William Soukoreff. 2003. Phrase Sets for Evaluating Text Entry Techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems (CHI EA '03)*. ACM, New York, NY, USA, 754–755. https://doi.org/10.1145/765891.765971

Anders Markussen, Mikkel Rønne Jakobsen, and Kasper Hornbæk. 2014. Vulture: A Mid-air Word-gesture Keyboard. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 1073–1082. https://doi.org/10.1145/2556288.2556964

Mark McGill, Daniel Boland, Roderick Murray-Smith, and Stephen Brewster. 2015. A Dose of Reality: Overcoming Usability Challenges in VR Head-Mounted Displays. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 2143–2152. https://doi.org/10.1145/2702123.2702382

M. Prätorius, U. Burgbacher, D. Valkov, and K. Hinrichs. 2015. Sensing Thumb-to-Finger Taps for Symbolic Input in VR/AR Environments. *IEEE Computer Graphics and Applications* 35, 5 (Sep. 2015), 42–54. https://doi.org/10.1109/MCG.2015.106

James Walker, Bochao Li, Keith Vertanen, and Scott Kuhl. 2017. Efficient Typing on a Visually Occluded Physical Keyboard. In *Proceedings of the 2017 CHI Conference* *on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 5457–5461. https://doi.org/10.1145/3025453.3025783

Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. 2011. The Aligned Rank Transform for Nonparametric Factorial Analyses Using Only Anova Procedures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 143–146. https://doi.org/10.1145/1978942.1978963 event-place: Vancouver, BC, Canada.

Xin Yi, Chun Yu, Mingrui Zhang, Sida Gao, Ke Sun, and Yuanchun Shi. 2015. ATK: Enabling Ten-Finger Freehand Typing in Air Based on 3D Hand Tracking Data. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology (UIST '15)*. ACM, New York, NY, USA, 539–548. https://doi.org/10.1145/2807442.2807504

Chun Yu, Yizheng Gu, Zhican Yang, Xin Yi, Hengliang Luo, and Yuanchun Shi. 2017. Tap, Dwell or Gesture?: Exploring Head-Based Text Entry Techniques for HMDs. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 4479–4488. https://doi.org/10.1145/3025453.3025964

D. Yu, K. Fan, H. Zhang, D. Monteiro, W. Xu, and H. Liang. 2018. PizzaText: Text Entry for Virtual Reality Systems Using Dual Thumbsticks. *IEEE Transactions on Visualization and Computer Graphics* 24, 11 (Nov 2018), 2927–2935. https://doi.org/10.1109/TVCG.2018.2868581