

Institute for Visualization and Interactive Systems

University of Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Diplomarbeit Nr. 3606

Modeling distant pointing for compensating systematic displacements

Sven André Mayer

Course of Study: Computer Science

Examiner: Jun.-Prof. Dr. Niels Henze

Supervisor: Stefan Schneegass M.Sc.

Commenced: 2013-12-18

Completed: 2014-06-19

CR-Classification: H.5.2, I.3.6, I.6.5

Abstract

People use gestures to give verbal communication more expression and also to replace the speech. One of the most concise and expressive gestures is the pointing gesture. Pointing gesture can be observed in the early childhood. In these early years, it is used to point at objects or people. Later, people use pointing gestures even for more complex things such as to visualize directions. Increasing pointing gestures are also used for interacting with computers. For example, gestures can be used to remotely interact with a display, without using an input tool. In this work we investigated how people point to objects and how the recognition accuracy can be improved by using a gesture recognition system. We performed a user study, where participants had to point on projected pointing targets. These gestures were recorded as reference data with the help of motion capture system. We used a variety of starting positions in which the study was carried out. For this the participants were placed with a distance of 2 to 3 meters to the pointing targets. At these two positions the participants had to sit and stand while pointing to the targets. From the recorded reference data we derived a pointing vector. Each vector describes the direction in which the gesture is directed. We generate these vectors out of different body parts. This is done to show that there are different ways to create these vectors but they behave all the same. In the optimal case, this vector would describe the path of the person pointing to the object, in this case, the projected point. By mathematical analyzes we show that in average over several experiments and over several participants a systematic deviation from this optimal vector can be detected. We specify models, which can compensate the systematic deviation. These models shift the pointing vector in the direction of the average distance between optimum and average study vector. Products of the consumer market can be used to detect pointing gestures. There gestures can be improved with the generated models. The focus here is, for example, products like the Kinect.

Kurzfassung

Menschen nutzen Gesten, um mündlicher Kommunikation mehr Ausdruck zu verleihen oder auch zu ersetzen. Eine der prägnantesten und ausdrucksstärksten Gesten ist die Zeigegeste. Diese lässt sich schon in jungen Jahren bei Kindern beobachten. In diesen jungen Jahren wird sie verwendet, um auf Gegenstände oder Personen zu deuten. Später nutzen Menschen Zeigegesten auch für komplexere Dinge wie z.B. um Wegbeschreibungen zu visualisieren. Zunehmenden werden Zeigegesten auch für die Interaktion mit Computern verwendet. Beispielsweise werden Zeigegesten verwendet, um aus der Ferne eine Interaktion mit einem Display zu realisieren, ohne ein Eingabegerät zu benutzen. In dieser Arbeit wird untersucht wie Menschen auf Objekte zeigen und wie die Erkennungsgenauigkeit durch ein interaktives System verbessert werden kann. In einer Benutzerstudie, in der Probanden auf projizierte Punkte zeigen mussten, wurden Referenzdaten aufgezeichnet. Damit präzise Daten aufgezeichnet wurden, kam ein Motion-Capture-System zum Einsatz. Eine Vielfalt von Ausgangspositionen wurde erzeugt, in dem die Studie mit einem Abstand von 2 und 3 Metern zu den Zielpunkten durchgeführt wurde. Von diesen zwei Positionen mussten die Teilnehmer im Sitzen und im Stehen auf die Punkte zeigen. Aus den aufgezeichneten Referenzdaten werden Mengen von Vektor abgeleitet. Jeder Vektor beschreibt die Richtung in welche die Geste gerichtet ist. Die Vektormengen werden jeweils aus der Ausrichtung unterschiedlicher Gliedmaßen abgeleitet. Dies geschieht, um zu zeigen, dass es verschiedene Möglichkeiten gibt, diese Vektoren zu erstellen, sich jedoch alle gleich verhalten. Im optimalen Fall würde dieser Vektor den Weg von der zeigenden Person zum Objekt, in diesem Fall dem projizierten Punkt, beschreiben. Durch mathematische Analysen wird jedoch gezeigt, dass auch im Durchschnitt über mehrere Versuche und mehrere Versuchsteilnehmer eine systematische Abweichung von diesem optimalen Vektor nachgewiesen werden kann. Es werden Modelle angegeben, welche die nachgewiesene systematische Abweichung kompensieren können. Diese Kompensation verschiebt den Zeigevektor um die mittlere nachgewiesene Distanz zwischen optimalem und gezeigtem Vektor. Produkte des Konsumermarkt können benutzt werden um Zeigegesten zu erkennen, diese können mit den erzeugten Modellen verbessert werden. Im Fokus hierbei stehen beispielsweise Produkte wie die Kinect.

Contents

1. Introduction	1
2. Related Work	3
2.1. Visual Feedback	3
2.2. Ray Cast Techniques	4
2.2.1. Hand Rooted Techniques	5
2.2.2. Head Rooted Techniques	5
2.2.3. Elbow Rooted Techniques	5
2.3. Tracking Techniques	6
2.4. Pointing Gesture	6
2.5. Pointing Tool	8
2.6. Natural Hand Tremor	8
2.7. Summary and Discussion	9
3. User Study	11
3.1. Design	11
3.2. Motion Capture System	17
3.2.1. Motive:Tracker Software	17
3.2.2. Calibration	17
3.2.3. Data Formats and Pre-Processing	18
3.3. Recoding Data	19
3.4. Participants	22
3.5. Mental Load	23
3.6. Result Software	25
4. Results	27
4.1. Data Preprocessing	28
4.2. Data Analysis	30
4.3. Angle Preparing	35
4.4. Model Fitting	37
4.4.1. Curve Fitting	38
4.4.2. Cross-Validation with Positions	42
4.4.3. General Model	43
4.5. Index Finger Ray Cast Fitting Graphics	45

4.6. Head Finger Ray Cast Fitting Graphics	46
4.7. Forearm Ray Cast Fitting Graphics	47
4.8. Summary and Discussion	48
5. Conclusion	51
A. Appendix	55
A.1. Study Questionnaire	56
A.2. Function Coefficients	60
Bibliography	63

List of Figures

3.1. Studies arrangement, the red dots is shown on the wall, the black dots represent the full grid, but not displayed on the wall all at the same time.	12
3.2. This figure shows a study setup. (The scene is recreated.)	13
3.3. A sample point as it is projected onto the wall. With extra lines to shorten the search time after the little red cross.	14
3.4. The sketch shows the marker position on the human body. The marks are represented through the red dots.	15
3.5. This shows how the hand markers are attached to the participant's hand.	16
3.6. This automaton shows the sequence of where the participants have to go through for each point.	16
3.7. A recorded track of one marker for almost 7 minutes that is displayed in the Motive:Tracker software. The three different coordinates are shown in in three different lines.	18
3.8. The screen shot of the recording program which shows the program in recording mode.	19
3.9. Sketch of the object diagram which stores all information captured in the study.	21
3.10. Body measurements of participants in cm. (A) is the height, (B) is the eye height, (C) is the shoulder height and (D) is the arm length. The height was measured from the ground.	22
3.11. Answers to Raw-NASA-TLX questions of the participants. The four phases are shown secretly (1. green, 2. yellow, 3. blue and 4. orange). For the questions (A) to (F) are presented in the following.	23
3.12. (a) Answers to NASA TLX questions of participants per pass. (b) The NASA TLX total result.	24
3.13. The UI of the result calculation software.	25
4.1. Here we visualize the calculation of the Δ_{lr} angle. The ray cast is projected into a plane parallel to the y plane. Than the angle Δ_{lr} is the angle between the projection of the ray cast and projection of the target ray.	27
4.2. Top view sketch of the data received in the modeling. The side view looks exactly the same.	29
4.3. Visualization of the markers we used in the three different ray cast techniques.	30
4.4. The absolute distance between the average intersection point (blue dot) and the pointing targets (red cross) for the index finger ray cast technique.	32

4.5. The absolute distance between the average intersection point (blue dot) and the pointing targets (red cross) for the head finger ray cast technique.	33
4.6. The absolute distance between the average intersection point (blue dot) and the pointing targets (red cross) for the forearm ray cast technique.	34
4.7. Δ_{lr} visualizations of IFRC technique	45
4.8. Δ_{bt} visualizations of IFRC technique	45
4.9. Δ_{lr} visualizations of HFRC technique	46
4.10. Δ_{bt} visualizations of HFRC technique	46
4.11. Δ_{lr} visualizations of FRC technique	47
4.12. Δ_{bt} visualizations of FRC technique	47
A.1. The general questions about the participant in the study.	56
A.2. The input for the body measurements of the participants. In the measurements, the participants was helped.	57
A.3. The first part of the TLX with extra input to verify the position.	58
A.4. The second part of the TLX.	59

List of Tables

3.1. The technical specification of the V100:R2 camera	17
4.1. The usable data sets from which the models will be generated. Theoretically 5040 pointing gestures were taken.	28
4.2. The table includes the average angle and the standard deviation of the correction angle Δ_{lr} and Δ_{bt}	36
4.3. The four fitting functions we use to generate our models. The coefficients are determined with the help of <i>fit()</i>	38
4.4. The fit results of the functions in Table 4.3 with index index finger ray cast, head finger ray cast and forearm ray cast.	41
4.5. The result of a leave-one-out cross-validation with four different parts, these are the four different positions on its own. All values are generated by using $LOOCV(y, train, test)$, see Equation 4.17. The results are independently for Δ_{lr} and Δ_{bt} as well as the four functions in Table 4.3 and the three ray cast techniques.	43
4.6. The result of a leave-one-out cross-validation	44
A.1. Random distribution of positions for each participant.	55

A.2. The coefficients for the IFRC technique models	60
A.3. The coefficients for the HFRC technique models	61
A.4. The coefficients for the FRC technique models	61

List of Listings

2.1. Simple definition of a pointing gesture like [LW98] defined.	7
---	---

1. Introduction

Pointing with the hand at distant objects is an important aspect when humans communicate with each other. Distant pointing is such an fundamental skill that humans acquire it in their early childhood [LR81]. Bolt proposed to use distant pointing for interacting with computing systems, in his seminal work on multimodal interfaces [Bol80]. A large body of human-computer interaction research applied it for various tasks and investigated the usability. At least since the introduction of the Wii Remote and the Kinect, distant pointing at virtual objects is also used in consumer products. After determining the orientation of a user's arm or an input device a position on the screen is determined through simple ray casting [VB05]. Previous work (e.g. [BMGX13]) reported a number of technical limitations that reduce the user's accuracy and investigated approaches to improve the performance (e.g. [KSMB08, KBSM10]).

A fundamental aspect of distant pointing has, however, been neglected in human-computer interaction research. Psychology showed that humans do not aim at targets with perfect accuracy [FH72]. Even if a person tries to aim straight at a distant target with his or her arm, a ray casted from the person's arm or finger does not necessarily hit the center of the target. A corpus of neuroscience research investigated aspects that affect pointing precision. A number of factors that lead to systematic errors have been identified. Neuroscience research, however, is mainly interest in deriving qualitative models of human behavior. While such models extend the understanding of human behavior they are of limited use to improve a users' performance. Predictive quantitative models for human-computer interaction tasks have successfully been used to compensate systematic errors. It has, for example, been shown that even very basic models can improve the accuracy of selecting targets on touchscreen [HRB11] and for entering text on mobile devices [HRB12]. It has also been shown that more complex models can further improve the target selection accuracy [HB11].

Humans use pointing gestures to directly referenced objects. Most real world objects cannot provide feedback. Therefore, a major challenge is enabling users to point at objects without giving feedback. In previous work, tools are typically used to select the target object. Use of such tool is not natural and such pointing tools are not always at disposal. Therefore it's necessary to develop a model without using feedback and without requiring a pointing tool. The aim of this thesis is to investigate if a users' performance when pointing at distant objects to interact his computing systems can be improved by a quantitative model of distant pointing that compensates systematic displacements. To develop such a model it is necessary to have precise data about human movements. A system that can precisely measure the position of relevant parts of a human's body should be implemented using a tracking system, like the OptiTrack

system and the software stack available in the “Institute for Visualization and Interactive Systems”. Humans do not point always in the same way. A target selection study should show how wide the range is. Precise data should be generated by a study which should be designed and executed based on the system, the first study and previous work on distant pointing. The relevant data to generate the model for distant pointing should be collected in this study. A simple model, similar to the basic vector field in [HRB11] and a complex model that considers the posture of the user should be derived. Depending on the models’ performance one or both models should be compared with a control condition in an experiment.

Outline

The thesis is structured as follows:

Chapter 2 – Related Work: In this chapter, we discuss related work in the field of distant pointing. Different techniques of interpreting pointing gestures are discussed.

Chapter 3 – User Study: In this chapter, we discuss the design and the implementation of our user study.

Chapter 4 – Results: In this chapter, we analyze the data collected during the study.

Chapter 5 – Conclusion: In this chapter, we summarize conclusions that we can draw from our work and discuss future work.

2. Related Work

This chapter discusses related work about state of the art distant pointing techniques. Therefore we divide the field in six subsets.

The first ideas in the field of HCI with distant pointing can be found in a system which is called 'Put-That-There' [Bol80]. In this system they build a 'Media Room' where it is possible to change the display with the help of point and speak. This system projects a cursor on the surface to symbolize the integration point. This cursor's projection principle, we will call in the further visual feedback. Related research on this field is described in Section 2.1. In the system, 'Put-That-There' no link between gesture and feedback is described. Vogel et al. [VB05] describes two basic types of a relationship, relative and absolute. Research is presented in Section 2.2. Vogel et al. [VB05] use an infrared tracking system others like Nickel et al. [NS03] use a stereo-camera. Different research on tracking methods are described in Section 2.3. Both detect the free hand gesture, see Section 2.4. In contrast, there are works which detect the user's gesture by a tool the user has to use. Research on different pointing tools are described in Section 2.5.

Our last field we make investigation in is on the natural hand tremor, this is done in Section 2.6.

2.1. Visual Feedback

Regular visual feedback is used to represent the position where the user points to. We can find a wide range of visual feedback techniques. Most of visual feedback researches are done in virtual environments (VEs). So they all give on-screen feedback. However there is also research to give real world objects a visual feedback.

Bold [Bol80] gives the user a '*small, white "x" cursor*' [Bol80] as feedback where the user points to. Argelaguet and Andujar [AA09] present their research on feedback on stereoscopic displays. They use 2D Cursor, 3D Cursor, ray based feedback and viewfinder metaphor to visualize the pointing feedback. They have carried out a study to evaluate these techniques. Matveyev and Göbel [MG03] do work with mouse cursor feedback on a projection surface. In their work they used an infrared beam as input. Also Jiang et al. [JOMS06] used in their research a cursor to give the user a feedback. Forlines et al. [FBB⁺05] also use a cursor as feedback. However, in

their research they make use of a zooming technique around the cursor to identify the right object better.

Wong and Gutwin [WG14] develop and rank feedback techniques in collaborative virtual environments. They have used five different methods to symbol a person's hand in to a 3D scene. These five are natural pointing, long arm, laser beam, spotlight, and highlight visualization. They have carried out a study with 24 participants to evaluate these five techniques.

Butz et al. [BSS04] build a real world search engine. They used a 'searchlight' to give feedback to the user. The 'searchlight' is a beam of light coming from a rotatable light source but from a fixed position. In this work they only give feedback on books. However, it is possible to use a beam of light as real world feedback for pointing gestures.

2.2. Ray Cast Techniques

The ray cast in general is vector. The ray cast is build out of two points. In our case these points are three-dimensional points. With the help of a ray cast direction of the pointing gesture is described. The point where a ray cast is incident on an object, there is the 'identifier' point.

In general Vogel and Balakrishnan [VB05] describe two techniques to move an identifier. An identifier is the place where a action would be executed. In the PC world this would be the mouse cursor, on a touch surface this would be the finger potion, in a 3D environment this is the feedback position. The two techniques are *relative* and *absolute* feedback. *Relative* feedback is when the input device is move somewhere and the identifier is moved in the same way but can be scaled and inverted, like the mouse. *Absolute* feedback is when the input device and the identifier react in exactly the same way, like on a touch screen. A ray cast technique is always an *absolute* input.

Mainly there are two different classes while pointing to make a ray cast. One class are hand rooted techniques and the others are eye rooted techniques. These are invented by Argelaguet et al. [AAT08]. We will not be that specific to eye rooted, because it is hard to detect eye orientation. So we will call these head rooted techniques. We consider eye rooted as a subset of head rooted. So we combine the eye rooted and head rooted techniques which are not that hard to detect. Head orientation tracking works the same way like hand tracking. For eye tracking special hardware will needed. The origin of the ray here is the hand, eye or head. However, we can find a third technique which is not so common. This technique is use by Nickel and Stiefelhagen [NS03] where the origin of the ray is the elbow use. So the techniques are classified by the origin of the ray.

2.2.1. Hand Rooted Techniques

Mine [Min95] work with 'hand orientation', he have no specification how this works in detail. However, he came up with a more general problem for hand rooted techniques. *'It can be confusing for novice users [...] to fully understand the relationship between hand orientation' and pointing target* [Min95]. We can extend this statement to elbow rooted techniques.

Others works do a more specific specification. Corradini and Cohen [CC02], define it in that way: *'passing through the base and the tip of the index finger'*. In this work a user can paint on a virtual paper with free hand pointing gesture and speech control. This is a common specification. Kranstedt et al. [KLP⁺06] use the same rule. In this work they measure the error between an object with lies on a table and the intersection of the ray. In the work of Vogel and Balakrishnan [VB05] they describe interaction techniques with a very large, high resolutions display. They also use this index finger technique. Argelaguet and Andújar [AA09] use this technique to generate a stereo feedback.

2.2.2. Head Rooted Techniques

Nickel et al. [NS03] show two general methods when using the head as origin for ray casting. One is the head orientations as ray. Here the head is the origin of the ray and the orientation of the view is the direction of the ray. The other option is when using the head also as origin and the direction is the vector to the hand. Mine [Min95] defines the possibilities in the same way. A detailed description give Jovic et al. [JBM⁺00] where they used the 'top of the head' and the 'tip of the fingers' to estimating the pointing direction. Pierce et at. [PFC⁺97] also work with head rooted technique combined with the hand. The specification is: *'from the user's eye-point through the location of the tip of the user's index finger'*. The user's eye-point is not a exact defined position. So Kranstedt et al. [KLP⁺06] and Pfeiffer et al. [PLW08] defined a exact position the 'Cyclops eye'. This is the point in the middle of the two eyes. The user's index finger is also not a exact position so they both use the tip of the index finger where the ray passes through.

2.2.3. Elbow Rooted Techniques

Nick et al. [NS03] introduce a third not common technique. They describe it as *'the orientation of the forearm'*, but we cannot classify this into the two common techniques. However when spiting the forearm into origin and direction, we end up with the elbow as origin and with the root of the hand as direction.

2.3. Tracking Techniques

If we have an example scenario like pointing at an object to select, we can use a pointing tool like we describe in Section 2.5 which emits a beam which can directly be detected. This section do not address on this case. Here we described techniques which detect tools or human body structures without a beam. For this purpose we have to be able to observe the environment to extract special body points from it. A common method is to use a marker based optical tracking systems, also known as optical six-degree-of-freedom (6DOF) tacker. Such a system can track several special marks. These markers can reflect or emit infrared light, like OptiTrack¹ or Qualisys². Through several cameras, the position of the marker can then be triangulated. Markers can be attached directly to the human body to detect the location of the body. This tracking techniques are used for pointing tasks by [VB04, VB05, KLP⁺06, PKL06]. Corradini and Cohen [CC02] use also a 6DOF tracker by Ascension Technology Corporation³, but this one use magnetic field detection to track markers. Also Wingrave et at. [WTW⁺05] used a magnetic tracker to detect gesture. However, these markers can also be attached to a pointing device to generate a pointing vector, this is done by [KSMB08, KBSM10, FKGR09].

Another technique is to extract information out of camera pictures. Jiang et al. [JOMS06] localized a pointing tool in RGB camera images. Also based on camera images it is possible to extract body part locations, this is done by [JBM⁺00] and [NS03].

2.4. Pointing Gesture

Pointing gestures are part of deictic reference or deixis. Which means the situation when people in a conversation use language and a gesture to reference objects around them [McN92]. Referencing objects with gestures is possible as long as the gesture can be seen and interpreted. Hindmarsh and Heath [HH00] described that the gesture has to be hold, so that the can be interpreted. In the following we present research on interpretation of pointing gestures.

Kendon [Ken08] says '*pointing are commonly done with the hands*' [Ken08, p. 199]. However research on none hand pointing is done too. What are pointing with eyes, lips [She73, Enf01] elbow, foot [JK00]. However, in addition, we will restrict ourselves to hand pointing.

In addition, Kendon [Ken08] says that there are there different main hand pointing options when people are in a conversation, *index finger extended*, *open hand in pointing* and *pointing with the thumb*. Pointing with the thumb is used to point '*when it is not important to establish the precise location or identity of what is pointed at*' [Ken08, p. 218]. Open hand in pointing is

¹<http://www.naturalpoint.com/optitrack/>

²<http://www.qualisys.com>

³<http://www.ascension-tech.com>

used when 'the object being indicated is not itself primary focus or topic of the discourse but is something that is linked to the topic' [Ken08, p. 208]. These both do not address our needs, so will not be describe further. Index finger extended pointing is used when a specific person, object or place is being meant. In a conversation '*the speaker will uses an explicitly deictic word, such as "here", "there", "this" or "that"*' [Ken08, p. 205]. These indication words are also used by Duranti and Goodwin [DG92]. Kendon defines an index finger extended gesture as a gesture with straight arm upward and hand shaped so that only the index finger is extended. Moreover, Kendon describes two sub-categories of index finger extended: index finger extended pone (palm down) and finger extended neutral (palm vertical). However in his work he '*do not see consistent difference in use between index finger extended pone and index finger extended neutral*' [Ken08, p. 208].

Haviland [Hav03] present research on pointing in the early childhood. Even at this age, children begin to be able to express themselves better through pointing gestures. However he mention that '*pointing may seem a primeval referential device, it is far from simple: It is complex*' [Hav03, p. 156].

```

1  function boolean isPointingGestre (IndexP, MiddleP, RingP, ↔
    PinkyP) {
2      return Radius (IndexP) / Length (IndexP) < threshI AND
3          Radius (MiddleP) / Length (MiddleP) > threshM AND
4          Radius (RingP) / Length (RingP) > threshR AND
5          Radius (PinkyP) / Length (PinkyP) > threshP;
6  }

```

Listing 2.1: Simple definition of a pointing gesture like [LW98] defined.

Latoschik and Wachsmuth [LW98] developed a function (like Figure 2.1) to detect a index finger extended pointing gesture. This definition matched the one from Kendon [Ken08] except for the position of the thumb. The thumb position is in [LW98] not specified. However, the index finger should be extended. Middle, ring and little fingers must be closed. Closed here means bent so that the fingertips point in the direction of hand heart. From this definition, the function in Figure 2.1 can be derived. As the input values IndexP, MiddleP, RingP and PinkyP are required. These are arrays of dots which represent the fingers, with at least three points. Radius returns the radius of a pointing sample. For a straight line the function returns infinity. The fact that fingers will be different sizes, the radii normalized to the length. So that it is possible to use a uniform threshold for all fingers.

2.5. Pointing Tool

A number of researches which investigate distant pointing use a tools to point, e.g. [ON01, MG03, WP03, MCM09, JOMS06, KBSM10, KSMB08]. They all use a tool to generate a pointing ray. The problem with pointing tools is that they change the real human body posture. To generate a model which does not require a pointing tool, it is important that the data, used to generate the model made without pointing tool. Thus, these works do not match directly to our work.

However, we want to introduce some tools for distant pointing. There are a number of devices which emit a beam. First one is a traditional tool, a laser pointer. The laser beam can be detected and used as an input for a virtual interaction surface, like a display [ON01]. Second is an infrared beam which can be used as input. [MG03] used an IR-Stylus as input device. Wilson and Shafer [WS03] build a device called 'XWand'. This device also use infrared, but it also has some other sensors e.g accelerometer, magnetometer and gyroscope. In [WP03] Wilson and Pham have done some evaluations on this device. MxArthur et al. [MCM09] analyze with a more common device, pointing with a 'Wii Remote'⁴.

Other works use devices that do not emit a beam. The tools are then localized by cameras. First is to extract the tool position out of RGB images [JOMS06] or second is to track special markers which are attached to the tool [KSMB08, KBSM10]. The second type, special marker, need an infrared tracking system like described in Section sec:TrackingTechniques.

2.6. Natural Hand Tremor

Elble [EK90] present research tremor in the medical field. Elble tries to find one definitions but he figured out that: *'Many authors have struggled with a definition of tremor. Most definitions are too restrictive to encompass all forms of tremor'* [EK90, p. 1] and *'existing classifications schemes are still vague and empirical'* [EK90, p. 2]. Elble defined tremor *'as any involuntary, approximately rhythmic, and roughly sinusoidal movement'* [EK90, p. 1]. We found more precise measurements in the field of neurophysiology. Motor units like the hand are firing at 8- to 12-Hz oscillations with an inter spike interval coefficients of variation that are less than 0.2 [CL80]. Basmajian and De Luca [BDL85] are not giving so accurate frequencies, they only say less than 13Hz.

Pointing studies like Olsen and Nielsen [ON01] have done describe this problem in a technical way. In their work they used a laser pointer to describe a pointing direction. They made the statement that *'the user can not accurately hold [...] position'* [ON01]. They do not give a precise measurement on natural hand tremor.

⁴Page 6 in 'Wii Operations Manual', http://www.nintendo.com/consumer/downloads/WiiOpMn_setup.pdf

Riviere et al. [RRK97] have done research on hand motion of eye surgeons. They could see that they cannot hold their hand completely quiet. They could identify 'a broad peak in the 8-12 Hz tremor band' [RRK97].

2.7. Summary and Discussion

After getting an overview about research that have been done, we make some chooses to classify our work.

First we think about visual feedback. We have shown a wide range of feedback possibility's, on screen and also one for the real world feedback. In general it is complicated to equip each real world object with a feedback. So our goal is to get along without feedback. For this purpose it is necessary to have a precise model to correctly interpret the pointing gesture. Next is pointing tools. Pointing tools does not address our needs. Tools change human's behavior but the model we will not build is to understand human behavior better. So it is impossible to use pointing tools in this case. With this choice it is clear that we want to track natural pointing gestures. The first choice implied a precise model. For this purpose we need to use a tracking tool gives us precise data. This cannot be done by RGB camera based location tracking. So we needed to use a 6DOF tracker, we used OptiTrack by Natural Points. Various persons have done research on ray cast techniques. In the further course we used to popular techniques and one more exotic, index finger ray cast, head finger ray cast and forearm ray cast. Hand posture (palm down or palm vertical) or finger positions (except the index finger) are not considered in this work. The forearm ray cast technique we have chosen because the forearm is larger than the finger and larger objects can be better detected in systems without precise markers. Detection without a 6DOF tracker is not part of this work but later we want to detect pointing gestures by using e.g. a kinec⁵.

⁵<http://www.xbox.com/de-de/Kinect>

3. User Study

We do a user study to then generate a model to compensating systematic displacements. In the study we record pointing gestures. Out of the pointing gestures we generate a model. Therefore it is necessary to have a wide range of different pointing directions. Because we aim to build a model which has a high accuracy in each direction. Out of the complexity of pointing gestures, we have a high probability that not every human makes the same pointing gesture to point at the same target. That is why our models has to be based on several pointing gestures with the same target. As a byproduct the study shows whether the assumption is true that there are several pointing possibilities for one direction. However, the main goal of the study is the collection of pointing gesture. Thus, there are two requirements. First we have the wide range of the data. Second is the redundancy of pointing gesture per direction.

To make the two requirements to the data securely, two properties are defined to the study. The first property is a regular point grid. This grid defines the positions where the participant has to point to. From now on, this is referred to as 'pointing target'. The second property is a fix position. This means a fixed place where the participant has to be while pointing to the pointing targets. Then the results are sets of various gestures with the same pointing target.

3.1. Design

The grid is project onto a flat wall by using a projector. In our case we use a point grid of 7 columns and 5 rows, see Figure 3.1 and Figure 3.2. The task of the participants is to point on the projected points. So they have to point to 35 different points. This follows 35 different pointing targets. These points are projected separately, so one appears after the other. The sequence of pointing targets is randomized for each pass. In this study, not just one fixed position is used. Rather, we used four different positions. The distance to the wall is thereby changed. We use 2 to 3 meters distance to the wall. The participant is always centered to the grid. Hereby, the influence of the distance will be measured. The two distances are combined with two different postures, standing and sitting. With the two various postures, other effects can be determined. As a result, we obtain four different positions. Each participant will guide throw four positions in the study. In the following, this operation will be described with phases. In each phase, the participant needs to change its position. Thus each participant has to point on all pointing target of each position. These four different positions are randomized for each

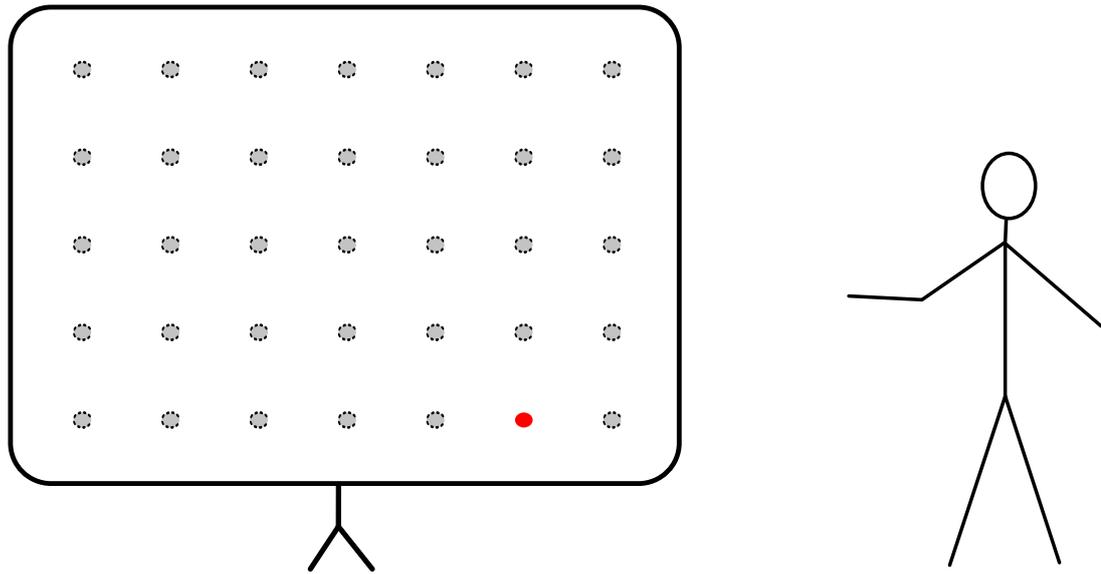


Figure 3.1.: Studies arrangement, the red dots is shown on the wall, the black dots represent the full grid, but not displayed on the wall all at the same time.

participant. The randomization of the passes is shown in Table A.1. We randomize the order to avoid learning or tiredness effect.

Out of the 35 pointing targets and the four phases results $35 \times 4 = 140$ gestures per participants. By the assumption that there will be differences while pointing, all pointing gestures are different and they are hard to combine. Also outliers of one participant are hard to analyze. For this reason, it is required that each participants points three times per phase to each target point. Thus there are $140 \times 3 = 420$ gestures per participants.

In order to create the same conditions for each gesture, the participants had to keep coming back to a starting position. This starting position is defined as follows: the arm is positioned along the right side of the body. This guarantees that no error of a gesture affects the next gesture.

In Section 2.6 we describe how the natural hand tremor works. The natural hand tremor has an influence on a pointing gesture. To minimize this influence the participants have to hold the gesture for at least one second. For this purpose, the participant was given an extra device in to their left hand. Here for we used a presenter. The presenter has a button. The signal button is use to guarantee the one second time span. So task of the participants was to click when they are ready for the next pointing target. They had to click if they were pointing to the pointing target on the wall. After the second click, the pointing target is viable for one more seconds. For this time the participant had to hold the gesture, so to point onto the wall. After this second the pointing target diapers and the participant had to move the right hand back in to start position. The process the test person goes through is shown in Figure 3.6. These entire



Figure 3.2.: This figure shows a study setup. (The scene is recreated.)

click actions are recorded with a special tool, described in Section 3.3. The data is used later combined with the tracking data, to filter the impotent sequences. In our case the impotent sequences are these there the participant hold the gesture. Each phase is stored for its one in a file. The combination of the click and tracking data is described in Section 3.6.

As a starting point we have defined so far the participants have to point on 420 points for one second. The hold time is $420 \text{ s} = 7 \text{ minutes}$. So point searching, targeting and return to start position is not included. After some testing, we came to the view that the time must be shortened to minimize the task load. And there is only one possibility to shorten the time. Find the point on the wall quicker. Because the cross is small, a circle and two lines have been added to the presentation. The circuit is based around the main point. The lines are going horizontally and vertically through the main point. This arrangement is shown in Figure 3.3. In this way the entire gesture time was minimized to about 30 minutes. This is around 25 % lower than without the viewfinder.

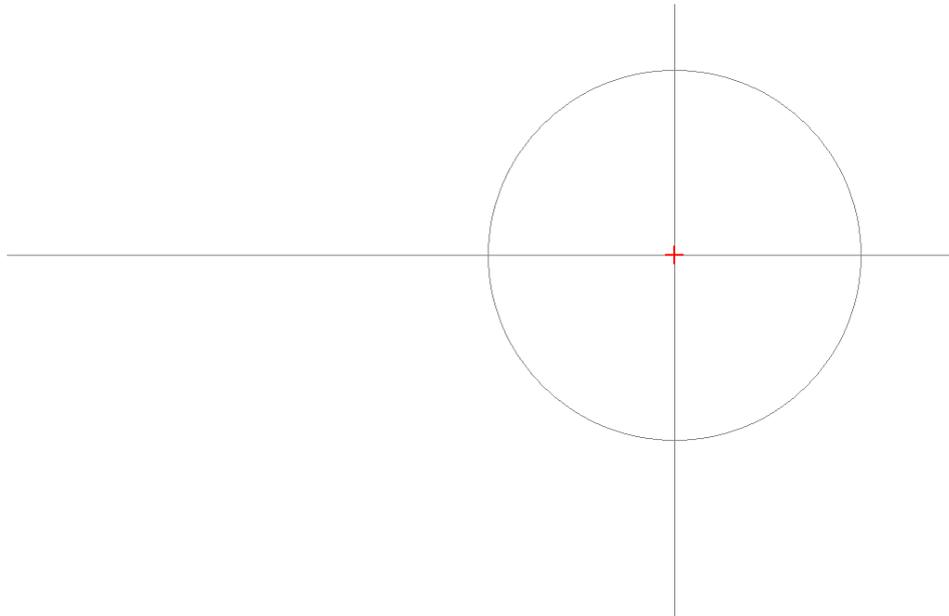


Figure 3.3.: A sample point as it is projected onto the wall. With extra lines to shorten the search time after the little red cross.

Each participant needs to start filling out a questionnaire. These are questions for the person. In addition, each participant is measured. The participants has to fill in an online questionnaire, see Figure A.1 and Figure A.2. To find out how exhausting this task is, each person will complete a questionnaire after each phase. Here we used a NASA raw task load index (NASA-RTLX) test. See [HS88] and [BBH89] for more details on the NASA-TLX and the NASA-RTLX. For our the RTLX questionnaire see Figure A.3 and A.4.

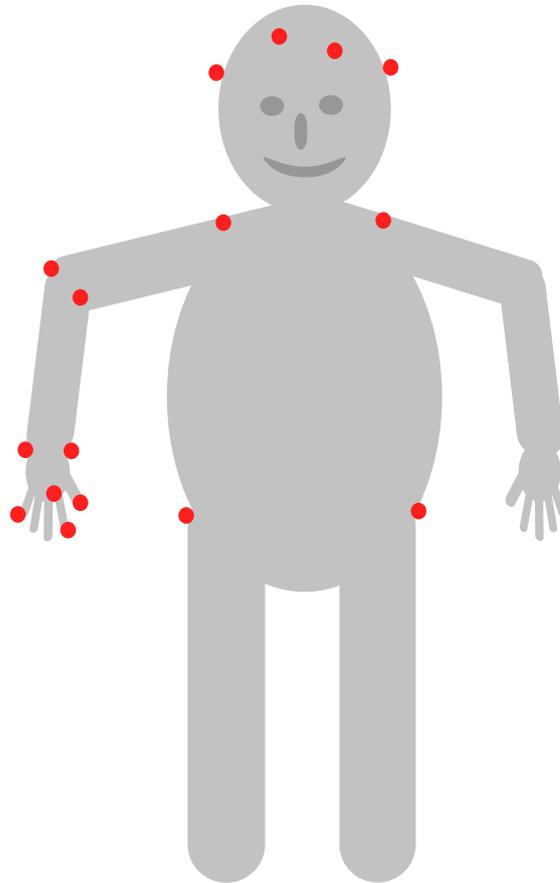


Figure 3.4.: The sketch shows the marker position on the human body. The marks are represented through the red dots.

Throughout the study, the test person is captured with the OptiTrack system. For basic information to the OptiTrack system see Section 3.2. For this purpose, 16 marker was attached on the participants body, see Figure 3.4. Four markers are attached to the head. On the shoulder and the hip are attached two each, one right and one left. At the elbow of the right arm also two markers are attached. On the wrist of the same arm are attached two more marker. To the hand four marker are attached, see Figure 3.5. To the tips of thumb, index finger and little finger are markers attached. Also one marker is attached to the root of the index finger. A further four markers were attached on the wall. This will help us later to reconstruct the pointing targets.



Figure 3.5.: This shows how the hand markers are attached to the participant's hand.

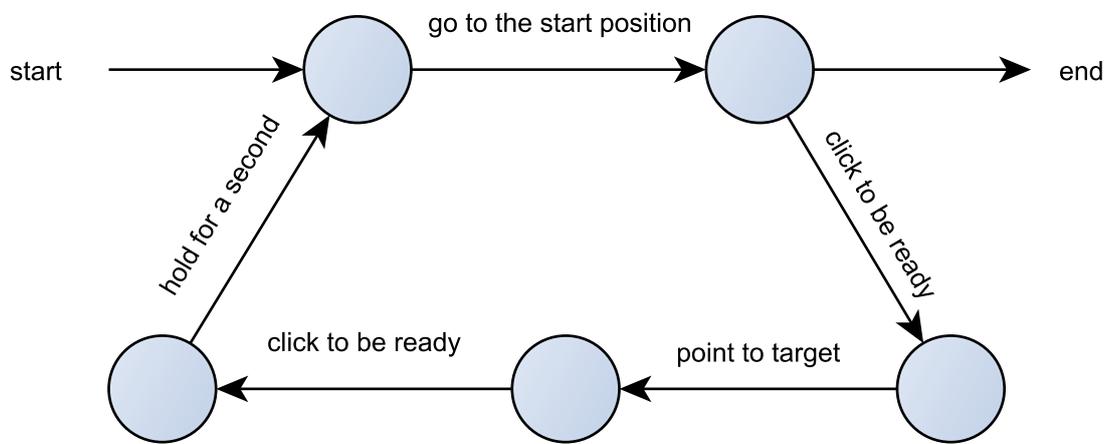


Figure 3.6.: This automaton shows the sequence of where the participants have to go through for each point.

3.2. Motion Capture System

The system that we used for recording is by the company Natural Points and. It is named OptiTrack¹. This system uses cameras and passive markers for position determination. Our setup consisted of 17 of the V100:R2 cameras (for technical specification see Table 3.1) and 3 of OptiHub2 USB-hub to synchronize the cameras. These 3 hubs was linked to on computer where the Motive:Tracker software is installed. The cameras emit infrared light which is reflected at the markers. With Motive:Tracker all data can be recorded for a period of several hours. To keep in mind is the memory to store all data. In a field test we figured out that the system required approximately 1.5 GByte disc space for a 10 minutes record.

	Value
Resolution	640 x 480 pixel
Frame Rate	100 FPS
Horizontal FOV	38°
Filter Switch	none
No. of LEDs	26
Latency	10ms

Table 3.1.: The technical specification of the V100:R2 camera

3.2.1. Motive:Tracker Software

This software processes the data of the cameras, stores, loads and calibrated the cameras. In this work the version 1.0.0 (64-bit) (Built #14948: March 18, 2013 3:58pm) is used. It is possible to save and load data in special and free standard data format. We used the C3D File Format² to export the tracking data. For live analyzes important is the streaming function, which send the data into the network. This data can then be taken up on a different program. The interface for this is called 'NatNet'. The library is available for .NET and C++.

3.2.2. Calibration

In principle, the calculation of the calibration is done automatically. This is done by a special tool called 'OptiWand'. This consists of a rod at the end of a 50cm long rod is attached, in turn, are mounted on the three markers. Because of the fixed distances between the markers, the system knows how they are in the room as soon as they are captured by three cameras. This

¹<http://www.naturalpoint.com/optitrack/>

²http://www.c3d.org/pdf/c3dformat_ug.pdf

3. User Study

means that a precise determination of the position is given when the full space can be covered by at least three cameras. The space may not be covered by the three cameras is approximated by the system. This means that the recording reflects not one hundred percent reality. With the right positioning of the cameras and through proper calibration, the accuracy can be improved. After calibration, the system provides a feedback on the correctness, the average deviation in millimeters. The last step is to set the 'ground plate'; there for a second tool is needed. With this the origin and the direction of the z axis can be set.

3.2.3. Data Formats and Pre-Processing

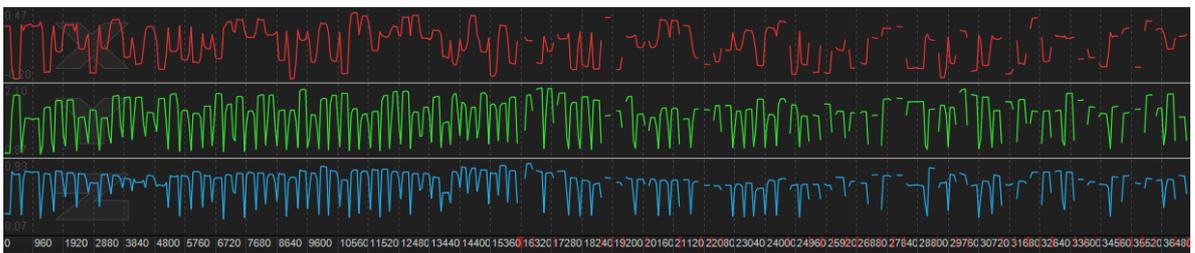


Figure 3.7.: A recorded track of one marker for almost 7 minutes that is displayed in the Motive:Tracker software. The three different coordinates are shown in in three different lines.

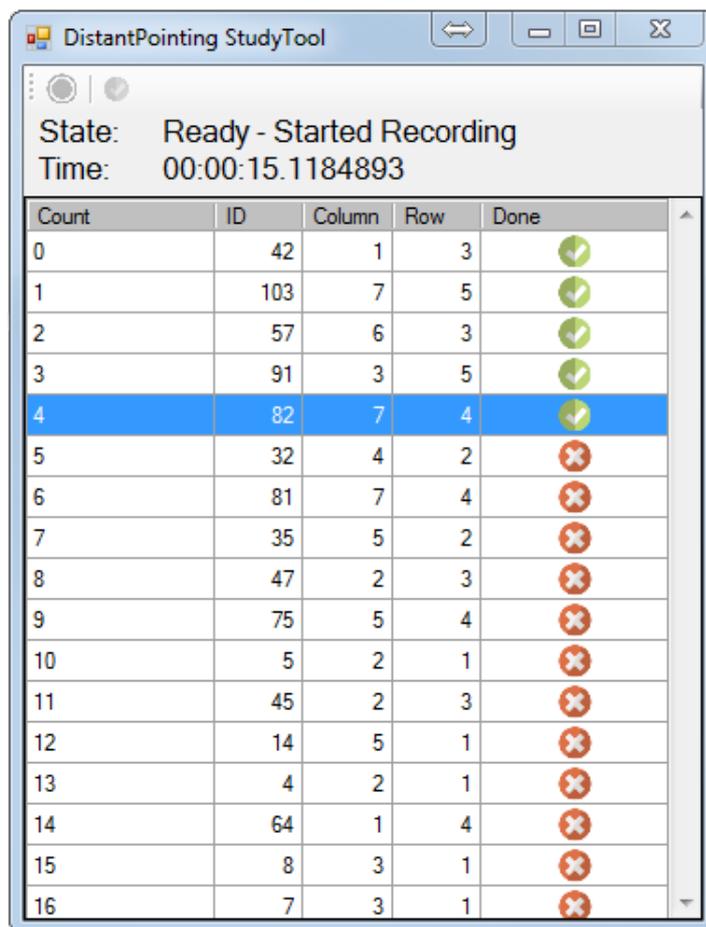
While recording data, it may appear that marker does not lie in the camera field of view for a certain time. Thus, gaps are created in the data. Smaller gaps can be filled by interpolation. This is done by using Motive:Tracker. Larger gaps may not be interpolated. Since the gaps are usually too large inaccuracies between data and real motion. For this work we will only use interpolation for gaps that are maximal 10 frames long. So we reconstructed maximal a tenth of a gesture. This is done because it is important to work with accurate data for our model development. The complicated part here is when a marker drops out of the camera view and gets back in, the marker gets a new identify number. So after a record it is necessary to combine all different identifier which represents the same marker. Here the identifier was represent through a name. When saving the data the names will map to integer values. Also it is necessary to clean up the data from makers that are no markers. Because markers are identifiable by reflection of infrared light, is may happen that other material also reflects infrared light. This work may take some hours. On average, it took two hours to revise one recording with 105 pointing gestures.

After reconstruction and interpolation a track of one marker can look like in Figure 3.7 shown. The figure shows a track of one marker for almost 7 minutes. The three different coordinates are shown in in three different lines, (x in rad, y in green, z in blue). The first half is nearly gap free, the second part of the track is full of large gabs. However, these gabs can have absolutely

no effect. Gabs do not fall necessarily into the scope of a gesture. This is evaluated in the further course and Table 4.1 shows the usable gesture count over all participants.

When all Marker of one record is revised, the data is export. As export format C3D files is chosen. Firstly, with C# can process this format with a special library. Secondly, this results in a massive compression, from partially 1% of the original.

3.3. Recoding Data



The screenshot shows the 'DistantPointing StudyTool' window. At the top, it displays 'State: Ready - Started Recording' and 'Time: 00:00:15.1184893'. Below this is a table with the following columns: Count, ID, Column, Row, and Done. The table contains 17 rows of data. Row 4 is highlighted in blue. The 'Done' column contains green checkmarks for rows 0-4 and red 'X' marks for rows 5-16.

Count	ID	Column	Row	Done
0	42	1	3	✓
1	103	7	5	✓
2	57	6	3	✓
3	91	3	5	✓
4	82	7	4	✓
5	32	4	2	✗
6	81	7	4	✗
7	35	5	2	✗
8	47	2	3	✗
9	75	5	4	✗
10	5	2	1	✗
11	45	2	3	✗
12	14	5	1	✗
13	4	2	1	✗
14	64	1	4	✗
15	8	3	1	✗
16	7	3	1	✗

Figure 3.8.: The screen shot of the recording program which shows the program in recording mode.

3. User Study

A program was developed especially for this study. It is written in C#³ with .NET Framework 4.5⁴.

This software controls not only the display of the points on the wall. It also manages the start times of the gestures. Also it randomizes the sequence of the 105 pointing targets. With the randomization no learning effect can occur. Figure 3.8 shows a screen shot of the running software. In the tool strip bar on the top are two icon buttons. The first is to start the record, here it is already started. The second is for testing purpose. Below this buttons the state of the program and the recording time is shown. In the area below a table with the randomized pointing list is shown. 'Count' is the after randomized order. 'ID' the order before randomization. 'Column' and 'Row' are the position on the grid. In the column 'Done' it is represented with a green chop which points are already done, all other have a red x.

Here the challenge is the synchronization time between this program and the tracking tool Motive by NaturalPoints. Normally one would use an `OnClickListener`. An `OnClickListener` listen on a mouse click on a specific button. Since the tracking software is proprietary software, this cannot be implemented retroactively here. In order to raise this issue a global click listener is used. Therefore a special library is used, it is called 'Application and Global Mouse and Keyboard Hooks .Net Library in C#'⁵. Thus it is possible to detect the next click after another click. This behavior is used to synthesizing the synchronization. When the record button of the studies tool is clicked, waiting for the next global click. After the second click the recording is started. The second click has to be on the record button in the tracking tool Motive.

All information about the recordings are stored in a text file. Therefore the serialization technique is used. This technique translates data structures and objects into a text stream. The Figure 3.9 shows a simple version of the original data structure. It shows as a UML object diagram with entity relations. All necessary values for the further calculations are including here. Special structures which are needed for serialization is not shown here. Especially the `List<>` and `Directory<>` construes are complicated to serialize. So this is a sketch to get an idea of the data structure. Here the entity relation counts the list and directory elements. With this structure it is possible to store on whole phase into one 'DistantPointStudyRecord' object. The tool described in this section does not fill the 'AllFrame'-list. This is part of the second tool which is described in Section 3.6.

³<http://msdn.microsoft.com/en-us/library/kx37x362.aspx>

⁴[http://msdn.microsoft.com/en-us/library/w0x726c2\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/w0x726c2(v=vs.110).aspx)

⁵<https://globalmousekeyhook.codeplex.com/>

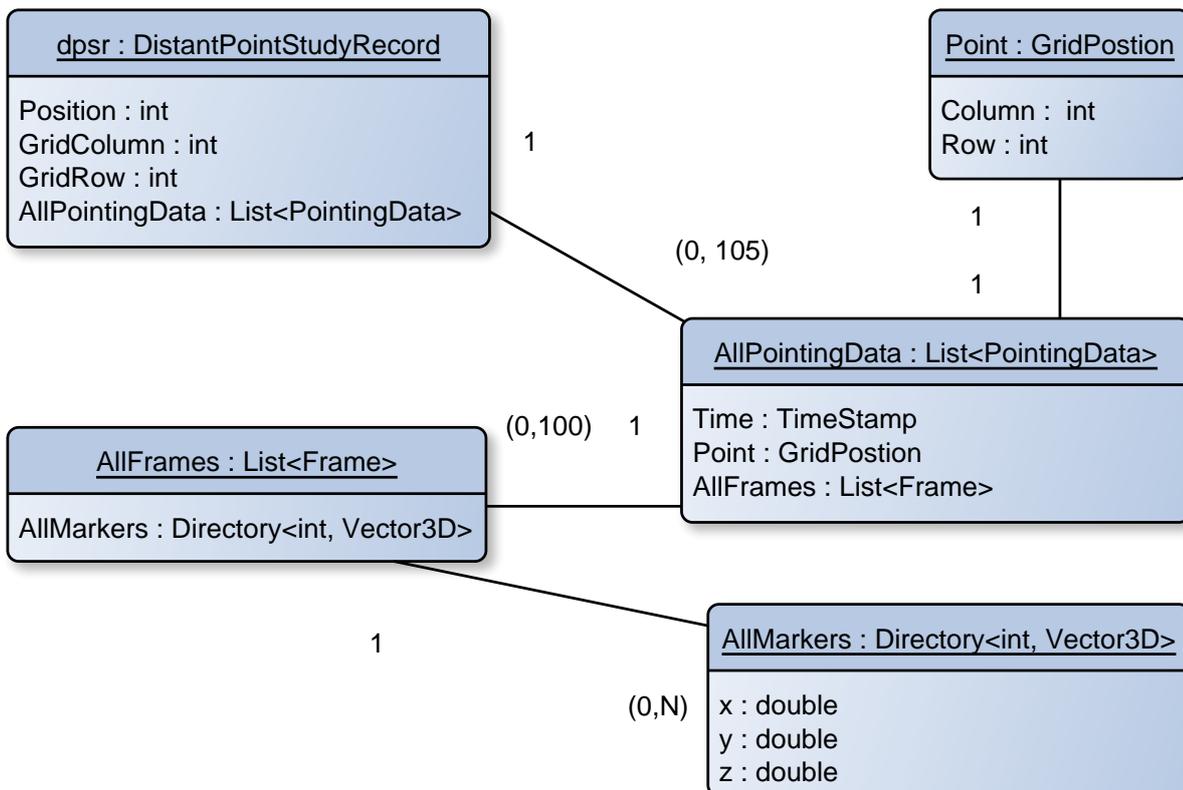


Figure 3.9.: Sketch of the object diagram which stores all information captured in the study.

3.4. Participants

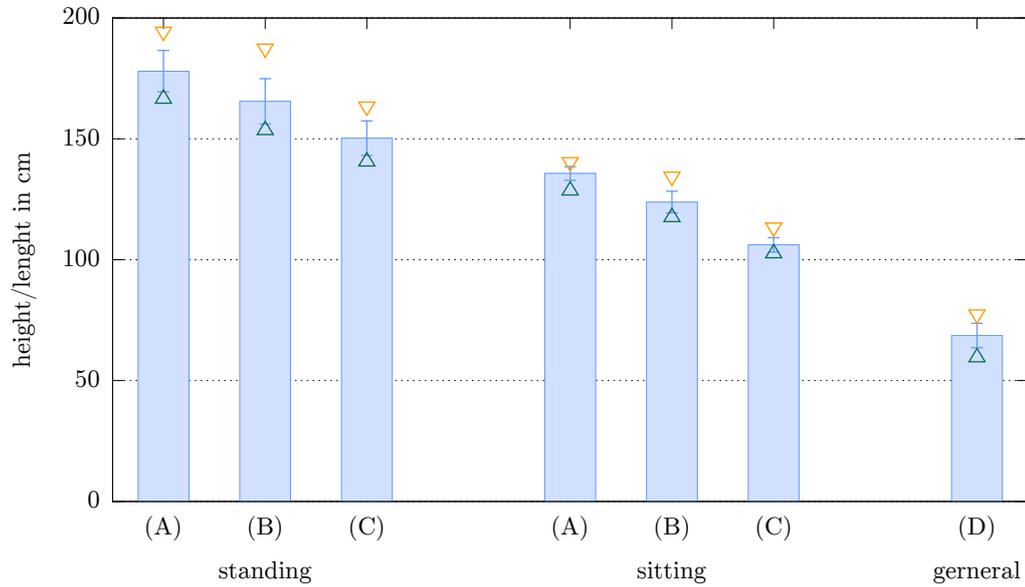


Figure 3.10.: Body measurements of participants in cm. (A) is the height, (B) is the eye height, (C) is the shoulder height and (D) is the arm length. The height was measured from the ground.

At the study 12 participants take part. Six of them were female, six male. The participants had an average age of 24.38 years. The standard deviation of age was 2.7 years. To build a single model, it is necessary either to have only right-handed or left-handed persons only. In this case, only right-handed participants took part. Since it is made in the further relative relations between participants and points, the size of the participants is of great interest. In the Figure 3.10 are all differed measured quantities shown. The bar chart shows the average value, the error bars represents the standard deviation. The two triangle under an over each bar represents the minimum and the maximum value. The figure shows a wide range of different participants, from 167 to 194 cm height.

3.5. Mental Load

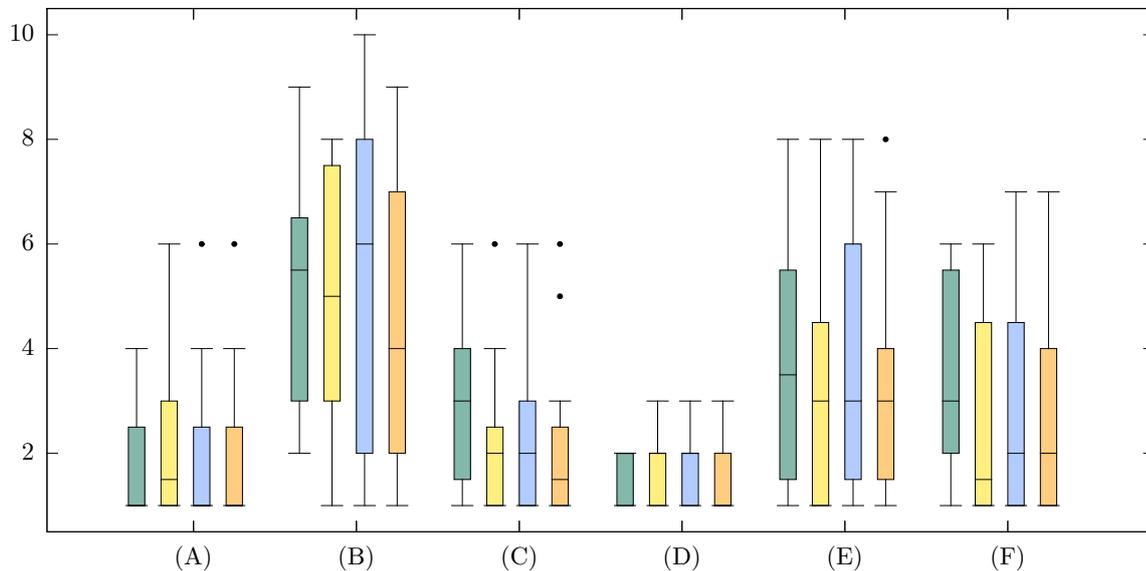


Figure 3.11.: Answers to Raw-NASA-TLX questions of the participants. The four phases are shown secretly (1. green, 2. yellow, 3. blue and 4. orange). For the questions (A) to (F) are presented in the following.

These are the six questions in the NASA-TLX:

- (A) How mentally demanding was the task?
- (B) How physically demanding was the task?
- (C) How hurried or rushed was the pace of the task?
- (D) How successful were you in accomplishing what you were asked to do?
- (E) How hard did you have to work to accomplish your level of performance?
- (F) How insecure, discouraged, irritated, stressed, and annoyed were you?

The participants were asked to fill out a RTLX questionnaire four times, each after the four pointing phases. The questionnaire contains six questions (A) to (F). The can answer on a scale from 1 to 10. Important to know is height values represents a height perceived workload. Thus, low values are aiming for. The results each question are shown separately for each time in Figure 3.11. Result of the first so fours time are colander in green, yellow, blue and the last in orange. The results are represented in a box-plot over all participants. The difference between TLX and RTLX is that in a RTLX the participants do not have to rate the importance of

3. User Study

each question. That is the reason why not only the global result is interesting. If the questions (A) to (F) are considered separately, one can recognize the answers do not vary much. This can be confirmed when the Figure 3.12a is considered. Here it can be seen that the values over the course of the study did not rise but fall. Figure 3.12b shows the total perceived workload. The average result is 2.93 with a SD of 2.20.

These low values may be said that the duration of the study does not decrease the performance of the participants.

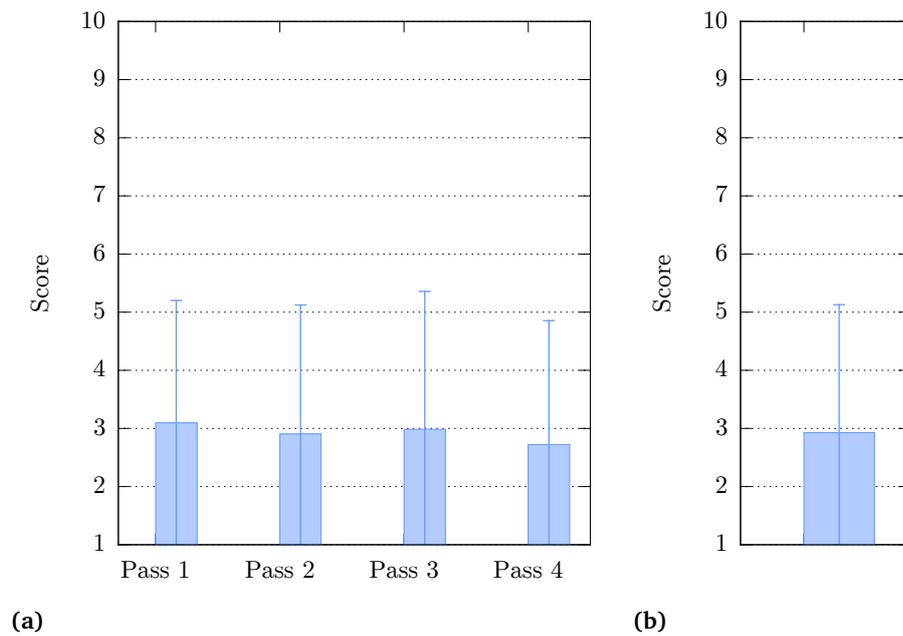


Figure 3.12.: (a) Answers to NASA TLX questions of participants per pass. (b) The NASA TLX total result.

3.6. Result Software

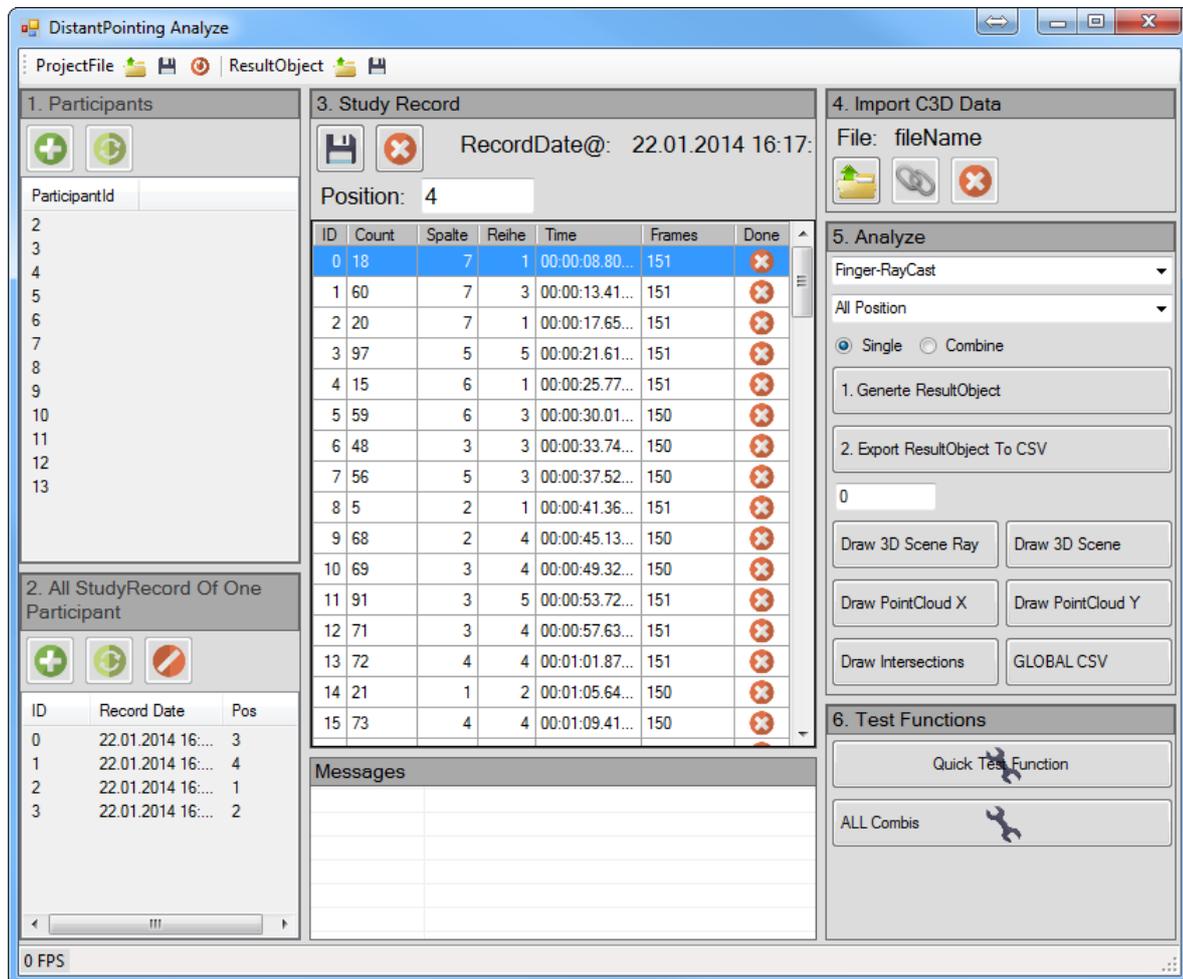


Figure 3.13.: The UI of the result calculation software.

To analyze the data we developed a special software tool. It is written in C#⁶ with .NET Framework 4.5⁷ like the study tool (see Section 3.3). Figure 3.13 shows a screen shot of the program. The tool is separated in seven parts, 1 to 6 and Messages. '1. Participants' and '2. All StudyRecord Of One Participant' organizes the project and the files returning from the study tool. More precisely, '1.' organizes the different participants and '2.' the different result files for one participant. The part '3. Study Record' can open a record and with part '4. Import C3D Data' the tracking data can sort to the record. This is done by using the time stamps

⁶<http://msdn.microsoft.com/en-us/library/kx37x362.aspx>

⁷[http://msdn.microsoft.com/en-us/library/w0x726c2\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/w0x726c2(v=vs.110).aspx)

3. User Study

from the synchronized record. This means that the 'AllMarkers' (see Figure 3.9) variable that has been left empty at the recording time (see Section 3.3) will now filled with the tracking data. Important here is that each first value (the integer vaule) of the 'AllMarkers' directory relays to a specific Marker. So the marker can be identified over all recordings by the number. This integer was set in the Section 3.2.3 by giving specified names to each marker. In part '5. Analyze' can show some 2D and 3D data visualizations. This is done with a library called ILNumerics⁸. This part is also do some groundwork which is described in Chapter 4, but in general it is possible to generate a CSV file. This file is the data base for all calculations Chapter 4 and can import into MatLab⁹.

⁸<http://ilnumerics.net>

⁹<http://www.mathworks.de>

4. Results

In this chapter we analyze the data obtained in the study. We calculate the deviations of the pointing ray cast and the pointing target. We use two angles to represent the deviations. In this work we use three different techniques to interpret a pointing gestures. All three techniques we works with are the ray cast technique. The three different methods are: index finger ray cast (IFRC), head finger ray cast (HFRC) and forearm ray cast (FRC). How these three work and perform is the goal of this chapter.

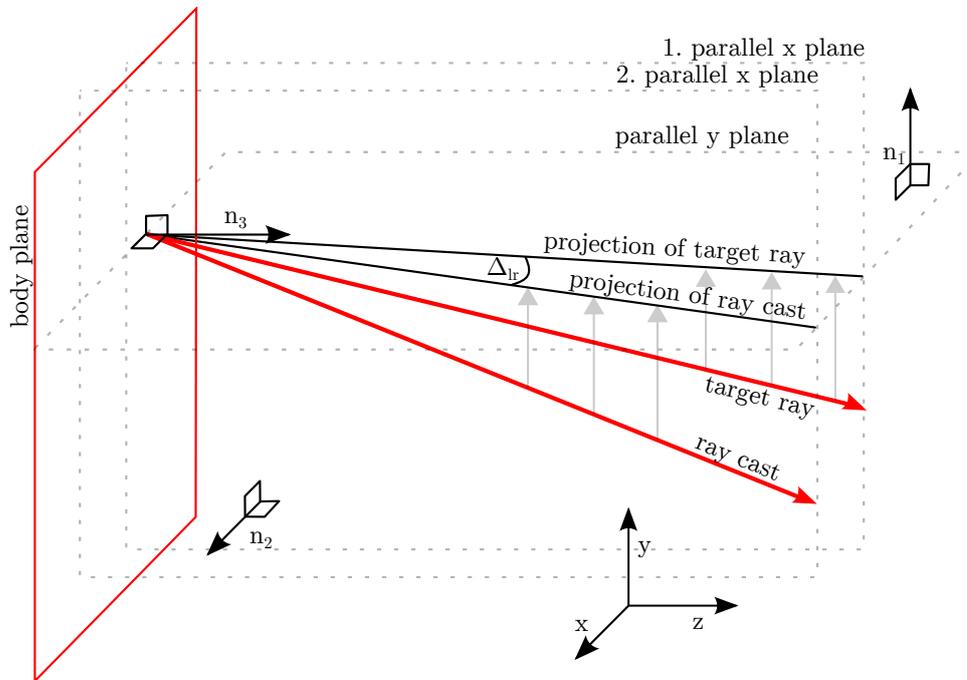


Figure 4.1.: Here we visualize the calculation of the Δ_{lr} angle. The ray cast is projected into a plane parallel to the y plane. Then the angle Δ_{lr} is the angle between the projection of the ray cast and projection of the target ray.

We will create models which compensates the deviations. Therefore we generate two vectors out of each pointing gesture. One vector is the target ray this goes from the pointing person to the pointing target. The other is the ray cast which is described by one of the three ray cast techniques we use. Between these two vectors we calculate two angles, Δ_{lr} and Δ_{bt} . These

are the angles between the vectors in left-right direction and the bottom-top correction angle. We visualized the calculation for Δ_{lr} in Figure 4.1. Based on all angles of all participants we generate our three models. One for each ray cast technique. Each model has two correction functions to correct the left/right and bottom/top displacements. To generate the models we will use a fitting technique.

4.1. Data Preprocessing

The study record is carried out with an average error of < 0.001 mm. This is the average variation of the markers while tracking. How this error value is generated see Section 3.2.2. The recording system delivers 100 frames per second. In each frame, the position of all visible markers is stored. The position of each marker is described by three coordinates. These are float values and describe the distance from the point of origin. We set the point of origin while calibration. The x axis is from left to right of the wall, the pointing targets are projected on this wall. The y axis runs from bottom to top of the wall. The z axis runs from the participants towards the wall.

Method	Usable Gestures	Percent
Index finger ray cast	4881	96.85 %
Head finder ray cast	4869	96.61 %
Forearm ray cast	4758	94.40 %

Table 4.1.: The usable data sets from which the models will be generated. Theoretically 5040 pointing gestures were taken.

In the study about 2 million of pictures were taken. The participants were instructed to click on a presenter when starting the pointing gesture. With the help of the time stamps of the click the important frames were extracted. The important frames are these where the participant is pointing on the pointing target. In total these are 504000 frames. Each gesture is captured in 100 frames. This is done to minimize the error arising from the natural hand tremor, see Selection 2.6. So theoretically the system recorded 5040 gestures. However, some markers were not visible all the time. This means that some of the gestures are not usable. The ray cast techniques rely on with different markers. Thus, the amount of usable data varies depending on the ray cast technique. Table 4.1 shows the usable gesture count. All models described in the following are based on all usable gestures.

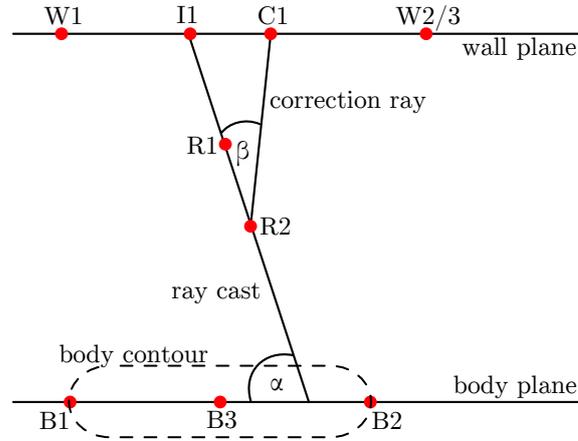


Figure 4.2.: Top view sketch of the data received in the modeling. The side view looks exactly the same.

We aim to get the coordinate system to be out of the perspective of the participant. This is useful to bring all different coordinate system to one point of origin to overlap them later. Therefore the coordinate system has to rotate two times. For this, a transformation matrix t_1 is generated. First it is rotated so that the normal line of the z plane corresponds to the normal of the body plane. The z plane is described by the function:

$$(4.1) (\vec{x} - \vec{0}) * (0, 0, 1)^T = 0$$

A straight line r_1 is constructed which passes through the two hip markers B_1 and B_2 . The second rotation is made around the z axis so that the straight line r_1 is parallel to the x axis.

Further is described how the correction angle can be calculated. Our models are based on these angles. The creation of the model is described in Section 4.4. We combine all 100 frames of one gesture by using the average position of each marker. By doing this the effect of the natural hand tremor is eliminated. All other calculations described here are calculated individually and independently for each gesture. So as input for the next calculations we have the average values of the markers.

First, a plane is calculated that represents the wall. Here for we use the wall marker. To create a plane three markers are sufficient. However, a marker more have been recorded. We detect some problems with recording the left bottom wall marker, this marker was not considered. These three markers are W_1 , W_2 and W_3 , the positions are shown in Figure 4.2.

Next we calculate the three ray cast techniques, index finger ray cast, head finger ray cast and forearm ray cast. The rays are represented by the two markers R_1 and R_2 . Figure 4.3 shows

all markers we used for the three ray cast techniques. We combined some markers to allow a better positioning of the rays. This is described below.

Index finder ray cast (IFRC): To create this ray, the two markers on the index finger are used.

Head finder ray cast (HFRC): The first marker that is used is the marker on the tip of the index finger. The second point is the center of the two outer head points, called head left and head right.

Forearm ray cast (FRC): These the first point is the center of two elbow marker. The second point is the center of the two wrist markers.

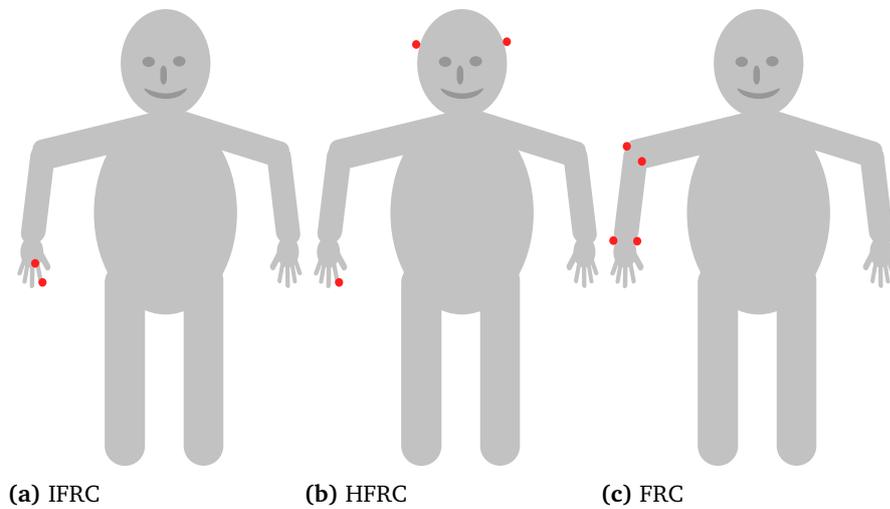


Figure 4.3.: Visualization of the markers we used in the three different ray cast techniques.

In addition, the pointing target $T1$ is needed. This is one of the pointing targets where the participant has to point to. The grid was designed symmetrical, so out of the four wall marker and the grid size we can calculate the position of the point $T1$. With the point $T1$ and the ray cast point $R2$ which is further away from the wall a second vector can be calculated. This ray is called 'correction ray'. The intersection point $I1$ is calculated by intersect the ray 'ray cast' into the plane 'wall plane'. We use $I1$ and $T1$ in the next section to visualize the distance between these two points.

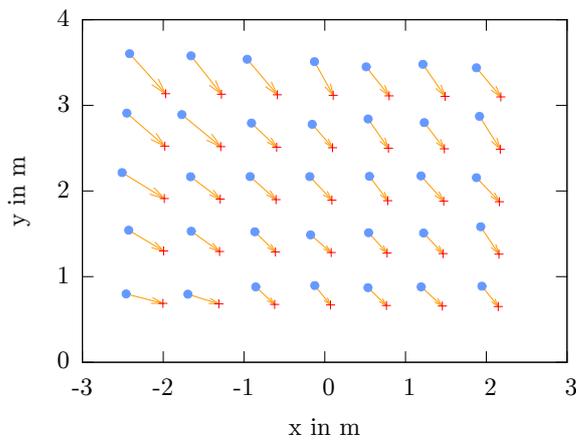
4.2. Data Analysis

In this section we visualize the distance between ray cast intersection and pointing target. These visualizations we used to analyze the different distances.

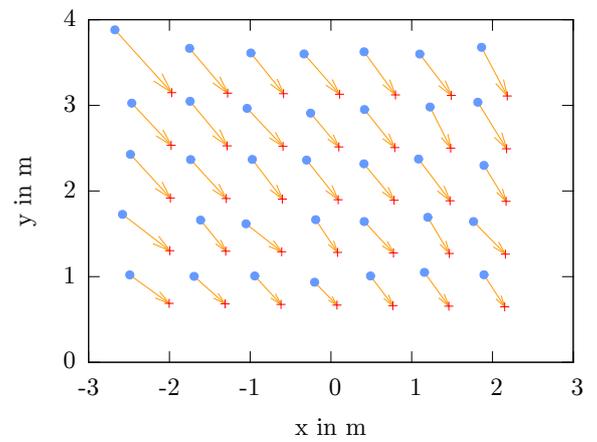
To provide an overview, how the participants act in average, we visualized the data in the following way. For each of the three ray cast technique we generated four figures. One for each position shown in the four sub figures in the Figures 4.4, 4.5, 4.6. The figure shows the whole grid of pointing targets. These are symbolized by red crosses like the red crosses which were displayed on the wall in the study. We calculated an average intersection point for each pointing target. These were calculate over all intersections points I_1 with relay to the same pointing target. The visualization represented all average intersections point by blue dots. The visualization also shows is the distance of the average intersection point and the pointing target, this is represented by a yellow arrow.

Now we look at the three Figures 4.4, 4.5 and 4.6 separately. First we look at the direction of the arrows, we can see in the four sub figures (a), (b), (c) and (d) the arrow all has the same direction. So for the same ray cast technique we have in general the same displacement behavior. The displacement behavior has a direction and a length from the pointing target. For index finger ray cast and forearm ray cast it is same direction, top left of the pointing target. For head finger ray cast the direction is bottom right of pointing target. These figures do not give early possibilities to say something about the absolute of the arrows. The ranges of the figures are only the same for one sub figure group. So only sub figures (a) to (d), in one figure, have the same axis length.

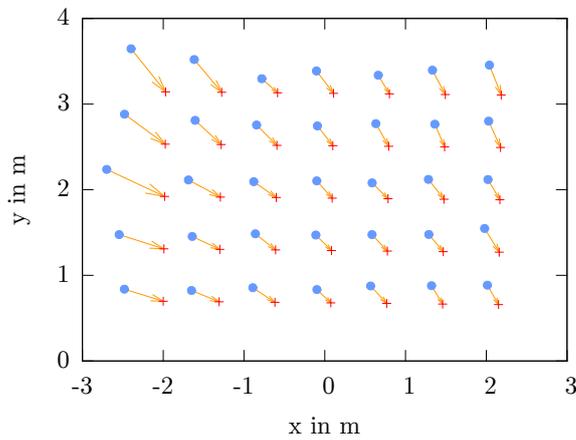
4. Results



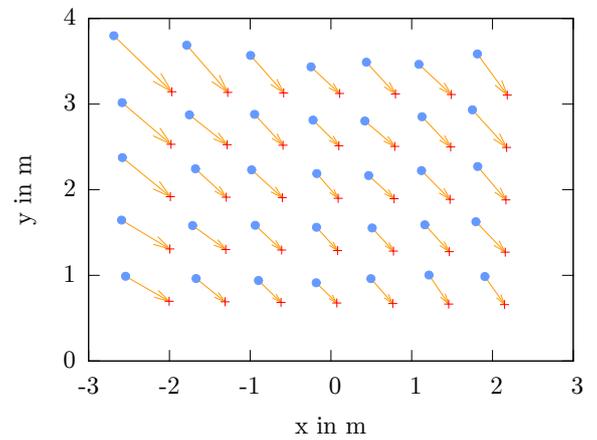
(a) Position: three meter standing



(b) Position: three meter sitting



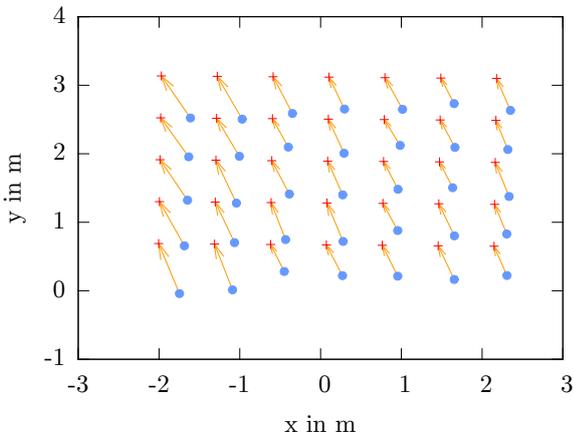
(c) Position: two meter standing



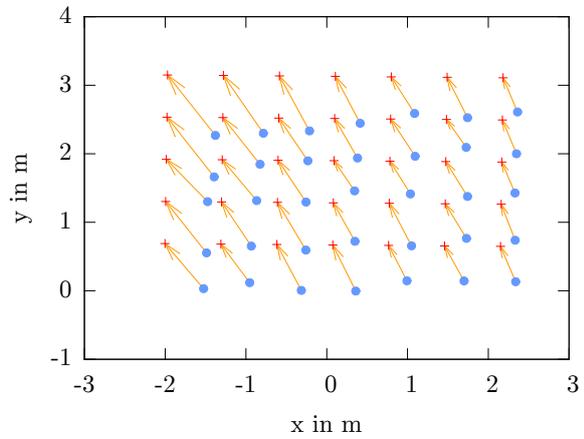
(d) Position: two meter sitting

Figure 4.4.: The absolute distance between the average intersection point (blue dot) and the pointing targets (red cross) for the index finger ray cast technique.

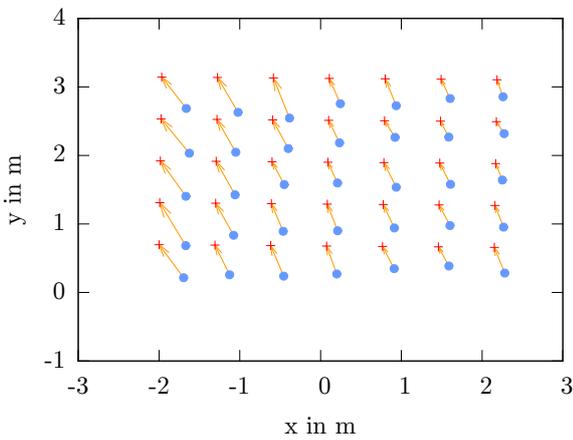
After knowing that the directions of the displacement are the same, we look at the length of the displacement. The two meter distance figures (c) and (d) compared to the three meter figures (a) and (b) are always shorter when we only compare the same body posture. So we first compared (a) with (c) and then (b) with (d). If the distance get increased the length of the displacement and thus the error increases as well. So we can say the distance has a significant influence on the absolute error. Finally, we can compare the same body posture at two and three distance. However, the height of the participants is different when the participants sit or stand, so the result is different input modality. So they are not comparable.



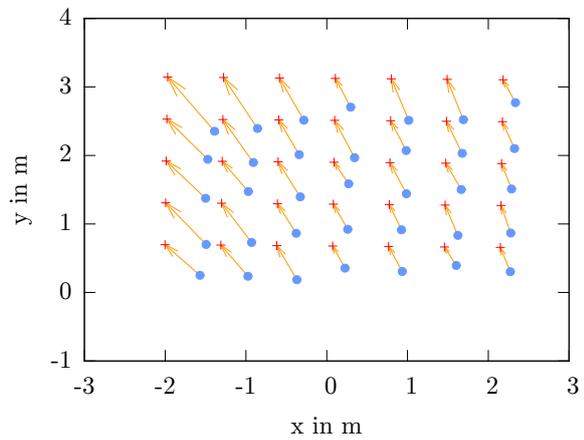
(a) Position: three meter standing



(b) Position: three meter sitting



(c) Position: two meter standing



(d) Position: two meter sitting

Figure 4.5.: The absolute distance between the average intersection point (blue dot) and the pointing targets (red cross) for the head finger ray cast technique.

In the next steps we analyze the data which are visualized in the sub figures, without comparing them. We have shown that all deviations extend in the same direction. We found one exception in the forearm ray cast technique at two meter sitting. There are three points in the upper left corner which are not in the same direction. We calculated these points over the average behavior over all participants. Thus, it must be a problem at recording time. This could be arising because of too bad camera overlapping. However, this anomaly will be part of the forearm model calculation. This is done because we cannot clearly say where it comes from.

So now the different length of the displacement can analyze. The maximum and minimum length of the arrow will be the first indicator. It is possible to identify the longest arrow over

4. Results

all figures at the same position, in the upper left corner. The minimum is harder to identify but in average the shortest length is in the lower right corner. Indeed it is possible to identify a pattern over all arrows. The error that is symbolized by the arrows is growing from the lower left to the upper right corner. The arrows which are orthogonal to this have the same length. This is the base behavior overall data visualized in the Figures 4.7, 4.8, 4.9, 4.10, 4.11 and 4.12. Figure 4.11 and 4.12 is out of line there we have the biggest standard deviation, see Table 4.2. So these results are not that meaningful.

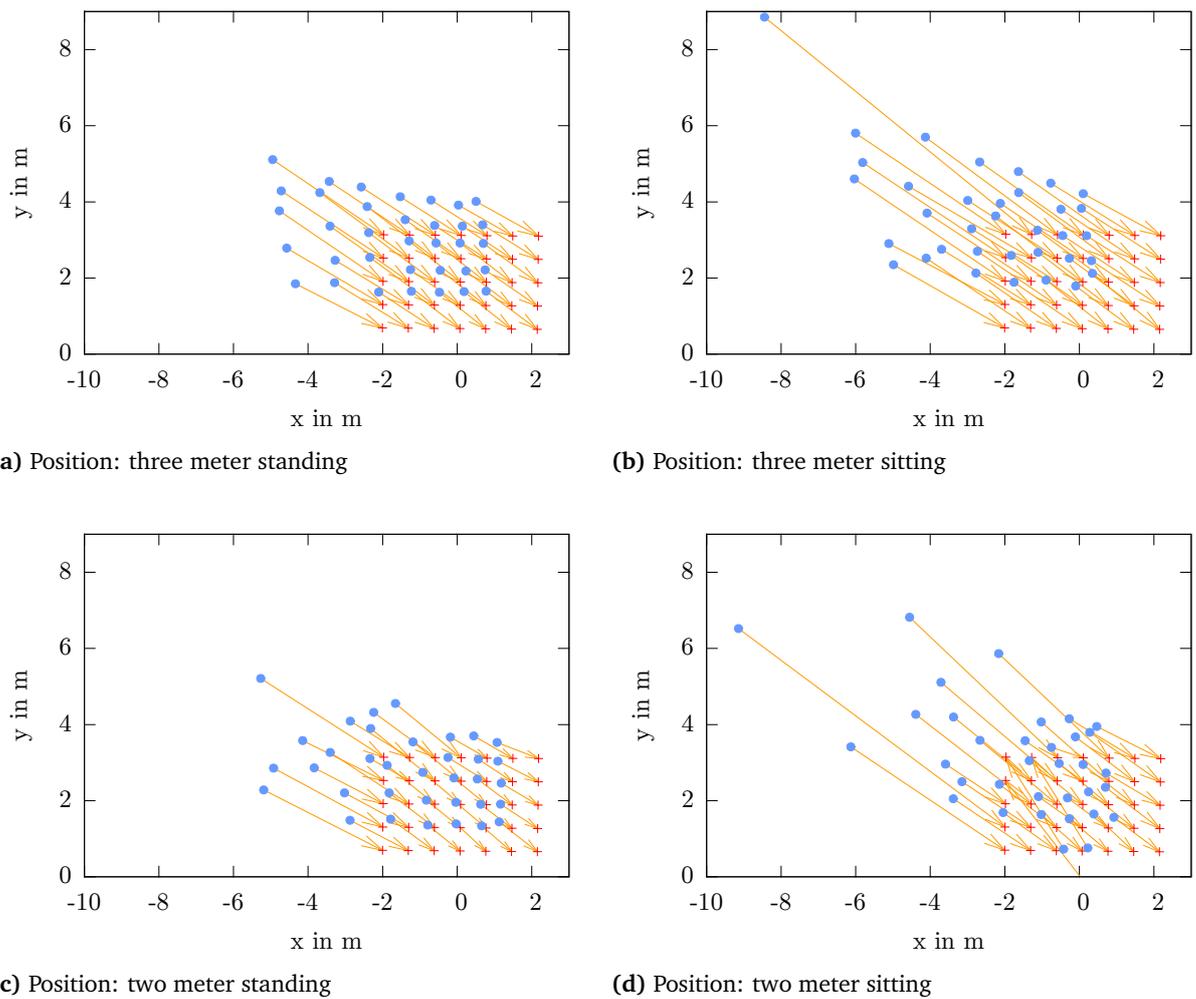


Figure 4.6.: The absolute distance between the average intersection point (blue dot) and the pointing targets (red cross) for the forearm ray cast technique.

4.3. Angle Preparing

The goal is to create a model that is not influenced by the distance to the Object. Because that, the distance is not known in general. For a global model it is necessary to combine all positions. With absolute values it is not possible to generate a global model. Because in Section 4.2 we have shown that the distance has an influence to the absolute values. The idea is to remove the distance out of the model. For that it is necessary to generate angle instead of absolute distances. These are relative to the object. Because of that, the greater distance and the same Angle the distance of the rays will grow. This is the behavior we like to have for our general model. However, still needs to be investigated whether the four different positions have the same gesture behavior. If this is not the fact, it is not possible to have a general model. In the next spats we first calculate a data set that includes the angles for each body gesture. After that we will test if the positions have the same behavior. This will be done by using cross-validation. This is described in Section 4.4.2.

To calculate the angles a plane is needed which describes the body posture. For this purpose, the center $B3$ of the two shoulder markers is used. The center together with the two hip markers $B1$ and $B2$ creates a new plane. This plane is the required 'body plane'. Figure 4.2 shows a marker sketch.

Next, we will describe how to calculate the angle. For this purpose first it is important to know how we use two angles to describe the direction the pointing direction. The root of the description is the person how perform the pointing gesture. The first angle α_{lr} describes the deviation to the left or to the right. The second angle α_{bt} describes the deviation to the bottom or to the top. Both angles have a value range from -90 to $+90$ degrees. $\alpha_{lr} = 0$ and $\alpha_{bt} = 0$ describes the straight line that goes from the person forward. $\alpha_{lr} = -90$ to $\alpha_{lr} = +90$ goes from the left to the right. $\alpha_{bt} = -90$ to $\alpha_{bt} = +90$ goes from directly down to straight up. With this, the entire area can be described which is in front of the person.

After calculating α_{lr} and α_{bt} it is necessary to calculate the correction angles Δ_{lr} and Δ_{bt} . These Δ angle are the angle β in Figure 4.2 where the left/right calculation is visualized. Δ_{lr} and Δ_{bt} have to be calculated separately for left/right and bottom/top movement.

With this data we create a new data set. This model includes all pointing gestures. It is a three-dimensional matrix. The dimensions give information about the position, the pointing targets which are projected on the wall and the multiple gesture count, $DS_1 := Matrix(position, girdId, multibleCount)$. The 'position' can have values from 1 to 4 to represent the four positions. The 'girdId' has values from 1 to 35 for the 35 different pointing targets which was projected on the wall. The 'multibleCount' is the count of a gesture with the same pointing target. So is depends on the 'girdId' and the repetition. However, the maximum is participants times point repeat time = $12 * 3 = 26$. The element in the matrix is a tuple with four values, $t := (\alpha_{lr}, \alpha_{br}, \Delta_{lr}, \Delta_{bt})$.

4. Results

On the base of DS_1 the values in Table 4.2 are calculated. These are the average correction angles and the average standard deviation (SD) based on the data set DS_1 . We used the Equations 4.2 and 4.3 for this purpose. The parameter x can be one of the tuple t variable. The function $mean()$ calculates the average value of a given list. The function $std()$ calculates the SD of a given list. \cup creates a new list. Here list induces all elements x out of the tuple which are in the Matrix DS_1 on entry i, j .

$$(4.2) \quad meanOf(x) = \frac{\sum_{i=1}^4 \sum_{j=1}^{35} mean(\bigcup_{k=1}^{maxCount} DS_1(i, j, k).x)}{4 * 35}$$

$$(4.3) \quad stdOf(x) = \frac{\sum_{i=1}^4 \sum_{j=1}^{35} std(\bigcup_{k=1}^{maxCount} DS_1(i, j, k).x)}{4 * 35}$$

Method	$meanOf(\Delta_{lr})$	$stdOf(\Delta_{lr})$	$meanOf(\Delta_{bt})$	$stdOf(\Delta_{bt})$
Index finger ray cast	6.2520	6.7354	-6.5067	9.4031
Head finder ray cast	-5.0168	5.3294	10.7086	13.2891
Forearm ray cast	26.7873	14.5562	-19.5630	17.0659

Table 4.2.: The table includes the average angle and the standard deviation of the correction angle Δ_{lr} and Δ_{bt} .

In the next steps we build a new data set out of DS_1 . This new data set DS_2 is defined by:

$$(4.4) \quad DS_2 := Matrix(position, girdId)$$

The multiple gestures dimension is removed and replaced by the average gesture posture. The Equation 4.5 shows the coherence. $mean()$ calculates the average tuple out of a tuple list. For the result tuple the elements are calculate separately.

$$(4.5) \quad DS_2(position, girdId) = mean(\bigcup_{k=1}^{maxCount} DS_1(position, girdId, k))$$

4.4. Model Fitting

We want to develop a general model, where the distance and the posture have no influence. This means we want to combine the four positions. We can do this if the positions does not affect the data. To show this we look at the average deviations for each position. We look at these values separately for each ray cast technique.

After we prepared our data set DS_2 we can generate our model m . DS_2 is defined by the Equation 4.4. The Equation 4.6 describes the model $m(\alpha_{lr}, \alpha_{bt})$. The first input value is the left/right angle α_{lr} and second value is the bottom/top angle α_{bt} . The model includes two functions. $f_{\Delta_{lr}}$ calculates the left/right correction angle Δ_{lr} . $f_{\Delta_{bt}}$ calculates the bottom/top correction angle Δ_{bt} . In general this function looks like $f(\alpha_{lr}, \alpha_{bt}) = z$ with $\alpha_{lr}, \alpha_{bt} \in \mathbb{R}$. However, in this case α_{lr} and α_{bt} are $\in [-90, 90]$, so these is the relevant range. The result z of the function is an correction angle.

$$(4.6) \quad m(\alpha_{lr}, \alpha_{bt}) = \begin{cases} f_{\Delta_{lr}}(\alpha_{lr}, \alpha_{bt}) & = \Delta_{lr} \\ f_{\Delta_{bt}}(\alpha_{lr}, \alpha_{bt}) & = \Delta_{bt} \end{cases}$$

In the further course we will use four different functions for $f_{\Delta_{lr}}$ and $f_{\Delta_{bt}}$, see Table 4.3. These functions have free coefficients which will be set later. f_1 is the only one-dimensional function. This is an attempt to justify the necessity for two-dimensional functions. The next three functions are two-dimensional. f_2 is a simple plane, the simplest polynomial that can describe the complete \mathbb{R}^2 . f_3 and f_4 are complex mathematics two-dimensional functions with more coefficients to have the ability to fit the data better. The functions were selected after a optical analyze of the 3d point cloud, here for see (a)-figures of the Figures 4.7, 4.8, 4.9, 4.10, 4.11 and 4.12. The result of this analysis is that a polynomial with a degree height of $(3 + n)$ with $n \in \mathbb{Z}$ does not match the point cloud. Because most of the figures looks more like a parable or a polynomial with a highest degree out of $(2 + n)$ with $n \in \mathbb{Z}$. More coefficients fit always better. However, with to match coefficients the function dose not describe the trend any more, it will than describe the noise of the data set. The trend is the general behavior of the data set. The noise can be described as the errors out of the measurement errors. Calculating the coefficients is a mathematical problem called 'curve fitting', this is described next.

4. Results

Position	function	Coeff
$f_1(x)$	$a * x^2 + b * x + c$	3
$f_2(x, y)$	$a * x + b * y + c$	3
$f_3(x, y)$	$a * x^2 + b * y^2 + c * x * y + d * x + e * y + f$	5
$f_4(x, y)$	$ax^4 + by^4 + cx^3y + dxy^3 + ex^3 + fy^3 + gx^2y^2 + hx^2y + ixy^2 + jx^2 + ky^2 + lxy + mx + ny + o$	15

Table 4.3.: The four fitting functions we use to generate our models. The coefficients are determined with the help of $fit()$.

4.4.1. Curve Fitting

In the next steps we will calculate the coefficients for the two functions $f_{\Delta lr}$ and $f_{\Delta bt}$ with the four different polynomials in Table 4.3. We will use a curve fitting technique. In further we will use a technique called ordinary least squares (OLS), it is a regression method. OLS is the standard technique for curve fitting. OLS is an iterative procedure which tries to minimize the error. All values are calculated with a program called 'MatLab'¹, where OLS is also the standard method. OLS predicts coefficients for function that 'best fit' the observed data. To use a curve fitting technique it needs a fitting function and data set as a input. A curve fitting function is defined as:

$$(4.7) \quad fit(y, d) = \vec{\alpha}$$

The first input y is a N -dimensional function. The function is defined as:

$$(4.8) \quad y(x_1, x_2, \dots, x_N)$$

$y_{\vec{\alpha}}$ is the function y with defined coefficients $\vec{\alpha}$. The second input d is the data set where the function y should fit to. d is a matrix with n rows and $N + 1$ columns. $d_{(a,b)}$ selects the value in row a and column b . We have n support points to fit the function. The output $\vec{\alpha}$ is a set of coefficients. If the count of free coefficients in function y is m then applies $|\vec{\alpha}| := m$. Curve fitting technique calculates the residual r as the distance between predict and observed model parallel to the x_{N+1} axis.

$$(4.9) \quad r := y_{\vec{\alpha}}(d_{(a,1)}, \dots, d_{(a,N)}) - d_{(a,N+1)} \text{ with } a \in [1..n]$$

¹<http://www.mathworks.de>

r is part of the error of the model. One option to define a global error is the sum of the squared errors (SEE). SEE is the variation which is not described by the model. SEE is defined as follows:

$$(4.10) \quad SSE := \sum_{i=1}^n r_i^2$$

Curve fitting is always trying to fit the data set in the 'best' way. When using OLS, best fitting is when the sum of the squared errors (SSE) is minimal. So OLS searches for a $\vec{\alpha}$ where SSE is minimal.

$$(4.11) \quad \min_{\vec{\alpha}}(SEE) = \min_{\vec{\alpha}} \sum_{i=1}^n r_i^2$$

Two other possible methods to describe the error of the model are $RMSE$ and R^2 . $RMSE$ is the root mean square error. R^2 is a coefficient of determination. This value is between 0 and 1. With $R^2 = 1$ the data set d is fully described by the function $y_{\vec{\alpha}}$

$$(4.12) \quad RMSE := \sqrt{\frac{1}{n} \sum_{i=1}^n r_i^2}$$

$$(4.13) \quad SST := \sum_{i=1}^n d_{(i,N+1)}$$

$$(4.14) \quad R^2 := 1 - \frac{SSE}{SST}$$

OLS is inappropriate when the observed data set contains outliers. Outliers are observation that are atypical or that are far removed from the rest of the data points. However, after we prepared the observed data all possible outliers should be gone, because we take the average angle over all pointing gesture with the same pointing target. Otherwise we had to use OLS with a robust regression technique like least-absolute-residuals (LAR). This technique minimize

the sum of the absolute value of the errors: $\min_{\alpha}(SAE)$. As result the function fits bad to the outlier and great to the rest of the data set.

$$(4.15) \min_{\vec{\alpha}}(SAE) = \min_{\vec{\alpha}} \sum_{i=1}^n r_i$$

In our case we used the data set DS_2 as input, which is generated out of the study data, see Section 4.3. In this case y are four different functions, see Table 4.3. One is 1-dimensional and three are 2-dimensional. OLS will predict coefficients that fit the data best. This means that the function predicts us a correction angle for every input, without having data in the data set for this special input. DS_2 has to be transformed to match the input of the $fit()$ function. By having three different ray cast techniques, we have also three different data sets DS_2 . Each data set for each ray cast technique. These are stored in $DS_{2,b}$ with $b \in [1, 3]$.

All tuple that are constraint in $DS_{2,b}$ has to be copied in a matrix d_b . So as result we have three matrices. They are generated in the way that for each tuple of the old matrix $DS_{2,b}$ the new matrix contains this tuple in one row. In our case d_b has 140 rows and 4 columns. When fitting the Δ_{lr} function it is necessary to leave the fourth column out. When fitting Δ_{bt} column three has to be left out. Then the first two columns are the x and y value the third column are the z values. Now it is possible to calculate all coefficients, for three different ray cast techniques and four differed fitting functions.

$$(4.16) \vec{\alpha} = fit(f_a, d_{b,c}) \text{ with } a \in [1, 4], b \in [1, 3], c \in \{\Delta_{lr}, \Delta_{bt}\}$$

The results for all combinations are stored in Table A.2, A.3 and A.4. The error values R^2 , SEE and $RMSE$ are shown in Table 4.4. All functions are shown in the Figures 4.7, 4.8, 4.9, 4.10, 4.11 and 4.12.

		$f_{\Delta lr}$			$f_{\Delta bt}$		
		R^2	SEE	RMSE	R^2	SEE	RMSE
IFRC	f_1	0.1086	353.35	1.6060	0.1956	445.03	1.8023
	f_2	0.1771	326.20	1.5431	0.1349	478.64	1.8692
	f_3	0.2357	302.95	1.5036	0.4047	329.38	1.5678
	f_4	0.3685	250.31	1.4151	0.4446	307.30	1.5679
HFRC	f_1	0.6375	236.32	1.3134	0.1019	873.23	2.5247
	f_2	0.6698	215.29	1.2536	0.4744	511.01	1.9313
	f_3	0.7072	190.87	1.1935	0.5540	433.68	1.7990
	f_4	0.7250	179.29	1.1977	0.5744	413.77	1.8194
FRC	f_1	0.1218	5.1433e+03	6.1272	0.0947	4.5206e+03	5.7443
	f_2	0.2588	4.3410e+03	5.6291	0.2935	3.5280e+03	5.0746
	f_3	0.3813	3.6234e+03	5.2000	0.4089	2.9518e+03	4.6935
	f_4	0.4744	3.0782e+03	4.9624	0.4994	2.4996e+03	4.4718

Table 4.4.: The fit results of the functions in Table 4.3 with index index finger ray cast, head finger ray cast and forearm ray cast.

A function with infinity coefficients dose describe 100% of the Noise. This is called overfitting. This happens because the result function matches all data points. We see no ability to say if we have to little or too much coefficients, because in generally it is not known what function is searched for. Because of this cross-validation (CV) is used to get a better understanding of how well the data set fits the curve. CV measure the quality or accurately of a function in relation to new input [DK82]. CV is part of data mining. CV is used in settings where the goal is prediction and one wants to estimate how accurately a predictive model will perform in practice. This is a technique where the data are divided into complementary subsets. The first subset (training set) trains the model. With the other subset (testing set) the model is tested and an error is calculated. This error is the distance between model and test data. In our case we use leave-one-out cross-validation (LOOCV) [HBM03]. This is a special version of cross-validation where the data set is divided in p parts. And the global error is the average error over the errors of all p parts. The SD is also interesting. A small SD represents that the errors of the p different models are nearly the same. This means that the new data have no big influence to the model, so the model preforms well. If the test data have a height SD the model represents to much noise and to less trend.

$$(4.17) \quad (error_{mean}, error_{sd}) = LOOCV(y, train, test)$$

y is a function like it is defined in Equation 4.8. When building the data set DS_2 we used average values. The average values were used because of differed pointing gestures by pointing

on the same point. These average values will be used to train the model, these are the input values $train$. However, to test the model we used the individual pointing gestures. These are bundled in the input variable $test$. $train$ and $test$ are both arrays of the same length, $|train| = |test| = x$. Each element of the two arrays contains a matrix with the support points to fit and to test. p is still the count of the different parts. So it will be trained and testes p times. The fit -function is called to generate p different model confections sets $\vec{\alpha}_i$:

$$(4.18) \quad \forall i \in [1, p], fit(y, train \setminus \{train_{(i)}\}) = \vec{\alpha}_i$$

These x different model will then be tested with the left out data $test_{(i)}$.

$$(4.19) \quad \forall i \in [1, p], y_{\vec{\alpha}_i}(test_{(i,1)}, \dots, test_{(i,N)}) - test_{(i,N+1)} = e_{last+1}$$

e is a array where all error values are stored into. Then is $error_{mean} = mean(e)$ and $error_{sd} = std(e)$.

4.4.2. Cross-Validation with Positions

In Section 4.3 we explain that it is necessary to test whether the pointing gestures have the same behavior four different positions. This is tested with LOOCV. Here for the data set DS_2 is used. The following calculations are done with the data sets of the three different ray cast techniques. In this case the data sets are split in four parts, so $p = 4$. DS_2 is split in four parts, each part contains all average gesture of one position. The four parts are stored in the array $train$. Also DS_1 is spitted in four parts and stored in $test$. Each will be one time the test set for the model. The model is created from the remaining parts. However it is still unknown which of the four functions in Table 4.3 fits the data set best. So LOOCV is done once for each function. This has to be done for all three different ray cast techniques. In Table 4.5 is shown the average error $error_{mean}$ and the standard deviation of the $error_{sd}$. With average error $error_{mean}$ near zero it is possible to say the four different positions works the same way. So it is possible to generate a model where all data can be used as input.

		IFRC		HFRC		FRC	
		$error_{mean}$	$error_{sd}$	$error_{mean}$	$error_{sd}$	$error_{mean}$	$error_{sd}$
$f_x(\Delta lr)$	f_1	0.0006	1.2999	-0.0028	1.3282	0.0089	4.6469
	f_2	-0.0077	1.2934	-0.0059	1.2932	-0.0168	3.8627
	f_3	-0.0104	1.3444	-0.0065	1.2183	-0.0070	3.5830
	f_4	-0.0263	1.4098	-0.0097	1.2621	0.1842	3.6332
$f_x(\Delta bt)$	f_1	-0.0216	1.7622	-0.0024	2.3984	-0.0539	4.4688
	f_2	-0.0212	1.7263	-0.0020	1.8439	-0.0531	3.4912
	f_3	-0.0022	1.4405	-0.0057	1.6298	-0.0198	2.9986
	f_4	-0.0787	1.5270	0.0295	1.6804	-0.0997	2.9099

Table 4.5.: The result of a leave-one-out cross-validation with four different parts, these are the four different positions on its own. All values are generated by using $LOOCV(y, train, test)$, see Equation 4.17. The results are independently for Δlr and Δbt as well as the four functions in Table 4.3 and the three ray cast techniques.

Now follows the interpretation of the Table 4.5. The mean error over all different functions is between -0.01 and 0.19. This is a indicator to say that the different positions do not have a influence on the data when they are expressed with angles. So by using angles it is possible to generate a model where it is not necessary to have the distance as an input variable. To show this was the goal of this section. However, the result gives another indicator. An indicator which functions preforms best. We use the $error_{sd}$ for this purpose. A standard deviation of zero is the best possible result. f_3 has four of six times the best performing $error_{sd}$. One time it is the second best result, it +3.4% away from the best result. The second time where f_3 is not the best result it is the third best result with a $error_{sd}$ of +3.4%. So it is possible to say that f_3 represents the most trend and the less noise of this four functions. This can be said for $f_{\Delta lr}$ and $f_{\Delta bt}$. To confirm that f_3 is the best performing function it is necessary to do another CV. Where the 12 participants will be uses as test set independently. This is done in the next section.

4.4.3. General Model

In the further steps we do the evaluation of the four different fitting functions. It is tested how the functions fit the data set. More specifically, a statement is made as how exactly a model for individual participants is. This is done by using LOOCV. Here the data is split into 12 parts this is the count of participants we had in the study, so $p = 12$. This is the number of different participants. So the parts are built this way that all pointing gestures of one participant belong to one separate part. This is done with the DS_2 data for the $train$ array and with the DS_1 data

4. Results

set for the *test* data. The principle of the creation of the data set is the same as in the section above. So the following result in Table 4.6 are out of 12 times training and 12 times testing.

		IFRC		HFRC		FRC	
		$error_{mean}$	$error_{sd}$	$error_{mean}$	$error_{sd}$	$error_{mean}$	$error_{sd}$
$f_x(\Delta lr)$	f_1	-0.0136	7.5169	-0.0193	6.9514	0.1361	19.4717
	f_2	-0.0316	7.5553	-0.0242	7.1673	-0.1552	17.1190
	f_3	0.0032	7.5621	-0.1501	7.0423	0.3973	16.1667
	f_4	0.9094	13.6730	-0.1240	7.1369	7.3326	66.6671
$f_x(\Delta bt)$	f_1	-0.0012	10.2137	0.0400	15.1175	-0.0176	22.1678
	f_2	-0.0068	10.2126	0.0647	15.4242	-0.0187	21.3413
	f_3	0.1077	9.9620	0.2209	15.3953	0.6956	21.5981
	f_4	-0.1992	11.9580	0.0008	16.9638	-11.7991	102.2813

Table 4.6.: The result of a leave-one-out cross-validation with twelve different parts, these are the twelve different participants on its own. All values are generated by using $LOOCV(y, train, test)$, see Equation 4.17. The results are independently for Δlr and Δbt as well as the four functions in Table 4.3 and the three ray cast techniques.

After calculating these values for all four functions it is possible to compare the results. When we look at the $error_{mean}$ it is possible to identify two outlier. When we use f_4 on the forearm ray cast data the $error_{mean}$ is bigger then 7, but the rest of all $error_{mean}$ are in rage $[-0.1552, 0.9094]$. So it is possible to say f_4 is inappropriate here. The next properties are hart to see. However, in Section 4.4.2 was shown that f_3 perform well. If we ignore f_1 , we can say f_3 preforms best. We can do this because it is the only one-dimensional function and works different. Then f_3 has 4 of 6 times the lowest standard deviation of the error. However, for unambiguous statement it is necessary to look at the global modules which was calculate in Section 4.4.1. This will be done in the next chapter.

4.5. Index Finger Ray Cast Fitting Graphics

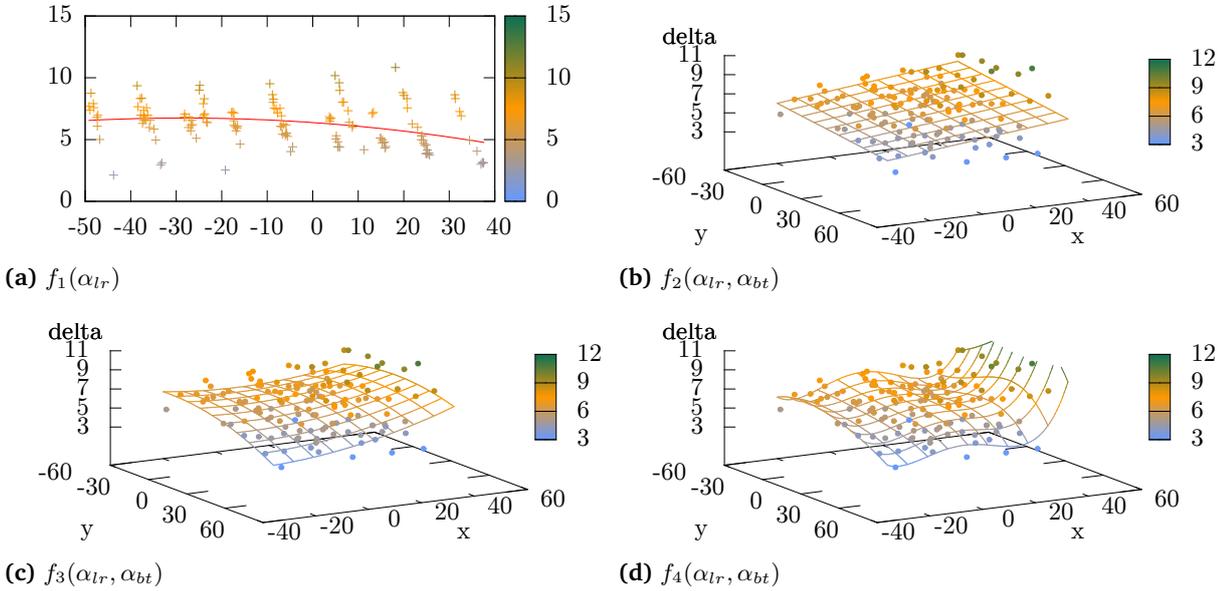


Figure 4.7.: All visualizations of the Δ_{lr} values for index finger ray cast (IFRC) technique with the functions in Table 4.3.

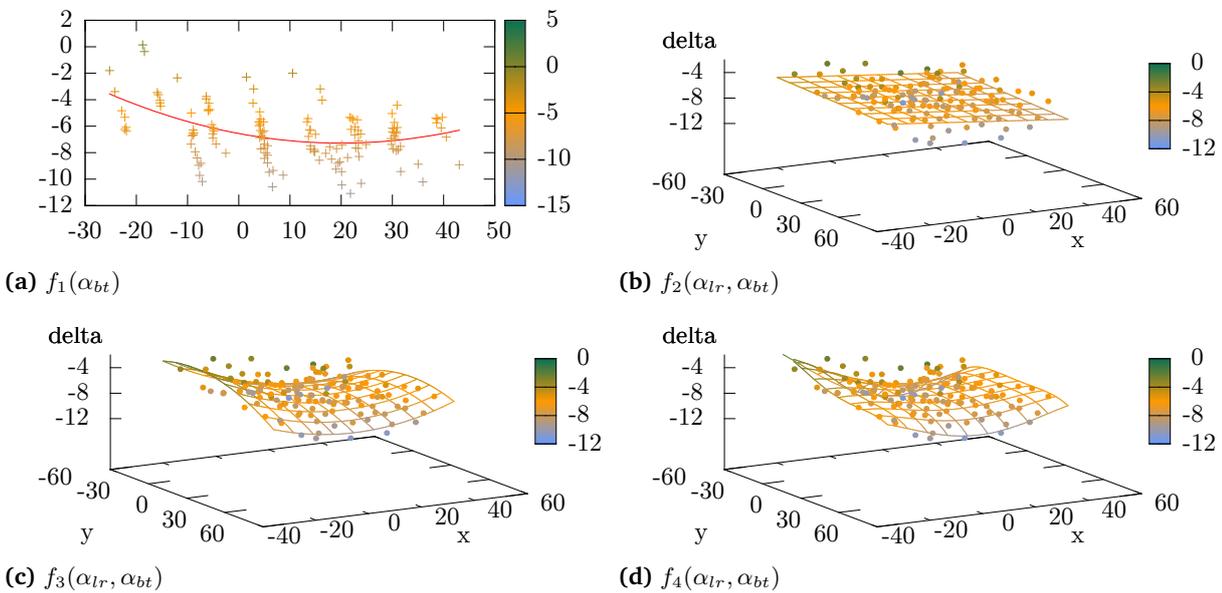


Figure 4.8.: All visualizations of the Δ_{bt} values of index finger ray cast (IFRC) technique with the functions in Table 4.3.

4.6. Head Finger Ray Cast Fitting Graphics

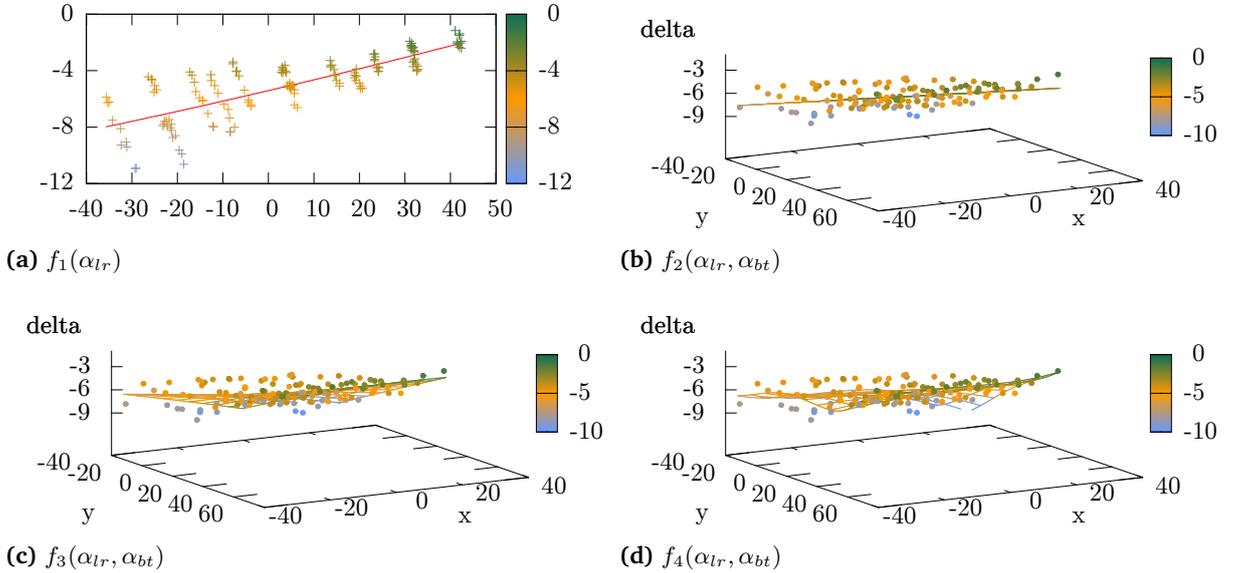


Figure 4.9.: All visualizations of the Δ_{lr} values for head finger ray cast (HFRC) technique with the functions in Table 4.3.

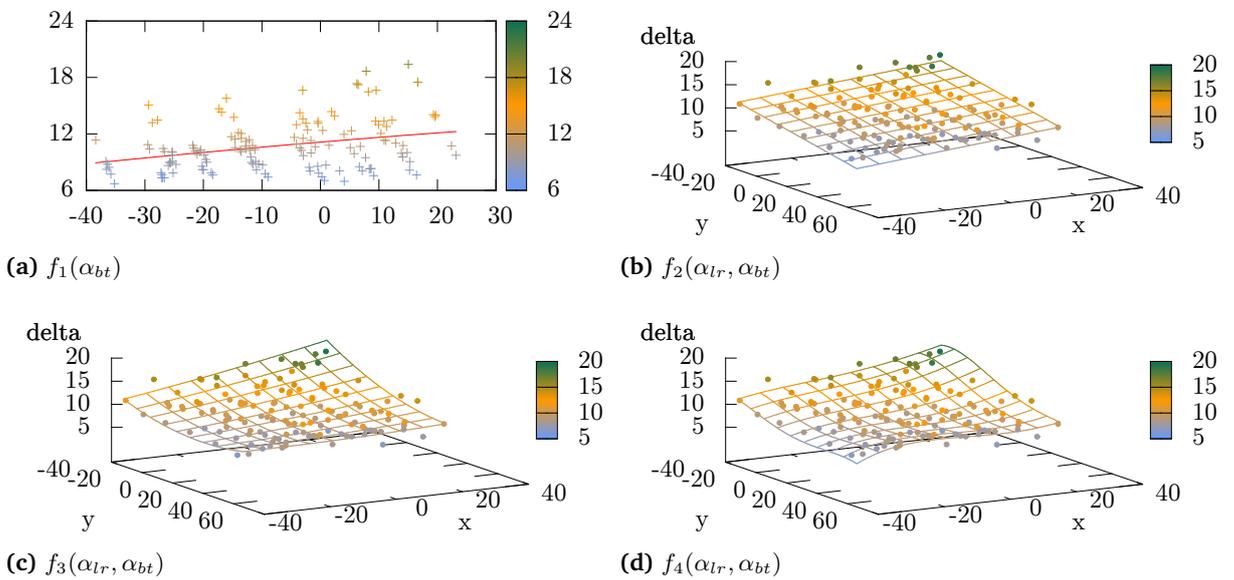


Figure 4.10.: All visualizations of the Δ_{bt} values for head finger ray cast (HFRC) technique with the functions in Table 4.3.

4.7. Forearm Ray Cast Fitting Graphics

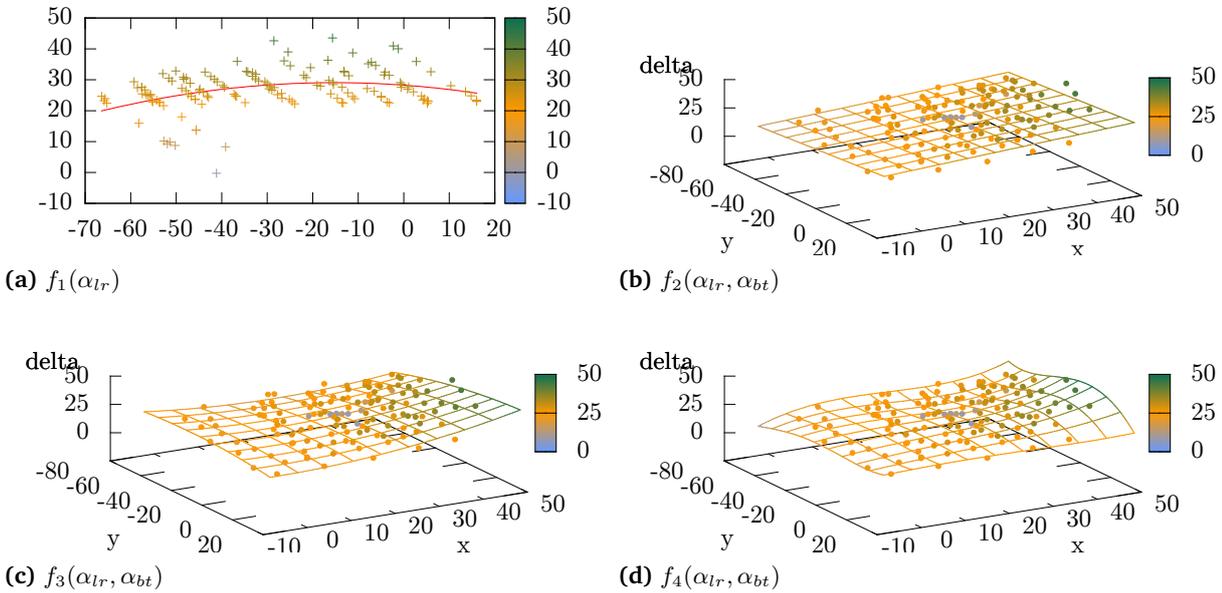


Figure 4.11.: All visualizations of the Δ_{lr} values for forearm ray cast (FRC) technique with the functions in Table 4.3.

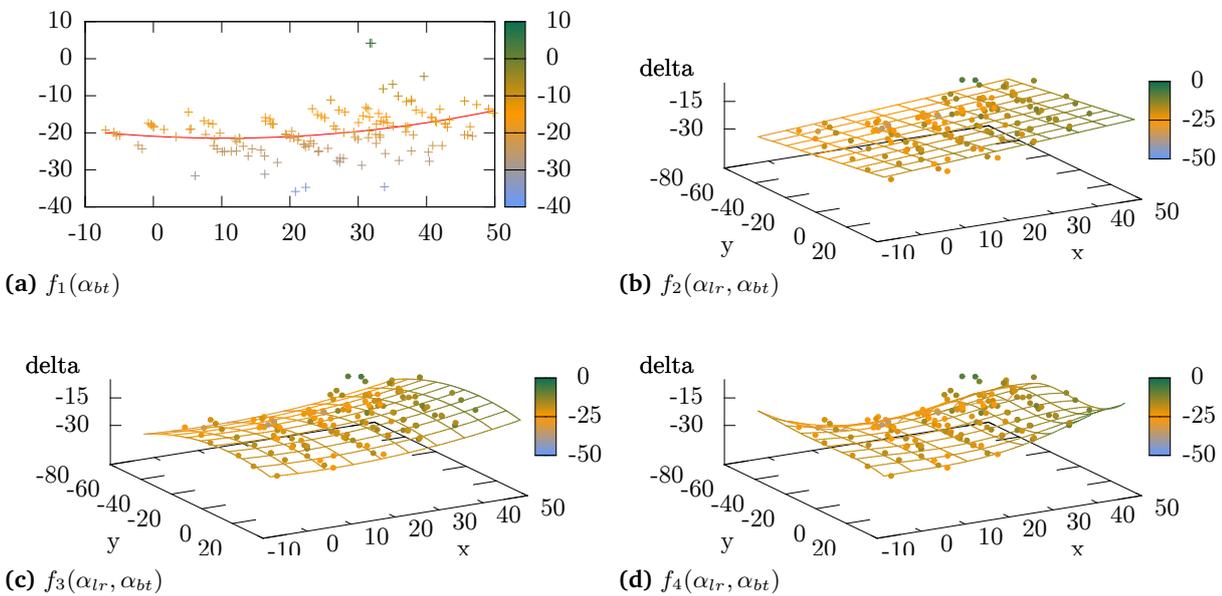


Figure 4.12.: All visualizations of the Δ_{bt} values for forearm ray cast (FRC) technique with the functions in Table 4.3.

4.8. Summary and Discussion

In this chapter, we have created four different models with the help of four different functions. This we have done for three different ray cast techniques. We have shown that when we would use more coefficients the accuracy increases. So we still do not know how many coefficients we need to describe the trend in our data set without describing noise. Next we argued which polynomial describes the human behavior in the best way. Therefore we look not only on the mathematical analysis. However, also on where the data came from and what is a logical function for that. The two input variables are angle which describes the arm posture. An arm movement is a continuous movement with some limitations: mobility and the body where the arm cannot move to. In the given input data set both limitations have played no role. Because the point set was chosen so that the limitations have no influence to the data set. More than that, the arm movement in general is linear in the not limited zone. The third model input variables are the pointing angle. The values are linked to the linear arm movements which have been done while pointing on the pointing targets. The pointing targets are arranged in regular grid. The x values are linear as well as the y values. Now we look at f_4 . This function is a polynomial, the degree of the polynomial f_4 is 4. Compared to f_3 where the degree of the polynomial is 2, f_4 has more theoretical local minima and maxima. This property helps to fit the data sets better. Because it fits the minima and maxima into the region where the data set is defined through values. So maybe the trend is a straight line and the fit has a too high degree then the fitting function jumps around the trend and produces noise. This is what we can see in all sub figures (d) in the Figures 4.7, 4.8, 4.9, 4.10, 4.11 and 4.12. This behavior is called overfitting. With all degrees higher than 4 this problem gets bigger and bigger. Now we are going back to input variable and why the trend has logical to be under degree 4. The input values are a neural linear pointing gesture. Then it is not logical that the error a pointing gesture has is something with a lot of minima and maxima. It is more logical that we have growing errors at the edges of the movement space and minima at the middle of the space.

In Table 4.2 we have figured out that overall data we have a $stdOf(\Delta_{lr})$ and a $stdOf(\Delta_{bt})$ which is near the mean value of $meanOf(\Delta_{lr})$ and a $meanOf(\Delta_{bt})$. This indicates the range of different pointing gestures that have been performed by the user's in the study. So in average our models compensate all systematic displacements. However for one sample it still could perform not that good. This comes from the variety of different gestures. To show the variety of gestures was one part of our study. This we have provoked by a not precise statement, as the participants should point to pointing targets. We say only that should point on the pointing targets, not how. All participants have point with the index finger extended but the right arm not always was extended. This is where the range of the different pointing results came from.

The last thing we want to make a note to our small range of age of the participants. Like [Hav03] has shown pointing is a behavior which humans know from the early childhood. That

is why we say that this behavior has strengthened so what it does not change over time of life. This applies as long as until disease-related changes in the behavior change fundamentally.

5. Conclusion

Previous work interpret the human pointing gesture by ray cast techniques and they measured the error between target and intersection point. We also measured the error but out of these errors we created model to compensate these errors. Therefor we used a fixed grid of points to compare the differences pointing gestures. To compare these differences we only used right-handed participants. The models we created are only usable for right-handed persons. This is true if the assumption is correct that right-handed behave symmetrical to left-handed persons. It is possible to reflect the data of left-handed persons on the vertical body axis so that they are virtual right-handed and then it is possible to use our models.

In addition, we let the models are based on different functions. We have used leave-one-out cross-validation to determine the goodness of the models. We have shown that functions with a degree greater than two fit better. However the noise of the models are higher. We have also argued that functions with a degree greater than two do not represent human behavior. Which is why we recommend a fitting function with a degree of two. Since a smaller degree reflect the measured values worse. We also recommend to use function f_3 with the two parameters Δ_{lr} and Δ_{bt} , see Table 4.3. We recommend this because this function offers the maximum degree of freedom at a function of the second degree.

Now we can define the global model like the general model m we defined in Equation 4.6. Here we set a specified function, like we argued we use f_3 , so it follows:

$$(5.1) \quad f_3(x, y) = a * x^2 + b * y^2 + c * x * y + d * x + e * y + f$$

$$(5.2) \quad m(\alpha_{lr}, \alpha_{bt}) = \begin{cases} f_{3, \Delta_{lr}}(\alpha_{lr}, \alpha_{bt}) & = \Delta_{lr} \\ f_{3, \Delta_{bt}}(\alpha_{lr}, \alpha_{bt}) & = \Delta_{bt} \end{cases}$$

The coefficients for the functions are represented in three tables, for index finger ray cast (IFRC) see Table A.2, for head finger ray cast (HFRC) see Table A.3 and for forearm ray cast (FRC) see Table A.4.

Next is the evaluation of the different ray cast techniques. The models which we developed correct the mean error of the deviation from the gesture to the pointing target. For this reason,

this mean error provides no good indicator of the quality. So we now use the SD as indicator of the quality. The SD describes the variance of the pointing gestures. In Table 4.2 we present the SD of Δ_{lr} and Δ_{bt} . We have demonstrated that IFRC has the smallest sum of SD followed by HFRC and then FRC. Thus, the input data IFRC have the smallest scatter of the gestures. However, this scattering depends largely on that the subjects had to make no defined gesture.

For this reason, the mean error and the SD of the LOOCV is more important. These two are the indicators for the quality of the models. So we compare the values for f_3 . It follows that the sum of mean error of HFRC is with 0.0122 the smallest but IFRC has almost the same mean error with 0.0126. However the sum of SD is smaller for IFRC. Thus, the IFRC model represents the sum of the gesture best but HFRC has almost the same values.

Our models do not contain system data of our system. The only components that influence our models are the positions of the marker. Units of measure does not affect the models because we use angles for our calculations. Thus, the models can be transferred to other system settings. The 6DOF tracker we used for our study is time-consuming to install. Likewise, the user must always be equipped with markers. For this reason, it is necessary to look for easier ways that allow us to track a user. One possibility would be to use a Kinect¹. From the depth images it is possible to extrapolate spatial coordinates. These can be used to make use of our models. When calculating the required coordinates is important to remember that points that are close to each other have a greater potential the small inaccuracies have large impacts on the correctness. Moreover, on a depth image small objects are more difficult to detect than larger objects. Thus is specific to the Kinect that head finger ray cast (HFRC) technique should be chosen. Since head and index fingers are further apart as two points on the index finger. In general forearm ray cast (FRC) is an alternative but if we compare the SD it is worse than the other two techniques.

However we have demonstrated that interpretation of pointing gestures from distant can be improved with our models. Herewith a first foundation for improved interaction from the distance is placed. In door interaction like switching on and off light from a distance without a tool can be improved. As well as on screen interaction with for example a TV can be improved too.

Future Work

Before proceeding further, we recommend a more detailed investigation of the variation of pointing gestures. In our study all participants used index finger extended gesture to point on the objects. Some of them also extended their arms, but not all. We assume that this has an massive effect on IFRC technique. For better results with the IFRC technique it will be

¹<http://www.xbox.com/en-US/Kinect>

necessary to consider the orientation of the arm. Kendon [Ken08] present in his work that he cannot find a difference between index finger extended pone (palm down) and finger extended neutral (palm vertical), this phenomenon should be investigated.

It should also be investigated on pointing to moving objects. It should be grounded determined whether the movement has an influence on the gesture. This is important in order to implement a correct behavior for drag and drop from a distance.

For better evaluation of the models it is necessary to implement the models into a live system. Usability tests (e.g., by using the System Usability Scale [Bro96]) should be made. Also a fitts' law task should be made [Mac92]. As we have already pointed out, such a system can be implemented with the help of kinect.

Furthermore, a good interpretation of pointing gestures have a use for commercial collaborative virtual environments (CVEs) and other virtual reality (VR) simulations. In these environments, for example, objects can be selected or manipulated by a pointing gesture. With the help of voice commands, a new spectrum of possibilities opens up.

A. Appendix

Participant Nr.	1. Position	2. Position	3. Position	4. Position
1	2	1	3	4
2	3	4	1	2
3	3	4	2	1
4	2	1	4	3
5	2	1	4	3
6	4	3	2	1
7	4	2	1	3
8	1	4	3	2
9	1	2	3	4
10	1	3	2	4
11	4	3	1	2
12	3	2	4	1

Table A.1.: Random distribution of positions for each participant.

A.1. Study Questionnaire

Study survey

General questions

* Erforderlich

Participant Nr *
Teilnehmernummer

Name *

Age *

Job/Study? *
Beruf/Studiengang?

Dominant hand *

right

left

Do you wear glasses for driving? *
Benutzen sie zum Autofahren eine Brille?

Yes

No

Bereitgestellt von  Dieser Inhalt wurde nicht von Google erstellt und wird von Google auch nicht unterstützt.
[Missbrauch melden](#) - [Nutzungsbedingungen](#) - [Zusätzliche Bestimmungen](#)

Figure A.1.: The general questions about the participant in the study.

Study survey

* Erforderlich

Study survey

measurements

Körpergröße im stehen *

Körpergröße im sitzen *

Augenhöhe im stehen *

Augenhöhe im sitzen *

Schulterhöhe im stehen *

Schulterhöhe im sitzen *

Armlänge *

Fingerlänge *

« Zurück Weiter »

 33 % abgeschlossen

Bereitgestellt von  Google Drive

Dieser Inhalt wurde nicht von Google erstellt und wird von Google auch nicht unterstützt.
[Missbrauch melden](#) - [Nutzungsbedingungen](#) - [Zusätzliche Bestimmungen](#)

Figure A.2.: The input for the body measurements of the participants. In the measurements, the participants was helped.

Study survey

* Erforderlich

Study TLX

Part 1

Distance? *
Abstand?

2m
 3m

Position? *

Stehend
 Stuhl

How mentally demanding was the task? *
Wie geistig anspruchsvollsten war die Aufgabe?

1 2 3 4 5 6 7 8 9 10

Very Low Very High

How physically demanding was the task? *
Wie anstrengend war die Aufgabe?

1 2 3 4 5 6 7 8 9 10

Very Low Very High

How hurried or rushed was the pace of the task? *
Wie hastige oder gehetzt war das Tempo der Aufgabe?

1 2 3 4 5 6 7 8 9 10

Very Low Very High

Figure A.3.: The first part of the TLX with extra input to verify the position.

How successful were you in accomplishing what you were asked to do? *
Wie erfolgreich waren sie im Lösen der Aufgabe?

1 2 3 4 5 6 7 8 9 10

Perfect Failure

How hard did you have to work to accomplish your level of performance? *
Wie sehr mussten sie sich ansträngen, um ihre Leistung zu erreichen?

1 2 3 4 5 6 7 8 9 10

Very Low Very High

How insecure, discouraged, irritated, stressed, and annoyed were you? *
Wie unsicher, entmutigt, irritiert, gestresst, und verärgert waren Sie?

1 2 3 4 5 6 7 8 9 10

Very Low Very High

 50 % abgeschlossen

Bereitgestellt von  Google Drive

Dieser Inhalt wurde nicht von Google erstellt und wird von Google auch nicht unterstützt.
[Missbrauch melden](#) - [Nutzungsbedingungen](#) - [Zusätzliche Bestimmungen](#)

Figure A.4.: The second part of the TLX.

A.2. Function Coefficients

		a	b	c	d	e	f
$f_x(\Delta lr)$	f_1	-0.0004511	-0.02572	6.377	-	-	-
	f_2	-0.01948	0.02963	5.829	-	-	-
	f_3	-0.0005073	0.0006495	0.0005508	-0.03096	0.02328	5.937
	f_4	a = -6.563e-08, b = 4.244e-06, c = -5.768e-07, d = 7.184e-07, e = -1.093e-05, f = -6.583e-05, g = -8.402e-07, h = -9.145e-06, k = -5.413e-06 l = -0.0002923, m = -0.00266, n = 0.0005939, o = -0.01955, p = 0.05277, q = 6.247					
$f_x(\Delta bt)$	f_1	0.001821	-0.07256	-6.555	-	-	-
	f_2	-0.009618	-0.04063	-6.223	-	-	-
	f_3	-0.001286	0.002156	0.001135	-0.03557	-0.06799	-6.035
	f_4	a = -6.261e-07, b = -9.815e-07, c = 1.607e-07, d = -2.767e-07, e = -8.96e-06, f = 4.384e-05, g = 1.391e-06, h = -4.202e-05, k = 5.213e-06, l = -0.0004098, m = 0.001252, n = 0.0008633, o = -0.02796, p = -0.05391, q = -6.125					

Table A.2.: The coefficients for the functions in table 4.3 with index finger ray cast data. All coefficients are the result of the equation 4.16.

		a	b	c	d	e	f
$f_x(\Delta lr)$	f_1	9.622e-05	0.07538	-5.414	-	-	-
	f_2	0.07802	-0.02529	-5.561	-	-	-
	f_3	-3.974e-05	-0.0004788	0.001158	0.08844	-0.03968	-5.555
	f_4	a = 1.011e-06, b = 2.338e-07, c = -1.18e-06, d = 1.189e-06, e = -2.266e-05, f = -1.145e-05, g = 8.075e-08, h = 2.486e-05, k = 4.227e-05, l = -0.001189, m = -0.001337, n = 0.002062, o = 0.09757, p = -0.0525, q = -5.301					
$f_x(\Delta bt)$	f_1	-0.0001436	0.05197	11.13	-	-	-
	f_2	-0.0716	0.06485	11.52	-	-	-
	f_3	0.00132	0.0003302	-0.001395	-0.0958	0.08095	10.95
	f_4	a = -6.8e-07, b = -4.552e-07, c = 8.779e-07, d = 8.498e-07, e = 3.052e-05, f = 4.446e-05, g = -7.177e-07, h = -3.409e-05, k = -5.704e-07, l = 0.001983, m = 0.002484, n = -0.002728, o = -0.1184, p = 0.08877, q = 10.63					

Table A.3.: The coefficients for the functions in table 4.3 with head finger ray cast data. All coefficients are the result of the equation 4.16.

		a	b	c	d	e	f
$f_x(\Delta lr)$	f_1	-0.003486	-0.1059	28.26	-	-	-
	f_2	0.1107	0.2133	24.67	-	-	-
	f_3	-0.002088	0.007764	0.005324	-0.121	0.006089	23.84
	f_4	a = 1.483e-06, b = 1.735e-06, c = -6.418e-06, d = -6.694e-06, e = 0.0001851, f = 5.531e-05, g = -5.806e-06, h = -0.0003548, k = 0.0001425, l = 0.004328, m = -0.004554, n = -0.003081, o = -0.0572, p = 0.2839, q = 22.99					
$f_x(\Delta bt)$	f_1	0.004765	-0.09916	-20.93	-	-	-
	f_2	0.1342	0.1685	-19.91	-	-	-
	f_3	-0.003959	0.005104	0.000249	-0.07867	-0.0624	-19.05
	f_4	a = -8.452e-07, b = -6.473e-06, c = 5.58e-06, d = 4.474e-06, e = -7.52e-05, f = 0.0004535, g = 5.458e-06, h = 0.0001878, k = -6.537e-05, l = -0.003065, m = -0.001432, n = 0.002357, o = -0.002435, p = -0.1122, q = -18.9					

Table A.4.: The coefficients for the functions in table 4.3 with forearm ray cast data. All coefficients are the result of the equation 4.16.

Bibliography

- [AA09] F. Argelaguet, C. Andújar. Visual feedback techniques for virtual pointing on stereoscopic displays. In S. N. Spencer, Y. Kitamura, H. Takemura, K. Kiyokawa, B. Lok, D. Thalmann, editors, *VRST*, pp. 163–170. ACM, 2009. URL <http://dblp.uni-trier.de/db/conf/vrst/vrst2009.html>. (Cited on pages 3 and 5)
- [AAT08] F. Argelaguet, C. Andujar, R. Trueba. Overcoming Eye-hand Visibility Mismatch in 3D Pointing Selection. In *Proceedings of the 2008 ACM Symposium on Virtual Reality Software and Technology*, VRST '08, pp. 43–46. ACM, New York, NY, USA, 2008. doi:10.1145/1450579.1450588. URL <http://doi.acm.org/10.1145/1450579.1450588>. (Cited on page 4)
- [BBH89] J. C. Byers, A. C. Bittner, S. G. Hill. Traditional and raw task load index (TLX) correlations: are paired comparisons necessary? *Advances in Industrial Ergonomics and Safety I.*, pp. 481–486, 1989. (Cited on page 14)
- [BDL85] J. Basmajian, C. De Luca. *Muscles alive: their functions revealed by electromyography*. Williams & Wilkins, 1985. URL <http://books.google.de/books?id=H9pqAAAAMAAJ>. (Cited on page 8)
- [BMGX13] S. Bateman, R. L. Mandryk, C. Gutwin, R. Xiao. Analysis and comparison of target assistance techniques for relative ray-cast pointing. *International Journal of Human-Computer Studies*, 71(5):511 – 532, 2013. doi:<http://dx.doi.org/10.1016/j.ijhcs.2012.12.006>. URL <http://www.sciencedirect.com/science/article/pii/S1071581913000049>. (Cited on page 1)
- [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, pp. 262–270. ACM, New York, NY, USA, 1980. doi:<http://doi.acm.org/10.1145/800250.807503>. URL <http://doi.acm.org/10.1145/800250.807503>. (Cited on pages 1 and 3)
- [Bro96] J. Brooke. SUS: A quick and dirty usability scale, 1996. (Cited on page 53)
- [BSS04] A. Butz, M. Schneider, M. Spassova. SearchLight - A Lightweight Search Function for Pervasive Environments. In *In Proceedings of the 2nd International Conference on Pervasive Somputing*, pp. 351–356. 2004. (Cited on page 4)

- [CC02] A. Corradini, P. R. Cohen. Multimodal Speech-Gesture Interface for Handfree Painting on a Virtual Paper Using Partial Recurrent Neural Networks as Gesture Recognizer, 2002. (Cited on pages 5 and 6)
- [CL80] C. N. Christakos, S. Lal. Lumped and population stochastic models of skeletal muscle: implications and predictions. *Biol Cybern*, 36(2):73–85, 1980. URL <http://www.biomedsearch.com/nih/Lumped-population-stochastic-models-skeletal/6243499.html>. (Cited on page 8)
- [DG92] A. Duranti, C. Goodwin. *Rethinking Context: Language as an Interactive Phenomenon*. Studies in the Social and Cultural Foundations of Language. Cambridge University Press, 1992. URL <https://encrypted.google.com/books?id=Jh59S6WIS98C>. (Cited on page 7)
- [DK82] P. A. Devijver, J. Kittler. *Pattern recognition: A statistical approach*. Prentice Hall, 1982. (Cited on page 41)
- [EK90] R. Elble, W. Koller. *Tremor*. The Johns Hopkins Series in Contemporary Medicine and Public Health. Johns Hopkins University Press, Baltimore, 1990. URL <https://encrypted.google.com/books?id=-LdrAAAAMAAJ>. (Cited on page 8)
- [Enf01] N. J. Enfield. Lip-pointing: A discussion of form and function with reference to data from Laos. *Gesture*, pp. 185–211, 2001. doi:10.1075/gest.1.2.06enf. URL <http://dx.doi.org/10.1075/gest.1.2.06enf>. (Cited on page 6)
- [FBB⁺05] C. Forlines, R. Balakrishnan, P. Beardsley, J. van Baar, R. Raskar. Zoom-and-pick: Facilitating Visual Zooming and Precision Pointing with Interactive Handheld Projectors. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*, UIST '05, pp. 73–82. ACM, New York, NY, USA, 2005. doi:10.1145/1095034.1095046. URL <http://doi.acm.org/10.1145/1095034.1095046>. (Cited on page 3)
- [FH72] J. M. Foley, R. Held. Visually directed pointing as a function of target distance, direction, and available cues. In *Perception & Psychophysics*, 12(3), pp. 263–268. 1972. (Cited on page 1)
- [FKGR09] S. Foehrenbach, W. A. König, J. Gerken, H. Reiterer. Tactile Feedback enhanced Hand Gesture Interaction at Large, High-Resolution Displays. *Journal of Visual Languages and Computing, Special Issue on Multimodal Interaction Through Haptic Feedback*, 20:341–351, 2009. (Cited on page 6)
- [Hav03] J. B. Haviland. *Pointing: Where Language, Culture, and Cognition Meet*, chapter Pointing Is the Royal Road to Language for Babies, pp. 139–169. Sotaro Kita, 2003. (Cited on pages 7 and 48)

- [HB11] C. Holz, P. Baudisch. Understanding touch. In D. S. Tan, S. Amershi, B. Begole, W. A. Kellogg, M. Tungare, editors, *CHI*, pp. 2501–2510. ACM, 2011. (Cited on page 1)
- [HBM03] D. M. Hawkins, S. C. Basak, D. Mills. Assessing Model Fit by Cross-Validation. *Journal of Chemical Information and Computer Sciences*, 43(2):579–586, 2003. doi:10.1021/ci025626i. URL <http://pubs.acs.org/doi/abs/10.1021/ci025626i>. (Cited on page 41)
- [HH00] J. Hindmarsh, C. Heath. Embodied reference: A study of deixis in workplace interaction. *Journal of Pragmatics*, 32(12):1855 – 1878, 2000. doi:[http://dx.doi.org/10.1016/S0378-2166\(99\)00122-8](http://dx.doi.org/10.1016/S0378-2166(99)00122-8). URL <http://www.sciencedirect.com/science/article/pii/S0378216699001228>. (Cited on page 6)
- [HRB11] N. Henze, E. Rukzio, S. Boll. 100,000,000 Taps: Analysis and Improvement of Touch Performance in the Large. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, pp. 133–142. ACM, New York, NY, USA, 2011. doi:10.1145/2037373.2037395. URL <http://doi.acm.org/10.1145/2037373.2037395>. (Cited on pages 1 and 2)
- [HRB12] N. Henze, E. Rukzio, S. Boll. Observational and Experimental Investigation of Typing Behaviour Using Virtual Keyboards for Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pp. 2659–2668. ACM, New York, NY, USA, 2012. doi:10.1145/2207676.2208658. URL <http://doi.acm.org/10.1145/2207676.2208658>. (Cited on page 1)
- [HS88] S. G. Hart, L. E. Staveland. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. *Human mental workload*, 1(3):139–183, 1988. (Cited on page 14)
- [JBM⁺00] N. Jovic, B. Brumitt, B. Meyers, S. Harris, T. Huang. Detection and estimation of pointing gestures in dense disparity maps. In *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pp. 468–475. 2000. doi:10.1109/AFGR.2000.840676. (Cited on pages 5 and 6)
- [JK00] A. de Jorio, A. Kendon. *Gesture in Naples and gesture in classical antiquity: a translation of La mimica degli antichi investigata nel gestire napoletano, Gestural expression of the ancients in the light of Neapolitan gesturing (1832), and with an Introduction and Notes, by Adam Kendon*. Advances in semiotics. Bloomington: Indiana University Press, 2000. URL <https://encrypted.google.com/books?id=hopoAAAAMAAJ>. (Cited on page 6)
- [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, pp. 1107–1110. ACM, New York, NY, USA, 2006. doi:10.1145/1124772.1124937. URL <http://doi.acm.org/10.1145/1124772.1124937>. (Cited on pages 3, 6 and 8)
- [KBSM10] R. Kopper, D. A. Bowman, M. G. Silva, R. P. McMahan. A human motor behavior model for distal pointing tasks. *International Journal of Human-Computer Studies*, 68(10):603 – 615, 2010. URL <http://dx.doi.org/10.1016>. (Cited on pages 1, 6 and 8)
- [Ken08] A. Kendon. *Gesture: visible action as utterance*. Cambridge University Press, Cambridge [u.a.], [repr.] edition, 2008. (Cited on pages 6, 7 and 53)
- [KLP⁺06] A. Kranstedt, A. Lücking, T. Pfeiffer, H. Rieser, M. Staudacher. Measuring and Reconstructing Pointing in Visual Contexts. In D. Schlangen, R. Fernández, editors, *Proceedings of the brandial 2006 - The 10th Workshop on the Semantics and Pragmatics of Dialogue*, pp. 82–89. Universitätsverlag Potsdam, Potsdam, 2006. URL http://opus.kobv.de/ubp/volltexte/2006/1048/pdf/brandial06_proceedings.pdf. (Cited on pages 5 and 6)
- [KSMB08] R. Kopper, M. G. Silva, R. P. McMahan, D. A. Bowman. Increasing the Precision of Distant Pointing for Large High-Resolution Displays. In *Technical Report TR-08-17, Computer Science, Virginia Tech*. 2008. (Cited on pages 1, 6 and 8)
- [LR81] E. H. Leung, H. L. Rheingold. Development of pointing as a social gesture. *Developmental Psychology*, 17(2):215–220, 1981. URL <http://books.google.de/books?id=8oSyLrmDdPcC>. (Cited on page 1)
- [LW98] M. Latoschik, I. Wachsmuth. Exploiting distant pointing gestures for object selection in a virtual environment. In I. Wachsmuth, M. Fröhlich, editors, *Gesture and Sign Language in Human-Computer Interaction*, volume 1371 of *Lecture Notes in Computer Science*, pp. 185–196. Springer Berlin Heidelberg, 1998. doi:10.1007/BFb0052999. URL <http://dx.doi.org/10.1007/BFb0052999>. (Cited on pages IX and 7)
- [Mac92] I. S. MacKenzie. Fitts' Law As a Research and Design Tool in Human-computer Interaction. *Hum.-Comput. Interact.*, 7(1):91–139, 1992. doi:10.1207/s15327051hci0701_3. URL http://dx.doi.org/10.1207/s15327051hci0701_3. (Cited on page 53)
- [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, pp. 203–208. ACM, New York, NY, USA, 2009. doi:10.1145/1570433.1570471. URL <http://doi.acm.org/10.1145/1570433.1570471>. (Cited on page 8)

- [McN92] D. McNeill. *Hand and Mind: What Gestures Reveal about Thought*. University of Chicago Press, Chicago, 1992. (Cited on page 6)
- [MG03] S. V. Matveyev, M. Göbel. The Optical Tweezers: Multiple-point Interaction Technique. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST '03*, pp. 184–187. ACM, New York, NY, USA, 2003. doi:10.1145/1008653.1008685. URL <http://doi.acm.org/10.1145/1008653.1008685>. (Cited on pages 3 and 8)
- [Min95] M. Mine. Virtual Environment Interaction Techniques. Technical report, UNC Chapel Hill CS Dept, 1995. (Cited on page 5)
- [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*, pp. 140–146. ACM, New York, NY, USA, 2003. doi:10.1145/958432.958460. URL <http://doi.acm.org/10.1145/958432.958460>. (Cited on pages 3, 4, 5 and 6)
- [ON01] D. R. Olsen, Jr., T. Nielsen. Laser Pointer Interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI '01*, pp. 17–22. ACM, New York, NY, USA, 2001. doi:10.1145/365024.365030. URL <http://doi.acm.org/10.1145/365024.365030>. (Cited on page 8)
- [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*, pp. 39–ff. ACM, New York, NY, USA, 1997. doi:10.1145/253284.253303. URL <http://doi.acm.org/10.1145/253284.253303>. (Cited on page 5)
- [PKL06] T. Pfeiffer, A. Kranstedt, A. Lücking. Sprach-Gestik Experimente mit IADE, dem Interactive Augmented Data Explorer. In S. Müller, G. Zachmann, editors, *Dritter Workshop Virtuelle und Erweiterte Realität der GI-Fachgruppe VR/AR*, pp. 61–72. Shaker, Technische Fakultät AG Wissensbasierte Systeme SFB 360 Situierete Künstliche Kommunikatoren, 2006. (Cited on page 6)
- [PLW08] T. Pfeiffer, M. Latoschik, I. Wachsmuth. Conversational Pointing Gestures for Virtual Reality Interaction: Implications from an Empirical Study. In *Virtual Reality Conference, 2008. VR '08. IEEE*, pp. 281–282. 2008. doi:10.1109/VR.2008.4480801. (Cited on page 5)
- [RRK97] C. Riviere, R. Rader, P. Khosla. Characteristics of hand motion of eye surgeons. In *Engineering in Medicine and Biology Society, 1997. Proceedings of the 19th Annual International Conference of the IEEE*, volume 4, pp. 1690–1693 vol.4. 1997. doi:10.1109/IEMBS.1997.757046. (Cited on page 9)

- [She73] J. Sherzer. Verbal and nonverbal deixis: the pointed lip gesture among the San Blas Cuna. *Language in Society*, 2:117–131, 1973. doi:10.1017/S0047404500000087. URL http://journals.cambridge.org/article_S0047404500000087. (Cited on page 6)
- [VB04] D. Vogel, R. Balakrishnan. Interactive Public Ambient Displays: Transitioning from Implicit to Explicit, Public to Personal, Interaction with Multiple Users. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, UIST '04, pp. 137–146. ACM, New York, NY, USA, 2004. doi:10.1145/1029632.1029656. URL <http://doi.acm.org/10.1145/1029632.1029656>. (Cited on page 6)
- [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, pp. 33–42. ACM, New York, NY, USA, 2005. doi:10.1145/1095034.1095041. URL <http://doi.acm.org/10.1145/1095034.1095041>. (Cited on pages 1, 3, 4, 5 and 6)
- [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 & Social Computing*, CSCW '14, pp. 1377–1387. ACM, New York, NY, USA, 2014. doi:10.1145/2531602.2531691. URL <http://doi.acm.org/10.1145/2531602.2531691>. (Cited on page 4)
- [WP03] A. Wilson, H. Pham. Pointing in Intelligent Environments with the WorldCursor. In *INTERACT*, pp. 495–502. IOS Res, 2003. (Cited on page 8)
- [WS03] A. Wilson, S. Shafer. XWand: UI for Intelligent Spaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pp. 545–552. ACM, New York, NY, USA, 2003. doi:10.1145/642611.642706. URL <http://doi.acm.org/10.1145/642611.642706>. (Cited on page 8)
- [WTW⁺05] C. A. Wingrave, R. Tintner, B. N. Walker, D. A. Bowman, L. F. Hodges. Exploring individual differences in raybased selection: Strategies and traits. In *Virtual Reality, 2005. Proceedings. VR 2005. IEEE*, pp. 163–170. IEEE, 2005. (Cited on page 6)

All links were last followed on June 17, 2014.

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