Institute for Visualization and Interactive Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Bachelorarbeit

Evaluation of typing performance in virtual reality on a physical keyboard

Florian Nieuwenhuizen

Course of Study:	Softwaretechnik
Examiner:	Prof. Dr. Albrecht Schmidt
Supervisor:	Pascal Knierim, M.Sc., Dipl.-Inf. Valentin Schwind
Commenced:	Febuary 1, 2017
Completed:	August 1, 2017
CR-Classification:	H.5.1

Kurzfassung

Wenn heutzutage über Virtual Reality (VR) gesprochen wird, dann verbinden die meisten Personen damit nur Computerspiele. VR hat jedoch weitere Anwendungsgebiete, wie beispielsweise Menschen zu trainieren oder bei der Arbeit. Eingaben, besonders Texteingaben, waren in VR schon immer problematisch. Es gab bereits verschiedene Ansätze, die Texteingabe in VR einzubringen. Es hat noch nie jemand versucht, das meist gebrauchte Eingabegerät für den Computer, die Tastatur, in Verbindung mit submillimeter genau gemessenen, virtuellen Händen zu verwenden.

Im Rahmen dieser Ausarbeitung wurde ein Prototyp entwickelt, welcher das Motion Capturing System OptiTrack mit VR Technologie und dem Unity Game Engine verbindet. Diese Kombination ermöglicht, dass wir Hände im virtuellen Umfeld in Echtzeit erleben können. Mit diesem Prototypen wollten wir herausfinden, welche Effekte das Sehen von Händen auf die Tippleistung in der VR haben. Außerdem hat es uns auch interessiert, ob es Leistungsunterschiede gibt, wenn in der VR und außerhalb der VR getippt wird. Während des tippens wurden den Benutzern verschiedene Arten von Händen vorgestellt. Weiterhin wollten wir herausfinden, ob die Tippleistung negativ beeinflusst wird, wenn überhaupt keine Hände sichtbar sind. Um dies zu beantworten, wurden verschiedene Handmodelle erstellt, mit welchen ein Tipptest im virtuellen Umfeld vollzogen werden sollte. Eine Studie mit 32 Teilnehmern wurde durchgeführt, bei welcher jeder Teilnehmer sieben verschiedene Konditionen mit VR und eine Kondition außerhalb von VR durchzuführen hatte. Jede VR Kondition beinhaltete ein anderes Handmodell. Die Teilnehmer wurden in zwei gleich große Tipp-Gruppen aufgeteilt, in eine schnelle und eine langsame.

Im Schnitt, wenn die Tippleistung aller Probanden gleichzeitig angeschaut wurde, wurde kein statistisch signifikanter Unterschied in Wörtern pro Minute (WPM) und Genauigkeit gefunden, ob in der VR getippt wurde oder außerhalb. Die Gruppe der langsamen Tipper hat einen statistisch signifikanten Unterschied in WPM und Genauigkeit bei diesem Vergleich gezeigt.

Weiterhin wurde entdeckt, dass das Sehen von einem beliebigen Handmodell während des Tippens in der VR wichtig ist. Überhaupt keine Hände zu sehen hat eine negative Auswirkung auf die Tippleistung, erhöht den Arbeitsaufwand und verringert die wahrgenommene Präsenz im virtuellen Umfeld.

Außerdem wurde herausgefunden, dass verschiedene Handmodelle innerhalb der VR keine statistisch signifikanten Unterschiede auf die Tippleistung haben, solange ein beliebiges Handmodell sichtbar ist.

Abstract

Nowadays, when people talk about virtual reality (VR), most think about playing computer games. VR has many fields of use, like for training and at work, that most people haven't thought about. Input, especially text input has always been difficult for VR. Many different attempts at allowing text input in VR have been made, but none have ever combined the most used input method for the computer, the keyboard, with sub-millimeter accurate, visible hands.

In the scope of this thesis, a prototype combining a motion capturing system, OptiTrack, with VR technology and the Unity game engine, was created to allow viewing hands in realtime, while inside of a virtual environment. Using this prototype, we wanted to find out, which effect hands have on typing inside of a virtual environment. Furthermore, we were interested, if there are differences in typing performance while typing inside of VR in comparison to typing outside of VR. Users will be typing while viewing different forms of hands. Is typing performance negatively impacted if no hands are visible? To answer this, different hand models were created for performing a typing test while immersed inside of the virtual environment. A study 32 participants was conducted, where each participant completed the typing test seven times while in VR and once outside of VR. Each VR condition contained a different hand model. These participants were divided into two equal typing groups, a slow and a fast one.

On average, when analyzing the data of all participants at once, we found no statistically significant difference in words per minute (WPM) or accuracy when comparing typing inside of VR to typing outside of VR. The slow typers, did show a statistically significant difference in WPM and accuracy in this comparison.

Furthermore, it was discovered, that seeing any sort of hand model while typing in VR is important. Not being able to see hands had a negative effect on typing performance, increased task load, and also decreased the perceived presence in VR.

We found that using different hand models inside of VR does not statistically significantly affect typing performance, as long as some variation of a hand is visible.

Contents

List of Figures

1 Introduction

In the age of modern computing, head mounted displays (HMDs) have become more popular, affordable and available to consumers. When talking about virtual reality, society always makes the connection to enthusiast gaming. Game developers attempt to create products where the presence, interaction and entertainment are increased, through the use of wearing an HMD.

Gaming is only one aspect where virtual reality can be utilized. Unfortunately, most people haven't made the connection of using VR for example, at work or in training. VR is, for example, already being utilized in simulators to train firefighters in extinguishing fires.

When imagining a general office workplace setup, we realize, that one employee needs a large table, often times two monitors and a chair. Besides that, the workplace environment is very often also enhanced through plants and other visually pleasing objects that improve the workplace. Since office buildings, especially in big cities, are often expensive and small, space is a desirable commodity. Replacing the aforementioned office setup through a VR office setup would save a lot of space. Instead of a large table and multiple monitors, we only require a chair and a small table to place a keyboard on. The employees with their HMDs can virtually work in whichever environment they prefer, while also having a virtual 360° screen space. Now instead of two monitors, everything could potentially be used as a display.

Besides using VR at the workplace, it would also be possible to take the VR setup with you, allowing you to work in your accustomed virtual work environment while sitting on a plane, train or even outside.

Another very interesting usage is for the development of VR applications. Nowadays, developers create and edit their programs outside of VR and then test the application through putting on the HMD. This quickly becomes tedious, which is why it would be nice, to be able to program while inside of the virtual environment, as well.

One big hurdle, when attempting to use virtual reality for the aforementioned cases, is the fact that text input is much more challenging while wearing an HMD. Without the ability to see hands or a keyboard, typing becomes much more challenging.

1.1 Motivation

VR input so far, especially for text input, has not been very well developed. Older attempts at bringing text input into VR often suffered because of latency problems, low computing power and less accurate sensors. Furthermore, slow input methods were used, which could never keep up with the input speed of using a physical keyboard.

Nowadays, the problems of latency are practically gone and computing power and sensors have greatly improved. This allows us to construct a setup, where presence, the feeling of being "there", is much higher than in older attempts at VR text input.

Bringing the workplace into VR could allow users to experience their regular workplace environment anywhere they go, for example while riding the train or while being outside.

When typing inside of VR, we are also able to alter the perceived reality. Humans are used to type while seeing their own realistic hands, but maybe those hands are not optimal for typing. Through exploring different hand models with different amounts of transparency, we can evaluate, with which hand model users have the highest typing performance.

1.2 Research Questions

The ability to see the text that was entered in virtual reality is one thing, but without the ability to locate hands, we believe that typing is much less efficient. Because of this, a setup was developed, which allows us to compare typing performance under different conditions. We wanted to find out which effects seeing hands while typing in VR have on typing performance. To answer this, a prototype was created which can display several different hand models in the virtual environment, with varying degrees of transparency. Besides comparing outside of virtual reality to inside, we also wanted to find out, how important hands are while typing inside of VR. Furthermore, we analyzed if a certain hand model could perform better than another inside of VR. Lastly, we were interested in how comfortable users are with typing inside of a virtual environment.

Outline

This thesis is structured in the following way:

Chapter 1 – Introduction: Here we introduce the reader into the subject.

Chapter 2 – Related Work: This chapter introduces related work and bridges a connection to our project.

Chapter 3 – Typing in VR on a Physical Keyboard: Here we explain all of the needed components to build the prototype.

Chapter 4 – Development: The development chapter gives insights into the work that went into the creation of the project.

Chapter 5 – Study: The study chapter describes the setup and procedure of the study as well as the results.

Chapter 6 – Summary and Future work: Here we summarize the thesis and introduce some ideas for future work.

2 Related Work

This chapter is used to describe earlier work that was performed in regards to typing, hands in VR and VR text input. Important findings from related projects are summarized and an overview, how they relate to this thesis, is given.

2.1 Typing

There have been many studies evaluating typing techniques and typing groups, many of which stem back to a time where all typing was performed on typing machines. Many of these older studies were held with participants which were mostly secretaries and learned ten-finger typing professionally. Nowadays, almost everyone knows how to use a keyboard, but many have never properly learned how to type. Despite that, plenty are still able to achieve a high input speed while maintaining a low error rate. This is attained through lots of practice through using Internet applications such as social media platforms or online games.

A recent paper from 2016, aptly named, how we type, analyzed and compared how people type nowadays. Back in the 70s and 80s almost everyone that was typing on a keyboard regularly, had professionally learned to type in a certain way. These people, mostly secretaries, utilized all ten fingers efficiently. This group is also known as the group of touch typists[FWO16; Gen88; Sal84].

Touch typists are able to enter text without having to look at the keyboard, by using precise defined movements. The methodology can theoretically be learned quite quickly, although it takes "hundreds of hours"[FWO16; YEYG03] to achieve the typing speeds that were reported in older literature.

In the age of modern computing, almost everyone has access to a computer and therefore, uses a keyboard, often times beginning at a very young age. This group of people, self-taught typists, often never learned how to type like touch typists. Instead they adapted a way, with which they were most comfortable with. Billions of people are not trained in touch typing today[FWO16].

When analyzing typing performance between touch typists and self-taught typists today, Feit et al. discovered, that self-taught typists are capable of achieving a comparable typing performance to touch-typists, even though self-taught typists were utilizing fewer fingers[FWO16].

With the knowledge that touch typists are not dependent on viewing the keyboard, we expect, that participants that achieve high input rates through using this technique, will be less influenced by wearing a HMD with generated VR hands than participants without this skill.

This paper also references another study where typing performance on a physical keyboard was compared to the speeds on a soft keyboard, like on a modern smart phone or tablet. They found, that their participants had an average of 60 WPM on a physical keyboard while achieving 30 WPM using a soft keyboard[FWO16; VLH12].

Originally, we also thought about only using a virtual keyboard. This would mean that participants would physically type on a flat surface, but still see their hands above a keyboard inside of the virtual environment. Besides the drawbacks of general typing speeds, like stated in the above related work, it also had the disadvantage of providing no haptic or acoustic feedback. This would have drastically decreased the perceived presence and comfort of typing, so we decided typing on a physical keyboard would be the better solution.

2.2 Hands in VR

One system of using hands in VR was developed during a project named "MuJoCo Haptix"[KT15], where, just like in our project, they utilized the OptiTrack system. They attempted to make it possible to interact with objects in VR. Unfortunately, they only used the OptiTrack system for hand rotations, their finger movement was achieved through a different product, the CyberGlove. This setup utilized real-time motion capure from the OptiTrack cameras, some physics simulation and stereoscopic visualization[KT15].

An older version of the OptiTrack cameras was used and the publishers had troubles with using motion capture due to occlusion and bad marker recognition because of other objects in the space[KT15]. The cameras that are used in our project are of a newer variant and much more accurate, although reflective objects like rings or watches had to be removed while typing.

When using hands in VR, it is also important to display hands that feel natural to the user. A project from the University of Stuttgart, "These are not my hands"[SKT+17] tackled the subject of gender perception of hands in virtual reality. They found that women perceive a higher level of presence when using non-human hands. Men perceive more presence when using human-like hands, no matter if the hands look male or female[SKT+17]. In general, they found that men and women feel discomfort when using hands of the opposite sex. To counteract this discomfort as strongly as possible, we chose to use an androgynous hand model for the realistic hand model.

2.3 Text Input in VR

Bowman et al. presents the results of a comparison of different input techniques for immersive virtual environments. In total, four different input methods were tested. Each method used a different input device[BRP02]. The first method, called the "Pinch Keyboard", was created by the publishers themselves. With this method the user was able to generate input through creating contact between the fingertips and the thumb of the respective hand. Different keys were chosen by rotating the hands and changing which finger came in contact with the thumb. The touches between the fingers were determined through the usage of a glove with sensors attached to the fingertips. The Pinch Keyboard also used a QWERTY keyboard layout, this allowed participants to find the keys they wanted, without having to learn a new keyboard layout[BRP02].

The next input method they compared, was a one-hand chord keyboard. These keyboards are commercially available. They used a very common one, known as the 'Twiddler' keyboard. This one-hand keyboard generates different inputs through pressing varying combinations, "chords", of keys on the device. These chords require some time to learn, although the developers state, that they are easy to learn. This keyboard carries the advantage, that only one hand is needed for input, leaving the other hand available for interaction inside the virtual environment[BRP02].

The third input method presented and compared was a soft keyboard in combination with a pen and tablet. The participants physically held a tablet and pen. Both objects were tracked and their movement and rotation was displayed inside of a virtual environment. The held tablet displayed a soft QWERTY keyboard and the user could select characters by moving the stylus over that character and pressing the stylus button. This approach is very similar to an older attempt from Poupyrev from 1998, the "Virtual Notepad"[PTW98]. The notepad provided a collection of interface tools that made it possible to take notes and annotate objects while still being inside of the virtual environment. The setup is very similar to that of the soft keyboard solution. The main difference being, that Poupyrev's solution directly drew the user's input into the virtual environment, without any character recognition. This made it possible to draw shapes and symbols, but the readability was always dependent on the quality of the original input.

Figure 2.1: The Virtual Notepad from Poupyrev et al.

The last used input method was speech. The spoken words of the participant were analyzed and then transformed into text. The speech was recognized as complete words, making it difficult to correct mistakes in the voice recognition[BRP02].

All of these variants were analyzed and compared to the typing performance while typing on a regular, non-VR keyboard. It was discovered, that speech is the fastest input method and that the pen and tablet approach produced the fewest mistakes. In comparison to a regular, non-VR keyboard, the methods were discovered to have a lower level of performance, usability and user satisfaction. Furthermore, they learned that users improve while using all input devices, except when using speech input[BRP02]. To counterbalance the learning effects for this thesis as greatly as possible, we decided to randomize the order of conditions by using the balanced latin square design[CG80].

In the past, VR applications had little to no user interaction, in terms of input. Most often, applications were just simple walkthroughs/flythroughs where, if anything, the head rotation was tracked. The user was just a passive observer. Bowman et al. state that this passiveness is currently changing, with more and more applications being developed, where different forms of input are necessary. They introduce the idea of using symbolic input in applications for designers, like in a 360° architectural walk-through, to leave quick annotations, or to quickly set the correct parameters in a scientific visualization. Even though speech was the fastest input method of the four compared methods, they believe, that it only allows a niche use in virtual environments and should only be used in scenarios where it can be used hands free, efficient and doesn't require any further attention.[BRP02]

2.4 Conclusion

Many studies have been held with a focus on VR input. Those generally used different input methods than using the most common input method known and used today, the keyboard. In these studies, the input speeds were compared, while the input methods were vastly different. All of those studies had proven, that these diverse input methods were slower than typing on a keyboard outside of a virtual environment. We also learned that typing on a soft virtual keyboard negatively influences the typing performance, which is why we decided to let participants type on a physical keyboard inside of the virtual environment.

There have been some attempts at combining the keyboard with virtual reality. Vertanen et al.[VME+15] conducted a study where users typed on a physical keyboard without seeing hands. Once a key was entered on the physical keyboard, the character that was pressed, lit up on a keyboard inside of the virtual 2D environment. Another attempt by Kim and Kim[KK04] used a physical keyboard with visible hands. Instead of sub-millimeter accurate finger tracking, they approximated the finger positions by only tracking the back of the two hands. This caused the typing accuracy to be greatly impacted while typing.

This thesis attempts to greatly diminish the input difference between inside and outside of virtual reality (VR), by using the same keyboard in both variants. This will allow us to compare error rates, input speeds, task load, as well as potential changes in typing techniques between the variants.

Taking into regard that participants might have troubles with certain variations of displayed hands, this thesis will also explore the effects of using different hand models on typing performance in VR. Participants will be typing while using a different hand model with a varying degree of transparency, each time. This will allow us to evaluate, how important hands are for typing in VR. Furthermore, this comparison will not only be restricted to outside of VR against inside of VR, we will also be able to evaluate which input variant is the highest performing inside of the virtual environment.

3 Typing in VR on a Physical Keyboard

Multiple systems have to be combined so that we are able to type in VR on a physical keyboard, while seeing hands with sub-millimeter accuracy. This chapter will explain all the required components at a meta-level.

3.1 Head Mounted Display

To be able to use virtual reality we require a virtual reality headset that allows us simulate a user's physical presence inside a virtual environment. This head mounted display needs to support head movement and rotational tracking, so that a user can experience all 360° of the virtual environment. The HMD should also have a high resolution, so that the user can perceive objects inside of the virtual environment as clearly as possible.

3.2 Keyboard

Text input inside of a virtual environment requires an input device. Related work showed, that a real, physical keyboard is the best input method for typing tasks. To be able to use a keyboard inside of VR, the user should be able to see and move the keyboard he is typing on inside of the virtual environment. Therefore, we need to use some form of a tracking system.

3.3 Tracking System

The next system that is needed to type in VR, is a system which displays real world movements in VR. The gaps between keys on a keyboard are quite small, therefore we require an accurate system with very low latency. Typing on a keyboard in VR with visible hands requires both tracking of the keyboard, as well as tracking of the hands and fingers. We learned that haptic plays an important role when typing on a keyboard, therefore we want to utilize a tracking system which does not obstruct the typing experience of the user, in any way.

3.4 Framework

In order to type on a keyboard inside of a virtual environment we need a software framework that can render the virtual environment while also reacting to the information from the HMD, keyboard and tracking system. This framework needs to visualize the user's hand movement as well as the keyboard movement. This combination will allow a user to type with his own hand movement above a physical keyboard inside of the virtual environment.

4 Development

The following sections will explain the process that went into developing the prototype that was used during the course of the study.

4.1 OptiTrack

OptiTrack is owned by NaturalPoint which is a company that focuses on optical tracking solutions. OptiTrack develops precision motion capturing and 3D tracking solutions and also sells their motion capturing equipment. For this thesis, we used eight OptiTrack Prime 13W cameras which record data at 240 frames per seconds (FPS) with a latency of only 4.2 ms while having a field of view (FOV) of 82 degrees[Optb]. The data from the OptiTrack cameras is captured in OptiTrack's software, called Motive. Motive can capture and stream rigid bodies, which are constellations of three or more markers. The constellations' pivot point, x,y and z positions and roll, pitch and yaw are streamed. The data is streamed through OptiTrack's own NatNet SDK, which is a client/server networking SDK that was developed for sending and receiving motion capturing data across networks[Opta].

The OptiTrack cameras are connected to a switch, which is connected to the main PC via Ethernet. OptiTrack developed their own Unity streaming client in Motive, which allows streaming of rigid bodies. Since the PC is directly connected to the cameras, the data is streamed with practically no latency.

Unfortunately, Motive does not support the live streaming of single (also knows as 'unlabeled') markers to Unity. After searching the OptiTrack forums for solutions, an attempt which was able to stream these unlabeled markers into Unity was discovered. With some modifications, a solution was acheived that created a Unity game object for every marker that was not part of a rigid body (unlabeled). These objects were named after the ID they were assigned in Motive. The problematic part was, that every time a marker disappeared for a frame it was assigned a new ID. Therefore, an automatic reassignment was necessary, which will be explained in subsection 4.2.1.

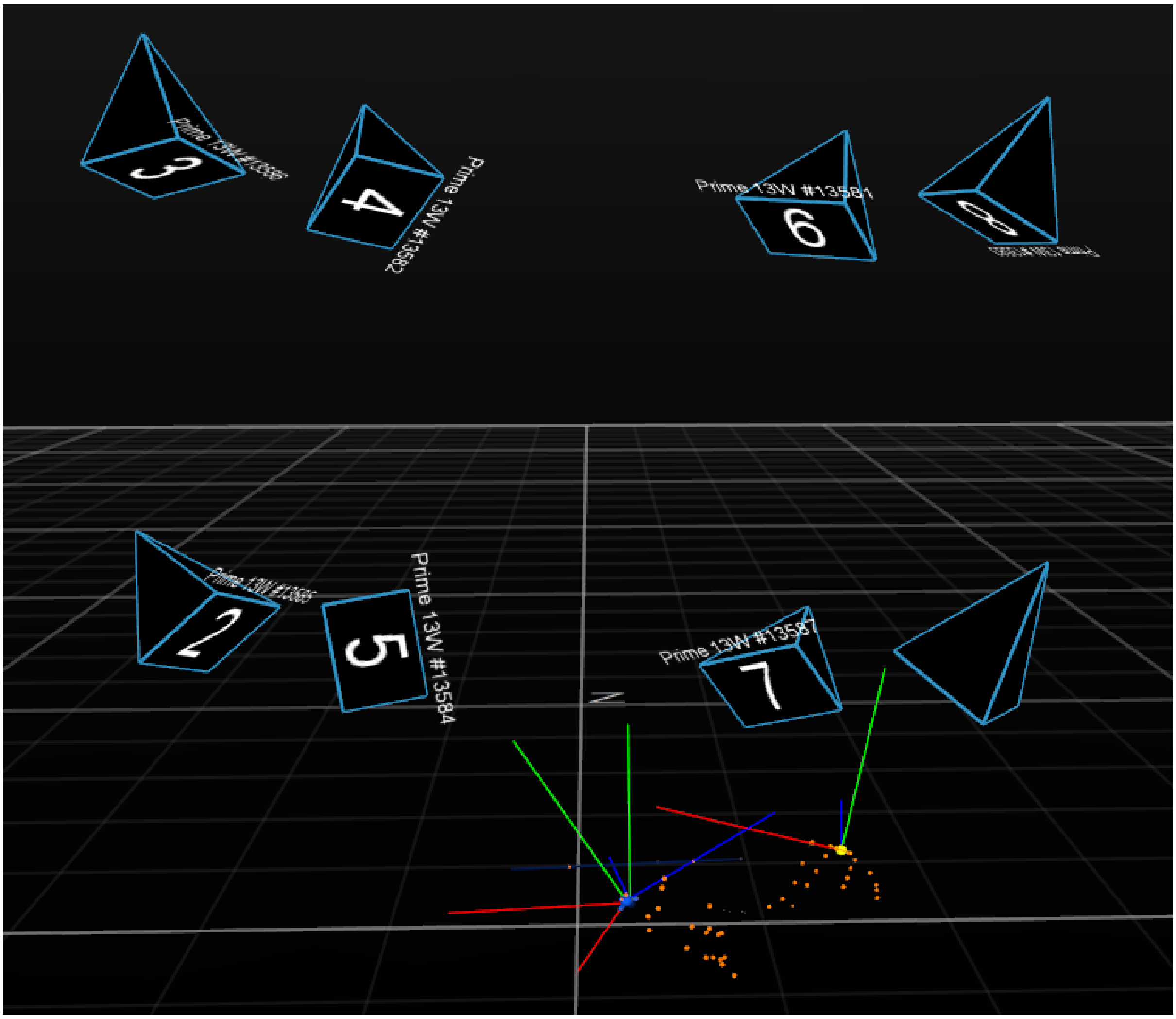

Figure 4.1: The layout of the OptiTrack cameras

For the cameras to work effectively, we needed to calibrate them. The cameras are placed around the target capture at different angles and heights. Each camera captures a 2D image, the 2D positions are calculated and the overlapping positional data to the other cameras is compared. This allows the computation of the 3D positions through triangulation as seen in figure 4.1.

The final calibration resulted in an overall average reprojection error of 0.111 pixels (px). With the worst camera having a maximum reprojection error of 0.212 px and the minimum reprojection error being 0.066 px. Reprojection errors are used to quantify the Euclidean distance between an object's real point and the measured 3D point. Furthermore, the suggested max ray length, the maximum distance between the cameras origin and the markers that are supposed to be tracked, should not exceed 2.1 meters.

The motive software rated the calibration as excellent, which allows us reliable sub-millimeter marker tracking.

4.2 Unity Development

This project was created in Unity for several reasons. The first reason being Unity's VR support for the Oculus Rift. Supporting VR in Unity is easily done through the activation of a check box. Furthermore, Motive supports direct streaming of rigid bodies into Unity. Lastly, the developer already had prior knowledge in creating VR applications for Unity, which helped during the creation of this project.

4.2.1 Hand Movement in VR

The main challenge of this thesis was transferring the real hand movements into Unity and displaying them as hands. Many different systems had to be brought together and integrated into one functional Unity project. The positional information from the OptiTrack markers had to be integrated with the Leap Motion hand models which had to be modified to work with the motion capturing system. The following sections will exemplify this process.

Assignment of Markers to Hands

All of the hand models used in this project are originally Leap Motion hand models, which were adapted to work with vector 3's on the joints and bones of the hand. These hands are 3D models, which were imported into Unity.

Leap Motion offers a small USB controller which is capable of tracking hands using two monochromatic IR cameras. We attempted to use the Leap Motion hand tracking system instead of using OptiTrack, but the Leap Motion system was not accurate enough. We decided to use the Leap Motion hand models since they offer a variety of hands ranging from realistic, to abstract hand models.

Once imported, the hands had to be modified to support a rigid body on the back of the hands, for rotation. Furthermore, a Unity object was needed per bone and joint of the fingers. Figure 4.2 shows the OptiTrack markers that were attached to the hands. The markers inside the red square were assigned to a rigid body, all the other markers were unlabeled markers. Since these hands are built to support Leap Motion, more modifications needed to be performed. Unfortunately, the hand models originated from different versions of the Leap Motion software. This prohibited us from using the exact same logic for all hands. The following sections will further explain the work that went into each hand model.

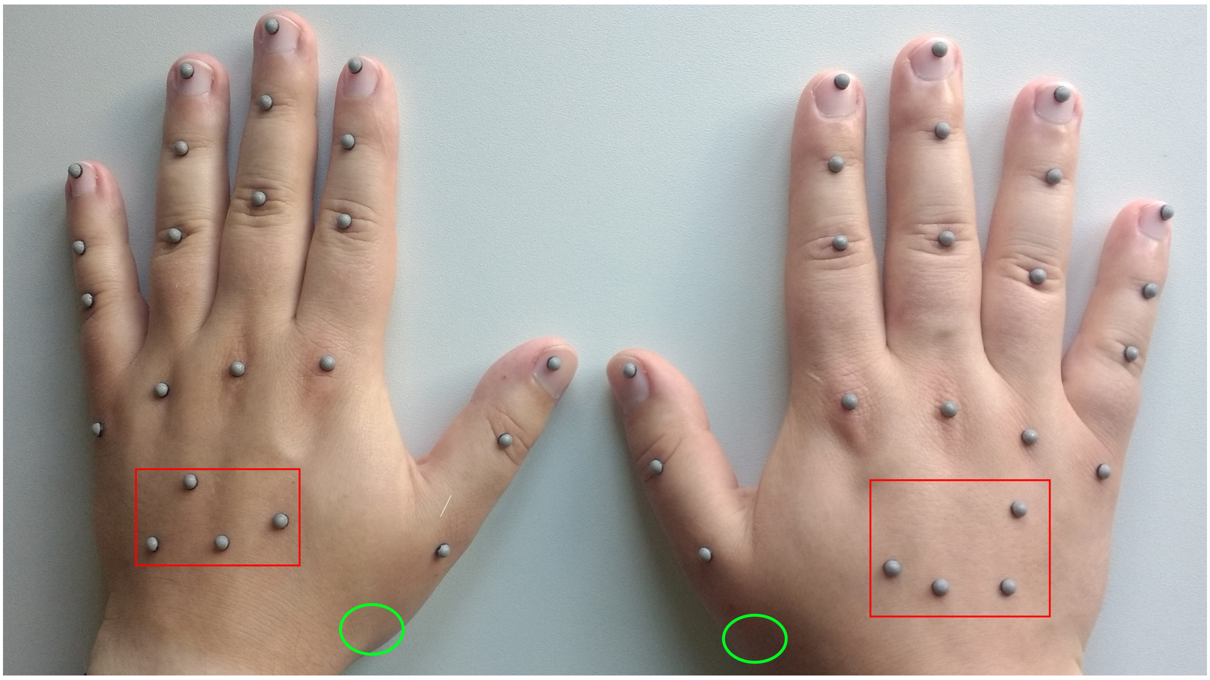

Figure 4.2: The OptiTrack body markers attached to the hands

Realistic Hands

The first hands that were integrated into Unity were the realistic looking hands. The Leap Motion hand model already had an object for every joint and for each finger's fingertip. It also already had a palm object, to which we could apply the OptiTrack rigid body. A big challenge was the fact that OptiTrack and Unity used two different coordinate systems. This caused the rotations that came from OptiTrack, to be falsely interpreted by Unity. To counteract this, we utilized Unity quaternions, which we rotated to neutralize the differences between the coordinate systems. Each finger had slightly different quaternions, but with some trial and error, it was possible to recreate the natural movements inside the virtual environment. At first, only the left hand was developed, with the thought, that mirroring the quaternions would be enough to support the right hand. This was not quite enough, but with some careful modifications to the right hand model, the logic of the left hand model only had to be slightly modified to support the right.

Figure 4.3: The realistic hands without transparency

Figure 4.4: The realistic hands with 50% transparency

The thumbs of each hand had proven to be difficult to display in a natural and realistic looking way. Unlike the other fingers, the thumbs only had one fingertip marker and two joint markers. We decided not to apply a marker to the thumb joint closest to the palm (marked green in figure 4.2), since the cameras had troubles locating it. Furthermore, it would also have interfered with the logic for the initial hand recognition scan. Therefore, that thumb joint was just always kept in its original position.

The original material and mesh of the hands were male hands with some hair on the back of the hands. We modified the model, mesh and material to a more androgynous hand since related work had proven, that androgynous hands perform better than if a gender group has to use hands from the opposite sex[SKT+17].

Minimalistic Hands

As soon as the development for the realistic hands was finished, we looked for Leap Motion models of more abstract, robot like hands. The hands we decided would work best sadly stemmed from a different Leap Motion version, than the realistic hands. Instead of having a position in the model for each joint, the minimalistic hands had positions for each bone, as well as the fingertips.

Therefore the script which was originally used for the realistic hands had to be modified. Instead of using the position of each marker, we applied linear interpolation between two neighboring markers of the same finger, to find the middle point, which was the center of the bone.

One advantage of using minimal hands was, that they needed less fine tuning to look realistic, since the hands themselves are an abstract model that only mimicked the bones inside of the hand. Like with the realistic hands, the functionality of the left hand was developed first. With the knowledge of creating the right hand with the realistic hand

Figure 4.5: The minimal hands without transparency

Figure 4.6: The minimal hands with 50% transparency

model, making the minimal right hand turned out to be quite simple. The actions that needed to be taken were similar to that of the realistic hand, except that the minimalistic hands needed less fine tuning.

One last consideration that was made, was the decision to color the hands black. This increased the contrast between the silver back plate of the keyboard and the hands themselves.

Fingertips

The development of the fingertip hand model was by far the easiest and fastest. The fingertip model is based on the same logic as the realistic hand, the position of the fingertips from the realistic hands are passed onto little spheres, which, in turn, represent the fingertips.

Since the initial scan of the hand is difficult when only little spheres as fingertips are visible, the full hand model of the human-like hand was used for the assigning phase. Afterward, the mesh render of the hand was deactivated, while the render of the fingertips was activated, leaving only five fingertips per hand.

Just like with the minimalistic hand, we also decided to use black spheres for the tips, since a high contrast made it easiest to differentiate the fingertips from the rest of the virtual environment.

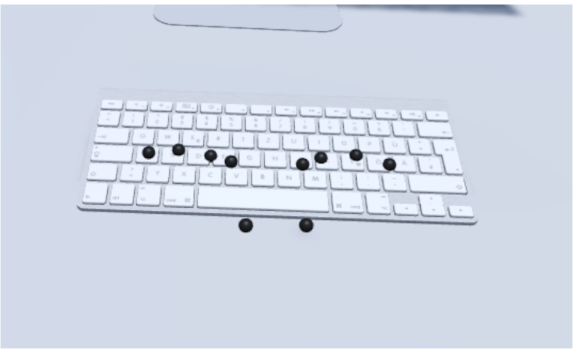

Figure 4.7: The fingertip hands without transparency

Figure 4.8: The fingertip hands with 50% transparency

Transparency

Another key aspect that we wanted to explore in this thesis, was to allow the possibility to look through your hands while typing, which is a factor that is impossible to generate in the real world.

Since Unity 5.0, adding transparency to game objects is possible through the use of the default material. This material has a special transparent rendering mode, where one can choose the transparency from 0 to 100% by changing the alpha value of the material. We decided to add 50% transparency to each hand model (real, minimal and fingertips) making it possible to see the keyboard and its keys through the hands.

The non-transparent hands use an opaque rendering mode, which is suitable for solid objects with no transparent areas. This rendering mode was then changed to a transparent rendering mode with an alpha value of 127, which is approx. 50% transparency in an alpha value range from 0-255.

Error Handling, Catching Lost Markers

There were many different approaches to increase performance of the hands. One big challenge was working with markers that were covered and thus, not visible to the OptiTrack cameras. Every unlabeled marker (all markers except the ones on the back of the hands) is internally assigned an ID. If that marker is not visible for a frame, it returns with a new ID. This made a one-time assignment impossible, therefore a runtime reassignment was necessary. The first approach consisted of two factors.

Optimizing the camera layout to maximize the coverage of the area and therefore reducing how often markers are lost. This was a bigger challenge than initially planned, since the documentation of how to perfectly calibrate the cameras for a small area was very sparse. Many attempts to set the cameras in a way that they covered the play area well, turned out to lead to very bad calibration results. A great layout was never achieved in the first iteration. Because of this, only four of the eight cameras were used, since those delivered an accurate calibration and an OK coverage.

The second part of the approach was to automatically assign the lost marker to the part of the hand that was missing a marker assignment. This worked in theory, although if two markers are lost in the same frame, the approach failed.

Since the camera setup had proven to be difficult, a lot of time was invested into the script based approach of keeping the hands intact.

To achieve an improved hand performance, the current script to catching lost markers was scrapped and a new approach had to be developed. After many different ideas and attempts, the nearest neighbor approach had proven to be the most efficient. Every position of the hand that had a marker missing was thrown into a list, with its last position of the joint before it lost its marker assignment. Every new marker compared its position with the positions of the "lost" bones and found its nearest neighbor. This approach had proven to be much more efficient and it remained the basis of the algorithm for all hand models. It still didn’t work perfectly, but with some extra comparisons at troublesome positions, the system was fine tuned to work very effectively. The problems with the motion camera persisted until the layout of the cameras was completely reworked. In addition, a new tracking wand was used, since some of the wanding markers were bent out of position. In the end, a camera calibration was finally discovered with all eight cameras and a good coverage of the action area (like seen in Figure 4.1). This combination was final and was also used during the course of the study. A reassignment was very rarely necessary during the typing test.

4.2.2 Typing Test

The development of the typing test also went through several iterations, until it reached the state that it had during the user study.

At first, we attempted to copy the exact mechanics of some popular online typing speed test, which had exercises where only the 200 most commonly used words were used, in no coherent sentence. Each given word was compared with the user input and if incorrect, the word was highlighted red. Backspace allowed correction of input but once the spacebar was pressed, it was impossible to correct the previous word.

Searching through related work on the subject of typing performance, we found that many used phrase sets, so we decided to change the test to work with phrases as well.

We used the phrase set from MacKenzie[MS03] since these sentences all had a moderate length and were easy to remember. This English phrase set contained no punctuation and very few uppercase letters.

Each run of the typing test contained ten randomly selected phrases. Using phrases allowed us to modify the error checking in a way, that the user was able to correct all the words in the phrase until the enter/return key was hit, to confirm the entry and thus displaying the next sentence. With using phrases we were then also able to log a few more interesting values, this will be explained in the following section.

4.2.3 Logging

The goals of logging data had two components. The first was to log as much data as possible and the second, to automate and make as many calculations at runtime as possible, to reduce manual calculation of data afterward. Besides logging the user input and comparing it with the given input, we also kept track of the position of each component of the hands and the camera.

Phrase and Letter Logging

When measuring typing performance, one value is most important, the words per minute (WPM). There are many different factors to take into account while measuring the WPM. One of these factors, for example, is how to deal with misspelled words. Another would be timing, how and when the timers are started and stopped.

While taking the typing test, two log files (.txt files) that analyze the typing performance were generated. One that is very basic and creates a log entry for every letter input, compares that with the character that is supposed to be entered, writes a boolean value depending on if the two match and attaches a time stamp to the entry.

The other log file creates an entry every time the enter/return key is hit. There, we analyzed the input over a whole phrase. Words that are misspelled are completely ignored and therefore do not increase the typing speed (WPM). Furthermore, we track the WPM with two different timers. One timer starts at the first key press and runs continuously. The other timer also starts at the first key press, but once enter/return is hit, the timer stops, until the first key of the next sentence is pressed. Both these measurements are calculated per phrase set and written into the logging file after the set is completed. Another entry per phrase is the total amount of keystrokes needed in comparison to the actual length of the phrase. This way, we can keep track of how often the user corrected his input. For example, one misspelled letter results in three extra inputs, one for the wrong character, one backspace and then the correct letter. Lastly, we also kept track of the time needed to enter each phrase, from the first key press until the enter/return key was hit.

Vector Logging

With vector logging, we understand the tracking of all important vector positions and rotations as Euler angles. Those include the position and rotation of the camera, which represents the position and rotation of the HMD. Furthermore, the position and rotation of every component (fingertips and joints) of the hand were logged. This information was added every Unity frame. All this data could be inserted into a system, which could recreate the movement of the hands as well as show how the HMD was positioned and rotated.

The vector logging file also adds the currently pressed key, if one is pressed in that frame, into the entry. This helps with recognizing when a key was actually struck.

4.2.4 Recreating the Lab in VR

To maximize presence felt while typing inside of VR, we attempted to recreate the room in which the study was held as closely as possible.

The most important aspect was the keyboard, since the positions of the hands over the real keyboard had to exactly match the movement that was visible in VR. To achieve this, we used an exact 3D model of the Apple keyboard, like shown in figure 4.9, on which participants also typed physically. Here we tried to match the movement of the hands inside and outside of VR.

Next, we added a chair and a table into the environment. We attempted to match the ratios between keyboard size and table in VR to that of the reality. At first we used a park bench and a wooden table, later those were switched to the same chair and a similar table than those inside the lab.

The walls, floor, and ceiling of the lab were photographed and put onto panes inside of Unity to build the walls inside of the virtual environment.

Adding virtual reality support into a Unity project was easy, only a check box needed to be activated. The camera was then positioned in front of the table, over the chair at a head-height. The movement of the head was tracked through an Oculus Sensor, which is placed in front of the participant. Rotations of the head are tracked in the HMD directly.

Figure 4.9: The keyboard model which was used for performing the typing task

We decided to display the phrases, countdown and user input on an iMac since we had those available for the outside of VR version as well. We added the 3D model of the iMac into the virtual environment and displayed all the needed information on it.

To aid the participants with the assigning phase, we also added two panes onto the table that show how the hands should be positioned. After the assigning phase was completed, the panes also disappeared.

4.2.5 Standalone for Outside of VR Typing

The development of the standalone version, which was used for typing outside of VR was very straightforward. All of the scripts, models, and materials that were used for moving the hands in VR were removed. Furthermore, all of the room objects were also deleted, since they would not be needed.

The goal was to display the screen area of the VR iMac onto a real iMac, on which the real world typing was performed. To achieve this, the three text-mesh renders for the countdown, phrases, and input were copied onto a white area, which resembled the VR iMac screen exactly. This area covers the whole screen and when the scene is viewed in full screen, the real world and VR iMac screens display exactly the same text in the same aspect ratios.

The typing test itself uses exactly the same logic and phrase sets outside of VR as inside. Vector logging is disabled for typing outside of VR but the character and phrase logging work exactly the same.

All of these aspects allow a fair comparison of typing performance outside of VR compared to inside virtual reality.

5 Study

The goals of the study were to figure out how well people accept virtual hands for typing, how the input speeds compare outside of virtual reality to inside as well as figuring out with which VR hand model the participants can perform the best. To achieve this, we logged all of the typing data to calculate results and participants filled out questionnaires to measure the task load and VR experience.

5.1 Setup

During the study, the participants were asked to type ten English phrases a total of 24 times. We had seven different virtual hand models, the eighth condition was typing outside of the virtual environment. Each condition had three repetitions, the first one being a practice run.

The phrases that had to be typed were randomly selected from a phrase set with 500 sentences that were easy to remember, of medium length, had no punctuation and little capitalization[MS03]. One phrase set of ten sentences finished either after all phrases were completed or after four minutes expired.

The virtual environment was modeled after the real room in which the study was performed. The walls, ground, and ceiling of the environment looked similar, as well as the keyboard and monitor.

To keep the haptics of typing inside of VR to outside as similar as possible, participants performed the typing test on the same keyboard model. Furthermore, the logic while typing outside of VR was exactly the same as inside of the environment.

5.2 Apparatus

For the head mounted display we used the Oculus Rift consumer version. We chose the Rift over the HTC Vive since the Vive depends on two IR sensors, called base stations, which are responsible for head tracking. These base stations had troubles with head-tracking because the IR signals of the OptiTrack cameras interfered with them. The Oculus Rift has a sensor called the Oculus Sensor, which looks similar to a microphone on a stand. This sensor was placed at the bottom of the capture area and none of the systems interfered with each other. The sensor is highlighted red in figure 5.1.

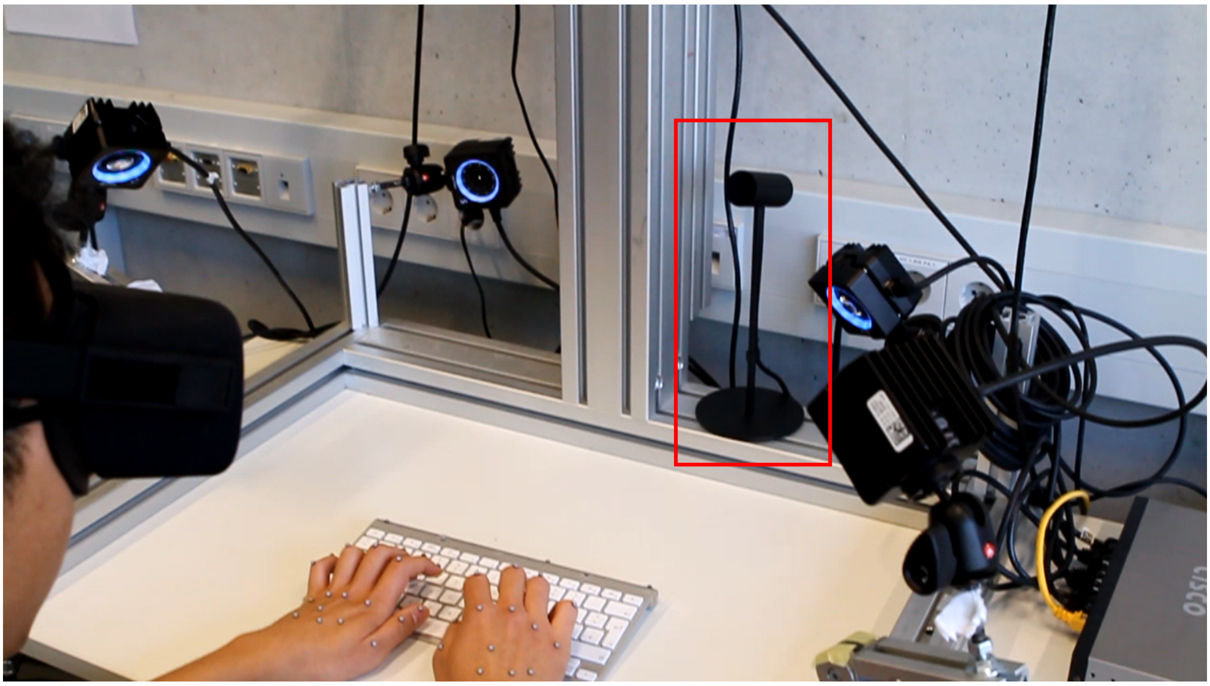

Figure 5.1: A user typing, the Oculus Sensor is highlighted in red

As a motion capturing system the OptiTrack motion capturing system was used. This system is capable of recording and streaming motion data at 240 FPS with a very low latency. The cameras were already available which was a big factor in why they were chosen. Why we chose to use the OptiTrack system over an already available VR hand system was explained in section 4.2.1.

Why Unity was chosen as the VR platform has already been explained in section 4.2.

5.3 Study Design

The purpose of the user study was to determine differences in typing performance in VR with different hand models in comparison to the outside of VR. Six key factors were analyzed. Firstly, we wanted to find if there are differences in input speeds between typing in VR in comparison to typing outside of VR. Next, we were interesed

Variables	
Independent Variables	**Dependent Variables**
Hand Model	Input Speed
Transparency	Error Rate
Typing group	Words/minute

Table 5.1: The independent and dependent variables

in differences in error rates (typing mistakes) between typing in VR and outside of VR. Furthermore, we looked for differences in needed effort and perceived presence between typing in VR and outside of VR. Another key factor that was analyzed was, if there are differences in input performance while using different hand models in VR. The fifth key factor was analyzing the differences in performance while changing the transparency of the hand models while typing in VR. Lastly, we wanted to find the amount of needed time to press the first character, from a neutral hand, starting position.

In total 32 participants were invited through a mailing list to perform the tests. The invited participants were divided into two groups, depending on their input speed in WPM while typing on the outside of VR condition. The slow typers were all typers who performed lower than 43.3 WPM. The fast typers were all typers who were faster than 43.3 WPM.

In total participants completed eight different conditions:

1. Realistic hands non-transparent (4.3)
2. Realistic hands 50% transparency (4.4)
3. Abstract hands non-transparent (4.5)
4. Abstract hands 50% transparency (4.6)
5. Only fingertips non-transparent (4.7)
6. Only fingertips 50% transparency (4.8)
7. No hands visible (VR)
8. Real hands (outside of VR)

The study was held under a mixed factorial design, with the within-subject variable being the different hand models and the between subject variable being the typing group. Table 5.1 contains all independent and dependent variables.

Since we have eight different conditions, we needed to invite a factor of eight times the amount of participants to counterbalance any learning effects as strongly as possible. Through the usage of the balanced latin square algorithm, the conditions were sorted in a way, that they were both row and column complete[CG80]. With 32 participants, we had a sample size of 16 participants per group.

5.4 Participant Profile

In total, 35 people were invited to participate in the study. One participant could not participate since his glasses would not fit under the HMD and his regular eyesight would not have been sufficient to generate meaningful data. Two other participants were excluded due to problems in time slot planning.

The mean age of the 32 participants was 21.91 years, with a standard deviation of 2.291. The youngest participants were 19, the eldest 27 years of age. Of the 32 participants, 30 were male and two female.

All of the participants were currently studying some form of computer science. 14 people have had at least some experience with virtual reality and had worn either a variant of the Google Cardboard or another HMD. 17 of the 32 participants wore glasses underneath the head-mounted display.

5.5 Procedure

This section will exemplify the standard procedure during the study with one example participant.

The first action that was taken for each new participant was to set the user ID on both PC's to the ID that was assigned to the participant. This ID was entered into a text file and then read at runtime. The next step was to prepare the setup for the participant. Once finished, the participant was invited in and given a consent form.

After the consent form has been read and filled out, the age and VR experience of the participant were recorded. Next, we thoroughly explained the task and general procedure of the study to the participant. Once the participant had understood the general gist of the task, the HMD was calibrated. This allowed the participant to see objects as clearly as possible while inside of the virtual environment.

Once the calibration was completed, the HMD was taken off and the OptiTrack reflective body markers were attached to the participant's hands, as seen in figure 4.2.

Since attaching the markers took a few minutes, the participant was encouraged to familiarize himself with the questionnaires. This allowed a better comprehension and a faster completion of the questionnaire during the course of the study. Any uncertainties in the questionnaires were explained in this phase as well.

After all of the markers were attached, we configured the forms on the back of the hands for the OptiTrack software and assigned the correct ID for the right and left hand.

These rigid bodies allowed the measurement of rotational and positional data which was transferred to the virtual hands.

All of the above steps generally took about 15 minutes to complete, once all of these were finished we began with the actual task. For this example procedure, the participant began with condition three, which is the minimal hand model without any transparency. The Unity scene was started. At this point, the participant was encouraged to look around and familiarize him/herself with the virtual environment.

Afterward, the hands were activated and the participant moved his hands into the assigning guide-panes. After the assignment was completed the user could move his hands and the keyboard to his desired position. As soon as he was ready, the countdown was started. Once the countdown displays go, the participant was allowed to move his hands and start typing. After he has finished typing all ten phrases, or once four minutes expired, the input was blocked and we reset the typing test. The test was repeated two more times, for a total of three times per hand model.

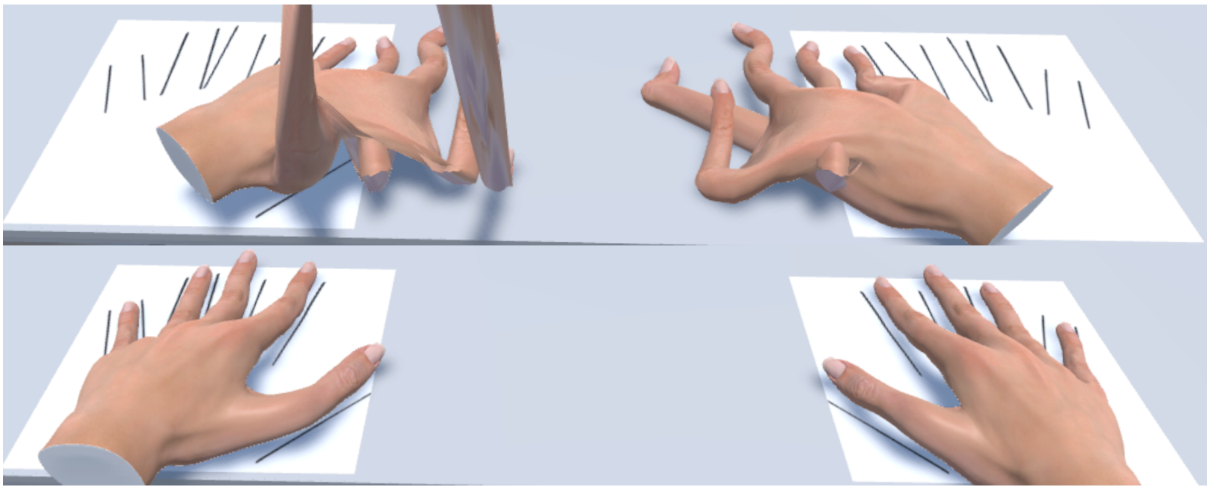

Figure 5.2: Screen-shots from the assignment phase if the hands are not aligned with the lines (top) and a correct alignment at the bottom

After the participant was finished with typing the ten phrases three times, he was asked to remove the HMD and fill out the first questionnaires. After the NASA TLX and the presence questionnaire (PQ) were filled out, we continue with the next typing condition. The following conditions inside of VR in this example were four, five, six and then seven. After typing was completed in these conditions and all of the questionnaires were filled out, we switch over to condition eight, which was typing outside of VR. The participant moved one seat over and completed three runs of the typing test outside of VR and then also filled out the questionnaires. The user then switched back to the VR setup to finish conditions one and two and then filled out the questionnaires accordingly.

To summarize, the participant took the ten phrase typing test three times, then filled out the questionnaires and moved on to the next condition.

Depending on the user's speed in typing and filling out questionnaires, this whole process took anywhere from 70 to 110 minutes. After all of the typing conditions were completed, a short semi-structured interview was held with the participant.

Lastly, the markers were removed from the hands and the participant received his compensation for participating.

5.6 Data Logging and Questionnaires

All of the input while typing inside of VR and outside was recorded into log files. While typing inside of VR, all of the movement and rotational data from the hands and HMD was logged. For a further explanation of exactly what input data is logged and when, refer to section 4.2.3. Besides input data, the participants also produced data through filling out the questionnaires. By using the NASA TLX questionnaire we can build an average number which represents the general task load of using each condition.

The Presence Questionnaire analyzes the virtual environment on four different factors. Each factor has multiple questions. The four factors are involvement, sensory fidelity, adaptation/immersion and interface quality. The sum of all responses represents the perceived presence[WS98].

After completing all of the tasks the participants were asked a few final questions about the project:

- Image it is now possible to easily take the hand tracking setup with you, when would you use it?
- If you use it, for which purposes would you use it?
- If you use it, in which virtual environment would you like to be?
- Which hand model did you think was generally the best, and why?
- With which hand model did you feel you had the fastest typing performance?

5.7 Measured Data

Each typing event was measured with a time stamp, like stated in subsection 4.2.3. This data was bound to a user ID and each user was divided into one of two typing groups. This division lead to a final sample size of 16 participants per group.

The words per minute (WPM) were calculated by building the sum of all correct words which was then divided by the required time in seconds. This was multiplied by 60 to generate the WPM. If a character in a word was misspelled the whole word was ignored.

The words per minute while typing are not only dependent on how fast keys are hit but also on how accurately the participants entered correct characters. Therefore, we measured the accuracy of each typing test. The accuracy is calculated by dividing the number of characters that are needed to be pressed by the amount which is actually entered.

Another factor that was measured was the amount of time needed to move your hands from a neutral position (hands next to or in front of the keyboard) to pressing the first key. We found that people use different approaches before typing the first letter. Some position their index fingers on the 'F' and 'J' key first and then begin typing, while others just directly aim for the first key and then continue typing with whichever finger feels most comfortable to them.

The presence questionnaire (PQ) was performed to measure the general presence during the typing conditions. This questionnaire is made for measuring the presence whilst inside of a virtual environment. For consistency purposes we also let participants fill out the PQ after conducting the typing test outside of VR.

Furthermore, participants also filled out a NASA Task Load Index (TLX). This questionnaire is used to assess the perceived workload while completing a task.

5.8 Results

5.8.1 Comparison of Means

The following sections will be used to present and analyze the average typing performances of all participants inside and outside of their groups. Since each condition was performed three times with the first test being a practice run, the average was derived from the final two runs. When comparing means while using statistical tests to check for significance we based our tests on a 95% confidence interval.

Words per Minute

The words per minute (WPM) were measured after each typing test with ten phrases. The means were built from the averages of each run for every condition. Figure 5.3 shows the WPM rates for every typing condition for group one and two in the form of a bar chart with 95% confidence interval error bars.

The average over all participants while typing outside of VR was measured at 47.06 WPM (± 13.00 WPM). The group of slower typers achieved a WPM rate of 37.86 (± 4.07 WPM) while the fast typers performed at 56.26 WPM (± 12.35 WPM).

A one-way ANOVA over all 32 participants, comparing all different hand models showed no statistically significant differences between the conditions (F(7,248)=0.990, p=0.439). Only the group of slow typers showed differences between the hand conditions (F(7,120)=2.523, p=0.019) while the fast typers did not (F(7,120)=0.449, p=0.869).

By far the slowest condition was typing inside of VR without being able to see the hands. The average typing rate of all participants was 37.51 WPM (± 18.87 WPM). The slow typers were much slower with 25.34 WPM (± 12.86 WPM) in comparison to the fast typers with 49.67 WPM (± 15.97 WPM).

The fastest VR typing condition turned out to be the minimalistic hands with 50% transparency with a typing speed of 45.92 WPM. While using this hand model, the group of slow typers showed a statistically significant difference in a one-way ANOVA (F(7,120)=2.523, p=0.019). The Tukey HSD post hoc test comparing the minimalistic hand with 50% transparency to the condition with no visible hands showed a statistically significant difference of 10.192 WPM (p=0.048).

This study found that the mean typing speed with 50% transparency (44.94 WPM ± 16.81) was 0.96 WPM faster than typing with no transparency (43.98 WPM ± 17.57). Figure 5.3 shows the WPM rates of the slow and fast typers for all transparencies.

The participants' typing speed while seeing hands (0%, 50% transparency and outside of VR typing) was 7.33 WPM higher than without the ability to see hands (100% transparency). A paired samples t-test showed that with hands visible (44.84 ± 16.06) in comparison to hands invisible (37.51 ± 18.87) there is a significant difference (t(31)=3.909, p<0.001).

Accuracy

Figure 5.4 displays the mean input accuracy in the form of a bar chart for every condition and both typing groups with 95% confidence interval error bars.

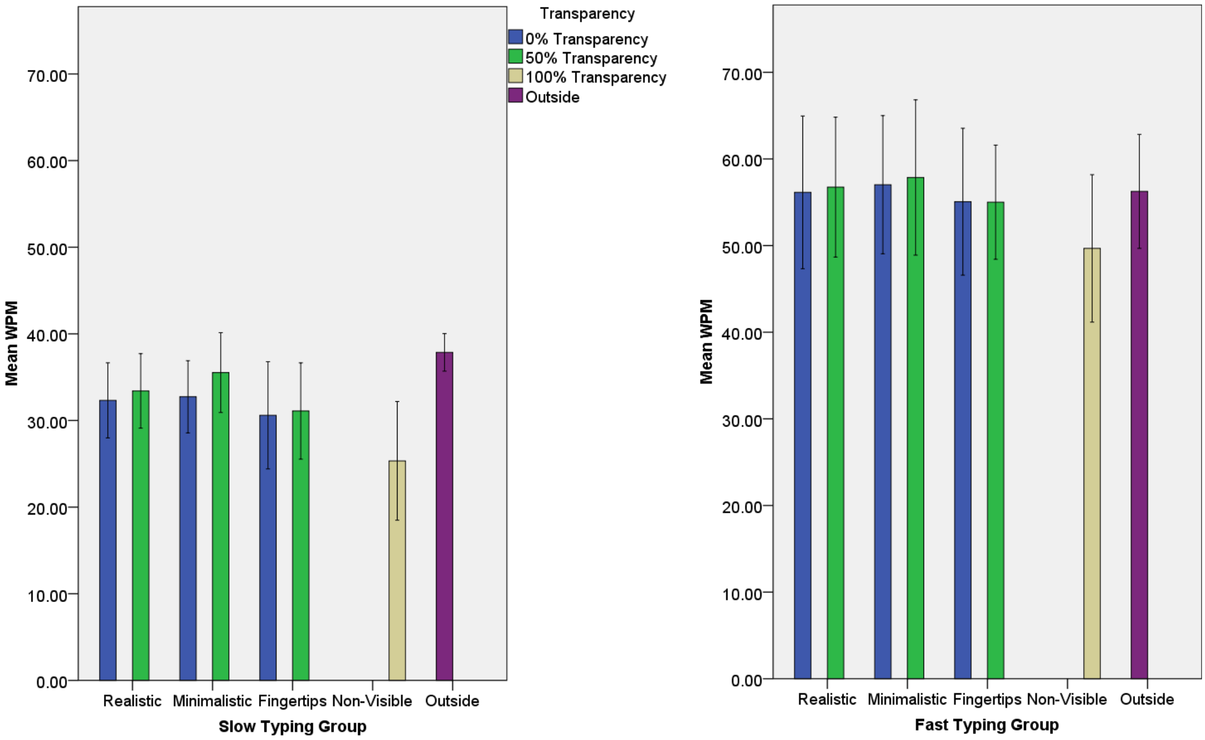

Figure 5.3: The words per minutes for both the slow and fast typers under each typing condition

The slower typing group, achieved an average accuracy over all conditions of 84.36% (± 9.34). The fast typing group was more accurate, having an average accuracy of 89.10% (± 6.24).

The highest average accuracy for the slow typers was achieved whilst typing outside of VR. A between samples t-test was conducted to compare the accuracy of the slow typers while typing inside of VR to outside. It showed a significant difference in accuracy while typing outside of VR (M=89.51% ± 6.44) in comparison to typing whilst wearing an HMD (M=83.62 ± 6.55), t(15)=3.619, p = 0.003. The same comparison for the fast typers resulted in a lower, non-significant mean difference of only 0.4% (p=0.660).

Another between samples t-test comparing the typing accuracy of the slow typers when hands are visible (M=85.61 ± 6.71) to non-visible (M=75.57 ± 13.20) showed that having visible hands while typing significantly influences the input accuracy (t(15)=2.761, p=0.015). The fast typers were also significantly slower with non-visible hands (M=85.00 ± 10.61) in comparison to visible hands (M=89.68 ± 4.35) with t(15)=2.210 and p=0.043.

Applying 50% transparency to the hand models did not significantly affect the input accuracy whilst typing.

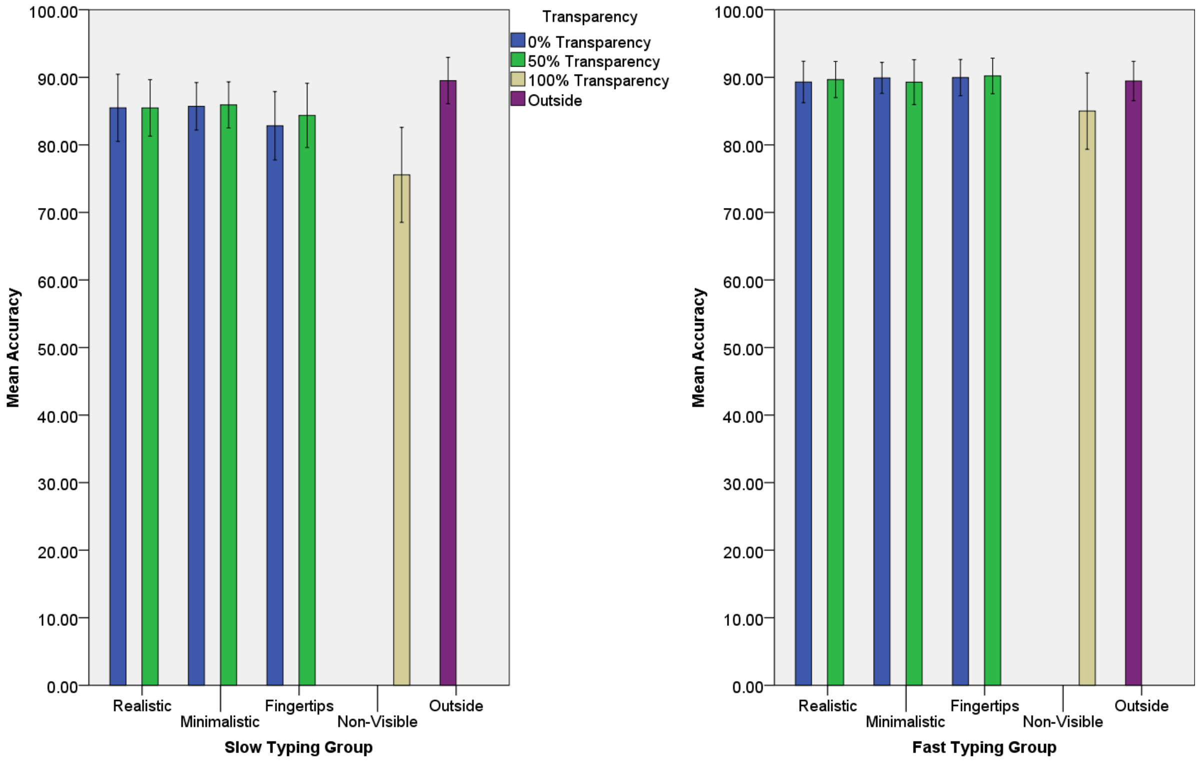

Figure 5.4: The mean accuracy for both the slow and fast typing groups under each typing condition

When performing a one-way ANOVA we found that there is a statistically significant difference between groups for the slow typers (F(7, 120)=3.339, p=0.003) while the same test for the fast typers did not show a statistically significant difference (F(7, 120)=1.180, p=0.319). A Tukey HSD post hoc test was performed on the data from the slow typers, which shows that the accuracy without hands visible was statistically significantly lower in comparison to using the realistic or minimalistic hands or while typing outside of VR (p<=0.038). The accuracy when using the fingertip hand model was a bit worse and therefore there was no statistically significant effect between those hands and the non-visible hands (p>=0.098).

Time to start

Figure 5.5 shows the average amount of time needed for the first key press for each condition. A statistically significant difference between condition was found through a one-way ANOVA (F(7, 248)=5.036), p<0.001. A Tukey HSD post hoc test shows that the condition with invisible hands (in figure 5.5 "Non-Visible") is statistically significantly

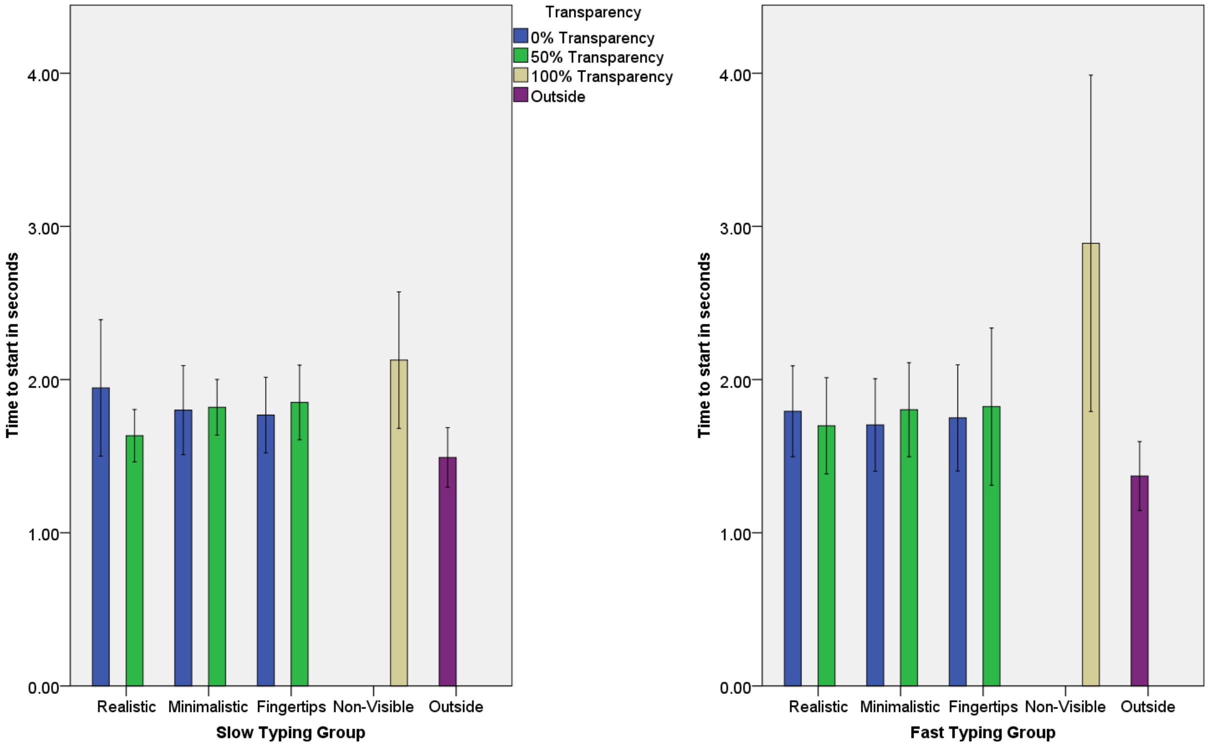

Figure 5.5: The average time needed until the first key press for every hand model with 95% CI error bars

slower than all other conditions (p<=0.024). There was no statistically significant difference between the VR typing conditions with hands visible (p>=0.966).

A very surprising result was, that participants of group one (1.805s ± 0.57s) needed almost exactly the same amount of time to press the first key than group two (1.854s ± 1.00s), although that difference is statistically non-significant. It is difficult to explain this difference, one limitation is, that the key press was not checked for correctness, while the other could be dependent on a users typing style. This will be further discussed in section 5.9.

5.8.2 NASA TLX

Figure 5.6 displays the means of the NASA Raw Task Load Index (TLX)[Har06] for every condition. A one-way between subjects ANOVA was performed to compare the means of the TLX scores. We found a statistically significant difference between groups with F(7, 248)=4.618 and p<0.001. A Tukey HSD post hoc test revealed that the TLX was statistically significantly higher with no hands visible in comparison to realistic hands (0% and 50% transparency), minimal hands (0% and 50% transparency) and whilst

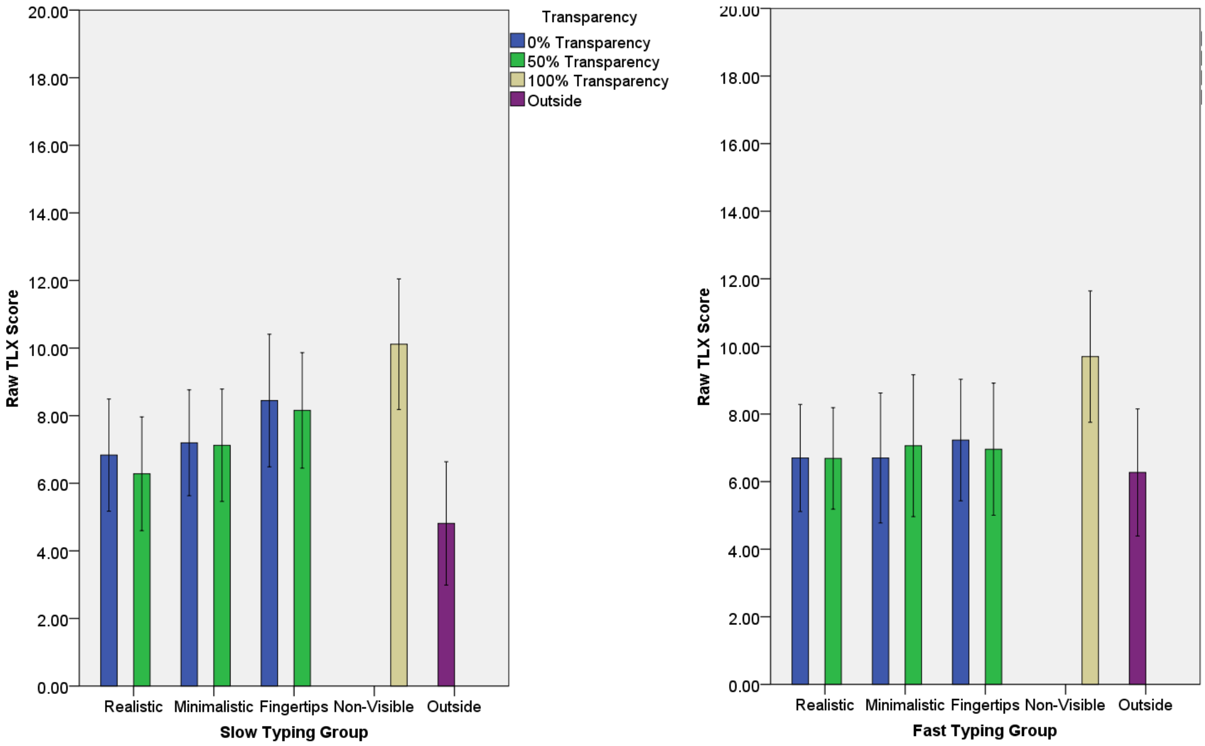

Figure 5.6: The mean results of the NASA Raw TLX for both typing groups

typing outside of VR. Interestingly, there was no statistically significant difference in TLX between non-visible hands and the fingertips hand model (0% and 50% transparency).

A between samples t-test was conducted comparing the raw TLX score of the slow typers between typing inside of VR(M=7.74 ± 2.53) and outside (M=4.81 ± 3.42). The test resulted in a statistically significant mean difference of 2.92 with p=0.001. When performing the same test on the data of the fast typers, we found a statistically non-significant mean difference of 1.02 with p=0.120.

Another between samples t-test was performed to compare the raw TLX score of the slow typers with hands visible (M=6.98 ± 2.53) to typing without having visible hands (M=10.11 ± 3.63). We found a statistically significant mean difference of 3.14 with p=0.001. The group of fast typers also showed a statistically significant difference between hands visible (M=6.80 ± 3.14) and invisible hands (M=9.70 ± 3.65) with p=0.001.

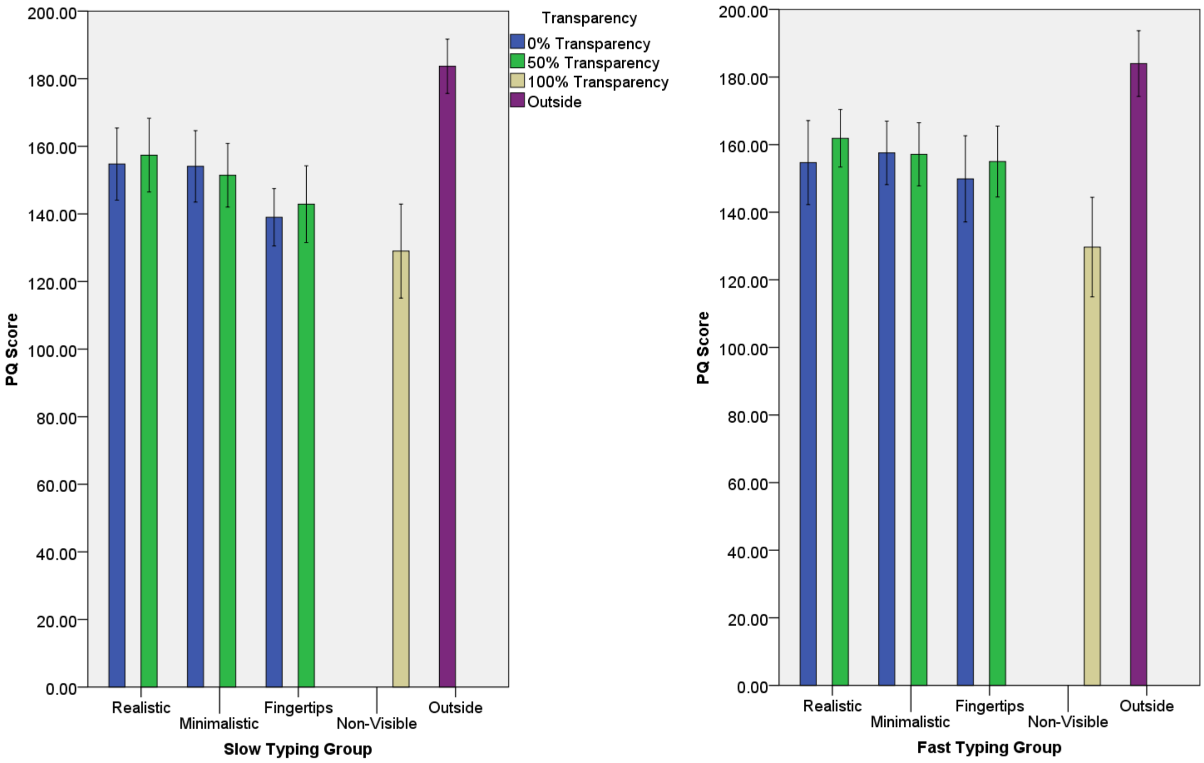

Figure 5.7: The mean results of the Presence Questionnaire for both typing groups

5.8.3 Presence Questionnaire

Figure 5.7 presents the average presence questionnaire[WS98] scores for every typing condition. A higher PQ score represents a higher perceived presence, the maximum score is 189. Another one-way between subjects ANOVA was performed to analyze the differences between the typing conditions. A statistically significant difference with F(7, 248)=11.97 and $p<0.001$ allowed us to perform a Tukey HSD post hoc test. It was found that typing outside of VR offered statistically significantly more presence than when typing with all other conditions ($p<0.001$).

Furthermore, we found the exact same results when comparing typing with no hands visible to the other conditions as we found with the NASA TLX questionnaire. Participants perceived statistically significantly less presence with no hands visible than with realistic hands (0% and 50% transparency), minimal hands (0% and 50% transparency) and whilst typing outside of VR. Again, no statistically significant difference in presence was found between non-visible hands and the fingertips hand model (0% and 50% transparency).

A between samples t-test was conducted to evaluate the difference in perceived presence between typing inside of VR and outside. The perceived presence when typing outside of VR was significantly higher for both groups. With the slow typers having a mean difference of 36.76 (± 15.70) and $p<0.001$. The fast typers reported a mean difference of 31.74 (± 12.78) with $p<0.001$.

Another between samples t-test was performed comparing the differences between typing while seeing hands and without visible hands. Just like before, both groups showed a statistically significant decrease in PQ score when hands were not visible. The slow typers showed a mean difference of 25.74 (± 22.40) with $p<0.001$. The fast typers reported a mean difference of 30.33 (± 25.74) with $p<0.001$.

5.9 Discussion

When designing the study we wanted to figure out, if there are differences in typing performance in VR with different hand models, in comparison to typing outside of VR. In the following section, we will go through and discuss the results for our list of key factors that we specified in our study design, section 5.3.

The first key factor of our study design was whether there is a difference in input speeds between typing in VR and outside of VR. We found, that on average there is no significant difference in WPM between typing inside of VR to outside. When focusing on the data from the slow typers, we did find a statistically significant difference. These results allow us to come to the conclusion that people with below average input speeds are negatively affected by typing inside of VR while the average and above average typists are not significantly impeded.

The next key factor we wanted to discern was the difference in typing mistakes between typing with an HMD against typing regularly. As words per minute are directly connected to the accuracy while typing, our results for accuracy mirror those from the above WPM speeds. Again, we found that on average there is no statistically significant difference in accuracy between typing inside of VR to outside. Just like with our first factor, the slow typers did have a significant difference in accuracy. We can again come to the conclusion, that below average typists are negatively influenced by wearing an HMD, while the average and above average typists were not significantly negatively affected.

To answer the third key factor, the differences in effort and presence between typing in VR and outside of VR, we let participants fill out both a presence questionnaire and a task load questionnaire to determine the perceived presence and the average necessary task load during the conditions (NASA TLX). We found that the task load for slow typers is statistically significantly higher while typing inside of VR. Furthermore, the results show that the perceived presence is statistically significantly higher for both groups whilst typing outside of VR in comparison to inside, although that was to be expected since the perceived presence, in reality, should always be as large as possible.

The fourth factor was, if there are differences in performance when using different hand models inside of VR. The only condition that was statistically significantly worse in accuracy, WPM and time needed until the first key press was the VR hand model where no hands were visible. Through this data, we can come to the conclusion, that having some sort of hand model definitely is important for typing. This also proves, that we can improve our typers through displaying virtual hands. When comparing the accuracy, WPM and time needed until the first key press between the other VR typing conditions (realistic hands, minimalistic hands, only fingertips) we were not able to find any statistically significant differences. When analyzing the questionnaires we found that there are statistically significant difference both for the presence and task load between seeing no hands and seeing the realistic, minimalistic and real world hands. The interesting part is, that this difference was not found when comparing no hands to the fingertip model. Through these results, we can determine that the realistic and minimalistic hands were easier to use and increased perceived presence in comparison to seeing no hands, while we cannot make the same statement about the hand model with only fingertips visible.

The next factor was finding if transparent hands cause any difference in performance while typing inside of VR. Here the results are very straightforward. A transparency of 50% did not influence the typing speed, accuracy, time needed for the first key press, perceived presence or task load, in comparison to a hand with no transparency. Through this, we can state that the ability to see the keyboard through the hands did not positively affect the typing experience. One limitation that has to be acknowledged, was the display resolution of the HMD. It was difficult to read the characters displayed on the keyboard and some participants specifically stated, that this factor influenced the gain of having transparent hands.

The last key factor that we wanted to answer was if any of the typing conditions influence the required time needed to press the first character when moving the hands from a neutral starting position (either hand next to or in front of the keyboard) to the first character. Here we found that it is important to see any sort of hand for quickly locating the hands. Furthermore, finding the first key outside of VR is also faster than when wearing an HMD. Between the two typing groups we did not find any statistically significant differences in time needed for the first key press. This is likely dependent on two factors. The first factor is that the first key press was not proven for correctness, so the slower typing group likely had more mistakes during their first key presses. This was an oversight during development and could not in hindsight be determined through the remaining log data. The second factor is likely dependent on the procedure that fast typers follow, before beginning to type. Like touch typists, our fast typers followed a similar approach and placed their fingers on certain anchor points (e.g. 'F' and 'J' key) above the keyboard before beginning to type, while slow typers often directly aimed for the first key with no prior finger placement. This process could explain why the faster typers were not quicker in pressing the first key than the slow typers.

6 Summary and Future work

With virtual reality growing in popularity and an increasing amount of developers creating content for the technology, we wanted to see, if VR could be used for productive work.

Many studies have been performed to test different text input methods for VR, but those were always much slower than typing on a keyboard outside of VR. Since the keyboard is the best and most commonly used input device for the computer, we wanted to make it possible, to use a physical keyboard inside of VR as well.

Typing while wearing an HMD is similar to typing while blindfolded. Nowadays, almost everybody's typing speed decreases when hands are not visible. Which is why we decided to create a prototype, which would allow us to compare the typing performance outside of VR, to the inside of VR, while using different hand models. This was made possible using the Oculus Rift HMD, the OptiTrack motion capturing system, and the Unity game engine. In total, we created six different VR models. These consisted of a realistic hand model, a minimalistic hand model, and a fingertip hand model. Each hand model had one version with no transparency and another with 50% transparency, which would allow the user to see the keyboard through his hands. A seventh variant was a hand with 100% transparency, which lead to no hands being displayed at all. To compare the typing performance of typing inside of VR to typing outside of VR we also recreated a version for typing outside of VR.

Once development was completed, a user study with 32 participants was conducted. These participants were divided into two groups of equal size, a group of faster typers and a group of slower typers. Each participant performed the typing tests under each of the eight conditions. Any learning effects were counterbalanced through the usage of the balanced latin square design. The typing performances were logged and compared to the other conditions. Furthermore, participants filled out a presence and a task load questionnaire, after completing each condition.

On average, there was no statistically significant difference in WPM or accuracy between typing outside of VR, to inside. When looking at the data of the slow typers, a statistically significant difference was found. Slow typers were significantly slower and less accurate while typing inside of VR.

We also found, that seeing hands is very important for typing. When participants were unable to see any hands while typing, their WPM and accuracy were statistically significantly lower in comparison to all the other typing conditions. The time needed to press the first key from a neutral hand position was also statistically significantly higher without hands visible, than with all other conditions.

Using different hand models inside of VR had no statistically significant effects on WPM, accuracy or on the time needed to press the first key from a neutral hand position.

The task load while typing inside of VR was higher than outside of VR, while the perceived presence was lower in VR than outside of VR. Furthermore, the task load with no hands visible was statistically significantly higher and the perceived presence was statistically significantly lower than with the realistic and minimalistic hand models as well as to typing outside of VR. We can conclude, that typing with the realistic and minimalistic hands inside of VR produces less task load and more presence than typing without seeing any hands. The same was not the case when comparing no visible hands to the fingertip hand model. We cannot state that the fingertip model produced less task load and more presence while typing than the model with no hands visible.

We found that transparency has no statistically significant effect on typing, task load or presence. There were indications of transparency positively influencing the slow typing group, but the sample size was not large enough for those indications to be statistically significant. Furthermore, participants mentioned that the inadequate display quality of the HMD made it difficult to recognize letters on the keyboard. This could have also countered the benefit of seeing the keyboard through the hands.

Future Work

The biggest problem with this project is the prototype's size. Currently, the necessary setup is a great deal too large, heavy and expensive. Furthermore, VR programs require a lot of computing power and electricity. All of these factors currently restrict us from actually being able to type anywhere in VR.

Because of this, one aspect of future work would be to create compact hardware that allows sub-millimeter accurate finger-tracking with very low latency. Maybe Leap Motion's device will soon be accurate and fast enough to fulfill these needs.

Some participants stated, that while using the realistic hands in VR, they were distracted by the fact that the displayed hands were not their actual own hands. It would be nice to conduct another study where the participants always type with a hand model exactly imitating their own hand.

Some participants also complained about having troubles with recognizing the letters on the keyboard inside of VR. With future improvements to the resolution of the VR lenses, it would be nice to see if typing in VR would improve even further.

Currently, the prototype only allows users to take a typing test inside of a Unity scene. To be able to work while in VR, the whole desktop would need to be accessible while also seeing hands above a keyboard in VR. To really gain an advantage of working in VR, the whole operating system and many programs would need to be optimized for working in VR. An optimized OS should allow features like displaying different applications, in whichever size the user prefers. Furthermore, it should also be possible, to stack as many windows behind each other, as the user desires.

Another interesting field of use of the complete VR desktop is the home office. A user could decide to work from home, put on his HMD and sit directly in his office. This office could be a live render of the actual office, allowing communication with colleagues without actually being in the workplace.

In our study, participants always typed on an Apple keyboard. Since people have different preferences in keyboards, it would be nice to support being able to choose any keyboard to type with in VR.

Bibliography

[BRP02] D. A. Bowman, C. J. Rhoton, M. S. Pinho. “Text input techniques for immersive virtual environments: An empirical comparison.” In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Vol. 46. 26. SAGE Publications Sage CA: Los Angeles, CA. 2002, pp. 2154–2158 (cit. on pp. 15, 16).

[CG80] G. Campbell, S. Geller. “Balanced latin squares.” In: *Purdue University Department of Statistics Mimeoseries* 80.26 (1980), pp. 3–1 (cit. on pp. 16, 35).

[FWO16] A. M. Feit, D. Weir, A. Oulasvirta. “How we type: Movement strategies and performance in everyday typing.” In: *Proceedings of the 2016 chi conference on human factors in computing systems*. ACM. 2016, pp. 4262–4273 (cit. on pp. 13, 14).

[Gen88] D. R. Gentner. “Expertise in typewriting.” In: *The nature of expertise* (1988), pp. 1–21 (cit. on p. 13).

[Har06] S. G. Hart. “NASA-task load index (NASA-TLX); 20 years later.” In: *Proceedings of the human factors and ergonomics society annual meeting*. Vol. 50. 9. Sage Publications Sage CA: Los Angeles, CA. 2006, pp. 904–908 (cit. on p. 43).

[KK04] S. Kim, G. J. Kim. “Using keyboards with head mounted displays.” In: *Proceedings of the 2004 ACM SIGGRAPH international conference on Virtual Reality continuum and its applications in industry*. ACM. 2004, pp. 336–343 (cit. on p. 17).

[KT15] V. Kumar, E. Todorov. “MuJoCo HAPTIX: A virtual reality system for hand manipulation.” In: *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*. IEEE. 2015, pp. 657–663 (cit. on p. 14).

[MS03] I. S. MacKenzie, R. W. Soukoreff. “Phrase sets for evaluating text entry techniques.” In: *CHI’03 extended abstracts on Human factors in computing systems*. ACM. 2003, pp. 754–755 (cit. on pp. 29, 33).

[Opta] OptiTrack. *OptiTrack - NatNet SDK - Stream motion tracking data across networks*. URL: http://www.optitrack.com/products/natnet-sdk/ (cit. on p. 21).

[Optb] OptiTrack. *OptiTrack - Prime 13W*. URL: http://www.optitrack.com/products/prime-13w/ (cit. on p. 21).

[PTW98] I. Poupyrev, N. Tomokazu, S. Weghorst. “Virtual Notepad: handwriting in immersive VR.” In: *Virtual Notepad: Handwriting in Immersive VR*. 14-18 March 1998, pp. 126–132. DOI: 10.1109/VRAIS.1998.658467 (cit. on p. 15).

[Sal84] T. A. Salthouse. “Effects of age and skill in typing.” In: *Journal of Experimental Psychology: General* 113.3 (1984), p. 345 (cit. on p. 13).

[SKT+17] V. Schwind, P. Knierim, C. Tasci, P. Franczak, N. Haas, N. Henze. “These are not my hands!: Effect of Gender on the Perception of Avatar Hands in Virtual Reality.” In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM. 2017, pp. 1577–1582 (cit. on pp. 14, 15, 25).

[VLH12] P. D. Varcholik, J. J. LaViola, C. E. Hughes. “Establishing a baseline for text entry for a multi-touch virtual keyboard.” In: *International Journal of Human-Computer Studies* 70.10 (2012), pp. 657–672 (cit. on p. 14).

[VME+15] K. Vertanen, H. Memmi, J. Emge, S. Reyal, P. O. Kristensson. “VelociTap: Investigating fast mobile text entry using sentence-based decoding of touchscreen keyboard input.” In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM. 2015, pp. 659–668 (cit. on p. 17).

[WS98] B. G. Witmer, M. J. Singer. “Measuring presence in virtual environments: A presence questionnaire.” In: *Presence: Teleoperators and virtual environments* 7.3 (1998), pp. 225–240 (cit. on pp. 38, 45).

[YEYG03] E. Yechiam, I. Erev, V. Yehene, D. Gopher. “Melioration and the transition from touch-typing training to everyday use.” In: *Human Factors* 45.4 (2003), pp. 671–684 (cit. on p. 13).

All links were last followed on July 27, 2017.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

place, date, signature