Input Devices in Immersive Environments

by

Duc-Minh Pham

B.Sc., VNUHCM University of Science, 2015

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

in the
School of Interactive Arts & Technology
Faculty of Communication, Art and Technology

© Duc-Minh Pham 2019
SIMON FRASER UNIVERSITY
Summer 2019

Copyright in this work rests with the author. Please ensure that any reproduction or re-use is done in accordance with the relevant national copyright legislation.
Approval

Name: Duc-Minh Pham

Degree: Master of Science (Interactive Arts and Technology)

Title: Input Devices in Immersive Environments

Examining Committee: Chair: Sylvain Moreno
Associate Professor

Wolfgang Stuerzlinger
Senior Supervisor
Professor

Bernhard E. Riecke
Supervisor
Associate Professor

Marek Hatala
External Examiner
Professor

Date Defended: August 29, 2019
The author, whose name appears on the title page of this work, has obtained, for the research described in this work, either:

a. human research ethics approval from the Simon Fraser University Office of Research Ethics

or

b. advance approval of the animal care protocol from the University Animal Care Committee of Simon Fraser University

or has conducted the research

c. as a co-investigator, collaborator, or research assistant in a research project approved in advance.

A copy of the approval letter has been filed with the Theses Office of the University Library at the time of submission of this thesis or project.

The original application for approval and letter of approval are filed with the relevant offices. Inquiries may be directed to those authorities.

Simon Fraser University Library
Burnaby, British Columbia, Canada

Update Spring 2016
Abstract

Recently, Virtual and Augmented Reality (VR and AR) head-mounted displays have become affordable. However, efficient solutions for pointing tasks and text entry in VR and AR remain a challenge.

Controllers are the typical input device in VR and many AR systems, but they are not as efficient as the mouse. Here, we investigate a pen-like pointing device that matches or exceeds the mouse’s performance. We performed a user study to compare several input devices and our results show that our 3D pen significantly outperforms modern VR controllers in all evaluated measures and that it is comparable to the mouse.

Text entry is a challenging task in modern VR systems, yet virtual keyboards are relatively inefficient. We introduce here a keyboard on a hawker’s tray worn in front of the user, which affords compact, simple, flexible, and efficient text entry solution. We ran a text entry study with standing users, involving both lower-case sentences as well as symbols. The results show that text entry rates are affected negatively by simplistic keyboard visualizations and that our video-based solution affords desktop text entry rates.

Keywords: 3D pointing; input devices; text entry; Virtual Reality; Augmented Reality
Dedication

I dedicate this thesis to my parents and grandmother, who have always been taking care of me and encouraging me to pursue my dreams; to my teachers and supervisors, who inspired my passion for science and technology and guided me to develop my skills and knowledge; to my best colleague and best friend, Thinh Nguyen-Vo, who has always been on my side and supporting me to achieve my goals. I also dedicate this thesis to my many friends for being there for throughout my entire master’s program.
I would like to thank my senior supervisor, Prof. Wolfgang Stuerzlinger, for the patient guidance, encouragement, and advice he has provided throughout my time as master’s student. His positive outlook and confidence in my research inspired me and gave me confidence.

I would like to thank my supervisor, Assoc. Prof. Bernhard Riecke for his invaluable guidance and feedback in different stages of my research. He did also teach me how to conduct scientific research at the very beginning of my master’s program.

I would like to mention my colleagues, Junwei Sun, Mayra Barrera, and Anil Ufuk for their support and collaboration in my research studies.
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approval</td>
<td>ii</td>
</tr>
<tr>
<td>Ethics Statement</td>
<td>iii</td>
</tr>
<tr>
<td>Abstract</td>
<td>iv</td>
</tr>
<tr>
<td>Dedication</td>
<td>v</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>vi</td>
</tr>
<tr>
<td>Table of Contents</td>
<td>vii</td>
</tr>
<tr>
<td>List of Tables</td>
<td>x</td>
</tr>
<tr>
<td>List of Figures</td>
<td>xi</td>
</tr>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Selection Using Pointing Devices in Virtual and Augmented Reality</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Text Entry Using Keyboard in Virtual Reality</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Thesis Outline</td>
<td>4</td>
</tr>
<tr>
<td>2 Is the Pen Better than the Controller? - A Comparison of Pointing</td>
<td>5</td>
</tr>
<tr>
<td>Devices in Virtual and Augmented Reality</td>
<td></td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>5</td>
</tr>
<tr>
<td>2.2 Related Work</td>
<td>7</td>
</tr>
<tr>
<td>2.2.1 Fitts’ Law and Throughput</td>
<td>7</td>
</tr>
<tr>
<td>2.2.2 Human Factors for Pointing</td>
<td>8</td>
</tr>
<tr>
<td>2.2.3 Input Devices for 3D pointing</td>
<td>8</td>
</tr>
<tr>
<td>2.3 Pilot Studies</td>
<td>9</td>
</tr>
<tr>
<td>2.4 Main User Study</td>
<td>10</td>
</tr>
<tr>
<td>2.4.1 Subjects</td>
<td>10</td>
</tr>
<tr>
<td>2.4.2 Apparatus</td>
<td>10</td>
</tr>
<tr>
<td>2.4.3 Pointing Task</td>
<td>13</td>
</tr>
<tr>
<td>2.4.4 Procedure</td>
<td>16</td>
</tr>
<tr>
<td>2.5 Results</td>
<td>16</td>
</tr>
</tbody>
</table>
### 2.5 Objective Measures
- **2.5.1 Objective Measures**: 16
- **2.5.2 Subjective Measures**: 19

### 2.6 Discussion
- **2.6.1 Pointing Performance in VR is Better than in AR**: 22
- **2.6.2 User preferences**: 23
- **2.6.3 Different Latency for the Mouse**: 24
- **2.6.4 Pen is Better than the Controller**: 24
- **2.6.5 Pen is Comparable to the Mouse**: 25
- **2.6.6 Applications of the Pen**: 25
- **2.6.7 VR/AR Pen Design Space**: 26
- **2.6.8 Limitations**: 27

### 3 Text Entry in Virtual Reality
- **3.1 Introduction**: 28
- **3.2 Related Work**: 30
  - **3.2.1 Mixed Reality Solutions**: 31
- **3.3 Study Design**: 31
  - **3.3.1 Text Entry Task**: 32
  - **3.3.2 Text Entry Interfaces**: 32
  - **3.3.3 Text Complexity**: 33
- **3.4 Experimental Setup**: 35
  - **3.4.1 Subjects**: 35
  - **3.4.2 Apparatus**: 35
  - **3.4.3 Phrase set**: 36
  - **3.4.4 Text Entry Conditions**: 36
  - **3.4.5 Procedure**: 37
- **3.5 Results**: 37
  - **3.5.1 Entry Rates**: 38
  - **3.5.2 First Character Statistics**: 40
  - **3.5.3 Error Rates**: 41
  - **3.5.4 Subjective Measures**: 43
- **3.6 Discussion**: 45
  - **3.6.1 Validity of Participant Group**: 47
  - **3.6.2 Familiarity with Lowercase Letters**: 48
  - **3.6.3 The Challenge of Complex Sentences**: 48
  - **3.6.4 Effect of Typing Experience**: 48
  - **3.6.5 User Feedback**: 49
  - **3.6.6 Support for Multiple Input Methods**: 49

### 4 Conclusion and Future Work
- **4 Conclusion and Future Work**: 51
4.1 Pen - A Promising pointing device for VR and AR .......................... 51
4.2 HawKEY - A Versatile Text Entry Solution in VR .......................... 51

Bibliography .......................... 53

Appendix A Post-study Questionnaire of Pointing Devices .................. 58

Appendix B Post-study Questionnaire of Text Entry Interfaces .............. 59
List of Tables

Table 3.1 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.

Table 3.2 Two-way Repeated-Measures ANOVA on Text Entry Interface (TEI), Text Complexity (TC), and their interaction (TEI×TC) (*p ≤ .05, **p < .01, ***p < .001, insig. p > .05).

Table 3.3 Analysis of different Text Complexities on Text Entry Interfaces for entry rate and first character (*p ≤ .05, **p < .01, ***p < .001, insig. p > .05). For averages across Simple and Complex, see table 3.2.

Table 3.4 The statistics of entry rate and first character of Text Entry Interfaces.

Table 3.5 The statistics of error rates of Text Entry Interfaces.
List of Figures

Figure 1.1 Illustrations of commercial VR pen devices. Left: HoloStylus (holo-light.com). Middle: VR Ink (logitech.com). Right: Massless Pen (massless.io). ................................................................. 2

Figure 1.2 Some VR keyboard solutions. Left: Drum-like virtual keyboard from Google[12]. Right: Using an optical tracking system and retroreflective markers to afford keyboard tracking. ............................... 3

Figure 2.1 Illustration of the input devices we compared in Virtual and Augmented Reality systems for pointing tasks. .......................... 5

Figure 2.2 Prototypes explored during the design process. Left: Pen tracked by Vive Tracker. Middle: HTC Vive controller used as a pen. Right: Ballpoint pen device. All these did not perform well in our pilot studies. .......................................................... 9

Figure 2.3 Left: HTC Vive Pro with 4 reflective markers. Right: Meta 2 with 4 reflective markers. ................................................... 11

Figure 2.4 From left to right: (a) Mouse, (b) Standard whiteboard pen, (c) Wrapped whiteboard pen with 4 reflective markers, and (d) Vive controller with 4 reflective markers. ........................................ 12

Figure 2.5 Left: All six combinations of IMMERSIVE ENVIRONMENTS and INPUT DEVICES investigated in our user study. Right: Screen shots of the pointing tasks in VR (top) and AR (bottom). ......................... 14

Figure 2.6 Screen shots of the pointing tasks in VR (left) and AR (right). ........ 14

Figure 2.7 Left: Mapping from the motion of the mouse to the movement of the cursor. Right: Ray-casting from the controller (or pen) to the cursor. 15

Figure 2.8 Objective measures (top-left: movement time, top-right: error rate, bottom-left: throughput, bottom-right: cursor speed) of input devices in VR and AR. Error bars represent 95% confidence intervals. (*p ≤ .05, **p < .01, ***p < .001) ........................................... 18

Figure 2.9 Subjective measures (top-left: comfortability, top-right: perceived speed, bottom-left: ease of interaction, bottom-right: user preference) for INPUT DEVICES and IMMERSIVE ENVIRONMENTS. Error bars represent 95% confidence intervals. (*p ≤ .05, **p < .01, ***p < .001) ....... 20
Figure 2.10 Illustration of a pen with touchpad and button magnetically snapped to the top of a headset for convenient pickup and storage. 26

Figure 3.1  
*Left*: RGB-D camera attached to VR headset.  *Middle*: Stabilizing straps.  *Right*: Text entry task. 28

Figure 3.2  
**Text Entry Interfaces in VR**: Baseline, None, Frame, Model, Video, and Point Cloud. 29

Figure 3.3  
*Top-left*: Baseline condition.  *Bottom-left*: Using HawKEY while sitting.  *Right*: A participant standing and wearing HawKEY and the VR headset during the study. 34

Figure 3.4  
Average text entry measures (*top-left*: words per minute, *top-right*: time to first key, *bottom*: first key correctness) with different keyboard representations in real and virtual environments (*p ≤ .05, **p < .01, ***p < .001*). Error bars represent 95% confidence intervals. 39

Figure 3.5  
Error rates of users (*top-left*: keystrokes per character, *top-right*: erroneous keystroke error rate, *bottom*: total error rate) with different keyboard visualizations in real and virtual environments (*p ≤ .05, **p < .01, ***p < .001*). Error bars represent 95% confidence intervals. 42

Figure 3.6  
NASA-TLX with different keyboard visualizations in virtual reality (*p ≤ .05, **p < .01, ***p < .001*). Error bars represent 95% confidence intervals. 44

Figure 3.7  
Subjective measures of users (*top-left*: ease of use, *top-right*: comfortability, *bottom-left*: perceived typing speed, *bottom right*: preference) with different keyboard visualizations in virtual reality (*p ≤ .05, **p < .01, ***p < .001*). Error bars represent 95% confidence intervals. 46

Figure 3.8  
Histograms of *WPM* (vertical axis) in Baseline condition with *Simple* (left) and *Complex Sentences* (right). 47

Figure 3.9  
A retractable design of HawKEY (in use). 50

Figure 3.10  
A retractable design of HawKEY (retracted). 50

Figure 4.1  
A design supporting multiple input methods: a pen magnetically snapped to VR headset, a keyboard, and two controllers. 52
Chapter 1

Introduction

Thanks to recent advances in technology, Virtual Reality (VR) and Augmented Reality (AR) applications have spread to many aspects of human life in the last few years, including education, research, training, and entertainment. Powerful computing hardware, high-resolution head-mounted displays (HMDs), and sub-millimeter tracking systems not only deliver a convenient and comfortable experience to users but also pave the way for professional applications of VR and AR. Despite those advantages, there are still strong concerns about the efficiency of current VR input methods in immersive environments. Traditional input devices used daily with computers, such as mouse and keyboard, are not immediately usable, as they do not support interaction directly or do not work well if the user’s eyes are completely or partially covered by headsets or obscured by virtual content, or if the user is not sitting at a desk. Many current VR systems use handheld controllers as the primary pointing device and a virtual keyboard as a temporary text entry solution, but these solutions are not optimal. This lack of efficient pointing and text input devices motivates us to propose and evaluate new solutions for data input in VR and AR that are potentially comparable to the performance of the traditional mouse and keyboard.

Users can easily use a pointing device like a controller to select a big menu button. However, the task becomes harder when they need to select a small data point in a 3D scatter plot. While the controller can be moved fast, the main option to achieve accuracy is to move the controller slower and target the data point more accurately, but this becomes time-consuming. While this speed-and-accuracy trade-off may not have a significant impact in some immersive games, it plays a key role in professional applications that require accuracy, including 3D modeling, engineering, and immersive analytics applications that display complex data. Current VR controllers typically do not enable the same kind of pointing performance found in desktop or touchscreen devices. VR/AR devices that exhibit good pointing performance would help to improve user productivity with such applications in various situations, such as selecting a small component in a 3D car model, drawing accurate geometry, and choosing a particular position in a 3D graph. Enhanced pointing devices for VR/AR are therefore a worthwhile research problem.
When discussing how to make input for immersive environments more efficient, it becomes clear that pointing is not the only mode of interaction. Text entry is one of the most neglected aspects for VR, and it afford complex and flexible communication between users and the virtual content. For example, immersive analytics software may require analysts to enter comments, captions, or even small reports; games may support in-game chat; or parts may have to be labeled in engineering contexts. An efficient text entry solution can improve user experience and productivity, but current alternatives, such as speech recognition, do not work well enough for professional work, except in very specialized circumstances.

All this motivated us to search for affordable, comfortable, and efficient VR/AR pointing devices and text entry interfaces to support various kinds of interactions in professional immersive applications.

1.1 Selection Using Pointing Devices in Virtual and Augmented Reality

There are several ways to select virtual objects in VR and AR. Some systems, such as the HoloLens and HMDs with an attached LeapMotion, recognize and translate hand gestures into designated actions. The HoloLens also permits voice commands. Current voice and gesture recognition systems can facilitate general selection tasks, such as choosing a menu button, picking an object, and performing a programmed animation. However, it takes a non-trivial amount of time to say a command or make a gesture and especially voice commands can disturb others in some environments. Also, the sub-optimal reliability and responsiveness of voice and (hand-)gesture recognition algorithms are still an issue that decreases usability. When an action fails, this becomes then frustrating for the user. Therefore, tracked devices, such as controllers, are still the most widely-adopted solution, to select and interact with virtual objects.

The HTC Vive and Oculus Rift utilize handheld VR controller to enable the user to point at or interact with the virtual content. Using a handheld device enables the user to perform on-demand selection instantly and intuitively. These controllers are nevertheless relatively big and must be held in a power grip, i.e., in the palm of the hand. Consequently,
whenever users want to change the orientation of the device, i.e., the direction it points to, they need to either rotate their wrist or even move their arm. This motivated us to think about smaller and more familiar designs of real-world pointing devices, especially devices similar to a pen/laser pointer. Several companies recently introduced such devices, including the Massless (massless.io), HoloStylus (holo-light.com), and VR Ink (logitech.com), see figure 1.1. These pen-like devices are held in precision grip, which potentially improves the precision of pointing and/or selection actions. This brings up the question how accurate a pen-like device is for selection tasks, when compared to the widely used VR controllers.

Hence, this thesis introduces a pen prototype and evaluates its performance. To more comprehensively assess this kind of device, we compare between three pointing devices: the pen, a VR controller, and the traditional mouse in both VR and AR. The results of this study contribute interesting insights about the potential for upcoming pen-like devices in the market.

1.2 Text Entry Using Keyboard in Virtual Reality

Pointing devices are only one option for users to communicate with immersive systems. They are suitable for simple input, such as choosing an option or direction, but inefficient for conveying more complex information, such as sentences, URLs, or whole documents. One commonly used solution is to employ a virtual keyboard and to select individual keys with one or two controllers, see figure 1.2 (left). While such virtual keyboards afford "hunt-and-peck" typing, physical keyboard support substantially higher input speeds and are thus a more appropriate choice. Though regular desktop computer users can easily use the keyboard in front of their 2D screens, VR users are typically unable to use a keyboard on a desk, as they cannot see the environment outside the VR headset. While an expert typist can touch-type without needing to see the keyboard or individual keys, they may still need to locate the keyboard and to look at the keyboard for unusual characters and/or key combinations. Also, being unable to see the keyboard and their hands to detect if they are typing the right key might frustrate an average or novice typist.
As a potential solution, several prototype systems have visualized physical keyboards in VR. Some of them take advantage of optical tracking to determine the position of the keyboard and even the user’s hand and then show their virtual models in VR [24], see figure 1.2 (right) for illustration. Others construct 2D or 3D images of the keyboard and/or the user’s hands based on external cameras and depth sensors. The relative complexity of all these solutions motivated us to investigate the potential for a simpler setup that requires less instrumentation and/or devices. Further, most of the previously presented solutions for high-speed text entry in VR only afford typing while seated in front of a desk. In contrast, most VR systems are designed to be used in a standing posture, or, even if the user is seated in a swivel chair, at least away from a desk. This motivated us to think about setups that facilitate typing while standing or while seated in a stand-alone chair.

Based on this reasoning, this thesis introduces HawKEY — our new versatile text entry solution, which affords efficient text entry for standing users. We also present a user study to determine the most effective keyboard representation and investigate how the complexity of the input text in terms of characters affects user typing performance. We did not focus on text entry in AR because most AR users can still perceive real-world objects, such as a keyboard, which makes virtual representations of physical keyboards in AR (largely) unnecessary.

1.3 Thesis Outline

This remainder of the thesis is structured into the following chapters.

- **Chapter 2: Is the Pen Better than the Controller? - A Comparison of Pointing Devices in Virtual and Augmented Reality.** This chapter presents the design of a pen-like pointing device for VR and AR. It also presents a comparison between the traditional mouse, a current VR controller, and a new pen-like device in term of performance and user preference for VR and AR pointing tasks.

- **Chapter 3: HawKEY - Efficient and Versatile Text Entry for Virtual Reality.** This chapter demonstrates a portable and convenient text entry solution in VR. It also includes an evaluation of text entry performance with different methods to visualize a physical keyboard in VR and analyzes how the complexity of the text affects the speed and error rates of typists.

- **Chapter 4: Conclusion and Future Work.** This chapter summarizes the findings of this thesis and proposes possible directions for the future to improve the design of pointing and text-entry devices in VR and AR further.

Chapters 2 and 3 of this thesis have been submitted together with my supervisor as two separate papers to the ACM Virtual Reality Software and Technology Symposium (VRST) 2019 [39, 38].
Chapter 2

Is the Pen Better than the Controller? - A Comparison of Pointing Devices in Virtual and Augmented Reality

2.1 Introduction

Based on recent technological advances, Virtual and Augmented Reality (VR/AR) applications have become more popular, in education, research, training, and entertainment. Powerful computing hardware, high-resolution head-mounted displays (HMDs), and sub-millimeter tracking systems not only deliver a comfortable experience but also pave the way for professional applications. VR/AR users typically manipulate virtual content through handheld devices, (e.g., HTC Vive controllers), their gaze, or their fingers (e.g., HoloLens). This primary interaction method plays a key role in different scenarios, including selecting, manipulating, and creating virtual content. Also, different applications may require different levels of selection accuracy. For example, selecting a small datapoint in a 3D scatterplot may need higher precision than choosing a large menu button. This motivates research on VR/AR selection methods.

Figure 2.1: Illustration of the input devices we compared in Virtual and Augmented Reality systems for pointing tasks.
Virtual environments can have any size and virtual objects can thus be far away from the user. Similarly, augmented content attached to real-world targets in AR can be distant from the user’s hands. Therefore, we focus on selection methods suitable for both near and distant objects in this work. Virtual hand techniques are limited to the reach of the human arm, which makes them a sub-optimal choice for some scenarios. Exocentric metaphors [2], such as automatic scaling or the world-in-miniature technique, can address the issue, but require transformations of virtual or augmented objects, which potentially break the immersive experience, increase interaction effort, and/or introduce motion sickness. 3D navigation methods apply only in VR scenarios and also increase interaction effort. Thus, we chose to evaluate user performance with ray-casting, the most popular egocentric method for distant object selection [6]. Most VR users are familiar with ray-casting as many applications in commercial systems use this technique to enable interaction with distant virtual items or the menu buttons.

Most current VR controllers are designed to be held in a power grip (and not in a precision grip). Thus, one might wonder if they are an optimal choice for selection tasks in term of accuracy and error rates. With ray-casting the ray typically starts from one end of the controller and users need to either rotate their wrist or even move their arm to control that ray. An unanswered question is if this the most efficient option for interaction. Different grips involve different muscle groups, which potentially results in different pointing performances. Due to the tripodal finger configuration, a precision grip affords more accuracy than a power grip, which is best illustrated by the progression of pen grips observed in children who are learning to write [42]. In essence, a pen is a cylindrical object small enough for users to manipulate with multiple finger tips, when held in the typical pen grip. Also, several VR/AR pens or stylii, such as Massless (massless.io) and HoloStylus (holo-light.com), have been recently introduced to provide more options for content creation and interaction. Thus, we evaluate a pen-like device held in a precision grip and compare its performance with a traditional VR controller, to provide timely design guidance for AR/VR developers.

For 2D content, a mouse or a touch screen are among the best options for interaction. Many users are already very experienced with these devices and can point at small targets with high accuracy. The technical properties of current mice in terms of latency, sampling rate, and sensor resolution support this, too. Given its ubiquity, this makes the mouse an ideal baseline condition, even though it is not a 3D input device. Also, previous work has identified that a controller is generally inferior relative to the mouse in terms of pointing performance [46]. Therefore, in this study, we investigate also if a pen could reach performance comparable to a mouse.

Hand-held devices like a controller or pen can be used to perform 3D pointing with ray-casting. This method does not work with the mouse. Users typically manipulate the mouse on a 2D surface to perform 2D pointing on a computer monitor. However, other work has shown that a mouse cursor can be used for 3D pointing with ray-casting [47, 11]. Hence, we
chose to perform our evaluation with a selection task where targets are located on a virtual plane that faces the user. Then, users can move a cursor on that plane to select the targets with any of our chosen input devices. Hence, they perform 2D interaction with the mouse but 3D interaction with the controller and the pen.

Other studies have evaluated different methods that use a human finger or the arm as a pointing device. Brown et al. identified that a bare finger tracked by the LeapMotion is not an efficient pointing method [9]. Other work used a motion capture system to track the user’s finger or arm in 3D space [32, 40, 49]. However, these techniques require retro-reflective markers attached to the user’s finger or a custom glove, which is not practical in many scenarios. As designated input devices can be shared between users with little concern for setup time and tracking quality, we decided to limit our work to device-based pointing.

Past work has evaluated the performance of distant pointing with wands and lasers in front of large displays, e.g., [25, 36, 35]. Yet, modern VR/AR systems involve HMDs and afford an experience that is different from large displays. Moreover, HMDs allow the user to be “inside” the virtual environment, which affords a higher degree of immersion.

In this paper, we report on a comparison of the mouse, a controller, and a pen input device for distant pointing for users wearing VR and AR headsets. We identify that the pen is a significantly better device for pointing in VR and AR compared to the controller and identify likely reasons for this outcome. We also discuss how pens could be integrated into today’s AR and VR systems.

2.2 Related Work

Here we first review evaluation methods for pointing devices, then discuss human factors related to pointing, and finally present the results of previous work for various input devices.

2.2.1 Fitts’ Law and Throughput

Fitts’ law [15] is well known in 2D user interfaces for its ability to accurately predict how long a pointing task to a target with a given size at a given distance will take:

\[ MT = a + b \log_2 \left( \frac{A}{W} + 1 \right) \]  
(2.1)

where \( MT \) is the time to point to a target, \( A \) is the distance, \( W \) the size of the target, and \( a \) and \( b \) are empirically determined constants. The logarithmic term is also called the index of difficulty of pointing (\( ID \)).

Extending Fitts’ law, the throughput measure better captures the natural speed-accuracy tradeoff [29, 1] of pointing. Throughput is defined through the use of effective measures that take the task that the users actually performed into account [29, 1]. Most work in 2D user interfaces has adopted this methodology to characterize pointing performance. Followup
work generalized this approach for measuring pointing performance to 3D [45, 46]. One important insight from this line of work is that latency can have significant impact on pointing performance [45, 37].

### 2.2.2 Human Factors for Pointing

Previous research investigated the influence of muscle groups on pointing performance. For 6DOF (degrees of freedom) devices, Zhai [54] identified that holding an ball-shaped input device with the fingers can afford faster 6DOF manipulation relative to a palm-held device. Balakrishnan and MacKenzie compared the performance of 2D input controlled with a single finger, the wrist, and the forearm and found that a single finger does not perform better than other options [4].

### 2.2.3 Input Devices for 3D pointing

Many input devices exist for 2D desktop applications. With proper mappings, such devices can also work well for 3D tasks [6]. Argelaguet et al. surveyed 3D object selection techniques [2], while Hoppe et al. surveyed the input and output devices and associated interaction techniques for 3D interaction [20]. Here we limit our discussion to those directly relevant for our work. Specifically, we do not discuss glove- or hand-based interfaces, also because virtual hand selection is outside of our scope. We focus mostly on comparison based on throughput, as this measure is more robust to the speed-accuracy tradeoff. This is important for comparisons of different input devices that are subject to different amounts of latency and other limitations of tracking technologies, such as tracking noise.

**Mouse**

The mouse is an ideal input device for 2D interfaces and some 3D interfaces [53]. Previous comparisons have identified that the mouse works well even for 3D manipulation [5]. Krichenbauer et al. [26] compared the mouse and a 3D input device for 3D manipulation and found no significant difference. For 3D pointing, Teather and Stuerzlinger identified that a mouse controlling a cursor performs better than other alternatives [46] and that displaying the cursor only to a single eye can mitigate potential depth conflicts for the mouse cursor in a 3D virtual environment [47].

**Controller**

Butterworth et al. [10] used a handheld 6D mouse as input device, which functioned as a 3D controller. Teather and Stuerzlinger [46] found that throughput of a controller-like device, was less than for the mouse, which was confirmed in recent work [43].
Pen

Early work with the throughput measure evaluated a pen-like device for 3D pointing \cite{46} in a fish-tank VR system, where all targets were within arm’s reach. Also, the authors stated that their implementation might have suffered from rotational jitter. Brown et al. investigated a chopstick (a pen-like device) tracked by the Leap Motion \cite{9} and found the throughput to be (almost) as good as the mouse. This promising result further motivated us to measure the throughput of a pen-like device in an immersive environment.

2.3 Pilot Studies

We performed several pilot studies to identify a pen-like device that works well. For each we used the same ISO 9241-411 methodology \cite{1} as in our main user study, see below.

We first tried to emulate a chopstick device \cite{9} by attaching a long thin rod to a HTC Vive standalone tracker, see figure 2.2. Yet, we found that the uneven weight distribution made the device uncomfortable to use, and we observed pointing performance substantially below a controller.

Subsequently, we tried to use the Vive Controller as a pen, by balancing the center of the device on the hand between the thumb and the side of the index finger and using its base as the pointing end, see figure 2.2. Yet, the controller was too heavy to be moved with the fingers and most participants naturally reverted to using their wrists to control the device. We thus were unable to identify a significant difference between the controller held normally and like a pen.

We also experimented with a ballpoint-pen-like device, as illustrated in figure 2.2. This version was too thin and the markers too close together, which affected performance negatively. This motivated us to look at bigger pens.

Finally, we attached a Vive tracker to an optical mouse and compared its native performance, the Vive-tracked version \cite{45}, and the Vive controller. This pilot only identified...
a significant difference for throughput (but not for time or errors) for the mice, with the normal mouse being better.

2.4 Main User Study

The main goal of our study was to compare the pointing performance of various pointing devices. As previous work, we use the ISO 9241-411 methodology [29, 1] to compare not only pointing time and errors but also throughput. ISO 9241-411 describes evaluation methods for the design of physical input devices and also provides tests to evaluate their performance, effort, and comfortability. In this standard, throughput is the most important measure, as it combines both speed and accuracy into one number, which indicates user performance in pointing tasks. For comparison purposes, including a baseline device such as a mouse is recommended.

In addition, we aimed to compare VR and AR conditions to investigate how the level of immersion would affect user pointing performance. In VR, participants only perceived virtual content, while they still saw the real-world background as well as their own hands (if they looked at the input device) in AR. Especially the fact that they could see their own hands, potentially introduces different levels of immersion and could affect performance.

2.4.1 Subjects

Twelve people (3 female), with ages ranging from 23 to 33 ($M = 27.92$, $SD = 2.84$), participated in this study. Only one of them was left-handed. Based on the Porta eye dominance test, 41.67% were left eye dominant. More than half, 58.33% played 3D games more than 5 hours a week. Participants were paid a small compensation for their participation.

2.4.2 Apparatus

To guarantee that machine performance was not a limiting factor, we used a PC with an Intel® Core™ i7-4790, 16GB RAM, and a nVidia GTX 1080, running Windows 10. The components of this PC far exceed the requirements for our VR and AR headsets. We chose VR and AR headsets with (roughly) similar specifications.

Virtual Reality Headset

The HTC Vive Pro is one of the newest VR headsets at the time of writing, see figure 2.3 (left). The total display resolution for both eyes is $2880 \times 1600$, with $90 \ Hz$ expected refresh rate. The horizontal field of view is approximately $100^\circ$. The Vive Pro weighs about 550 g.

Augmented Reality Headset

The Meta 2 is currently a state-of-the-art AR headset, as shown in figure 2.3 (right). We chose this tethered AR headset over un-tethered alternatives, as it has a much larger hori-
Horizontal field of view, approximately 90°. The total resolution is 2560 × 1440, with an expected refresh rate of 60 Hz. Yet, with our setup the display usually maintained 80 Hz. The Meta 2 weighs about 500 g, which is a little bit lighter than the Vive Pro.

**Optical Tracking System**

Both the HTC Vive Pro and the Meta 2 have their own tracking systems. The HTC Vive Pro and its controllers can be tracked with its own outside-in optical system, which uses two base stations emitters. The Meta 2 uses front-facing RGB cameras and IR-based sensors to support inside-out positional tracking. These two tracking systems are based on radically different technologies, which might result in vastly different latency and/or accuracy. In our work, we did not use either of these two tracking systems, but used a different optical tracking system to keep the tracking quality consistent between our VR and AR conditions. To avoid interference, we blocked the IR illuminator of the Meta 2, see figure 2.3 (right).

We used an external outside-in optical tracking system from OptiTrack. In our setup, there were eight OptiTrack S250e, 250 Hz IR cameras, which were hung from a 2.4 × 1.8 m rectangular metal frame 2.1 m above the tracked area. We attached optical markers to each headset, see figure 2.3, in different configurations to ensure they can be tracked reliably. We placed the markers at the top of the headsets to avoid potential occlusions, even by the user’s hands. To achieve the best possible tracking performance, participants were asked to sit at the center of that space. Participants sat in a swivel chair, with a Mobo keyboard and mouse tray, see figure 2.5, onto which we placed a wireless keyboard and mouse.
The OptiTrack cameras were connected over a Gigabit Ethernet switch to a secondary tracking computer, which sends the tracking results to the main experimental system over the network. While this increased the latency, it ensured that the CPU of the main experimental system was not fully loaded, which reduced the potential for dropped frames. We also tracked several input devices through the OptiTrack system in our work, as described in the following.

**Mouse**

We chose a wireless Logitech M215 mouse, as shown in figure 2.4 (a), as input device for the mouse conditions. This fairly average mouse is easy to operate and fits most people’s hands well. To ensure good tracking performance, we regularly checked the two AAA batteries and replaced them to avoid potential tracking issues during the pointing task.

We chose not to show a virtual model of the mouse in the VR condition during the pointing task. The mouse was located on the Mobo mouse pad attached to the chair, which is well below the participant’s view when they are looking straight forward, at the targets. This is similar to how most people use a mouse in their daily work on a computer. We did not track the mouse by the OptiTrack system, see also the discussion below. To keep internal validity high, we required that participants indicated selection through hitting the space bar of the keyboard placed on the Mobo keyboard tray, with the their non-dominant hand, see figure 2.5. The mouse button had no function in the trials. Participants used this mouse naturally through a combination of elbow, wrist and (finer) finger movements.
Controller

We used a HTC Vive controller in this study, one of the most popular VR controllers, which weighs 470g. Instead of tracking it via the Vive lighthouse base stations, we attached a set of four reflective markers on its upper surface, as shown in figure 2.4 (d), and tracked the device with the OptiTrack system. The positions of the markers were tested carefully so that they were chosen so that participants could still hold the controller comfortably without hiding the markers. Because the controller was potentially visible in the user’s field of view, we showed its 3D model in VR to improve the consistency between VR and AR conditions. As the controller is held in the palm, participants used this device with a combination of wrist and elbow movements. As with the mouse, participants indicated selection through the space bar of a keyboard like the mouse condition. We chose to use an external button, as the buttons on the controller can cause “dips” in the selection, which can affect selection performance negatively.

Pen

Based on the results from previous work, which identified pen-like devices as a good alternative for mid-air input, we chose to compare such a pen with the mouse and controller. As mentioned above, we iterated through several versions before ending up with the final design, shown in figure 2.4 (c). To create this pen device, we took a normal whiteboard pen, figure 2.4 (b), and attached four reflective markers so that OptiTrack cameras could determine its position and orientation. To eliminate tracking interference due to the reflective surface of the pen, we wrapped it in black tape. The weight of this current pen was about 60g, with an average diameter of approximately 1.8 cm in average. The balance and weight of the pen enables users to hold it like a real pen, which means that they were able to use multiple fingers in coordination as well as the wrist to move the device. Similar to how participants held the controller, the pen was often held so that it was visible in the field of view of the VR headset. Hence, we displayed a virtual 3D model of the pen in VR so that they could be aware of its position and direction. As with the other input devices, participants also indicated selection via the space bar of the keyboard like the mouse and controller conditions.

2.4.3 Pointing Task

In this study, participants were asked to perform pointing tasks in different combinations of Immersive Environments and Input Devices. Each pointing trial required participants to use one of the input devices to point to 11 spherical targets, while wearing either the VR or AR headset. Corresponding to a 3D version of the ISO 9241-411 task [1], targets were shown in a circular arrangement in a plane parallel to the view plane, with the center aligned to the eye level of participants at the beginning of the trial, as shown in figure 2.6.
Figure 2.5: *Left:* All six combinations of **Immersive Environments** and **Input Devices** investigated in our user study. *Right:* Screen shots of the pointing tasks in VR (top) and AR (bottom).

Figure 2.6: Screen shots of the pointing tasks in VR (left) and AR (right).
We also displayed the transparent plane containing the targets, so that participants could easily recognize the planar arrangement of those targets in 3D space. The diameter of the circle (and thus the distance between targets) and the size of the targets varied across trials. We chose a pointing task on a 2D plane in 3D, as ray-casting involves mostly rotating the wrist in 2DOF, making it comparable to 2D manipulation. Moreover, pointing to 3D targets at different distances is equivalent to projecting all targets to the same plane [47].

Participants manipulated a green spherical cursor on that target plane with the input devices: mouse, controller or pen. For the mouse, we simply mapped motions to cursor movements on the target plane. With the controller and pen, we used ray-casting based on the current 6DOF position of the device and displayed the cursor at the ray-plane intersection. The goal of the task was to move the cursor to hit each target to select them in the order specified by the ISO 9241-411 task [1], see figure 2.7 for an illustration. Although the mouse was controlled horizontally on a surface to manipulate the cursor on the vertical virtual plane, previous work has shown that this mapping should not result in a difference in pointing performance [44]. Also, users are very adept at using a mouse on a horizontal desk to operate a cursor on a vertical screen.

Initially, all targets were grey except a random yellow one indicating the current objective. Participants then control the cursor to hit the target by moving it inside the objective. When the cursor center is inside the target, it was highlighted dark yellow, as highlighting makes it easier to see which object will be selected [48]. Once participants hit the space bar in this state, see figure 2.7, we counted a “hit” and the target changed to black. Otherwise, if the cursor was outside of the target, the objective became grey and an error sound was triggered to indicate a failed selection. After each selection, the next objective became highlighted. In the prescribed sequence, consecutive targets are (approximately) opposite in the
target circle. Participants continued to select objectives until all 11 targets were selected. Figure 2.6 shows different target states during a pointing trial.

2.4.4 Procedure

After greeting the participants, we asked them to fill the consent form and then informed them about the general study procedure. Next, they were asked to fill a pre-study questionnaire to record their age and experience with 3D virtual environments. Then, we introduced the pointing task and gave them a demonstration. Then they experienced a practice section where they could try both headsets and all input devices. These initial steps took about 15 minutes.

When participants indicated that they were ready for the main experiment, we positioned the swivel chair at the center of our tracking area. To maintain optimal tracking performance participants were asked not to move or rotate the chair (much) during the study. Then, they performed all pointing tasks with one of the two headsets and the three input devices before taking a 3-minute break and proceeding to the other headset. This sequence helped maintain immersion with each headset and avoided potential negative effects due to repeatedly switching headsets. We counterbalanced the order of headsets across participants. For each headset condition, they used all three input devices (mouse, controller, and pen) in counterbalanced order. This order was the same for both VR and AR conditions of a participant.

For each of the $2 \times 3 = 6$ system configurations, participants were asked to complete a series of 18 pointing tasks. Different pointing tasks had varying parameters of target size, target circle size, and target plane distance. Target size is the diameter of the 11 spherical targets in the circle. This sub-condition had three unique values: 1.5, 2.5, or 3.5 cm. The diameter of the target circle sub-condition was either 15, 20, or 30 cm. The target plane distance, between the participant and the whole target circle, was either 75 or 150 cm. Thus, the total number of target circles for each device was $3 \times 3 \times 2 = 18$. The order of these target circles was also counterbalanced via a Latin square for each participant. In summary, there were 2 headsets $\times$ 3 devices $\times$ 18 trials $= 108$ target circles for each participant, corresponding to 1188 individual target selections. When the experimental section finished after about an hour, participants were asked to answer a post-study questionnaire to elicit their preferences for the Immersive Environments and Input Devices.

2.5 Results

2.5.1 Objective Measures

We first present the results for the objective measures followed by the subjective ones. As all pre-conditions for ANOVA, including normality, were met by the data for all objective measures, we applied repeated measures ANOVA unless noted otherwise. If sphericity
did not hold for any factor, we used either Greenhouse-Geisser or Huynd-Feldt correction depending on their estimated $\epsilon$ [14].

To gain deeper insights into the data, we also performed ANOVA on VR and AR separately to investigate the effect of INPUT DEVICE, for every measure. In addition, we used Holm-Bonferroni correction to conduct post-hoc tests.

**Movement Time**

The movement time is the average time measured in seconds between the selection of two consecutive targets, excluding the time for the first target. We observed significant effects of both IMMERSIVE ENVIRONMENT, $F(1, 11) = 10.554$, $p = .008$, $\eta^2 = .490$, and INPUT DEVICE, $F(2, 22) = 10.869$, $p = .001$, $\eta^2 = .497$, on movement time. Their interaction, however, was not significant, $F(1.370, 15.065) = 3.775$, $p = .060$, $\eta^2 = .256$.

Investigating each IMMERSIVE ENVIRONMENT separately, we observed the significant effect of INPUT DEVICE on VR, $F(2, 22) = 8.838$, $p = .002$, $\eta^2 = .446$, and AR, $F(2, 22) = 10.850$, $p = .001$, $\eta^2 = .497$. Post-hoc tests indicated that the movement time in VR of controller was significantly larger than mouse and pen. In AR, pen required significantly smaller movement time when compared to mouse and controller, see figure 2.8.

**Error Rate**

We computed the error rate as the ratio of missed selections over the number of targets in a circle. There was no significant difference between VR and AR conditions with regard to the error rate, $F(1, 11) = .020$, $p = .889$, $\eta^2 = .002$. However, the INPUT DEVICE had a significant effect, $F(2, 22) = 5.114$, $p = .015$, $\eta^2 = .317$. The interaction IMMERSIVE ENVIRONMENT $\times$ INPUT DEVICE was significant, $F(2, 22) = 3.450$, $p = .050$, $\eta^2 = .239$.

Investigating each IMMERSIVE ENVIRONMENT separately, we observed the significant effect of INPUT DEVICE on VR, $F(2, 22) = 11.125$, $p < .001$, $\eta^2 = .503$, but not on AR, $F(2, 22) = .684$, $p = .515$, $\eta^2 = .059$. Post-hoc tests showed that participants produced significantly higher error rate with controller when compared to mouse and pen in VR, see figure 2.8.

**Throughput**

The ANOVA indicated that different IMMERSIVE ENVIRONMENTS yielded significantly different throughput, $F(1, 11) = 5.652$, $p = .037$, $\eta^2 = .339$. The main effect of INPUT DEVICE on throughput was also significant, $F(2, 22) = 14.034$, $p < .001$, $\eta^2 = .561$. Their interaction was not significant, $F(2, 22) = .727$, $p = .495$, $\eta^2 = .062$.

Investigating each IMMERSIVE ENVIRONMENT separately, we observed the significant effect of INPUT DEVICE on VR, $F(2, 22) = 9.454$, $p = .001$, $\eta^2 = .462$, and AR, $F(2, 22) = 13.880$, $p < .001$, $\eta^2 = .558$. Post-hoc tests revealed the same pattern of throughput in both
Figure 2.8: Objective measures (top-left: movement time, top-right: error rate, bottom-left: throughput, bottom-right: cursor speed) of input devices in VR and AR. Error bars represent 95% confidence intervals. (*\(p \leq .05\), **\(p < .01\), ***\(p < .001\))
IMMERSIVE ENVIRONMENTS, which indicated that controller generated significantly lower throughput than mouse and pen.

Cursor Speed

Due to the different control mappings, it is challenging to characterize how efficiently participants used each device. As the targets are all in a plane, cursor speed in that plane is a more objective measure. It captures how fast the cursor on said plane is moving, and is computed as $\text{CursorSpeed} = \frac{S}{T}$ where $T$ is the completion time and $S$ the total cursor travel distance on the target plane between the first and the last selection. It is measured in cm/s. The ANOVA revealed that the main effect of IMMERSIVE ENVIRONMENT was significant, $F(1, 11) = 6.676$, $p = .025$, $\eta^2 = .378$. The main effect of INPUT DEVICE was also significant, $F(2, 22) = 10.791$, $p = .001$, $\eta^2 = .495$. Their interaction was not significant, $F(2, 22) = 1.127$, $p = .342$, $\eta^2 = .093$.

Investigating each IMMERSIVE ENVIRONMENT separately, we observed the significant effect of INPUT DEVICE on VR, $F(2, 22) = 8.723$, $p = .002$, $\eta^2 = .442$, and AR, $F(2, 22) = 7.308$, $p = .004$, $\eta^2 = .399$. In VR, post-hoc tests indicated that pen manipulated the cursor significantly faster than controller. In AR, pen resulted in larger cursor speed than mouse and controller, see figure 2.8.

2.5.2 Subjective Measures

Here we present the results of the post-study questionnaires, see appendix A. Our subjective data used a 0-100 Likert-scale. We applied Aligned Rank Transform [51] on the data before performing ANOVA.

Comfortability

A good IMMERSIVE ENVIRONMENT or INPUT DEVICE for a pointing task should not only allow users to achieve high performance but also feel comfortable to use. Thus, participants were asked to rate the comfortability on a 0-100 scale for each IMMERSIVE ENVIRONMENT and INPUT DEVICE. 0 indicates very uncomfortable and 100 indicates very comfortable. For IMMERSIVE ENVIRONMENT, a t-test showed that participants felt significantly more comfortable in VR than with the AR headset, $t(1) = 9.305$, $p = .011$, $\eta^2 = .458$. The INPUT DEVICES received significantly different comfortability scores, $F(2, 22) = 6.426$, $p = .006$, $\eta^2 = .369$. Post-hoc tests showed that pen and mouse were rated significantly more comfortable than controller, see figure 2.9.

Perceived Movement

To validate the objective measure of cursor speed, we asked participants to rate their perception of how fast they manipulated the cursor. The rating ranges from 0-very slow to
Figure 2.9: Subjective measures (top-left: comfortability, top-right: perceived speed, bottom-left: ease of interaction, bottom-right: user preference) for Input Devices and Immersive Environments. Error bars represent 95% confidence intervals. (*$p \leq .05$, **$p < .01$, ***$p < .001$)
100-very fast. A t-test indicated that there was no significant difference in perceived cursor movement speed between VR and AR conditions, \( t(1) = 2.067, p = .178, \eta^2 = .158 \). An ANOVA revealed a significant effect of Input Device, \( F(2, 22) = 3.530, p = .047, \eta^2 = .243 \). Post-hoc tests showed that participants perceived they could move the cursor significantly faster with the pen compared to the controller, see figure 2.9.

Ease of interaction

We also recorded ease of pointing, by asking participants to rate how easy they felt that the pointing task was for each Immersive Environment and Input Device. The range is from 0-very difficult to 100-very easy. Participants found the task significantly easier in VR than in AR, \( t(1) = 5.007, p = .047, \eta^2 = .313 \). To compare different Input Devices, participants were asked to rate how easy they found each device for pointing. An ANOVA identified a significant main effect of Input Device, \( F(2, 22) = 4.216, p = .028, \eta^2 = .277 \). Post-hoc tests showed that controller was perceived to be significantly more difficult to use than pen, see figure 2.9.

Preference

We also asked participants to rate their preference for each combination of Input Device and Immersive Environment. The value ranges from 0-strongly not recommended to 100-strongly recommended. An ANOVA revealed that Immersive Environment did not have significant impact on user preference, \( F(1, 11) = .745, p = .407, \eta^2 = .063 \). On the other hand, the main effect of Input Device was significant, \( F(2, 22) = 9.683, p = .001, \eta^2 = .468 \). The interaction was not significant, \( F(1.164, 12.802) = .202, p = .698, \eta^2 = .018 \). Post-hoc tests revealed that the pen was significantly more preferred than mouse or controller in either VR or AR, see figure 2.9.

2.6 Discussion

Overall, we identified that the pen is an input device that has similar performance to the mouse for 3D pointing tasks. This is very promising as, to our knowledge, no other work has identified an input device that is usable in immersive environments and is comparable to (or non-significantly exceeds) the mouse in terms of throughput. Also, participants generally liked the pen as an input device. Our results also confirm the outcomes of previous work [46] in that the controller has significantly worse performance in terms of throughput than the mouse. In the following we discuss more specific findings and how they relate to potential limitations of our apparatus.
2.6.1 Pointing Performance in VR is Better than in AR

We aimed to compare the pointing performance between VR and AR to examine how the levels of immersion would affect user pointing performance. The appearance of hands and real-world objects could potentially cause some differences between these immersive environments. We found that the pointing performance was significantly better in VR than in AR in most measures. Before isolating and exploring possible causes, we present how we reduced potential confounds for the different Immersive Environments.

Similarities of Headset Specifications and Tracking System

To mitigate the effect of external factors the Immersive Environment, we choose the HTC Vive Pro and Meta 2 headsets for our comparison of pointing tasks in VR and AR. Both headsets have reasonably comparable fields of view, resolutions, and weight. We used them on the same computer, and we achieved an average refresh frequency for the HTC Vive Pro and Meta 2 of 90 Hz and 80 Hz, respectively. The similarity in specifications for the two headsets leads us to expect (roughly) comparable outcomes.

Early pilots made us only too aware of the differences in the headset tracking systems, with different latencies and different degrees of accuracy. To avoid these potential confounds, we chose to use an external OptiTrack system to track both VR and AR headsets as well as the pen and controller. Thus, the tracking latency and quality should be comparable for both headsets as well as the two input devices.

Different Display Latencies

The results of the main study revealed that in the AR condition participants took longer to point, manipulated the cursor slower, and achieved less throughput compared to VR. The most likeliest cause for this is different headset display latencies. To determine the display latencies for both headsets, we measured the delay between the movement of a physical input device and that of a virtual object shown on the device, by moving a controller in front of each headset display showing the manipulated sphere and recording both with a 240 Hz camera [45, 37]. By observing the delays in the movement in the display device in several trials, we measured the average latency of the VR condition to be 51 ms and 79 ms for AR. Given these different latencies and that previous work identified a clear effect of latency on pointing performance [45, 37], we believe that latency is a good explanation for the differences observed between the AR and VR conditions.

Other Differences Between Headsets

The VR headset allows users to see only the virtual scene, while the AR headset allows them to also see the real world. To reduce potential issues due to the (virtual or real) environment visible in the headset, we replicated the general appearance of the physical room in VR.
Yet, there might be still some differences due to different lighting and material properties. One third of participants identified that they sometimes got distracted by real objects. One participant said “The fact that the room was [free] of objects made it easier to perform the task in VR.” Another reported “It was easier to concentrate in VR without [the] real world image.” While users might have been able to concentrate less in an AR environment, we point out that the pointing task is fairly repetitive. Thus, it is not surprising that participants pay more attention if they interact only with virtual content.

During the experiment the experimenter also observed that user concentration seems to be a notable factor that affect the ability to perform the pointing task efficiently. The presence of real objects in the AR environment can distract the focus of participants. Virtual objects whose appearances are "not real" and limited by the resolution of the headsets may distract less. It is thus possible that the performance of pointing tasks in AR is possibly affected by distractions caused by the surrounding environment.

Participants also found the AR condition to be less comfortable. One explanation for this is that, although lighter than the Vive, the Meta 2 AR headset has a relatively unbalanced design, with the front being substantially heavier.

Unbalanced design of AR headset

Although the weight of both headsets is pretty similar, the Meta 2’s design is not as balanced as the HTC Vive Pro. A third of participants complained that the AR headset was too heavy. However, it is actually slightly lighter than the VR headset (500g vs 555g). The most possible explanation is the unbalanced distribution of weight on the Meta 2. Its front part is bigger and heavier than its rear part, causing considerable pressure to be put on the user’s forehead. Although we carefully fit and tightened the AR headset on their head, this fact seems to have caused some discomfort for our participants. The reduced comfort might have contributed to the worse performance in the AR condition.

2.6.2 User preferences

Interestingly, participants perceived the pen to be more comfortable than the controller. One participant even explicitly identified that "[the] pen-like device is smaller and lighter compared to the HTC [Vive] controller." Although they had never tried a pen in VR and AR before, they got quickly used to it, as "it felt like using your finger [for pointing]". While it achieved equivalent performance to the pen, the mouse received poor reviews from the participants. It is more familiar, but "it wasn’t convenient since the movement was horizontal [on the mouse pad]' and 'the cursor seemed to appear from nowhere which made it difficult [to point]'.
2.6.3 Different Latency for the Mouse

A notable limitation is that we used a different tracking method for the mouse than the other two input devices. The mouse’s light emitter and detector track its relative movement on a surface, while the position and orientation of controller and pen are determined by OptiTrack – an external optical tracking system. As latency reduces pointing performance [45, 37], at least above an end-to-end latency of about 50 ms, this can potentially reduce pointing performance. For simplicity, we discuss only the AR condition here, results for VR are analogous.

We measured the end-to-end latency for the mouse and the pen with the AR headset and observed 55 ms respectively 79 ms, i.e., a 24 ms difference. Given that the latency for the pen was higher, a low-latency implementation of a pen should perform better than our apparatus. This means that the throughput measurements in our study form a lower bound for the pen and we expect that future implementations might perform even better.

Latency of OptiTrack system

Interestingly, and given that the Vive tracking system is known to have an latency of approximately 22 ms [34], we can expect that a mouse tracked by the Vive system should perform similarly. Given this additional latency, and based on the fact that we expect that the controller and pen would suffer relative to the mouse, as both of these devices use the OptiTrack tracking system. Yet, our results show that the outcomes for the pen match or even (non-significantly) exceed the results for the mouse. We believe that this means that a truly low-latency implementation for the pen has the potential to even exceed the throughput for the mouse.

2.6.4 Pen is Better than the Controller

Overall, our results indicate that a pen is better, i.e., “mightier”, than a controller in all objective measurements. Although the Vive controller is specifically designed for VR interaction, the pen helped users to complete pointing tasks significantly quicker, to manipulate the cursor faster, to make fewer errors, and to reach higher throughput. 83.33% of our participants had no or only a little experience with VR and AR at the time of participation. Consequently, most of them were not familiar with a controller nor did they expect that one could use a pen in such systems. The result, therefore, was not biased towards either of these devices.

User behaviors

Analyzing videos captured during the experiment, we saw that participants usually used their wrist and sometimes their arms to control the direction of the controller, as shown in figure 2.5. On the other hand, they used the movement of at least three fingers and
sometimes their wrist to manipulate the mouse and the pen, as shown in figure 2.5. This matches observations from previous work on 2D input [4], but extends their results to 3D pointing. We believe the pen/precision grip using multiple fingers is the likeliest explanation for the better pointing performance of the pen compared to the controller, which is grasped with the palm. Another indication is that our pen is controlled similarly to how people use a real pen to write/draw on paper. This similarity is another explanation for higher accuracy of the pen, and thus higher throughput, compared to the controller.

2.6.5 Pen is Comparable to the Mouse

Our results identified no significant difference between the pen and mouse in most measurements, including task completion time, error rate, and throughput. While this lack of a significance does not mean that there is no difference (one “cannot prove the null”), we point out that the average performance of our pen exceeds that of the mouse. Given that the mouse is used daily as an input device, it is likely more familiar than a controller or pen. However, this potential advantage still does not yield a significantly better performance than the pen. Finally, participants liked the pen better than the mouse for pointing tasks in VR and AR.

2.6.6 Applications of the Pen

The pen achieved an average throughput of 4.7 bit/s, compared to 4.0 bit/s of the controller. This difference may not have a considerable impact on most current VR and AR games where game objects are usually big enough for players to easily see and interact with. The distances of these objects from the player are also adjusted so that the users can easily point at and select them unambiguously with the controller.

However, for applications, such as engineering, that require accuracy, the performance of pointing devices is important to VR and AR users. Consider a user drawing a part in AR that has to match the dimensions and shape of a real object, e.g., to add a handle or to replace a broken part. Pointing to accurate locations may require a device as good as the mouse, which is very familiar to computer users. As indicated in the results of our experiment, the pen is at least comparable to and potentially better than the mouse, 4.7 bit/s vs. 4.5 bit/s on average. The pen is, therefore, a promising alternative for professional applications, such as immersive analytics and 3D engineering software. It enables VR and AR users to, e.g., select a specific data point on a complex graph, move a virtual part to match another, or draw a line between two points in a fast and accurate manner. The pen also increases user productivity and the quality of the experience relative to the controller, while removing the need for a desktop-like surface for operating the mouse.
2.6.7 VR/AR Pen Design Space

Although our outcomes identify the pen to be comparable to the mouse in pointing performance, the specifics of its design have a strong impact on user performance. Given the different results we observed in our pilots, we believe that weight and shape, i.e., ergonomics factors, are among the biggest issues. Also, the current design of our pen lacks one or more buttons. Yet, it is fairly easy to augment a pen with several buttons, as evidenced by the pens used for pen tablets. To support additional input one can even add a touchpad, see figure 2.10. Alternatively, we could add a touchstrip along its length to at least support forward-backward scrolling/movements. Such a pen can then alternatively be used like a controller, simply by holding it in a different grip. This then makes a pen equivalent to a controller as a 6DOF input device. Alternatively, future 6DOF controllers could also be made slimmer, so that they could be used as both controllers and pen-like devices.

Looking at current headsets, we identified that many have surfaces where a pen could be magnetically snapped to, either to the side of the headset (say roughly at the temple) or the top, see figure 2.10. This permits the user to store a pen temporarily while they are not using it, but to also quickly and easily retrieve it when they need it. A simple extension of this concept is to use two pens, one on either side of the headset. Given that a pen can also be held like a controller, as discussed above, this two-pen system then creates a system that is very similar to the two-controller setup offered by several current VR systems, e.g., the HTC Vive, but also affords precision input.
2.6.8 Limitations

Motion Sickness

While we used an external tracking system to reduce potential issues introduced by different tracking systems, we recognize that this likely affected display latencies for the VR and AR conditions, as we could not benefit from the built-in motion compensation in the HTC Vive and Meta 2 headsets. However, such compensation systems are (mostly) targeted at reducing motion sickness during head movements. As participants sat in a chair and kept their head (relatively) stable during the experiment, we believe that the lack of motion compensation was not a main factor. Moreover, none of our participants reported motion sickness symptoms. Still, better tracking might further reduce input latencies.

Participant Group

Our study involved only a small number of participants. Still, we observed not only significant differences, but also medium or large effect sizes for our main results. This makes us believe that our results are reasonably robust.

Design of Pen

We chose the shape of a typical whiteboard pen to create the prototype of our VR pen, as this kind of pen is familiar to many people. Such a pen is also lighter than the controller, which enables people to easily control the device with three fingers. In our pilot studies, we asked participants to hold the controller like a pen. The results showed that the controller was too heavy to be held in a precision grip and thus did not result in significantly higher pointing performance than when held in a power grip. Hence, we believe that the weight and how it is distributed on the pointing device matter. However, how exactly the weight and its distribution in a pointing device affects the performance is still unknown and needs further investigation.

We asked participants to hold the pen in a typical pencil grip, more specifically a dynamic tripod grasp, which is only one of many precision grips. We acknowledge that there are other pen grips, e.g., where the pen passes underneath or over the thumb. We do not yet know if these forms have different effects on the pointing performance of a pen.

The pen in our study did not have any button, as we wanted to avoid the Heisenberg effect [7]. We asked participants to confirm target selection with another device, i.e., by pressing the space bar on a keyboard. However, a more commercially viable design of the pen should includes buttons, although the number of such buttons may have to be smaller than the controller because of limited area on the pen’s surface. Consequently, it is worth exploring how clicking a button on the pen affects the accuracy of selection, how many buttons should be supported, and where they are placed on the device.
Chapter 3

Text Entry in Virtual Reality

3.1 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 built-in touchpads to enter text. Such solutions afford typing in a “hunt-and-peck” style, where the typist (potentially) finds each key by sight and presses each key individually. Yet, this typing style does not approach the text entry performance afforded by touch-typing with ten fingers on regular keyboards. We focus on physical keyboards and how users can efficiently use them in VR.

Figure 3.1: Left: RGB-D camera attached to VR headset. Middle: Stabilizing straps. Right: Text entry task.
Figure 3.2: Text Entry Interfaces in VR: Baseline, None, Frame, Model, Video, and Point Cloud.

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 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 [50]. For this both the keyboard and the user’s hands need to be visible in the VE [24, 16]. Yet, the VR representation of the physical keyboard should obstruct the VE as little as possible. Alas, 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 [33], markers on the hands [17, 24], or special keyboard covers [22]. 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.

With a portable keyboard, users are then also free to move around in their space. While previous work requires users to sit at a desk [28, 22], we propose a new portable keyboard setup, HawKEY, that allows users to either sit or stand while still being able to enter text. This gives the user the freedom to stand while being immersed into VR, but also to sit
down when having to enter larger amounts of 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 [24, 28, 50, 33]. 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. 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.

3.2 Related Work

Previous work has investigated various approaches to text entry in VR. Some explored voice or gesture recognition to input words and phrases, but did not identify reasonably good performance [8, 21, 27, 41]. Others showed not only the presence of the keyboard in the VE but also provided visual and/or haptic feedback [16, 24, 52, 18].

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 [12]. 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 [52], 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. HoVR [23], 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 [18].

Vulture, a word-gesture virtual keyboard, uses optical tracking to determine the users’ gestures for enable key selections [31]. 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 [13].

3.2.1 Mixed Reality Solutions

The HiKeyb system [22] 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. [24] 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. [16] 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. [33] 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. [28] 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 3.1 summarizes typing performance in words per minutes (WPM) in previous works and our study. It listed the results of the real-world baseline condition and the VR condition with the highest WPM.

3.3 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.
Table 3.1: 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.

<table>
<thead>
<tr>
<th>Study</th>
<th>Pose</th>
<th>Typist</th>
<th>Text</th>
<th>Baseline</th>
<th>VR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jiang [22]</td>
<td>seated</td>
<td>-</td>
<td>lowercase</td>
<td>32.5</td>
<td>23.1</td>
</tr>
<tr>
<td>McGill [33]</td>
<td>seated</td>
<td>-</td>
<td>lowercase</td>
<td>58.9</td>
<td>38.5</td>
</tr>
<tr>
<td>Grubert [16]</td>
<td>seated</td>
<td>-</td>
<td>lowercase</td>
<td>-</td>
<td>38.7</td>
</tr>
<tr>
<td>Lin [28]</td>
<td>seated</td>
<td>-</td>
<td>lowercase</td>
<td>39.8</td>
<td>28.1</td>
</tr>
<tr>
<td>Google [12]</td>
<td>standing</td>
<td>expert</td>
<td>lowercase</td>
<td>-</td>
<td>50.0</td>
</tr>
<tr>
<td>Knierim [24]</td>
<td>seated</td>
<td>expert</td>
<td>lowercase</td>
<td>67.2</td>
<td>69.2</td>
</tr>
<tr>
<td>HawKEY</td>
<td>standing</td>
<td>expert</td>
<td>lowercase</td>
<td>76.1</td>
<td>77.7</td>
</tr>
<tr>
<td>HawKEY</td>
<td>standing</td>
<td>novice</td>
<td>lowercase</td>
<td>50.3</td>
<td>44.1</td>
</tr>
<tr>
<td></td>
<td>standing</td>
<td>expert</td>
<td>complex</td>
<td>45.7</td>
<td>41.5</td>
</tr>
<tr>
<td></td>
<td>standing</td>
<td>novice</td>
<td>complex</td>
<td>27.9</td>
<td>21.6</td>
</tr>
</tbody>
</table>

3.3.1 Text Entry Task

We evaluated the performance of different Text Entry Interfaced 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 3.1. 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 [3].

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.

3.3.2 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 3.2.

- **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 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.

### 3.3.3 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**.
Figure 3.3: Top-left: Baseline condition. Bottom-left: Using HawKEY while sitting. Right: A participant standing and wearing HawKEY and the VR headset during the study.

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 exemplar 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.
3.4  Experimental Setup

3.4.1  Subjects

We recruited 16 participants from the local university population for our study, 7 of them were female. Their ages ranged from 19 to 30 (\(M = 22.686\), \(SD = 3.321\)). All of them had limited experience with VR. They had tried VR a few times but did not use VR regularly. They earned 1% of course credit through this study.

3.4.2  Apparatus

To guarantee that machine performance is not a limiting factor for our study, we used a PC with an Intel® Core™ i7-4790, 16GB RAM, and a NVIDIA Geforce GTX 1080, running Windows 10. The components of this PC far exceed the requirements for text entry tasks.

Virtual Reality Headset

We used a HTC Vive Pro, one of the newest VR headsets at the time of writing. The total display resolution for both eyes is 2880 × 1600, with 90 Hz expected refresh rate, see figure 3.1. The horizontal field of view is approximately 100°. The Vive Pro weighs about 550 g.

Large display

The large display is only used in the baseline condition where we examine the normal typing performance of participants. It is a 4K 85” display on a movable stand, positioned 1.5 meter from the center of our experimental area, see figure 3.3. We show the panel containing the presented text and the text box for transcription at the same (relative) locations as they were displayed in the virtual world.

RGB-D Camera

To collect 2D and 3D information of the physical keyboard and user’s hands, we use Intel® RealSense™ Depth Camera D435, see figure 3.1. It has both color and depth sensors. The horizontal field of view of these sensors is at least 70°, wide enough to capture the whole keyboard located on our tray. Thanks to the auxiliary USB port on the Vive Pro headset, we can easily attach the camera at the front.

Physical Keyboard

We used a Logitech K480 Bluetooth keyboard, see figure 3.3. This has an appropriately small form factor, which allows the RGB-D camera to fully capture the keyboard while still reserving space for the user’s hands when they are positioned at both sides of the keyboard. The lack of a designated numeric keypad also helps to ensure study validity as users do not have a choice as to which key to press when entering a number. To ensure the best
connectivity, we installed a Bluetooth receiver close to the experimental area. Moreover, to enhance the readability of the keys, we used customized label stickers with larger symbols.

HawKEY

The body-attachable tray plays an important role in our proposed versatile text entry solution, see figure 3.3. It includes a surface where the keyboard rests on and two adjustable straps which helps to stabilize the tray in front of user’s body and parallel to the ground. We call this design HawKEY because of the similarity between it and a hawker’s tray. We use a controller rigidly attached to the tray to track the keyboard’s position in space and to enable us to display its virtual representation in the correct location, too.

3.4.3 Phrase set

Inspired by the creation process for MacKenzie’s phrase set [30], we collected 67 Simple and 72 Complex sentences on various topics. Simple sentences had a mean of $M = 65.7$ 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 [30], as used in previous work [28, 24, 33]. 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 $M = 19\%$, $SD = 0.08$. In addition, some of these complex characters required the Shift modifier key, $M = 9\%$, $SD = 0.05$ of the total.

3.4.4 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 counterbalanced 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.

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.

**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 counter-balanced. We enforced a break of 10 seconds between sentences.

3.4.5 **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 survey [19] to provide their perceived workload and a questionnaire to record ratings and comments. Each condition took around 10 minutes and there was a final post-study survey to ask for any suggestions or recommendations at the end. Depending on the participant’s typing performance, it lasted between 60 to 90 minutes.

3.5 **Results**

We performed two-way repeated-measures ANOVA on all collected measures, including words per minute (WPM), keystrokes per character (KSPC), erroneous keystroke error rate (EKS ER), and total error rate (Total ER). Greenhouse-Geisser (if its epsilon was smaller than .75 [14]) 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 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 [51] 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
Table 3.2: Two-way Repeated-Measures ANOVA on Text Entry Interface (TEI), Text Complexity (TC), and their interaction (TEI×TC) (*p ≤ .05, **p < .01, ***p < .001, insig. p > .05).

<table>
<thead>
<tr>
<th>Measure</th>
<th>Factor</th>
<th>ANOVA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>df1</td>
<td>df2</td>
</tr>
<tr>
<td>WPM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEI</td>
<td>2.549</td>
<td>38.230</td>
</tr>
<tr>
<td>TC</td>
<td>1.000</td>
<td>15.000</td>
</tr>
<tr>
<td>TEI×TC</td>
<td>5.000</td>
<td>75.000</td>
</tr>
<tr>
<td>Time to First Character</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEI</td>
<td>2.095</td>
<td>31.426</td>
</tr>
<tr>
<td>TC</td>
<td>1.000</td>
<td>15.000</td>
</tr>
<tr>
<td>TEI×TC</td>
<td>2.978</td>
<td>44.676</td>
</tr>
<tr>
<td>First Character Correctness</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEI</td>
<td>2.224</td>
<td>33.367</td>
</tr>
<tr>
<td>TC</td>
<td>1.000</td>
<td>15.000</td>
</tr>
<tr>
<td>TEI×TC</td>
<td>5.000</td>
<td>75.000</td>
</tr>
<tr>
<td>KSPC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEI</td>
<td>1.647</td>
<td>24.712</td>
</tr>
<tr>
<td>TC</td>
<td>1.000</td>
<td>15.000</td>
</tr>
<tr>
<td>TEI×TC</td>
<td>1.823</td>
<td>27.347</td>
</tr>
<tr>
<td>EKS ER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEI</td>
<td>1.514</td>
<td>22.712</td>
</tr>
<tr>
<td>TC</td>
<td>1.000</td>
<td>15.000</td>
</tr>
<tr>
<td>TEI×TC</td>
<td>2.236</td>
<td>33.537</td>
</tr>
<tr>
<td>Total ER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEI</td>
<td>1.624</td>
<td>24.363</td>
</tr>
<tr>
<td>TC</td>
<td>1.000</td>
<td>15.000</td>
</tr>
<tr>
<td>TEI×TC</td>
<td>2.744</td>
<td>41.153</td>
</tr>
</tbody>
</table>

text, respectively. To present all results compactly, we list statistical results in tables 3.2, 3.3, 3.4, and 3.5.

3.5.1 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.

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 3.2. Investigating Simple and Complex Sentences separately, ANOVA revealed that the effect of Text Entry Interface was only significant for Complex ones, see table 3.3.
Figure 3.4: Average text entry measures (*top-left: words per minute, top-right: time to first key, bottom: first key correctness) with different keyboard representations in real and virtual environments (*$p \leq .05$, **$p < .01$, ***$p < .001$). Error bars represent 95% confidence intervals.
Table 3.3: Analysis of different Text Complexities on Text Entry Interfaces for entry rate and first character (*p ≤ .05, **p < .01, ***p < .001, insig. p > .05). For averages across Simple and Complex, see table 3.2.

<table>
<thead>
<tr>
<th>MEASURE</th>
<th>Text Comp.</th>
<th>ANOVA on Text Entry Interfaces</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>df1</td>
<td>df2</td>
</tr>
<tr>
<td>WPM</td>
<td>Simple</td>
<td>2.750</td>
</tr>
<tr>
<td></td>
<td>Complex</td>
<td>3.124</td>
</tr>
<tr>
<td>Time to First Character</td>
<td>Simple</td>
<td>2.337</td>
</tr>
<tr>
<td></td>
<td>Complex</td>
<td>2.075</td>
</tr>
<tr>
<td>First Character</td>
<td>Simple</td>
<td>2.483</td>
</tr>
<tr>
<td>Correctness</td>
<td>Complex</td>
<td>2.994</td>
</tr>
<tr>
<td>KSPC</td>
<td>Simple</td>
<td>1.861</td>
</tr>
<tr>
<td></td>
<td>Complex</td>
<td>1.690</td>
</tr>
<tr>
<td>EKS ER</td>
<td>Simple</td>
<td>1.855</td>
</tr>
<tr>
<td></td>
<td>Complex</td>
<td>1.769</td>
</tr>
<tr>
<td>Total ER</td>
<td>Simple</td>
<td>1.810</td>
</tr>
<tr>
<td></td>
<td>Complex</td>
<td>2.071</td>
</tr>
</tbody>
</table>

Table 3.4: The statistics of entry rate and first character of Text EntryInterfaces

<table>
<thead>
<tr>
<th>MEASURE</th>
<th>Baseline (1)</th>
<th>None (2)</th>
<th>Frame (3)</th>
<th>Model (4)</th>
<th>Video (5)</th>
<th>Point Cloud (6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIMPLE WPM</td>
<td>M SD</td>
<td>M SD</td>
<td>M SD</td>
<td>M SD</td>
<td>M SD</td>
<td>M SD</td>
</tr>
<tr>
<td></td>
<td>63.203 .4902</td>
<td>59.149</td>
<td>6.682</td>
<td>54.640</td>
<td>5.914</td>
<td>56.299</td>
</tr>
<tr>
<td>Time to 1st Ch.</td>
<td>1.280 .094</td>
<td>1.446</td>
<td>.222</td>
<td>1.325</td>
<td>.144</td>
<td>1.486</td>
</tr>
<tr>
<td>1st Ch. Corr.</td>
<td>.984 .016</td>
<td>.844</td>
<td>.060</td>
<td>.797</td>
<td>.076</td>
<td>.891</td>
</tr>
<tr>
<td>COMPLEX WPM</td>
<td>M SD</td>
<td>M SD</td>
<td>M SD</td>
<td>M SD</td>
<td>M SD</td>
<td>M SD</td>
</tr>
<tr>
<td>Time to 1st Ch.</td>
<td>1.714 .240</td>
<td>1.382</td>
<td>.132</td>
<td>1.452</td>
<td>.140</td>
<td>1.505</td>
</tr>
<tr>
<td>1st Ch. Corr.</td>
<td>.906 .031</td>
<td>.719</td>
<td>.072</td>
<td>.734</td>
<td>.066</td>
<td>.797</td>
</tr>
</tbody>
</table>

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 3.4.

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.

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 3.2.
Table 3.5: The statistics of error rates of Text Entry Interfaces

<table>
<thead>
<tr>
<th>MEASURE</th>
<th>Baseline (1)</th>
<th>None (2)</th>
<th>Frame (3)</th>
<th>Model (4)</th>
<th>Video (5)</th>
<th>Point Cloud (6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIMPLE</td>
<td>M SD</td>
<td>M SD</td>
<td>M SD</td>
<td>M SD</td>
<td>M SD</td>
<td>M SD</td>
</tr>
<tr>
<td>KSPC</td>
<td>1.057 .015</td>
<td>1.272 .106</td>
<td>1.128 .078</td>
<td>1.228 .073</td>
<td>1.127 .029</td>
<td>1.144 .044</td>
</tr>
<tr>
<td>EKS ER</td>
<td>.028 .007</td>
<td>.133 .052</td>
<td>.128 .038</td>
<td>.112 .036</td>
<td>.063 .014</td>
<td>.070 .022</td>
</tr>
<tr>
<td>Total ER</td>
<td>.026 .007</td>
<td>.094 .030</td>
<td>.099 .025</td>
<td>.085 .024</td>
<td>.055 .011</td>
<td>.058 .015</td>
</tr>
<tr>
<td>COMPLEX</td>
<td>M SD</td>
<td>M SD</td>
<td>M SD</td>
<td>M SD</td>
<td>M SD</td>
<td>M SD</td>
</tr>
<tr>
<td>KSPC</td>
<td>1.203 .018</td>
<td>1.795 .206</td>
<td>1.384 .076</td>
<td>1.384 .081</td>
<td>1.384 .076</td>
<td>1.379 .058</td>
</tr>
<tr>
<td>EKS ER</td>
<td>.056 .009</td>
<td>.277 .065</td>
<td>.224 .036</td>
<td>.224 .036</td>
<td>.130 .031</td>
<td>.127 .025</td>
</tr>
<tr>
<td>Total ER</td>
<td>.050 .008</td>
<td>.183 .032</td>
<td>.192 .033</td>
<td>.165 .022</td>
<td>.103 .020</td>
<td>.104 .067</td>
</tr>
</tbody>
</table>

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 3.2. Investigating Simple and Complex Sentences separately, ANOVA revealed that the effect of Text Entry Interface was only significant for Complex ones, see table 3.3. However, Holm-Bonferroni corrected post-hoc tests revealed no significant differences for Complex Sentences, see figure 3.4.

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.

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 $KSPC = \frac{|IS|}{|IT|}$, 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 3.2. 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 3.3. 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 fig-
Figure 3.5: Error rates of users (top-left: keystrokes per character, top-right: erroneous keystroke error rate, bottom: total error rate) with different keyboard visualizations in real and virtual environments (*$p \leq .05$, **$p < .01$, ***$p < .001$). Error bars represent 95% confidence intervals.
ure 3.5. On the other hand, post-hoc tests for Simple Sentences showed that KSPC with Baseline was significantly smaller than Frame and Video.

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 $EKS\ ER = \frac{IF}{|T|} \times 100\% = \frac{IF}{|T|}$, 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 3.2. 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 3.3. 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 3.5. Post-hoc tests for Simple Sentences indicated that EKS ER with Baseline was significantly smaller than Frame and Video.

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 [3]. 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 3.2. 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 3.3. 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 3.5. Post-hoc tests for Simple Sentences indicated that Total ER with Baseline was significantly smaller than Frame and Video.

3.5.4 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.
Figure 3.6: NASA-TLX with different keyboard visualizations in virtual reality (*\(p \leq .05\), **\(p < .01\), ***\(p < .001\)). Error bars represent 95% confidence intervals.

**Task Load Index**

We asked participants to complete the NASA-TLX [19] 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 3.6.

**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 3.7.

**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 3.7.
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 3.7.

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 3.7.

3.6 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 [24], Grubert [17], or McGill [33]. 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.

We identified the Video condition as the most preferable Text Entry Interface in VR as it was comfortable and easy to use. Although our results did not reveal any significant
Figure 3.7: Subjective measures of users (top-left: ease of use, top-right: comfortability, bottom-left: perceived typing speed, bottom right: preference) with different keyboard visualizations in virtual reality (*p ≤ .05, **p < .01, ***p < .001). Error bars represent 95% confidence intervals.
difference in WPM between Video and None, the former condition helped participants to reduce their Total Error Rate from 9.4% to 5.5% with Simple Sentences and significantly from 18.3% to 10.3% with Complex Sentences. With no keyboard visualization, participants tended to make more mistakes and they often tried several keys until they pressed the correct one. Also, according to the NASA-TLX results, seeing a keyboard reduced their mental workload.

As the Video condition is technically (substantially) simpler to implement, we see this condition currently as the overall best choice. 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.

3.6.1 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 3.8 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.
3.6.2 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 3.3. 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*.

3.6.3 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.

3.6.4 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. [24], with 67.2 WPM and 69.2 WPM, respectively. This is very notable, since our participants did this while standing, whereas 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.

### 3.6.5 User Feedback

According to the NASA-TLX results, see figure 3.7, 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”.

**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”.

### 3.6.6 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 3.9. 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, see figure 3.10.
Figure 3.9: A retractable design of HawKEY (in use).

Figure 3.10: A retractable design of HawKEY (retracted).
Chapter 4

Conclusion and Future Work

Here we summarize the contributions of this thesis and discuss future avenues.

4.1 Pen - A Promising pointing device for VR and AR

We presented a pen-like VR interaction device and a comparison of different pointing devices with VR and AR headsets. Overall, interaction in VR was faster than in AR, potentially due to the higher latency in the AR system. We also identified that a VR pen device can afford throughput at least as high as the mouse, both of which are significantly better than a typical VR controller. Also, participants liked to use a pen device in VR and AR. Finally, we presented several design options for pens in VR and AR systems.

In general, the results of our user study strongly support the introduction of pen-like devices into the VR market. We demonstrated the potential of pens for easy-to-use and accurate distant selection in modern fully- or semi-immersive environments. Compared to the popular controller held in a power grip, a pen held in a precision grip is lighter, slimmer to hold, and easier to control, which helps VR/AR users to deliver higher performance and potentially also to make fewer errors. Though it did not outperform the traditional mouse, our study identified its potential for achieving even higher performance, which paves the way for further improvements through pen-like VR pointing devices.

In the future we also plan to measure the performance of (virtual) hand-based interaction in VR and AR.

4.2 HawKEY - A Versatile Text Entry Solution in VR

We presented HawKEY, a new text entry method usable either while standing or sitting in a VR system and which affords text entry rates that are comparable to those achievable while sitting in front of a desk. We also examined different representations of physical keyboards in VR and found that a see-through video condition is overall likely the best solution. It not only affords desktop-level typing speeds for lowercase content, but also yields acceptable
entry speeds and error rates for less frequently used symbols, numbers and other characters. Also, the video condition is technically easier to implement than most other previously presented solutions. Users got quickly used to HawKEY and felt comfortable typing on it, especially since they could see their own hands. Our participants also appreciated that that the keyboard automatically disappeared when they looked straight ahead in the video condition, which makes HawKEY unobtrusive during a VR experience.

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. We also plan to explore adding “controller holders” at the side of the HawKEY tray and a pen snapped to the headset, see figure 4.1.
Bibliography


[34] Diederick C. Niehorster, Li Li, and Markus Lappe. The accuracy and precision of position and orientation tracking in the htc vive virtual reality system for scientific research. i-Perception, 8(3):2041669517708205, 2017.


Appendix A

Post-study Questionnaire of Pointing Devices

For each pointing device (mouse/controller/pen), we asked participants the following questions:

- Please rate how easily you control [the given pointing device] to move the cursor to hit the targets (from 0-very difficult to 100-very easy).
- Please rate how comfortably you hold and control [the given pointing device] in the pointing tasks (from 0-very uncomfortable to 100-very comfortable).
- Please rate your perceived speed between two consecutive target selections with [the given pointing device] (from 0-very slow to 100-very fast).

For each immersive environment (VR/AR), we asked participants the following questions:

- Please rate how easily you get used to [the given environment] to perform the pointing tasks (from 0-very difficult to 100-very easy).
- Please rate how comfortably you are in [the given environment] while performing the pointing tasks (from 0-very uncomfortable to 100-very comfortable).
- Please rate your perceived speed between two consecutive target selections in [the given environment] (from 0-very slow to 100-very fast).

For each combination of pointing devices (mouse/controller/pen) and immersive environments (VR/AR), we asked participants the following question:

- Please rate your preference for using [the given pointing device] to perform the pointing tasks in [the given environment] (from 0-less preferable to 100-more preferable).
Appendix B

Post-study Questionnaire of Text Entry Interfaces

For each text entry interface (None/Frame/Model/Video/Point Cloud), we asked participants the following questions:

- Please rate how **easily** you get familiar to [the given interface] to perform the text entry tasks (from 0-very difficult to 100-very easy).
- Please rate how **comfortable** you feel with [the given interface] during the text entry tasks (from 0-very uncomfortable to 100-very comfortable).
- Please rate your **perceived typing speed** while performing the text entry tasks with [the given interface] (from 0-very slow to 100-very fast).
- Please rate your **preference** for using [the given interface] to perform the text entry tasks in VR (from 0-less preferable to 100-more preferable).