

Institute for Visualization and Interactive Systems

University of Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Bachelorarbeit

Influence of Real World and Virtual Reality on Human Mid-Air Pointing Accuracy

Robin Schweigert

Course of Study:	Informatik
Examiner:	Jun.-Prof. Niels Henze
Supervisor:	Dipl.-Inf. Sven Mayer, Dipl.-Ing. (FH) Valentin Schwind
Commenced:	April 4, 2017
Completed:	October 4, 2017
CR-Classification:	H.5.2

Abstrakt

Freihändiges Zeigen ist eine mächtige Geste, um nonverbal Richtungsangaben auszudrücken. Diese Arbeit wird ihren Fokus auf absolutes Zeigen legen um Objekte oder Personen mit direktem Sichtkontakt referenzieren zu können. Wir sehen freihändiges Zeigen für die Zukunft als eine gute Möglichkeit, um mit Objekten und Smart Home Umgebungen zu interagieren. Jedoch könnte freihändiges Zeigen genauso Contoller für die Steuerung virtueller Umgebungen ersetzen. Bisherige Arbeiten haben bereits gezeigt, dass Menschen bei freihändigem Zeigen ungenau sind. Diese Arbeiten konnten weiterhin einen systematischen Fehler bei freihändigem Zeigen nachweisen. Mit dieser Arbeit werden wir diese Fehler reproduzieren und weiter zeigen, dass die gleichen Fehler auch in virtuellen Umgebungen auftreten. Wir werden weiter zeigen, dass Menschen signifikant anders in der Echtwelt und virtuellen Umgebungen zeigen. Daher entwickeln wir unterschiedliche Modelle für das Berechnen der tatsächlichen Zeigerichtung. Diese Modelle bauen auf Daten aus einer ersten Studie auf, während welcher wir Probanden beim Zeigen aufgezeichnet haben. Des weiteren verifizieren wir diese Modelle durch eine zweite Studie mit neuen Probanden. Unsere Ergebnisse zeigen, dass wir den Fehler signifikant reduzieren können. Des weiteren können wir zeigen, dass ein Cursor, welcher die Zeigerichtung anzeigt, den Fehler weiter reduzieren kann. Jedoch steigt die benötigte Zeit für die Zeigegesten durch diesen Cursor an.

Abstract

Mid-air pointing is a major gesture for humans to express a direction non-verbally. This work focuses on absolute pointing to reference an object or person which is in sight of the person who performs the pointing gesture. In the future, we see mid-air pointing as one way to interact with objects and smart home environments. However, mid-air pointing could also replace the controller to interact with a virtual environment. Recent work has shown that humans are imprecise while mid-air pointing. Furthermore, previous work has shown a systematic offset while mid-air pointing. In this work, we are reproducing these results and further reveal that the same effect is present in virtual environments. We further show that people point significantly different in a real and virtual environment. Therefore, to correct the systematic offset, we develop different models to determine the actual pointing direction. These models are based on a ground truth study in which we recorded participants' body posture while mid-air pointing. Finally, we validate the models by conducting a second study with 16 new participants. Our results show that we can significantly reduce the offset. We further show that when displaying a cursor indicating the pointing direction the offset can be further reduced. However, when displaying a cursor the pointing time increased in comparison to no cursor.

Contents

1	Introduction	15
2	Related Work	17
2.1	Pointing	17
2.2	Pointing Devices	18
2.3	Visual Feedback	20
2.4	Ray Casting	21
2.5	Hand Tremor	21
2.6	Summary	22
3	Data Collection Study	23
3.1	Study Design	23
3.2	Apparatus	23
3.3	Procedure	32
3.4	Participants	34
3.5	Summary	34
4	Modeling	35
4.1	Fatigue Effect	35
4.2	Preprocessing	35
4.3	Model Creation	37
4.4	Cross Validation	39
4.5	Summary	40
5	Evaluation Study	41
5.1	Study Design	41
5.2	Apparatus	42
5.3	Procedure	43
5.4	Participants	44
5.5	Summary	45
6	Results	47
6.1	Fatigue Effect	47
6.2	Accuracy	47

6.3	Task Completion Time	48
6.4	Summary	49
7	Discussion	51
7.1	Models	51
7.2	Task Completion Time	52
7.3	Limitations	52
7.4	Summary	53
8	Conclusion	55
A	Appendix	57
	Bibliography	59

List of Figures

3.1	The marker arrangement on the participant.	25
3.2	The Marker Arrangement on the HTC Vive.	26
3.3	Schematic side view of the arm.	29
3.4	Main graphical user interface for entering all measured data for each participant.	30
3.5	Virtual study scene from the participant's view.	31
3.6	Overview of the virtual study scene.	32
3.7	A single target on the projector screen.	33
4.1	Different plots of offset between pointing gesture and actual target. . . .	38
5.1	A target on the projector screen with visual feedback.	43
A.1	The raw NASA TLX questionnaire.	58

List of Tables

3.1	Results of eye dominance tests for data collection study.	34
4.1	Remove outliers for the different ray casting methods.	36
4.2	Absolute distances in centimeters for fingertip and forearm on the projector screen.	39
4.3	Absolute offsets after cross validation in centimeters.	40
5.1	All different conditions for the evaluation study.	42
5.2	Results of eye dominance tests for evaluation study.	45
6.1	All raw NASA Task Load Index (raw NASA TLX) scores from the second study in range 0 to 100.	47
6.2	Remaining offsets between intersection and actual target in cm for second study.	48
6.3	Task completion time (TCT) in seconds for second study.	49

List of Abbreviations

- EFRC** cyclops eye fingertip ray casting. 21, 35
- FARC** forearm ray casting. 21, 35
- FOV** field of view. 51
- FPS** frames per second. 35
- FTRC** fingertip ray casting. 21, 35
- HMD** head mounted display. 23
- HRC** head ray casting. 21, 35
- LOOCV** leave one out cross validation. 39
- raw NASA TLX** raw NASA Task Load Index. 11
- RM-ANOVA** repeated measurement analysis of variace. 47
- RW** real world. 15
- TCT** task completion time. 11
- VR** virtual reality. 15

1 Introduction

Mid-air pointing is a natural and powerful gesture. Already in childhood, humans learn how to point. An infant learns how to communicate with other persons, even without knowing anything about language. The infant learns to point at things and starts to laugh or cry. Surrounding people like the parents directly know what is meant with the pointing gesture. Later in the adult life, pointing still is a powerful gesture to clarify a specific direction. In conversations, people point at things to support verbal communication. However, recognizing a mid-air pointing gesture still is an open question for computers. Using context and deduction, humans are well skilled in solving potential inaccuracies or ambiguities while mid-air pointing at a specific target. However since an interactive computer system does not have this kind of context or at least it is very hard to obtain context-based information for today's computers, they try to determine the direction by analyzing the body posture only. Related work showed that determine the pointing direction out of pointing gestures is a complex task for a computer. Further related work presented methods to cast a ray for the pointing posture to determine the object of interest.

Over the last few years virtual reality (VR) arrived in both the consumer sector and research. The hardware is finally ready for a wide area use. For controlling virtual environments today's VR-Headsets are shipped with separate controllers. Those controllers have to be held with the hands. This enables people to interact with the virtual world and also to point at things in this world. Nevertheless, those controllers interfere with the naturalness of pointing and pointing gestures. There is currently some hardware, like the LeapMotion¹ which enables users to use their own bare hands as controllers. However, there are no models which let computers interpret the pointing gestures recorded with hardware like the LeapMotion accurately.

We will record pointing gestures in both real world (RW) and VR to study how humans actually point. With those recorded ground truth pointing gestures, we aim to compare different ray cast techniques to determine which suites best to identify the pointing direction. We will train a model on the remaining offset of the best ray casting method to minimize the error generated during the recognition of the pointing gesture by the

¹<https://www.leapmotion.com/>

computer. We want to compare those Models for RW and VR. In a second study, we then want to verify our models and see, how good they perform with new participants. In the last step, we want to find out, how good participants can point while provided with a visual feedback in form of a mouse cursor on the projector screen.

Structure

This thesis separates into following parts:

Chapter 2 – Related Work: In this chapter, we discuss the related work.

Chapter 3 – Data Collection Study: This chapter we present the first Study for data collection.

Chapter 4 – Modeling: In this chapter, we present our approach for modeling the data.

Chapter 5 – Evaluation Study: This chapter contains our second study for verifying our models.

Chapter 6 – Results: In this chapter, we present our results from the second study.

Chapter 7 – Discussion: This chapter will discuss all results and will try to explain them.

Chapter 8 – Conclusion: In this chapter, we come to an overall conclusion of this thesis.

2 Related Work

In this chapter, we will present the current state of research. We first present works that are dealing with pointing as a possibility of expressing a direction. We continue with works where a pointing device was used to enable interaction with systems. Followed by works which investigated visualizations for the pointing direction. We then present works which investigated how to determine a pointing direction without devices followed by how this pointing is influenced by natural hand tremor.

2.1 Pointing

Pointing is a natural way of interaction between humans. Every time in a conversation about objects when someone wants to make clear, what object he actually means, he would just point at it. This is understood by almost every human, never the less which language they are speaking. This pointing gesture is learned and developed in early childhood. Haviland et al. suggested a "natural history of pointing" [Hav03]. They stated, that "the often assumed simplicity of "pointing gestures" [Hav03] is wrong, and that pointing is a complex task. But on the other hand, they showed, that even young children, who can not even barely speak but use single words, are already able to make their intentions clear by using pointing gestures. But pointing is not only a powerful gesture in early childhood. Also in the later adult life, pointing is still used all the time in conversation. McNeill et al. stated, that pointing and speech in a conversation can support each other by "convey[ing] information about the same scene, but each can include something the other leaves out." [McN92].

Distance pointing, "pointing directly at a target that is remotely situated with respect to the input device" [KBSM10] how Kopper et al. defined it is the gesture of pointing at objects "that enables the user to move around freely while still being able to interact with an application" [KSMB08]. Since distance pointing is such a natural and intuitive gesture the next step was to combine pointing with interaction on computers. The first bigger attempts on this were done by Bolt et al. [Bol80]. They created a big room with multiple projectors where the users could choose between different input methods, including pointing without any additional devices. But they did not only rely

on the pointing gestures but also provided the possibility of entering input via speech recognition and also manual input via joystick. They also provided visual feedback in form of a cursor. This "Media Room" mainly started the research in the field of pointing gestures for computers.

So far there were many different studies which investigate the pointing behavior of humans. Pfeiffer et al. [PKL06] tried to find a way of analyzing pointing gesture information. In their paper, they formulated different difficulties while processing pointing gestures. They stated, that the interpretation of recorded data done by humans is time-consuming and the translators easily make mistakes while interpreting the data. Hence we will only use a fixed computer software for interpretation without the input of a human. Furthermore, they state that the conditions around an experiment do have a big influence on human communication. They refer this to general human interaction but this also includes pointing gestures. Hence we have to eliminate every possible difference between our scene in virtual reality and the real world setup or at least make it as small as possible. This will ensure comparable data over both environments. Another study by Schwind et al. [SKT+17] deals with different hand models in VR. They investigated how different hand models influence the presence and acceptance of the VR-scene. They used six different hand models, an abstract hand, a cartoon hand, a robotic hand, a male hand, a female hand and an androgynous hand. The participants had to conduct three different tasks and fill out a presence questionnaire afterward. Out of the realistic human hands, in average the androgynous hand performs the best. Especially the female participants did not accept the male hand. Hence we decided to use exactly the same realistic human androgynous hand.

2.2 Pointing Devices

When trying to record pointing gestures you can mainly distinguish between two possible ways of recording, the device free recording and one where you provide the user with a pointing device and record the pointing gestures according to this device. For example, Olsen et al. [ON01] investigated how good pointing gestures work in collaborative tasks in group meetings. They provided the participants with a small laser pointer. The tasks were to interact with a software during a group meeting. The software contained interaction tasks, like scrolling through lists, clicking buttons, entering text or just move the mouse cursor. The laser pointer was recognized with a camera directly on the projector screen. This has the advantage of not having to recognize anything about the participants at all but just rely on the laser pointer. On the other hand, you still have to use a device. A different device was used by McArthur et al. [MCM09]. They tried to use a Wii remote as stand-alone and combined with two different attachments, the "Intec Wii

Combat Shooter" [MCM09] and the "Nintendo Wii Zapper" [MCM09]. They conducted a two-dimensional target selection task. In this task, the participants were provided with multiple small circle arranged in a big circle on a projector screen. They had to select one highlighted circle with the provided input devices. Even that they found out, that you can get good results regarding the accuracy you still have to use an input device. In 2010 Kopper et al. [KBSM10] tried to find a model, similar to the Fitt's Law, to describe the relation between the pointing gesture and the movement time. In their experiment, they used a wireless mouse for interacting with a large display array. Attached to this mouse were several reflective markers for full three-dimensional tracking. With this three-dimensional mouse, the participants were asked to select different targets on the screens in a given order. But they did not try to find a model for resulting offsets, they were only looking into the actual speed. And as a second downside, respectively to our goals, they also did use a pointing device, namely the three-dimensional mouse. Kopper et al. [KSMB08] tried to increase the accuracy of distance pointing at large screens. They stated different problems with distance pointing. First of they presented the "Heisenberg Effect" [KSMB08]. This effect describes the fact, that if a participant uses a pointing device while clicking to accept the input, he will often move a bit with his hand. According to them this effect also appears if participants do not use a pointing device but just point with their hand and make a gesture to accept the input. They suggest using a technique they call "framing" [KSMB08]. This means, that the pointing gesture while clicking is ignored and only the gesture at the beginning of the click is taken into account. However, we decided to prevent this effect by taking the other hand to accept the input. This way the participants will have to point with one of their hands and click on a device in their other hand. By doing this, the pointing hand will not move while clicking. One other problem they address is the differences in pointing accuracy when pointing from different distances. They state, that pointing from a closer distance is more accurate than pointing from further away. This effect is based on the fact that from a smaller distance a movement of the pointing apparatus results in less movement on the target than the same movement would do further away from the target. This could disturb the recorded data, but only if different distances to the targets are used to create the models. Hence we will limit our study to one overall distance.

On the other side, there are not only studies done for pointing with a device. Many studies do not use such devices. Pfeiffer et al. [PLW08] for example conducted a study in 2008 where they did not use a special pointing device but just tracked the hand of the participant with an optical tracking system and then they used the fingertip as ray casting root. During their study they let two participants play an "identification game" [PLW08] how they call it. During this study, one participant had to point at an object on a table and the other participant had to identify at which object the other was pointing at. During the evaluation, they created two different cones in the direction of pointing one proximity cone and one distal cone. The proximity cone is for pointing at objects

below a given threshold and the distal cone for objects over this threshold. According to them, this can result in a good identification of objects, but since they use those cones, they are limited to identifying whole objects and can not get a single point where the participant is pointing at. This is not the way we want to go, we actually want a single target point. Another study dealing with pointing without any controller device by Vogel et al. [VB05] describes different pointing methods. In this study they were using the index finger ray casting, meaning casting the ray for interaction from the fingertip in the same direction than the index finger. With this technique, it is possible to get a single point as the target from the participant, in comparison to being limited to only returning objects from the participant's ray cast like it was with Pfeiffer et al. [PLW08]. This gave Vogel et al. the possibility to attach a mouse cursor to the extended ray on the screens. For controlling the mouse they defined three different input methods, the absolute pointing, the relative pointing and a mixed form of both. With absolute pointing, the target is directly calculated from the ray cast from the participant. An advantage of this is, that it is rather intuitive for the users since they just point where they want to aim at. This method also does not require any feedback at all. The relative pointing instead uses the current position of a cursor and then uses the pointing gesture to control the cursor. A direct disadvantage of this is, that this method always requires a visual feedback since the participants can only control a cursor. On the other hand, an advantage of the relative pointing is that it can be way more accurate than the absolute pointing. With relative pointing, you can just modify the ratio between movement of the arm and movement of the cursor to get more accurate results. The mixed form combines the absolute pointing for rough pointing and the relative pointing while more accurate results are necessary. Since we do not want to always provide visual feedback, we decided to only use the absolute pointing.

2.3 Visual Feedback

Visual Feedback describes the mechanism which provides the participants with any sort of visual markers as feedback. In a classical way, this is just a cursor similar to a standard mouse cursor on a computer. This is done for example by Jiang et al. [JOMS06] and Kopper et al. [KSMB08] and Zhai, Morimoto, and Ihde [ZMI99] and Bolt [Bol80]. They all use some kind of input methods, from cameras like Jiang et al. [JOMS06] did to eye trackers and the gaze input like Zhai, Morimoto, and Ihde [ZMI99] used to directly control the mouse cursor. Bolt et al. [Bol80] for example displayed "a small white 'x' cursor [...] provid[ing] running visual feedback" [Bol80]. Even that this is a good way to control computers, it can not be used in all situations. Since this feedback method is limited to two dimensions it will always need some kind of monitor or projector screen to provide the feedback, at least in RW. In VR a cursor can be accomplished without any

problems, even in a three-dimensional setup. Another way for three-dimensional visual feedback in a virtual scene is described by Wong et al. [WG14]. They use different ways of pointing and visual feedback in a virtual environment. First of they have the "Long arm" [WG14]. This feedback extends the virtual arm of the participant to the calculated target in the scene. A second Method is the "Laser Beam" [WG14]. In this scenario, a laser beam is sent from the extended index finger of the virtual arm pointing in the calculated direction to the target. Thirdly they used a method called "Spotlight" [WG14] that uses the same ray cast like the "Laser Beam" but only show a dot on the intersection point. Finally, they have the "Highlight" [WG14]. In this method, they highlight the intersected object calculated by the ray casting. If you want to accomplish similar feedback in RW you can use a laser pointer like Olsen et al. [ON01] did. Even that they only used the laser for controlling the mouse cursor, something like this could be used as a real-world feedback and not only on one screen.

2.4 Ray Casting

For calculating the actual ray cast many studies used different methods. Some, like Corradini and Cohen [CC02], describe the possibility of index finger ray casting. With this method, the ray cast starts from the finger root and continues through the fingertip. Some other described a ray casting, where the ray starts at the head and continues through the fingertip. Jovic et al for example call this "line-of-sight pointing" [JBM+00], Mine calls it "Crosshair mode" [Min95], Pierce et al. call it "sticky finger" [PFC+97]. Mine also presents the "gaze direction" [Min95] where the ray is calculated from the forward head direction thus further referenced as head ray casting in this thesis. The same method is described by Argelaguet et al. [AA09]. The last method we want to use in this thesis is described by Nickel and Stiefelhagen [NS03] as "the orientation of the forearm". With this method, the ray cast is calculated from the forward direction of the extended forearm. We will further refer to this as fingertip ray casting (FTRC), forearm ray casting (FARC), head ray casting (HRC) and cyclops eye fingertip ray casting (EFRC).

2.5 Hand Tremor

Riviere, Rader, and Khosla [RRK97] showed with their studies, that every participant has a small tremor in their hand. This means, that the hand is trembling a little bit while the participant tries to hold it still. Riviere, Rader, and Khosla [RRK97] originally only investigated the hand tremor with eye surgeons, but you can transfer this behavior to

every other human. Since we also want to get as accurate results as possible we have to take the natural tremor of our participants into account. Therefore we will record multiple data points for one pointing gesture and average them out. This will lead to a small influence of the natural hand tremor.

2.6 Summary

In this chapter, we presented the current state of research. We showed different possibilities for pointing with and without devices, different forms of visual feedback and some ray casting methods. Finally, we described problems out of older research and how we plan to overcome those. Mayer et al. [MWSH15] have already accomplished lots of those aims. Nevertheless, their study had some disadvantages which we want to overcome with this thesis. First of their system did not have the possibility of live correction of recorded pointing gestures. Secondly, they only conducted one study, built their models around this data but never tested it with new participants. Finally the study of Mayer et al. [MWSH15] only used RW.

Our target will be to record pointing gestures, build a model around this data and create a system capable of live compensation of offsets. We then want to verify this system with a second study and new participants. We also want to compare visual feedback against no visual feedback with this second study. Finally, we want to achieve an overall comparison between pointing in RW and in VR.

3 Data Collection Study

The goal of the data collection study was to record ground truth pointing gestures. In this chapter, we will present our approach to both the hardware and software setup, our study design and which data we plan to record for later investigation. With this recorded information we will then be able to train our models which we then want to test later in a second study. Finally, we will present our participants from the first study.

3.1 Study Design

The main aim of the first study was to collect as many as possible pointing gestures of different participants. To get pointing gestures from all over the projector screen, we showed targets spread evenly over the whole projector screen. We had seven target positions in horizontal and five targets in vertical. The targets had a horizontal distance of 44.9cm between each other and to the border of the projector screen and a vertical distance of 34.05cm once again between each other and to the border of the projector screen. We only showed one target at the same time and we randomized the order of appearance of those targets. Our independent variables was the setting, meaning RW and VR. As dependent variables, we recorded all necessary points on our participants to calculate the pointing gestures. As conditions, we had a part in RW on the projector screen and a second part while wearing the head mounted display (HMD). We counterbalanced those condition, which means, participants with odd ids started with the RW part and participants with an even id started with the VR part.

3.2 Apparatus

Our apparatus separates into different parts. First of we have the hardware setup consisting of a tracking system, a projector, the VR glasses and a central computer. Running on this computer we have our second part, the software including both the tracking software and our own software for evaluation. In our software and in our real laboratory we have arranged a study environment.

3.2.1 Hardware

Our hardware setup consists of three main parts, OptiTrack, the VR glasses and the central computer running our software.

OptiTrack The first part is an OptiTrack system for recording the gestures. OptiTrack is a motion capturing system widely used by both consumer-oriented companies mainly in the film industry and research institutes. It consists of multiple cameras, synchronized over a central hub, the tracking markers, small spheric infrared mirrors, which can be attached to the tracked object and the Motive software by OptiTrack, which calculates the positions of the tracked markers out of the camera pictures. Although with a normal calibration the system is only capable of an approximately tracking, when calibrated over a longer time, the accuracy is sufficient for scientific needs.

In our setup we used the Flex 3¹ system with 14 cameras. Each camera has a resolution of 640×480 , which overall results to 0.3 MP. The cameras have a field of view of 46 degrees. Each camera records images with a frame rate of 100 frames per second. Around the lenses, each camera has a ring of infrared LEDs for better illumination of the tracked space. The cameras are connected to the main synchronizing hubs via USB 2.0 cables. The hubs, on the other hand, are synchronizing over a cinch connection. We arranged those cameras in a circle over the head of the participants. The main focus point of the cameras was on the position of the right arm, while completely stretched out towards the projector screen. We calibrated the system with a 40cm wand and the OptiTrack Motive software. After calibrating we got a mean error of the calibration by OptiTrack of 0.513mm.

For tracking, we used rigid body markers from OptiTrack combined with some 3D-printed markers from us. Each rigid body marker consists of at least three infrared markers. Those markers have to be arranged in a unique way. That means that no combination of distances between the infrared markers should appear in another rigid body. The rigid body markers were arranged mainly on the right arm, on the head, and on the shoulders (see Figure 3.1 on the facing page). The head marker was for the RW part of the study to keep track of the head movement. The markers on the shoulder were to record the participant's movement of the upper body. On the right arm, we attached marker to the upper arm, to the forearm, to the hand, and to the index finger. We used those markers mainly for recording the actual pointing gestures.

¹<http://optitrack.com/products/flex-3/>



Figure 3.1: The marker arrangement on the participant.

Visualization The second part of our hardware setup was the VR glasses. We decided to use a HTC Vive² as VR hardware.

The HTC Vive has a resolution of 2160×1200 . Its screen has a refresh rate of 90 Hz. The HTC Vive has a field of view of 110 degrees. It weighs about 590 grams. The Vive is shipped with two controllers for hand tracking and interaction and two tracking stations for positional tracking. This tracking stations contain two rotating infrared lasers each and an infrared flash each. Those lasers build up a laser mesh in the tracking space with which the headset and the controller can calculate their position and orientation inside the tracking space. However, our tests showed, that the tracking system of the HTC Vive is not good enough for exact scientific studies. The actual tracking space, calculated by the Steam software, shifts around by multiple centimeters, even if you ensure, that the tracking lasers are fixed to a robust rig system. Hence we decided not to use the internal laser tracking of the HTC Vive but just use the same OptiTrack system we are using for the body tracking. To use the OptiTrack system with the HTC Vive we had to add some infrared markers to the VR headset. For the arrangement see Figure 3.2.

²<https://www.vive.com/uk/>

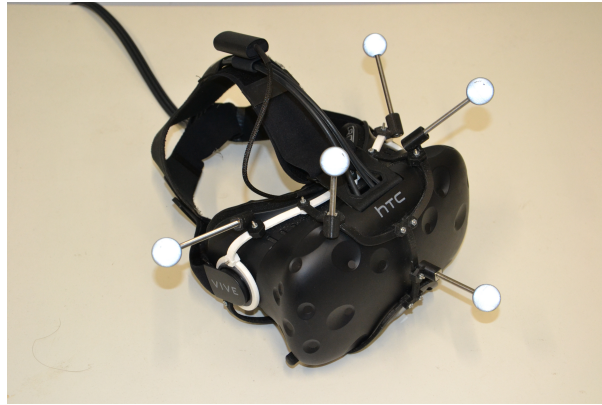


Figure 3.2: The Marker Arrangement on the HTC Vive.

Beside the VR headset we used a standard full HD projector to display the study scene in RW on a projector screen. This projector was mounted above the head of the user in the middle to the projector screen. It was mounted as close as possible to the projector screen so the participants would not interfere with the image and cast a shadow on the projector screen while pointing at a target.

Computer The third part of the hardware setup was a central computer. It consisted of an i7-5820k, 32Gb of RAM and a NVIDIA GeForce GTX 1080. This computer was running a 64bit version of Windows 10. The central station was for handling all information out of the OptiTrack system and for rendering the testing scene on both the VR headset and the projector. It was connected to all synchronizing hubs via USB 3.0 and to the VR headset and the projector via HDMI. For the study scene, it ran our study software which will be described later in this thesis.

3.2.2 Room Setup

Our testing room was a laboratory. We used laser distance measurement tools to get the exact size of the laboratory and the projector screen. The laboratories' dimensions were 5.99m in width, 5.55m in length and 3.52m in height. On one side, we had the projector screen. Its dimensions were 3.592m in width and 2.043m in height. The participants were standing in two meters distance in front of the projector screen on a small platform with a height of twelve centimeters. We used this platform for the user to stand in about the middle of the projector screen. With those dimensions, we were able to exactly rebuild the whole laboratory in 3D on the computer. We tried to make the differences between the real laboratory and the virtual scene as small as possible. Hence we scaled the virtual room to the exact same size as the real laboratory and we

positioned the projector screen at the exact same position in the virtual environment as in the real laboratory. Furthermore, we positioned the platform at the exact same spot in the virtual environment and in the real world. This was important, because while the user wears the HMD he does not see his real surroundings. Hence he does not see, that he is standing on a platform. To protect the participants from falling off the edge of the platform we aligned the platform as perfect as possible with the real platform, so the participants do know, that they are standing on an elevated part and not on the flat ground. To make the differences even smaller we decided to arrange multiple objects like tables, chairs, and computers at the same position in the virtual environment and the real laboratory. This helps the participants getting less distracted by the surrounding when entering the virtual environment.

3.2.3 Software

On the software side, we had to manage all the tracking data coming from OptiTrack, saving positions and orientations of all parts of the body we needed, managing the study procedure and rendering the whole virtual environment on both the HMD and the projector screen.

For handling the tracking and calibration of the tracking space, we used OptiTracks Motive Software³. To calibrate the software we had to arrange the cameras in a first step. In a second step, we had to walk inside the whole tracking space and move a calibration wand through the whole space. This calibration wand is a rod with three infrared markers with a fixed distance to each other on top of it. When walking around in the tracking space the Motive software can calculate the position of the camera in relation to each other as soon as at least three different cameras can see all three infrared markers on the wand. It is just important to move the wand to every position in the tracking space and also to every position inside the camera images of each camera. This will then increase the chances of a perfect calibration. After calibrating the positions of all cameras we had to set up the floor plane. OptiTrack offers a small piece of hardware with three infrared markers on top of it and a function inside the Motive software with which it is possible to calibrate the floor plane, the zero point and the direction of Motive's coordinate system all at once. We decided to place the zero point somewhere in the middle in front of the projector screen and we let the coordinate system point straight to the projector screen.

The version of Motive we were using comes with the full capability of body tracking, meaning it is possible to track a whole human body with infrared markers attached to

³<http://optitrack.com/products/motive/>

it. But since this body tracking uses interpolation to get better-looking results, it is not accurate enough for our needs. But the software is also able to create rigid bodies out of multiple infrared markers and stream the position and orientation of those rigid bodies to our Software via a network. A rigid body is a set of at least three infrared markers with fixed positions and lengths relative to each other marker in the set. With this constraints given, Motive is then able to calculate a position and also an orientation for this rigid body which would not be possible with single infrared markers. Furthermore, it is possible to arrange the center points of this rigid bodies to fit any special needs. Each of those final markers can get an id then, which makes it identifiable in the network stream.

Furthermore, in next step, we had to decide which rendering engine we wanted to use. Currently, there are two main engines on the market, the Unity3D Engine, and the Unreal Engine. We decided to use the Unity3D engine because of the well working plugin by OptiTrack and the possibility of writing code in C#. Furthermore, the Unity3D asset store offered lots of 3D objects, which we used in our main scene. After we decided to use the Unity 3D engine and not to use the full body tracking of OptiTrack we started to plan an own software system for tracking all part of the participant's bodies which we needed movement data from.

For streaming the position and orientation data from OptiTracks Motive software we used the official Unity3D plugin by OptiTrack. This plugin is able to handle the network stream out of Motive and to convert the data to the Unity3D engines build in types of position and orientation. In the Unity3D Engine, it is possible to just enter the corresponding id out of Motive with which it will then return all the data for this specified rigid body. One problem we had was the alignment of the scene root of Unity3D and OptiTracks zero point. Since earlier we had to calibrate the zero point of OptiTrack by hand, it will almost never match perfectly with the origin point in Unity3D. To calibrate this, we displayed a cross on the projector screen in RW and attached a rigid body to the exact position on the projector screen. With the streamed position of this rigid body in Unity3D, we were able to calculate the offset between both zero points. We then verified this offset with multiple crosses displayed on the projector screen at different positions. After verifying the offset, we could then apply this offset to all streamed data for aligning both zero points.

We decided to only use right-handed people for our study, hence we only needed markers on the right arm. Like described in Section 3.2.1 we put a marker on each shoulder for getting the body position, a marker on the upper arm, forearm, hand and index finger for getting the position of the right arm and a marker on the head while the participants did not wear the HMD for keep track of the head position and orientation. But since those markers are on top of the skin of the participants and always only roughly on the same position on each spot between the participants we had to build a system which

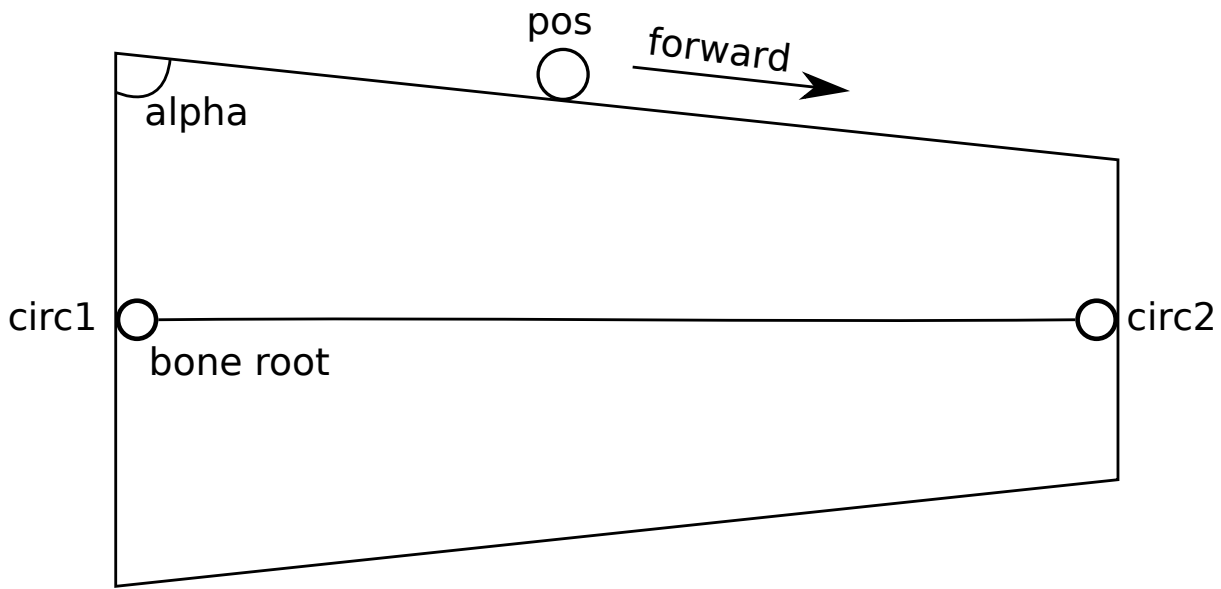


Figure 3.3: Schematic side view of the arm.

allows us to enter some measurement data for each participant to adopt our software to each individual body of each participant. For the head, we wanted to find out where exactly the cyclops eye for each participant is located. The cyclops eye is an imaginary third eye located in the middle between both actual eyes somewhere at the top of the bridge of the nose. In VR we simply used the main camera in the scene as cyclops eye since this camera is located directly between the eyes. For RW we measured the head circumference and the eye height for each participant. With these values, we were then able to calculate the actual position of the cyclops eye in relation to the head marker. To get this position, we used the tracked position of the head marker, which was located in the center of the head circumference. We then calculated the radius of the head with the following formula: $r = \frac{\text{headcircumference}}{2*\pi}$. With this value, we went exactly the length of the radius to the forward vector of the head marker. From there we went the length of the eye height to the down vector of the head marker. This was the position we used as cyclops eye.

For the shoulder markers, we could just use the position and orientation of both rigid bodies themselves since they were located directly on the shoulders.

For the arms, we did not want the position and orientation of the actual rigid bodies but the position and orientation of the bones inside the arm (see figure 3.3 – Schematic side view of the arm.). For the upper arm, we measured the circumference of the arm at the shoulder (*circ1*) and at the elbow (*circ2*). For the position of the rigid body marker, we measured the distance *pos* between the shoulder and the marker. With this measured values we started at the position of the rigid body marker and went the length of *pos* in

Figure 3.4: Main graphical user interface for entering all measured data for each participant.

the direction of the backward vector of the rigid body to get to the point at the top of the arm bone root. With both circumferences, we could then calculate both diameters on the shoulder side and on the elbow side. With this diameters, we could calculate the angle α . After calculating this angle we could move down by half the length of the diameter at the shoulder to reach the point of the bone root. With the angle α given we could calculate the orientation of the bone with the orientation of the rigid body and the angle α . For the forearm, we did the same thing but with the elbow circumference as *circ1*, the wrist circumference as *circ2* and the distance between the elbow and the forearm marker as *pos*.

For the hand once again we did not want to have the position of the actual rigid body marker but the position of the bone root. For this, we measured the length of the hand and attached the marker to the center of the back of the hand. For calculating the actual bone root we just went half the hand length backward and half the wrist diameter downwards relative to the hand marker. As the orientation, we simply used the orientation of the rigid body. On the index finger, we attached the marker directly to the fingertip and then went back by the length of the finger to get the bone root. The orientation, once again, was the orientation of the finger rigid body.

With all these bone positions and orientations, we were then able to track the upper body and the right arm for each individual participant. In the VR scene we added a t-shirt with a movable right arm. The t-shirt itself was attached to the shoulder markers.

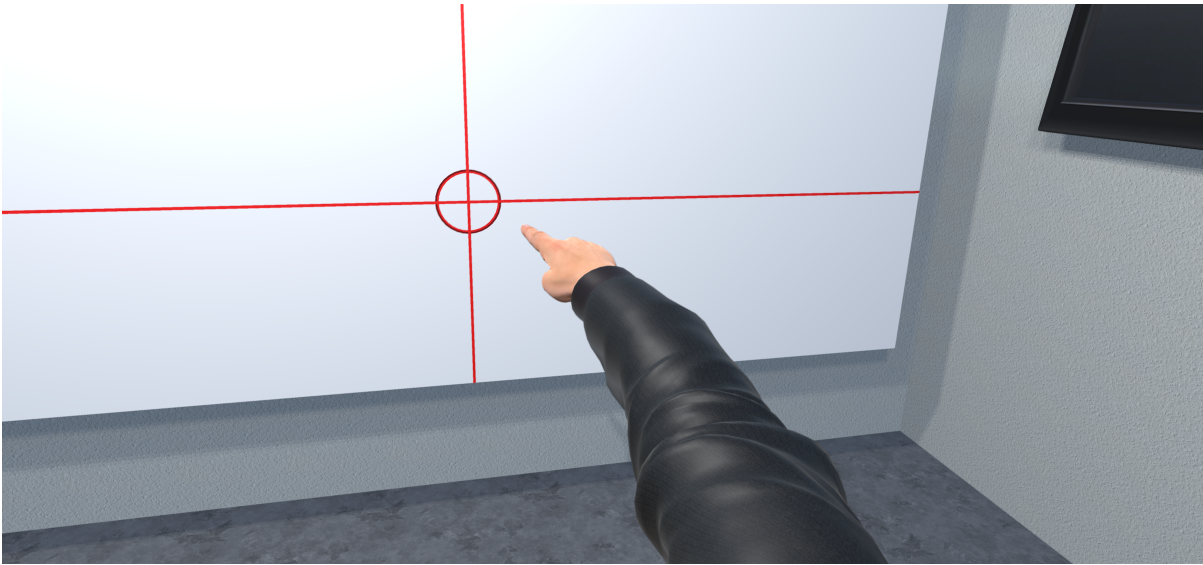


Figure 3.5: Virtual study scene from the participant's view.

For the upper arm and forearm of the t-shirt, we used the calculated positions and orientations of the bones. At the end of the t-shirt, we added a human hand to the model attached to the calculated position and orientation of the hand bone. At the fingers, only the index finger was tracked, once again bound to the calculated position and orientation of the finger rigid body.

Our software itself consists of three different views. In the first view, see Figure 3.4, we could enter all the participant data, like the id and all the measurements.

The second view is a preparation scene, similar to the actual study scene. In this scene, the participants can try out the study setup. We do not record any data here. After the participants feel comfortable with the study mechanics, we can start the actual study. In this scene, all the movement data is recorded. We present an overview of the actual study scene in Figure 3.6 and from the participant's perspective in Figure 3.5. In this scene, there is also the possibility of showing texts on the projector screen in both RW and VR for example for questionnaires which have to be filled out by the participant. Besides this three views for the actual study, there is always a view on a separate monitor for us to supervise the study. In this view, we can see the whole room from different perspectives to be able to always see the whole tracked body of the participants and to see if anything goes wrong.

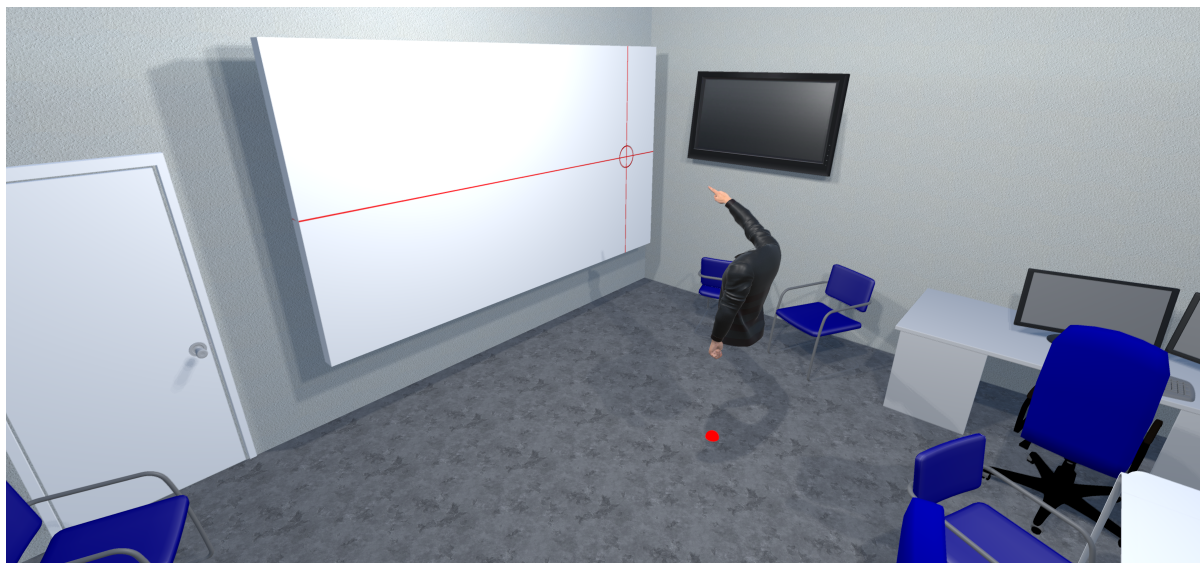


Figure 3.6: Overview of the virtual study scene.

3.3 Procedure

At the beginning of the study, before doing anything else, we welcomed the participants and asked them to read through our consent form and sign it, if everything was all right from their side. The consent form included standard legal information like that the participants are always allowed to interrupt or cancel the study, information about what data we want to collect and finally, a part, where the participants could choose if it was ok for us to take and publish pictures while they were conducting the study. After signing this consent form we started to explain what the study was about without telling them too much about the actual aims of the study to not influence them. As a first part, we then started to measure all the necessary lengths of the participants. This included the length of the upper arm, the forearm, the hand and the index finger, the circumference of the upper arm at the shoulder, the elbow, the wrist and the index finger, the shoulder height, the head circumference and the eye height. After we collected all this data we had to enter it into our software. After this, we conducted two different tests for eye dominance with the participants. The first one was the pipe test, the second one the target test. In the pipe test, we asked the participants to take a small paper pipe and hold it next to their body. Then they should take the pipe in front of their face and search for a distant object, in our case a street sign, through this pipe. The eye in front of which they put the pipe is the dominant eye for this test. In the target eye dominance test, the participants were asked to point at a distance target. If they thought they hit the target we asked them to successively close both eyes and see with which opened eye they still hit the target. This eye is the dominant eye for this test.

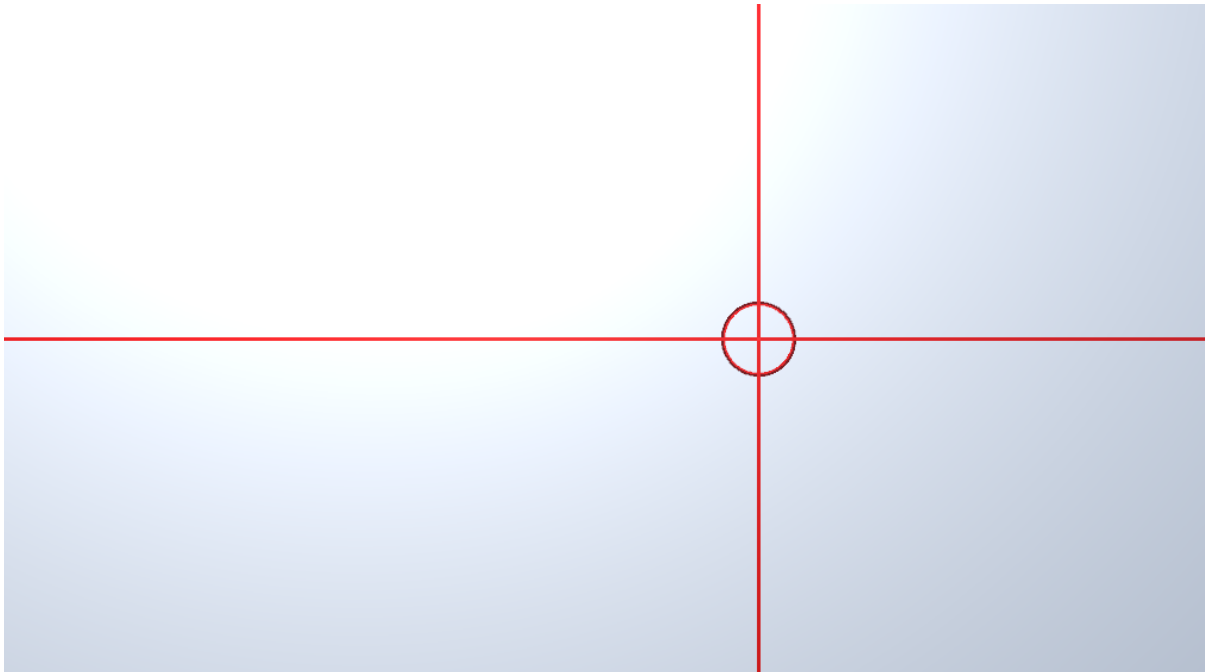


Figure 3.7: A single target on the projector screen.

After measuring and testing everything we attached all markers to the participants. With this done, we explained the participant the actual procedure of the study. We showed them how a target looks like (see Figure 3.7) and told them to always hit the center of the target. The remaining lines were only for the participants to find the target faster. We told them to always keep the arm stretched out while pointing and to keep both eyes open. While conducting the study we did not allow the users to move with their feet but freely move everything else. We attached a small dot on the floor to the point in the middle of the projector screen in two meters distance in both RW and VR. The participants should always stand exactly on this point. Since we did not want to get a single pointing value per target but multiple values which we then could average out, we asked the participants to hold the pointing gesture for one second. After pointing on a target they should lower their arm totally next to their body to always start from the same position again. For controlling the study software we handed the participants a presenter to their left hand. With this presenter, they could click through the study. The procedure always started with a blank projector screen. After the participants clicked the first time on the presenter, the first target was shown. We told them to start pointing as soon as they see the target. When they were sure to hit the target we told them to press the presenter again and hold the pointing gesture for this one second. After this second the target on the projector screen disappeared and they should lower their arm again. With the next click on the presenter, the next target showed up on the projector screen. Since we did not want one single average value per target, we displayed every target

	Eye Dominance		
	L	R	A
Pipe Test	5	6	9
Target Test	7	10	3

Table 3.1: Results of eye dominance tests for data collection study.

six times in total, summing up to 210 targets per condition and 420 targets in total. Since we wanted to avoid any learning effects we randomized the order of the targets. To control for any fatigue effects, we asked the participants to fill out a raw NASA TLX after every quarter of the study. We showed them a text on the projector screen telling them to interrupt the study. During the whole study, we allowed the participants to take a break whenever they wanted. After they pointed at all 420 targets we removed all markers from the participants and thanked them for their participation. As the last step, we asked them for remaining questions about the study and answered earlier questions, which we could not answer before the study because of a possible influence on the study data.

3.4 Participants

For the data collection study, we invited students from our internal study volunteer mailing list. Overall we had 20 participants with a mean age of 22.1 and a standard deviation of 3.19. Our youngest participant was 17 the oldest 30. From this 20 participants, 16 were male and 4 female. We only had right-handed participants with no movement restrictions and no glasses. However, some of the participants were wearing contact lenses. Our participants were in average 175cm tall. For the results of both eye dominance test see Table 3.1.

3.5 Summary

In this chapter, we presented our approach to the data collection study and its study design. We showed which hardware we will use. Furthermore, we described our software with which we will process the data out of the hardware. Next of we presented the procedure of our first study and finally, we presented the demographic data of our participants for this first study.

4 Modeling

In this chapter, we present the process of evaluating our recorded data, the preparation and the build process of our models. We will show the preprocessing of our recorded data in a first step. Furthermore, we will then expound what was necessary for our method of training our models with different functions. We will finally compare these functions and then give a short discussion of the results in the end of this chapter.

4.1 Fatigue Effect

Since the study took about 45 to 60 minutes for only the pointing and about 60 to 90 minutes for the overall procedure, including the measurement of the participant, a big concern is the fatigue of the participants. This means, after a while, the participants could start getting tired and less accurate in their pointing. A second outcome of getting tired is that the participants could get less concentrated and start messing up the study procedure with for example clicking too early on the presenter and moving the arm back down to early before the target actually disappeared. To prevent an impact on our recorded data we decided to counterbalance the conditions. With this done the impact should be as small as possible. As a second step we conducted a raw NASA Task Load Index (raw NASA TLX) [Har06] as mentioned earlier. A one-way repeated measures analysis of variance (RM-ANOVA) was conducted to assess the participant's perceived workload. Since Mauchly's test indicated that the assumption of sphericity had been violated, $\chi^2(28.125) = 0.205, p < 0.001, \epsilon = 0.544$, we used Greenhouse-Geisser correction to adjust the degrees of freedom of the raw NASA TLX score. As the analysis did not reveal a significant effect, $F(1.623, 30.999) = 0.047, p = 0.927$, we assume that the effect of the four conditions on the participants' workload is negligible.

4.2 Preprocessing

As a first step, we calculated the actual intersection point of every single pointing gesture. For this, we just loaded all our saved movement data into a database. To get the relevant

	FTRC		FARC		HRC		EFRC	
	RW	VR	RW	VR	RW	VR	RW	VR
Count removed gestures	136	185	157	186	159	155	2	2
Percentage (%)	3.24	4.40	3.74	4.42	3.79	3.69	0.05	0.05

Table 4.1: Remove outliers for the different ray casting methods.

movement data out of the recorded raw movement data, we matched the time stamp of the presenter click with the time stamps of the raw movement data. But since we recorded the raw movement with about 70 frames per second (FPS) to 90 FPS and the participants had to hold the pointing gesture for one second, for every pointing gesture we had multiple data points. To get better results we did not use the movement data from the whole second. Most of the time the participants were still moving a little bit after clicking on the presenter. And after a while, the participants learned how long they had to hold the pointing gesture and they started to move their arm down again, even if the target was still on the projector screen. Even telling them to concentrate once again and really wait until the target really disappeared before they should lower their arm did not help much. Hence we decided to not use the first 100 ms and also not the last 100 ms for every pointing gesture. After retrieving all relevant movement data for a single pointing gesture we averaged the remaining values. With this averaged pointing gesture we could then calculate all necessary values. We choose to use four different ray casting methods for calculating the actual point, where the participants pointed at. These are the fingertip ray casting (FTRC), with has its root at the fingertip of the index finger and points in the same direction as the index finger, the forearm ray casting (FARC), which has its root at the bone inside the forearm at the elbow and points in the same direction than the forearm itself, the head ray casting (HRC), which has its root at the cyclops eye and points straight to the front from this point, and finally the cyclops eye fingertip ray casting (EFRC), which has its root once again on the cyclops eye but this time points in the direction of an imaginary line between the cyclops eye and the fingertip. For each of this ray casting methods we calculated the root point, the pointing direction, the angles between the pointing direction and an imaginary plane parallel to the projector screen at the point, where the participant was standing in both up-down direction and left-right direction, the angle between this plane and a line from the root of the pointing gesture to the actual target, the position on the projector screen of the actual target, the intersection of the ray casting method with the projector screen and the offset on the projector screen between the actual target and where the participant pointed at, both in x- and y-direction. With this values we calculated the error for each pointing gesture one time as an absolute offset on the projector screen, separated in x- and y-direction, and secondly as the angle between the line of the pointing gesture and the line of the

actual target, once again separated in the left-right direction and the up-down direction. With the absolute overall distance on the projector screen, we then removed outliers. We calculated the mean distance and the standard deviation of every single ray casting method and condition and removed data points where the distance was bigger than the average plus twice the standard deviation. For every ray casting method and condition, we have 4200 different data points in total. This value comes from 35 targets on the projector screen times six repetitions per target times 20 participants. For the number of removed values see Table 4.1.

4.3 Model Creation

After removing the outliers the actual data processing started. In a first step, we calculated every average point, where the participants actually pointed with the distance to where the actual target was. We repeated this separately for every ray casting method and every condition. We present the plotted results for all four ray casting methods in Figure 4.1a to Figure 4.1d. We repeated the same procedure but this time with the angles calculated from the imaginary parallel plane to the projector screen and the calculated ray cast. These angles are necessary because we want to get rid of the restraint of distance. In our study, the participants were only standing in two meters distance to the projector screen. Hence if we would train a model to only the absolute distances on the projector screen it is only valid for users standing in the exact same distance to the projector screen, thus two meters, like our participants did in our study. With using angles instead the distance does not play any role anymore. This is because if standing further away from the projector screen, the angle between the imaginary parallel plane and your arm will just get smaller if you want to point at the same target. The models will then just return a different value independently from the distance to the projector screen. For better readability, we will continue from now on to present most of the result values in centimeters and meters.

The average distances of the targets and the averaged pointing gestures are presented in Table 4.2. We can directly see, that at least without any correction the EFRC method performs the best. We have an average distance of 10.21cm in RW and 8.45cm for VR. This is less than the half of the next best ray casting method, which is the standard FTRC with 29.15cm in RW and 27.67cm in VR.

The next thing you can see in this data is, that there is a difference in the offsets between RW and VR. For FTRC with an average difference of 1.48cm, for FARC a difference of 9.33cm, for HRC a difference of 9.37cm and finally for FTRC a difference of 1.76cm. Since there is a difference between RW and VR we decided to use a separate model for both conditions. For creating the models we used a function for both x-direction

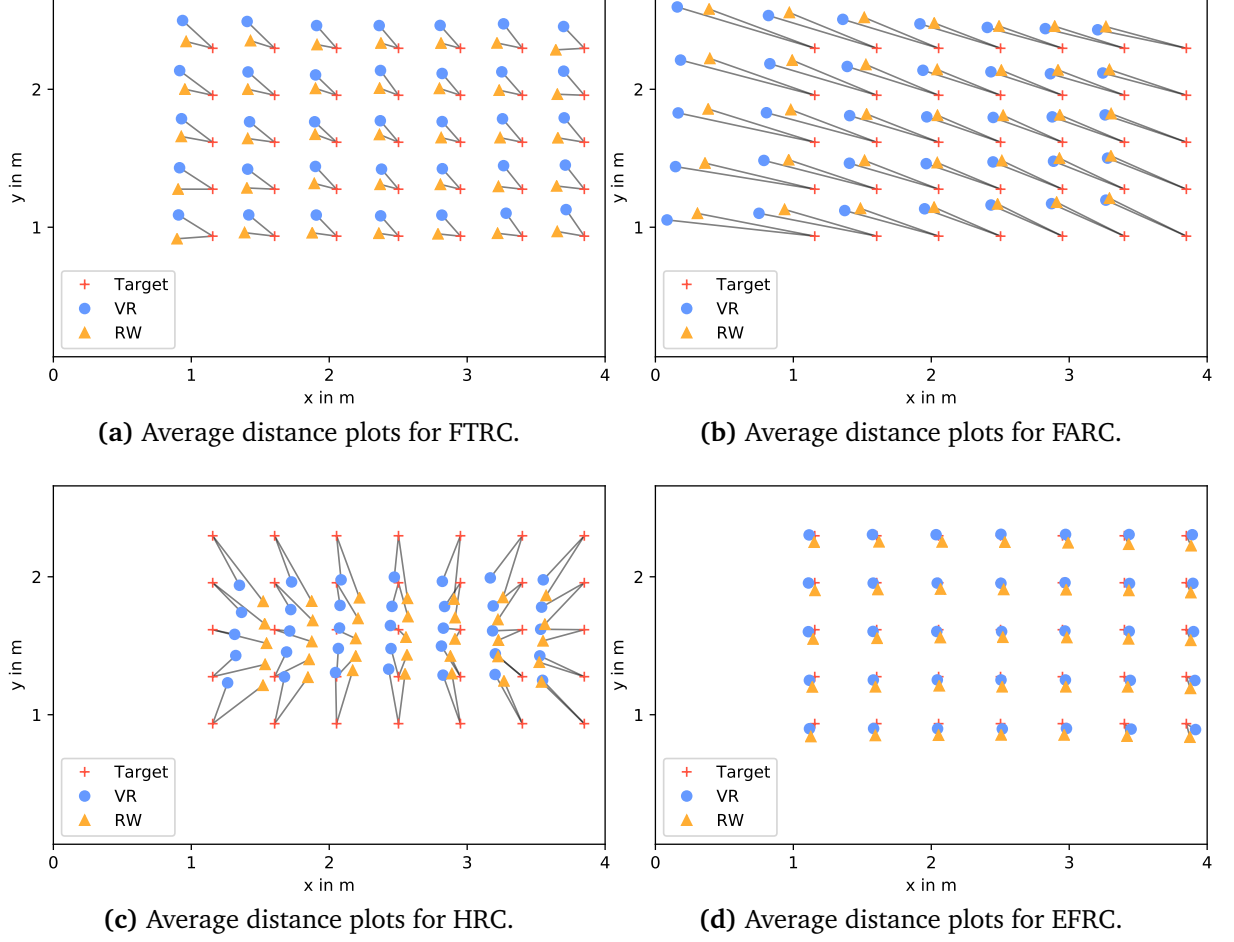


Figure 4.1: Different plots of offset between pointing gesture and actual target.

(left-right) and y-direction (up-down). We trained both of those functions separately on the according error. Like mentioned earlier we used the offset of angles in degrees for these functions. As functions we used a straight line f_1 which only corrects with given one direction as input, a plane f_2 which corrects with given both the x and y offset, a full function of second degree f_3 and finally a function of fourth degree f_4 but not with all coefficients. The function f_4 is the same like the function Mayer et. al. [MWSH15] used. We trained all those four functions on the same set of offsets.

$$f_1 : z = a + bx \quad (4.1)$$

$$f_2 : z = a + bx + cy \quad (4.2)$$

Ray cast	RW		VR	
	M	SD	M	SD
FTRC	29.15	19.40	27.67	19.06
FARC	60.33	20.16	69.66	25.42
HRC	47.14	23.71	37.77	19.19
EFRC	10.21	5.56	8.45	4.54

Table 4.2: Absolute distances in centimeters for fingertip and forearm on the projector screen.

$$f_3 : z = a + bx + cy + dxy + ex^2 + fy^2 + gx^2y^2 \quad (4.3)$$

$$f_4 : z = ax^4 + by^4 + cx^3y + dxy^3 + ex^3 + fy^3 + gx^2y^2 + hx^2y + ixy^2 + jx^2 + ky^2 + lxy + mx + ny + o \quad (4.4)$$

We took all those four functions and optimized the coefficients to the smallest error. As a starting point, we used zeros for all coefficients.

4.4 Cross Validation

For validating the created models we used a leave one out cross validation (LOOCV). This cross validation takes all participants and leaves one of those participants out. It then trains the model on the rest of the participants and uses the created model on the left out one. It repeats this procedure separately for every participant. We used the average remaining offset after all steps after correcting the single left out participant with the created model as the measurement how good the model performs. We present the results after cross validation for all four different functions in Table 4.3.

The cross validation shows, that in almost every case we have an improvement in remaining offset and in average function f_4 works the best. That is the same result, like Mayer et. al. [MWSH15] found out. With those results, we decided to only use function f_4 in the further process.

	FTRC		FARC		EFRC		HRC	
	RW	VR	RW	VR	RW	VR	RW	VR
f_1	15.62	13.45	28.23	31.79	8.07	8.37	46.80	38.30
f_2	15.28	13.14	27.53	30.76	8.07	8.39	46.88	38.19
f_3	15.21	13.00	27.07	30.03	8.05	8.37	46.80	38.11
f_4	15.17	12.90	27.03	30.18	8.02	8.41	46.92	37.77

Table 4.3: Absolute offsets after cross validation in centimeters.

4.5 Summary

In the preceding chapter, we described the preprocessing of our raw recorded data from the data collection study. We furthermore described which effects the fatigue can have on the results and that based on our raw NASA TLX questionnaires we assume that the effect is negligible. Next off, we described how we built our models from the preprocessed data. Finally, we expound our method for validating the models.

5 Evaluation Study

The main aim of the evaluation study was to test and verify our built models. Secondly, we wanted to investigate how participants can perform when they are provided with a visual feedback both in RW and VR. We wanted to find out if our participants can get significantly faster or more accurate if they are provided with a visual feedback like a mouse cursor on a computer. Since the participants do not have to think of where they are pointing anymore but instead get an immediate feedback, where they are actually pointing at, they might just start to use this feedback and control it instead of just pointing. Thanks to this, at least the accuracy might get better. On the other hand, this pointing with a visual feedback might be too far away from a natural pointing. In our models, we have chosen the best working ray casting method regarding the minimum absolute remaining error after correction. A possible problem of our models combined with the visual feedback might be that the task completion time increases. Our models are only averaged what might lead to participants starting with their natural pointing, then recognize, that the cross is not exactly where they want to point at and then start to correct the position of the cross. We plan to investigate this behavior with this second study.

5.1 Study Design

We kept the study design as close to the first study as possible. Our independent variables was once again the setting, this time expanded by the visual feedback and the correction model. As dependent variables, we once again had the same points on our participants like in the data collection study. The independent variables resulted in eight different conditions (see Table 5.1). Since we did not provide the participants with any absolute feedback while no visual feedback was provided, like a distance error after every pointing gesture, we were able to reduce the conditions while recording to six different. We were able to do this with recording both with and without model while not providing any visual feedback. We shuffled these six remaining conditions to get rid of any possible learning effect. We used the same 35 targets on the projector screen like in the first study. Combined with two repetitions per condition this time and six different conditions, we once again had 420 targets in total per participant.

with feedback				without feedback			
with model		without model		with model		without model	
RW	VR	RW	VR	RW	VR	RW	VR

Table 5.1: All different conditions for the evaluation study.

While creating the models for this study we used four different ray casting methods, the FTRC, the FARC, the HRC and the EFRC. Since the absolute remaining error after correction for EFRC is the smallest of all four ray casting methods (see chapter 4 – Modeling) we decided to use only this ray casting method.

5.2 Apparatus

Since we wanted to get comparable data to the data collection study we used the same software like in the data collection study only with some modifications. We did not change anything about the room setup (see section 3.2.2 – Room Setup). Hence the user was again standing on a point two meters in front of the middle of the projector screen and once again not directly on the floor but on a podium with a height of twelve centimeters.

Since we were using rigid bodies for tracking the movement of the participant and those rigid bodies do not need any post labeling we were able to apply our correction in real-time on the tracking data. We achieved this by implementing the model function, which we created earlier, in our software for the second study in the Unity3D Engine. These functions are able to apply the correction value in real time on the tracked data.

As visual feedback, we used a cross on the projector screen (see Figure 5.1). Since we were only using the EFRC method, the cross was always bound to this ray casting method. But we did not tell the participants how the cross was actually working and which ray casting method we were using to not influence the participants in their natural way of pointing and to prevent any effects, where the participants would start to no longer point but just use their knowledge about the cross and start controlling it. This might, for example, result in a pointing gesture, where the participants would no longer stretch their arm while pointing but just keep the fingertip in front of their face and point in this way.

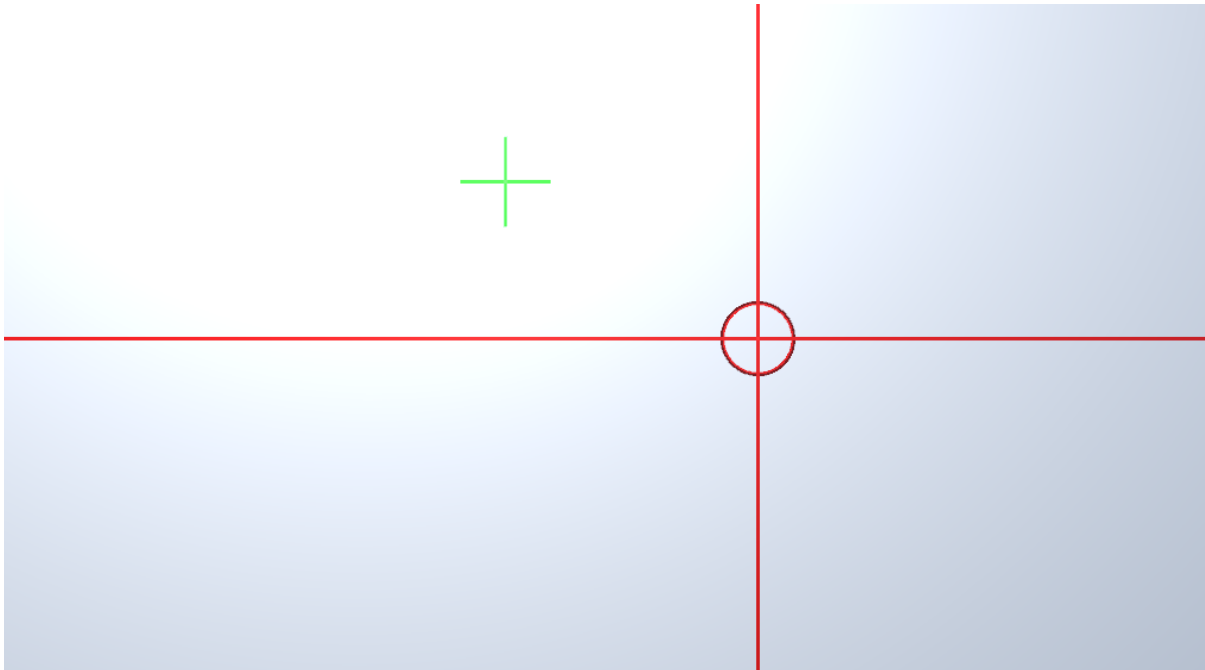


Figure 5.1: A target on the projector screen with visual feedback.

5.3 Procedure

We kept the study procedure of this second study as close as possible to the procedure of the data collection study (see Chapter 3 – Data Collection Study). For the second study, we once again restricted our participants to right-handed people with no glasses and no movement restrictions in the right arm. At the beginning of the second study, we once again welcomed the participants and asked them to read through the consent form and sign it, if everything was ok. The consent form was the same form like in the first study. After signing the consent form we asked the participants to ask every question they had and then we started to explain to them what the study was about but without telling them that we are using our models sometimes in the background. We only told them that sometimes there will be a visual feedback in form of a cross on the projector screen and if there is a feedback that they should also use it. Next of we measured all the necessary lengths of the participants, which were the upper arm length, the forearm length, the hand length, the index finger length, the circumference of the upper arm at the shoulder, the circumference of the elbow and the wrist, the circumference of the index finger, the shoulder height, the head circumference and the eye height. The software used these values just like the software from the first data collection study (see section 3.2.3 – Software). After measuring all the lengths we once again conducted the eye dominance tests, both the pipe dominance test and the target dominance test. After attaching all markers to the same position like in the first study, we gave the participants

some time to get into the study scene and procedure. We showed the participants the RW and VR scenes and described them how they could go through the study. Like in the first study we gave them a presenter to the left hand and clicking on either of the buttons continued the study. This time we had a small delay between the click for showing the next target and when the target showed up. This delay was randomized between 0.5 and 1.0 second. In addition to the normal tryout of the study procedure we also showed them the visual feedback without telling them how the cross was actual working, and let them try out the visual feedback in both RW and VR. After the participants felt comfortable with the whole procedure we told them, that they are allowed to interrupt or cancel the study at any time, especially if they feel uncomfortable with VR and that they can take a break whenever they want. Secondly, we told them, like in the first study, that they should always stand on the center point and not move with their feet but furthermore, they should point as natural as possible. Once again the only restriction was a stretched arm while pointing. Finally, we told them to be as fast and accurate as possible.

During the study after every of the six conditions, we asked the participants to have a small break and we let them fill out the raw NASA TLX once again to control for a possible fatigue effect. As a second effect of this small break, the participants did not carry anything they learned in an earlier condition to the next one.

After the participants finished all six conditions and filled out the last questionnaire, we removed all rigid body markers from the participant. Now that we had finished the study, we could answer any remaining questions without influencing anything in the results. After answering everything we finally thanked the participants for their participation.

5.4 Participants

We once again invited participants over the mailing list of study voluntaries. Just like for the first study parts of the students in this list have to participate in a study to get admitted to the final exams of their lecture. This time we also had some other persons since not enough students were able to come. Those other persons were compensated for their time with ten Euro. In total, we had 16 participants for the second study, 15 male, and one female. The participants had a mean age of 22.69 with a standard deviation of 1.78 years. The youngest participant was 19, the oldest 26. We once again had, like in the first study, only right-handed participants with no movement restrictions and no one was wearing glasses. The mean body size of the participants was 170.36cm with a standard deviation of 7.09cm. The smallest participant was 155.94cm tall and the biggest was 181.47cm tall. We finally present the results of both eye dominance test for the evaluation study in Table 5.2.

	Eye Dominance		
	L	R	A
Pipe Test	2	7	7
Target Test	4	11	1

Table 5.2: Results of eye dominance tests for evaluation study.

5.5 Summary

This chapter dealt with our evaluation study. We presented the design for it and furthermore showed which parts of the data collection study including hardware, software, and procedure we took over to the evaluation study and which parts we had to change. Finally, we presented our participants of this second study since we did not take the same persons like in the data collection study.

6 Results

In this chapter, we will present the results of the evaluation study. We will show the effects of our model on a new set of participants. Besides this, we will present how the visual feedback in form of a cursor influences pointing. This time for this study we will also evaluate the effects of our model and the visual feedback on the TCT.

6.1 Fatigue Effect

Just like in the first study we conducted a raw raw NASA TLX to find out if there are any fatigue effects over time by the participants. We present the mean values of the raw NASA TLX in Table 6.1.

We once again conducted a one-way repeated measurement analysis of variace (RM-ANOVA) to find out if there are any significant differences in the task load over time. The RM-ANOVA resulted in following values, $F_{5,15} = 0.654$ and $p = 0.659$. Hence there is no significant difference over time. So we once again assume, that the fatigue effect on the participants for the second study is negligible.

6.2 Accuracy

We present the mean offsets with the corresponding standard deviations in Table 6.2.

Just like in the first study, we conducted a RM-ANOVA. As conditions we had the setting (RW and VR), the visual feedback and the correction model. The RM-ANOVA for the

First		Second		Third		Fourth		Fifth		Sixth	
M	SD	M	SD	M	SD	M	SD	M	SD	M	SD
36.25	11.37	38.28	13.46	39.84	15.10	37.81	16.32	39.74	15.73	37.76	17.68

Table 6.1: All raw NASA TLX scores from the second study in range 0 to 100.

Feedback	Correction	RW		VR	
		M	SD	M	SD
<i>False</i>	<i>False</i>	6.89	3.08	6.35	3.40
<i>False</i>	<i>True</i>	5.71	3.04	5.74	3.23
<i>True</i>	<i>False</i>	1.19	1.95	1.29	0.82
<i>True</i>	<i>True</i>	1.13	0.91	1.13	0.66

Table 6.2: Remaining offsets between intersection and actual target in cm for second study.

setting resulted in following values $F_{1,15} = 1.3$ and $p = 0.027$, hence we have a significant difference between RW and VR. The same thing happens for the visual feedback. The RM-ANOVA resulted in following values $F_{1,15} = 131.9$ and $p < 0.001$, so we once again have a significant difference between with and without visual feedback. Finally for the last condition, the correction model we got following values $F_{1,15} = 5.321$ and $p = 0.027$, so we also have a significant difference for the correction model. For visual feedback \times setting the RM-ANOVA resulted in the values $F_{1,15} = 15.61$ and $p < 0.001$, hence in a significant difference. On the other side we have no significant difference in correction model \times visual feedback, with $F_{1,15} = 0.067$ and $p = 0.799$, and correction model \times setting with $F_{1,15} = 1.291$ and $p = 0.274$. And also for setting \times visual feedback \times correction model, with the results $F_{1,15} = 1.163$ and $p = 0.298$, we do not have a significant difference.

6.3 Task Completion Time

We present the average TCTs with corresponding standard deviations in Table 6.3.

We once again conducted multiple RM-ANOVAs. Just like with the offsets we, have the conditions setting, visual feedback, and the correction model. The RM-ANOVAs for setting and correction model resulted in no significant difference with values $F_{1,15} = 0.004$ and $p = 0.956$ for setting and $F_{1,15} = 0.158$ and $p = 0.697$ for the correction model. However the visual feedback resulted in $F_{1,15} = 7.834$ and $p = 0.013$ hence in a significant difference. For the combinations we found no significance for setting \times correction model ($F_{1,15} = 1.291, p = 0.274$), visual feedback \times correction model ($F_{1,15} = 0.067, p = 0.799$) and setting \times visual feedback \times correction model ($F_{1,15} = 1.163, p = 0.298$). Only setting \times visual feedback resulted in a significant difference with $F_{1,15} = 15.61$ and $p < 0.001$.

Feedback	Correction	RW		VR	
		M	SD	M	SD
<i>False</i>	<i>False</i>	1.48	0.43	1.64	0.61
<i>False</i>	<i>True</i>	1.48	0.43	1.63	0.61
<i>True</i>	<i>False</i>	1.83	0.43	1.76	0.56
<i>True</i>	<i>True</i>	1.89	0.73	1.68	0.45

Table 6.3: TCT in seconds for second study.

These results show, that neither the setting nor the correction model has a significant influence on the TCT. Only the visual feedback influences the time significantly.

6.4 Summary

In this chapter, we presented the results of our evaluation study. We showed that our model significantly increases the accuracy of pointing gestures. That applies in both RW and VR. To get even better results using a visual feedback, in our case a cursor on the projector screen, is a good way. The visual feedback once again increases the accuracy significantly. However like shown in the TCT chapter, the visual feedback increases the TCT significantly. Hence we suggest using pointing without visual feedback in cases where a perfect accuracy is not necessary, like pointing at big objects, and activate a cursor if the user should point at small areas where accuracy is important. This results in a good balance between fast pointing at lower accuracy and accurate interactions if necessary.

7 Discussion

This chapter will try to give an overall summary and of the results of both our studies. It will furthermore discuss those results and their implications. We will furthermore try to give some suggestions on how to use our results. Finally, we will present some limitations of our studies and the created models.

7.1 Models

We found a significant difference in the data collection study between RW and VR. We assume that this is due to the HMD. Currently all available HMDs are limited in their field of view (FOV). Even that they have developed to a FOV of about 110° , this is still less than the normal FOV of a human eye. An average human without any seeing limitations has a FOV of about 200° [Doh07]. We also did use a HMD with only 110° FOV. This results in more head movement in VR since the participants are limited in their peripheral vision. Due to this they cannot view all targets on the border of their FOV with just eye movement but they have to move their head around. Since we use the cyclops eye in our ray cast calculation, more movement in the head results in a different model for the pointing gestures. Another downside of current HMDs are the cables. So far there is no wireless hardware on the market capable of transferring all image data without restrictions wireless to the headset. Those cables and the HMD itself result in different movement of the participant with their head while wearing the headset. We assume that this once again results in a different calculation of our ray casts. Since we have a difference between RW and VR we suggest using different models for both settings.

Nevertheless, with different models for RW and VR we were able to achieve significantly better results for both settings with models than without those models.

We created our models on one of the best tracking systems currently on the market. This setup requires multiple hours of installation and calibration and thus it is almost impossible to use this setup without further training. But when calibrated in this way it delivers perfect tracking results. Hence we can ensure, that all offsets we recorded with this setup, are actually offsets in the pointing gesture and not offsets from an inaccuracy

of our hardware setup. Thus, our models, which are trained on this offset, represent the actual pointing offset nearly without any influence of the tracking hardware.

7.2 Task Completion Time

Considering the TCT values in Table 6.3 we learned that the visual feedback increases the time the participants took for completing a single target in average. We assume this comes from the participants no longer pointing at the target in their natural way of pointing but now, since they are provided with a visual feedback in form of a cursor, try to point with the visual feedback. Since this feedback is always adapting to the current pointing gesture in real time, the participants can start to just control the cursor and try to align it with the target. This alignment took more time than without a cursor. Hence we suggest using the visual feedback only if TCT is no concern and in situations where a nearly perfect accuracy is necessary. The model instead had no significant influence on the TCT even though the TCTs were a little bit smaller with models while providing a visual feedback.

7.3 Limitations

In our studies, we restricted us in different ways. In the selection of participants, we restricted us to only right-handed people. This restricts our model to also only right-handed people. Furthermore, our participants did not wear glasses at all. This is due to our HMD. Wearing glasses under the HMD is not always possible. But since it is a problem related to the hardware, we can not change anything about it.

Another limitation of our models is the fact, that we only created them on data recorded with targets on our projector screen. Thus we are limited with our models to possible targets in the same window in front of the users like our projector screen represented. Outside this area, our models do no longer represent useful correction values but on the contrary, make the recorded gestures even worse and destroy them pretty soon over the border of this area. This can be easily overcome by just use our method of creating the models but record the data with targets in the necessary area for the given use case.

In our studies, we also did limit ourself to only a simple pointing task, were our participants had to point at single targets. Thus our results only represent this task. It might be interesting to use the models on a different kind of task more related to the everyday life. The accuracy might not decrease but the TCT might change.

7.4 Summary

In this chapter, we discussed our results from the overall thesis. We presented the results of both our studies in detail and our approach of interpreting the individual outcomes. Furthermore, we presented some advice on how to use our models. Finally, we presented some limitation which we identified during our studies, some related to the hardware and some other related to our study design.

8 Conclusion

This thesis dealt with human mid-air distant pointing without any additional pointing devices. At the beginning, as a first step, we introduced our main ideas and goals within this thesis. As a next step, we searched for similar works in research which have already been conducted by different other people. Though this related work we identified different studies with similar goals and problems they had to overcome during their work. Finally, we came to a conclusion where our work should start and from which studies it should build upon. The next step was to introduce our approach for the first study for recording natural ground-truth pointing gestures. We presented our used hardware and which new parts of the software we had to develop for ourself. We furthermore presented our study design and how we planed the procedure around the study. After conducting this study we started with the evaluation of the recorded gestures. This included a first step of removing outliers and calculating the offsets with different ray casting methods. With those offsets we the trained multiple corrections models. For verifying those models we carried out a LOOCV. With the best ray casting method, namely the EFRC, we then continued the second study. This study had the aim to test our models with new participants and to achieve a comparison of pointing with and without visual feedback. As feedback, we used a cross on the projector screen bound to the ray casting method. With the data from this second study, we were then able to verify our models and also accomplish the wanted comparison.

After both studies and both evaluations of our recorded data, we were then able to reproduce the results of older research works. We were able to reproduce the offset between targets and the recognized pointing gesture in RW of older research. Furthermore we were able to reproduce this offset also in VR. With those offsets we were able to prove a difference in pointing between RW and VR. Due to this, we created two different models, one for RW and one for VR. We were then able to show, that pointing with those models increases the accuracy significantly without increasing the TCT. Besides this, we were able to show, that a visual feedback increases the pointing accuracy again significantly. Nevertheless, on the other hand, this visual feedback also increases the TCT significantly.

For the future, it would be interesting to go deeper in the process of the model creation to achieve better fitting models. Our models currently only take into account the angles

of the pointing direction. We could try to refine those models by providing them with more information about the participants like for example the eye dominance. Another step would be to test our models with left-handed participants and investigate if we would need different models for different handedness or if it is possible to create an average model for both.

Furthermore, it might be interesting to try out our models in real-world tasks. The first step towards this would be to create models with a full coverage of 360° in both up-down direction and left-right direction. With these models it would be possible to replace the current state-of-the-art controllers shipped with VR-headsets. Those controllers interfere with a natural interaction with the virtual environment. With our models we present a possibility of using the bare hands as controllers and therefore preserve the natural way of interaction. On the other hand, a possible real-world task could then be something in the field of home automation. Home automation is a constantly growing market. More and more household switch over to connected devices which can be controlled wireless with remote controls. The next step here would be to get rid of those controls and use pointing gesture alone or in combination with speech input to control those home automation devices. In this scenario, it would not be necessary to get a single target point but just to identify an object. Another scenario where a single target would be necessary are big computer displays or projector screen. This kind of hardware is already used in many offices but is still pretty hard to control via standard input methods like a mouse. Combining those standard methods with pointing gestures for controlling the mouse cursor might result in faster working.

As a next study, we might try out, how our models can perform on different hardware. In this work, we recorded pointing gestures in a first study with the best tracking system currently on the market. We simplified the target calculation by not taking into account the whole body posture but only two different points on the user's body to calculate the target with a ray cast. In the best working method, the EFRC, these points were the cyclops eye and the fingertip. Hence we do not need a full body tracking for the target calculation. This opens the possibility of using different tracking hardware, like for example the Microsoft Kinect sensor¹. This sensor is way cheaper and used in both the consumer sector and research. Since we only need two points on the user's body, we do not have to rely on the bone models of the Kinect sensor, we only need to track these two points. In the study, we could try our current models on this sensor and also compare them with newly created models with ground truth pointing gestures recorded with the Kinect sensor. This would offer the possibility of using pointing gestures to a wide spread of people.

¹<https://developer.microsoft.com/en-us/windows/kinect>

A Appendix

Figure 8.6

NASA Task Load Index

Hart and Staveland's NASA Task Load Index (TLX) method assesses work load on five 7-point scales. Increments of high, medium and low estimates for each point result in 21 gradations on the scales.

Name	Task	Date

Mental Demand How mentally demanding was the task?

Very Low Very High

Physical Demand How physically demanding was the task?

Very Low Very High

Temporal Demand How hurried or rushed was the pace of the task?

Very Low Very High

Performance How successful were you in accomplishing what you were asked to do?

Perfect Failure

Effort How hard did you have to work to accomplish your level of performance?

Very Low Very High

Frustration How insecure, discouraged, irritated, stressed, and annoyed were you?

Very Low Very High

Figure A.1: The raw NASA TLX questionnaire.

Bibliography

- [AA09] F. Argelaguet, C. Andujar. “Visual Feedback Techniques for Virtual Pointing on Stereoscopic Displays.” In: *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology*. VRST '09. Kyoto, Japan: ACM, 2009, pp. 163–170. ISBN: 978-1-60558-869-8. DOI: [10.1145/1643928.1643966](https://doi.org/10.1145/1643928.1643966) (cit. on p. 21).
- [Bol80] R. A. Bolt. ““Put-that-there”: Voice and Gesture at the Graphics Interface.” In: *Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '80. Seattle, Washington, USA: ACM, 1980, pp. 262–270. ISBN: 0-89791-021-4. DOI: [10.1145/800250.807503](https://doi.org/10.1145/800250.807503) (cit. on pp. 17, 20).
- [CC02] A. Corradini, P. R. Cohen. “Multimodal speech-gesture interface for hand-free painting on a virtual paper using partial recurrent neural networks as gesture recognizer.” In: *Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on*. Vol. 3. 2002, pp. 2293–2298. DOI: [10.1109/IJCNN.2002.1007499](https://doi.org/10.1109/IJCNN.2002.1007499) (cit. on p. 21).
- [Doh07] K. K. C. Dohse. *Effects of field of view and stereo graphics on memory in immersive command and control*. Iowa State University, 2007 (cit. on p. 51).
- [Har06] S. G. Hart. “Nasa-Task Load Index (NASA-TLX); 20 Years Later.” In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 50.9 (2006), pp. 904–908. DOI: [10.1177/154193120605000909](https://doi.org/10.1177/154193120605000909) (cit. on p. 35).
- [Hav03] J. B. Haviland. “How to point in Zinacantán.” In: *Pointing: Where language, culture, and cognition meet*. Ed. by S. Kita. Psychology Press, 2003 (cit. on p. 17).
- [JBM+00] N. Jojic, B. Brumitt, B. Meyers, S. Harris, T. Huang. “Detection and estimation of pointing gestures in dense disparity maps.” In: *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*. 2000, pp. 468–475. DOI: [10.1109/AFGR.2000.840676](https://doi.org/10.1109/AFGR.2000.840676) (cit. on p. 21).

- [JOMS06] H. Jiang, E. Ofek, N. Moraveji, Y. Shi. “Direct Pointer: Direct Manipulation for Large-display Interaction Using Handheld Cameras.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’06. Montreal, Quebec, Canada: ACM, 2006, pp. 1107–1110. ISBN: 1-59593-372-7. DOI: [10.1145/1124772.1124937](https://doi.org/10.1145/1124772.1124937) (cit. on p. 20).
- [KBSM10] R. Kopper, D. A. Bowman, M. G. Silva, R. P. McMahan. “A Human Motor Behavior Model for Distal Pointing Tasks.” In: *Int. J. Hum.-Comput. Stud.* 68.10 (Oct. 2010), pp. 603–615. ISSN: 1071-5819. DOI: [10.1016/j.ijhcs.2010.05.001](https://doi.org/10.1016/j.ijhcs.2010.05.001) (cit. on pp. 17, 19).
- [KSMB08] R. Kopper, M. G. Silva, R. P. McMahan, D. A. Bowman. “Increasing the Precision of Distant Pointing for Large High-Resolution Displays.” In: (2008) (cit. on pp. 17, 19, 20).
- [MCM09] V. McArthur, S. J. Castellucci, I. S. MacKenzie. “An Empirical Comparison of “Wiimote” Gun Attachments for Pointing Tasks.” In: *Proceedings of the 1st ACM SIGCHI Symposium on Engineering Interactive Computing Systems*. EICS ’09. Pittsburgh, PA, USA: ACM, 2009, pp. 203–208. ISBN: 978-1-60558-600-7. DOI: [10.1145/1570433.1570471](https://doi.org/10.1145/1570433.1570471) (cit. on pp. 18, 19).
- [McN92] D. McNeill. *Hand and Mind: What Gestures Reveal about Thought*. University of Chicago Press, 1992, 1992. ISBN: 0226561321 (cit. on p. 17).
- [Min95] M. R. Mine. *Virtual Environment Interaction Techniques*. Tech. rep. Chapel Hill, NC, USA, 1995 (cit. on p. 21).
- [MWSH15] S. Mayer, K. Wolf, S. Schneegass, N. Henze. “Modeling Distant Pointing for Compensating Systematic Displacements.” In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. CHI ’15. Seoul, Republic of Korea: ACM, 2015, pp. 4165–4168. ISBN: 978-1-4503-3145-6. DOI: [10.1145/2702123.2702332](https://doi.org/10.1145/2702123.2702332) (cit. on pp. 22, 38, 39).
- [NS03] K. Nickel, R. Stiefelhagen. “Pointing Gesture Recognition Based on 3D-tracking of Face, Hands and Head Orientation.” In: *Proceedings of the 5th International Conference on Multimodal Interfaces*. ICMI ’03. Vancouver, British Columbia, Canada: ACM, 2003, pp. 140–146. ISBN: 1-58113-621-8. DOI: [10.1145/958432.958460](https://doi.org/10.1145/958432.958460) (cit. on p. 21).
- [ON01] D. R. Olsen Jr., T. Nielsen. “Laser Pointer Interaction.” In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’01. Seattle, Washington, USA: ACM, 2001, pp. 17–22. ISBN: 1-58113-327-8. DOI: [10.1145/365024.365030](https://doi.org/10.1145/365024.365030) (cit. on pp. 18, 21).

- [PFC+97] J. S. Pierce, A. S. Forsberg, M. J. Conway, S. Hong, R. C. Zeleznik, M. R. Mine. "Image Plane Interaction Techniques in 3D Immersive Environments." In: *Proceedings of the 1997 Symposium on Interactive 3D Graphics*. I3D '97. Providence, Rhode Island, USA: ACM, 1997, 39–ff. ISBN: 0-89791-884-3. DOI: [10.1145/253284.253303](https://doi.org/10.1145/253284.253303) (cit. on p. 21).
- [PKL06] T. Pfeiffer, A. Kranstedt, A. Lücking. "Sprach-Gestik Experimente mit IADE, dem Interactive Augmented Data Explorer." In: *Dritter Workshop Virtuelle und Erweiterte Realität der GI-Fachgruppe VR/AR*. Ed. by S. Müller, G. Zachmann. Shaker, 2006, pp. 61–72 (cit. on p. 18).
- [PLW08] T. Pfeiffer, M. E. Latoschik, I. Wachsmuth. "Conversational Pointing Gestures for Virtual Reality Interaction: Implications from an Empirical Study." In: *2008 IEEE Virtual Reality Conference*. Mar. 2008, pp. 281–282. DOI: [10.1109/VR.2008.4480801](https://doi.org/10.1109/VR.2008.4480801) (cit. on pp. 19, 20).
- [RRK97] C. N. Riviere, R. S. Rader, P. K. Khosla. "Characteristics of hand motion of eye surgeons." In: *Proceedings of the 19th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. Vol. 4. Oct. 1997, 1690–1693 vol.4. DOI: [10.1109/IEMBS.1997.757046](https://doi.org/10.1109/IEMBS.1997.757046) (cit. on p. 21).
- [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*. CHI '17. Denver, Colorado, USA: ACM, 2017, pp. 1577–1582. ISBN: 978-1-4503-4655-9. DOI: [10.1145/3025453.3025602](https://doi.org/10.1145/3025453.3025602) (cit. on p. 18).
- [VB05] D. Vogel, R. Balakrishnan. "Distant Freehand Pointing and Clicking on Very Large, High Resolution Displays." In: *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*. UIST '05. Seattle, WA, USA: ACM, 2005, pp. 33–42. ISBN: 1-59593-271-2. DOI: [10.1145/1095034.1095041](https://doi.org/10.1145/1095034.1095041) (cit. on p. 20).
- [WG14] N. Wong, C. Gutwin. "Support for Deictic Pointing in CVEs: Still Fragmented After All These Years'." In: *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work and Social Computing*. CSCW '14. Baltimore, Maryland, USA: ACM, 2014, pp. 1377–1387. ISBN: 978-1-4503-2540-0. DOI: [10.1145/2531602.2531691](https://doi.org/10.1145/2531602.2531691) (cit. on p. 21).
- [ZMI99] S. Zhai, C. Morimoto, S. Ihde. "Manual and Gaze Input Cascaded (MAGIC) Pointing." In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '99. Pittsburgh, Pennsylvania, USA: ACM, 1999, pp. 246–253. DOI: [10.1145/302979.303053](https://doi.org/10.1145/302979.303053) (cit. on p. 20).

All links were last followed on September 28, 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