Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

innovation center virtual reality

# Pointing, Pairing, and Grouping Gestures Recognition in Virtual Reality

Cecily Merkle

Bachelor Thesis
January 2022

Valentina Gorobets

Prof. Dr. Andreas Kunz

# Abstract

This bachelor thesis aims to develop a special gesture recognition for the inclusion of visually impaired persons in group meetings for brainstorming. To support the brainstorming the Metaplan method is used, where each group member can write notes, which then will be arranged on a whiteboard by a moderator during a subsequent discussion. During this discussion, the moderator uses three specific gestures particularly frequently: pointing, pairing and grouping. The gesture recognition is supposed to distinguish these movements and recognise which notes have been gestured at. In a user study in which the participants performed all three gestures several times, it was proven that the gesture recognition already works very well for the pointing gesture, but that there are some difficulties for the more dynamic movements, such as grouping and pairing. This is because these movements are rather vaguely defined and their execution depends on the performing person. Thus, many special cases have to be considered. Another challenge is to define the start of a gesture from the whole movement.

# Zusammenfassung

Diese Bachelorarbeit setzt sich zum Ziel, eine spezielle Gestenerkennung für die Inklusion von sehbehinderten Personen in Gruppenmeetings für Brainstorming zu entwickeln. Zur Unterstützung des Brainstormings soll dabei die Metaplan Methode eingesetzt werden. Dabei kann jedes Gruppenmitglied Notizen verfassen, welche bei einer anschliessenden Diskussion von einem Moderator an einer Tafel angeordnet werden. Während dieser Diskussion verwendet der Moderator drei spezielle Gesten besonders häufig: das Zeigen, Paaren und Gruppieren der Notizen. Die Gestenerkennung soll diese Bewegungen unterscheiden und erkennen, auf welche Notizen gestikuliert wurde. Bei einer Anwenderstudie, bei der die Teilnehmer alle drei Gesten mehrfach ausführten, wurde erwiesen, dass die Gestenerkennung für die Zeigegeste bereits sehr gut funktioniert, aber einige Schwierigkeiten, für die eher dynamischen Gruppier- und die Paarbewegungen auftreten. Denn diese Bewegungen sind eher vage definiert und ihre Ausführung abhängig von den unterschiedlichen Personen, sodass einige Spezialfälle in Betracht gezogen werden müssen. Eine weitere Herausforderung ist es, den Anfang der Geste aus der Bewegung heraus zu erkennen.

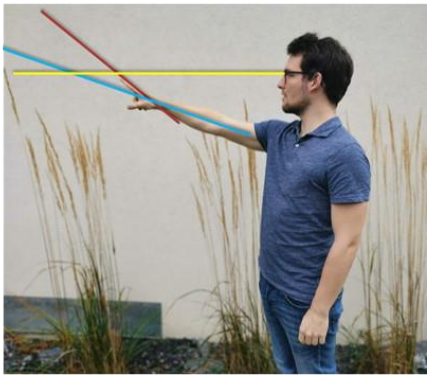
IWF
Institut für Werkzeugmaschinen und Fertigung
Institute of Machine Tools and Manufacturing

# Pointing, Pairing, and Grouping Gestures Recognition in Virtual Reality (VR)

Keywords: Gestures recognition in VR, interactions in VR.

## Overview

Human communication includes verbal and non-verbal communication (NVC). To provide blind and visually impaired people with a useful tool to interpret NVC, it is possible to use VR tools. Using additional tracking hardware on the human body allows tracking the position and orientation of the particular part of the body.
As an application, you will be working with the pointing gestures recognition during group meetings. The setup includes the virtual whiteboard with the virtual notes, with which a user can interact by pointing, pairing, and grouping gestures.
The setup and the algorithm will be provided for the thesis. The main goal of the thesis is to improve this recognition algorithm and prove it with a subsequent user study.



## Tasks

Your task is to improve the implementation of pointing gestures recognition in VR. You need to find the optimal tracker location on the human body. Also, you need to work on the gesture recognition algorithm to improve its accuracy and overcome some of the current limitations. Further, you set up a virtual testing environment, conduct a user study and evaluate your implementation. Finally, you present your findings to the ICVR lab and hand in a written report.

## Workpackages

- Literature research on the pointing, pairing, and grouping gestures recognition in VR
- Improve the existing gesture recognition algorithm in VR
- Find the best locations for the trackers on the human body to improve the accuracy of the algorithm
- Set up and conduct a user study, incl. data analysis
- Intermediate and final presentation
- Written report

## Skills

- Basic programming skills, preferably in C#/C++
- Unity and VR experience is a plus
- Strong communication and interpersonal skills

## Results

The results of this thesis have to be summarised in a written report and will be presented to the ICVR in a 20min talk.

## Contact

Valentina Gorobets, LEE L201 vgorobets@ethz.ch
Andreas Kunz, LEE L208 kunz@iwf.mavt.ethz.ch

icvr
innovation center virtual reality

# Acknowledgment

I would like to thank Professor Andreas Kunz for making this bachelor thesis possible and especially Valentina Gorobets for her great support and advises. Special thanks also to Rosalie Merkle and Roman Ludwig for proofreading this report and helping me staying motivated. Finally, I would like to thank all the participants who took their time to take part in the user study.

# Contents

# List of Figures

# List of Tables

# List of Acronyms

**BVIP** Blind and Visually Impaired People.

**FN** False Negative.

**FP** False Positive.

**HMD** Head-Mounted Display.

**NASA TLX** NASA Task Load Index.

**NVC** Non-Verbal Communication.

**SSQ** Simulator Sickness Questionnaire.

**SUS** System Usability Scale.

**TP** True Positive.

**VE** Virtual Environment.

**VR** Virtual Reality.

# Introduction

This first chapter starts with a short introduction to the importance of gestures in communication and explains the issues, when these gestures cannot be perceived. Moreover, related works are presented, which refer to the topic of pointing gesture recognition. Finally, the core question of this thesis is stated and the outline of this project is given.

## 1.1 Motivation

One big difference between humans and other animals is the ability to communicate and hence the possibility to exchange information efficiently. It is widely believed that for humans communication and language development started with gesturing and the ability to interpret the displayed gestures [BGD97]. In a study of gestural behaviours of chimpanzees, the next human-like animals, it was observed that apes do not use gestures to share information, despite being able to learn it [DJP03].

By Definition of the Oxford dictionary gesture is a "a movement of part of the body, especially a hand or the head, to express an idea or meaning". Kendon suggested a more specific definition, where he considered conventional gestures, gesticulations and signing movements, but excluded posture shifts, self touchings or incidental object manipulations [Ken97]. Moreover gestures can be performed in silence, but they are often combined with speech. Thus, gestures with an relation to the spoken words are called deictic and are often used e.g. to describe the meaning of directions or nearby objects. For example, a person could ask, "how do you like this?" and point to the direction of the target. When the gesture is not seen by the listener, the main content of the sentence is lost. A similar issue occurs when only the gesture is seen but words are not heard.

The relevance of gestures in communication is a widely researched topic. In language development of children, it was found that infants learn to use deictic gestural communication about three month before they learn to understand words [OGGM13]. Furthermore, it was found that children learn to build sentences and connecting words only after combining single words with gestures [CV10]. Thus, the amount of children's speech can be predicted by the amount of pointing gestures they perform [GM07].

Rogers found in a study that gestures improve verbal comprehension for listeners significantly, especially in noisy environments [ROG78].

The significance of gestures becomes obvious again when they cannot be perceived, as it occurs for example for Blind and Visually Impaired Peoples (BVIPs). A sighted person talking to a BVIP might adapt to the situation by cautiously expressing themselves more verbally without using gestures. But when BVIPs participate in group discussions of mostly sighted persons, conversations involving gestures become hard for them to comprehend - especially when deictic phrases are involved in the conversation.

Inclusion of visually impaired persons in society progressed significantly in the past years and assisting technologies evolved. Nevertheless, Non-Verbal Communications (NVCs) used in group meetings remain a challenge. An international group, including researchers from the Technical University of Darmstadt (Germany), the Johannes Kepler University Linz (Austria), and the ETH Zürich- Swiss Federal Institute of Technology (Switzerland), addresses this issue in the project *MAPVI: Meeting Accessibility for Persons with Visual Impairments* [GKD+19]. One of the addressed issues in this project is the accessibility of brainstorming meetings when a whiteboard is used, like suggested in the Metaplan method: each group member can write notes with their ideas on cards. The cards are then collected by a moderator and sorted on a whiteboard throughout a group discussion. During the process of discussing the topics, and thus sorting and rearranging the notes, the moderator uses many deictic gestures like pointing, pairing and grouping. For BVIPs, these gestures cannot be seen and a recognition system is needed to make them accessible. Finally, the question arises as to how such a recognition system can be implemented for three similar deictic gestures.

## 1.2 Related Work

This section gives an overview of some publications that are related to the recognition of deictic gestures. There are several works that cover the topic of pointing gesture recognition, but non can be found about pairing and grouping, although they can be seen as a dynamic version of pointing.

First, the work by Wnuczko and Kennedy investigates the question on how people point in different circumstances [WK11]. By conducting various user studies, it was found that a person uses different trajectories for pointing in a certain direction. When the pointing target is in sight, a line from the eye through the tip of the finger was used to determine the pointing direction, as depicted in Figure 1.1. When pointing while blindfolded, the person was more likely to use the line of their elongated arm to point at the target . Furthermore, Wingrave et al. found that the pointing trajectory is also dependent on provided feedback. For example, when using a laser pointer there is a visual feedback on the pointing location provided and the pointing direction is automatically adapted [WBR02].

Later, Akkil and Isokoski investigated the accuracy of interpreting pointing gestures in egocentric view [AI16]. They found that pointing is most accurate when targets are straight in front of the person. When moving to a eccentric pointing target, subtle head movements or ocular dominance can influence the accuracy of pointing at a target. These findings where confirmed by Nickel et al. in their work about pointing gesture recognition based on 3D-tracking on the face, hands and head orientation [NS03]. In this project, the researchers used a camera connected to a PC to determine the positions of the head and hands of a person via image processing. To Identify these body parts, two steps were included: First, a dense disparity map was applied to determine the outline of the person and therefore the most reasonable location of the head. Second, to distinguish the face and hands from the rest of the body, a colour filter was used for detecting the skin. Then, an algorithm to detect the pointing direction was applied. After an analysis of three different ways to approximate the pointing direction, it was found that using the "Eye-

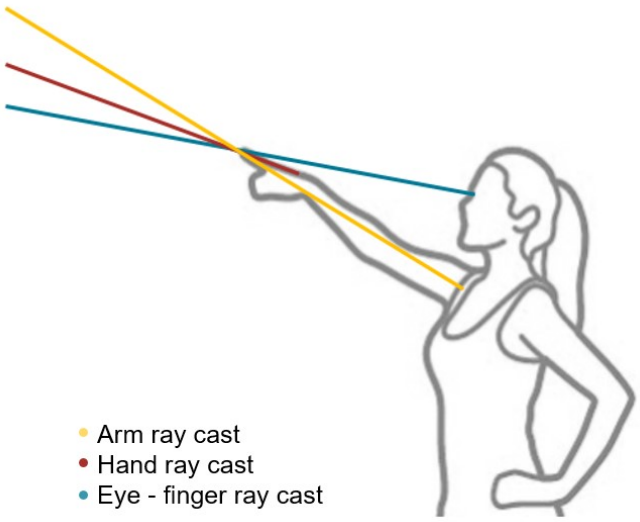



***Figure 1.1:*** *Ray casts to determine the pointing direction, adapted from [Poi]*

finger ray cast" as seen in Figure 1.1 resulted in the most accurate recognition. Later on, a magnetic sensor was added to determine the head orientation more precisely. Finally, it was found that this system delivered a 90% correct recognition of the pointing direction. However, it must be added that this system still has some limitations. The camera works on a two dimensional basis. Thus, the area in the room where the person can move and point, while remaining visible to the system, is limited.

A different approach would be to use virtual reality devices. Virtual Reality (VR) is mainly known in the gaming industry, but it is recently applied in a growing number of scenarios: From online meetings in business, landscape modelling in architecture to simulations in training etc.. In recent years, VR technologies have evolved greatly and nowadays not only provide a visual and acoustic experience, but also feature integrated tracking systems to detect movements. Using trackers, the integration of a recognition system to determine the pointing direction has become a rather simple task that has already been used in several projects [MRS+20] [KJP+20].

The challenge remains of how gestures can be detected out of context and how multiple similar gestures like pointing, pairing and grouping can be differentiated. Simon Liechti proposes a method which covers these challenges. In his work, he suggests a method how a gesture recognition could be implemented and the gesture targets detected. Therefore a brainstorming application was used, which allowed a However, the accuracy of this method was not verified sufficiently.

## 1.3 Task Formulation

The task, which is a sub task of the above mentioned project MAPVI, is to develop a recognition system for the pointing, pairing and grouping gestures, when the application for brainstorming (described in Section 2.2.4) is used. Based on this previous project, the research question of this thesis can be formulated as:

**How can the given gesture recognition system be improved?** To improve the system, first, the limita-

tions of the previous work were elaborated and an optimal positioning for the motion trackers determined. Then, some changes in the recognition algorithm were implemented. In addition, a virtual environment was generated and a user study conducted to test the recognition system.

In the following Chapter 2, the methods used to implement the gesture recognition are introduced. First, an overview of the gestures, which are detected, is given. Afterwards, the used hardware and software are presented, and the methodology of the previous project shortly summarised. Moreover, the implementation of the recognition algorithm and also some integrated features of the system are explained. Thereafter follows the concept of the conducted user study and within it, an explanation of the utilised questionnaires and the data collection.

Chapter 3 presents the results of the user study and discusses them. After the questionnaires and the objective results are evaluated, the most important and some unexpected findings are emphasised and possible explanations for these behaviours are given.

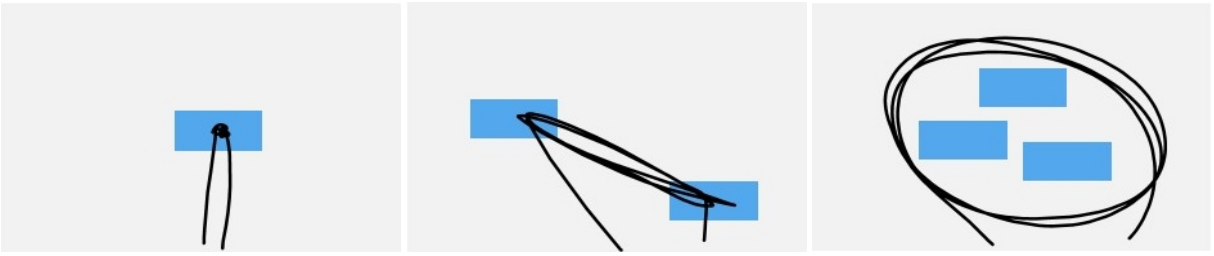
Finally, in Chapter 4.1, the conclusion of this thesis is given and ideas for the future work are presented.

# Methodology

In this chapter the methodology that has been used to implement the gesture recognition is presented. First, a definition of the gestures which are detected is given in Section 2.1. Second follows an overview of the used hardware and software in Section 2.2. In addition, a the previous used methodology, which was given in the beginning of the project is summarised in Section 2.3. Thereafter, the implementation of the improved gesture recognition is given in Section 2.4. Furthermore, the user study which was conducted to test the gesture recognition is described in Section 2.5. Finally, a description how the data is evaluated is given in Section 2.6.

## 2.1 Gesture Definitions

This subsection is about the definitions of the three gestures, which need to be determined. They start with raising the arm and finish with lowering and relaxation. The differences of the three gestures are found by the movements, that are made while the arm is raised. In Figure 2.1, all three gestures are depicted next to each other for comparison.



*(a) Pointing* *(b) Pairing* *(c) Grouping*

***Figure 2.1:*** *Representation of the three gestures, where the black line depicts the movement trajectory and the blue rectangles are the notes that are gestured at.*

#### Pointing

The pointing gesture is the most simple movement of the three tested gestures. It is performed by simply holding still the arm and hand into a certain direction for a short time. This gesture can differ by using slightly different hand positions, whereas the movement itself is very clear. The depiction of this movement can be seen in Figure 2.1(a).

#### Pairing

The pairing gesture is a dynamic extension of pointing, as it aims to connect two notes. The gesture starts with pointing to one note and moves further to a second note. This movement is often repeated, resulting into a back and forth motion. An example of a pairing movement can be seen in Figure 2.1(b).

#### Grouping

The grouping gesture is defined as a dynamic movement to raise the attention of a listener on multiple notes. Therefore, an elliptic movement is often made over or around the notes, unlike in pointing and pairing, where the person performing this gesture is aiming at the notes directly, as can be seen in Figure 2.1(c). The difficulty here is that line of the movement trajectory is not necessary touching the notes that are gestured at like for pointing or pairing. Thus, the recognition system has to detect any notes, that are included in the ellipse, that is drawn by the movement trajectory.

## 2.2 Hardware and Software

The hardware and software, that was used for the gesture recognition are introduced in this section. First, the VR-devices *HTC Vive Pro* and *HTC Vive Trackers* are described. The following section presents the software *Unity* and *Steam VR* that are needed for the Virtual Environment (VE). Subsequently, an application for digitising a meeting, using the Metaplan method, is presented.

### 2.2.1 HTC Vive Pro and HTC Vive Tracker

The *HTC Vive Pro* is a set of VR devices, including a Head-Mounted Display (HMD), lighthouses, and two controllers. In the user study, these were used with two additional *HTC Vive* trackers. These devices were developed in collaboration of *HTC* and the *Valve Corporation*.

The HMD is equipped with an AMOLED screen with a resolution of 1440 x 1600 pixels per eye. Furthermore, it has a refresh rate of 90 Hz and a field view of 110 degrees. The headset is connected to the computer through HDMI and USB cables and needs an additional power supply. While wearing the HMD the user can only see the VE and hence dive into virtual reality.

For the motion tracking, at least two base stations, so called "lighthouses", need to be installed on two opposite sides of a room. Inside the lighthouses there are two stepper motors that rotate two infrared lasers with a constant speed and send out signals with information of their clock time and rotation speed. The tracking system of the HMD, controllers and trackers have sensors to detect the infrared signals. Thus, via triangulation, they can calculate their own position, using the given information about the


***Figure 2.2:*** *HTC Vive Pro set and additional trackers adapted from [Viv]*

difference between the signals and the current time and also the rotation speed of the motor. For higher tracking accuracy a gyroscope and a G-sensor are integrated [Viv].

The trackers and controllers are connected wireless to the HMD, which exchanges the given positioning information to the processor of the computer. The trackers have an internal thread for attachments and can only be used for tracking the position and orientation of the attached object. The controllers can be used for the same, but are designed to be handheld and additionally, they have multiple buttons and a touch pad for input for interactions with the system.

### 2.2.2 Unity

*Unity* is a game engine developed by *Unity Technologies*. It allows a fast and easy way to build 2D and 3D environments and enables interactions with the environment. Originally created and developed for the video game industry, *Unity* is nowadays used for over 25 different platforms including desktop, mobile, web, TV and also VR. Thus, it has expanded significantly and is used in various fields visualise landscapes and models in the construction industry, for architecture as well as for cinematic techniques.[Uni] To implement interactions *Unity* provides a scripting API in the programming language C#. Therefore, it can be connected to *Visual Studio*, an integrated development environment for easy text editing and debugging. In this project, *Unity* 2019.4.16.f1 was used for designing the VE and scripting the recognition algorithm.
A VE build in *Unity*, consists of multiple 3D models, which are integrated as so called "game objects". which have certain properties, including the dimension and position . In addition, parts like a *mesh renderer*, or a *collider* can be added to the properties to visualise the object with a chosen visual material or to give the object a solid body that is detectable. The shape of the solid is independent of the actual shape of the object. Furthermore, a script can be attached to the object for adding functionalities, like reactions to movements or inputs in the VE or a behaviour of the game object. A script contains two main functions that can be used; the *start* and the *update* function. The start function is only called once, at the beginning, when the associated game object is enabled. In contrast, The update function is accessed at every frame, as long as the script is set active and can be used e.g. to keep tracking a movement or check for new inputs. Besides, to categorise all the different game objects, a *Tag* can be added to its properties.

### 2.2.3 Steam VR

Steam VR is an application which makes it possible to enable VR and connect the software to the VR devices. It was released by *Steam*, a digital distribution service developed by *Valve Corporation*. For connecting Steam VR with Unity, the *Steam VR Unity plugin* can be inserted. It eases the handling of the VR- devices and their input, loads 3D models for the devices and also estimates the form of the hand when using the devices[Ste]. In this project it was used to create a glove model in *Unity* and to represent the position of the real hand in the VE and to obtain the position of the trackers.

### 2.2.4 Brainstorming Tool

The "Brainstorming Tool" is an application created by Reinhard Koutny [KGD+20] and can be used as an digital version of the Metaplan method. Therefore, group members can access a meeting platform through their individual devices and create notes in it. These notes are shown on a whiteboard, where they can be sorted, connected or deleted by the moderator. An example of how such a whiteboard could look like is given in Figure 2.3.
The information about the content, form and size of the notes and also their position on the whiteboard are then given in a REST API.

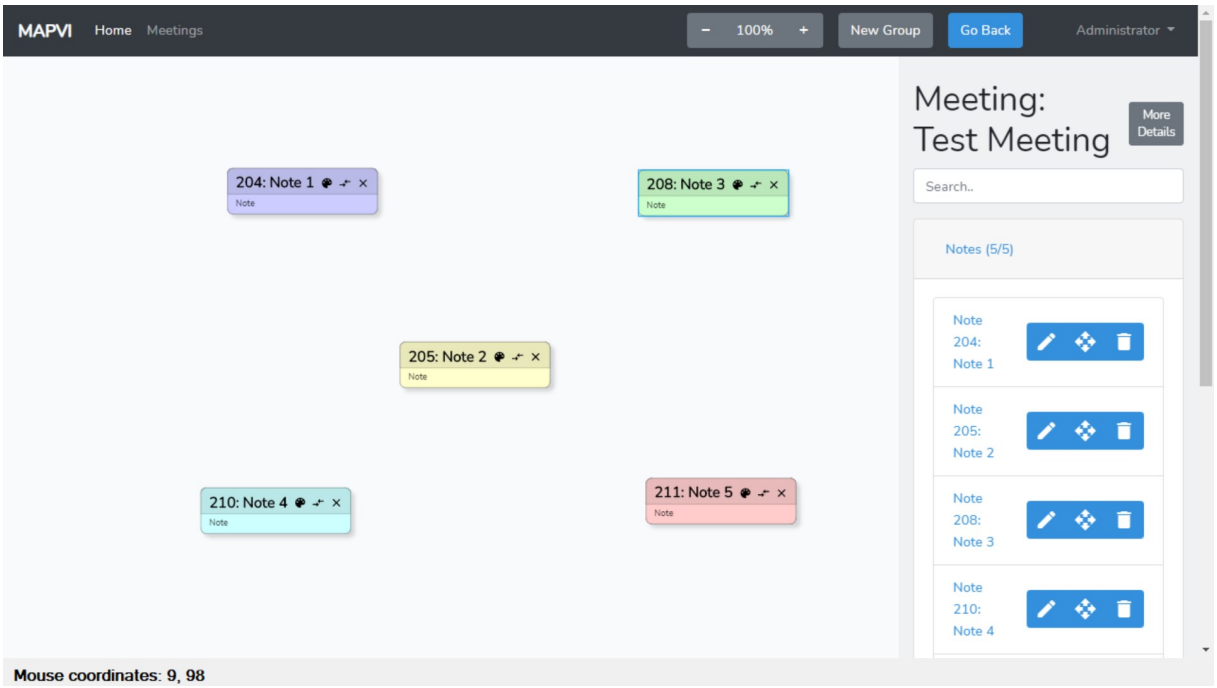



***Figure 2.3:*** *Window of the Brainstorming Tool*

## 2.3 Methodology of the Predecessor Project

Following presents the implementation of the gesture recognition in the previous project by S. Liechti, which was used as a starting point for this thesis. In the following it is explained how the notes are implemented from the meeting application and the main functions of the recognition system are shortly summarised. Finally the limitations of the system are stated. A more detailed review of this project can be found in [LDK21].

### 2.3.1 Implementation in the VE

To determine the gesture direction, two controllers from the *HTC Vive Pro*-set are used to follow the orientation of the arm axis. For this, one controller is held in the crook of the arm and the other in that arms hand as depicted in Figure 2.4. In addition, a "recognition ray" is created in *Unity* from one controller through the other.

Furthermore, the notes created in the meeting application are implemented into *Unity*. Using the access to the Rest API of the *Brainstorming Tool*, the information of the exact position and size of the notes can be simply calculated and copied as game objects on a virtual board. This can be seen as a virtual twin of the real whiteboard in *Unity* as depicted in Figure 2.5. The virtual environment only consists of a greyish stone floor with a virtual board floating in the air. All of the note game objects are automatically assigned to the tag called "note", so that they can be differed from the whiteboard, e.g. when detecting colliding objects.

When the recognition ray hits the virtual whiteboard, small recognition spheres with the size of 5% of the whiteboard size are created with a rate of 0.1 sec. These spheres are given the additional Boolean variable called "activation", which turns to true one second after the sphere is created. This property is used to start the recognition algorithm. So, when a sphere collides with an activated one, the gesture recognition for the pairing or grouping movement starts. One additional second later, when the *spheres lifetime* ends, the sphere is deleted again. This allows to track only the most recent part of the trajectory of movement.

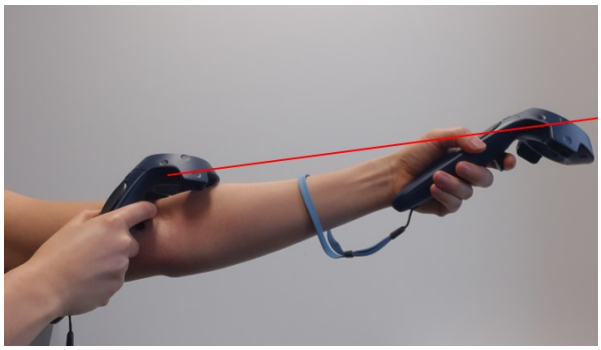


***Figure 2.4:*** *Usage of the controllers and depiction of the recognition ray*

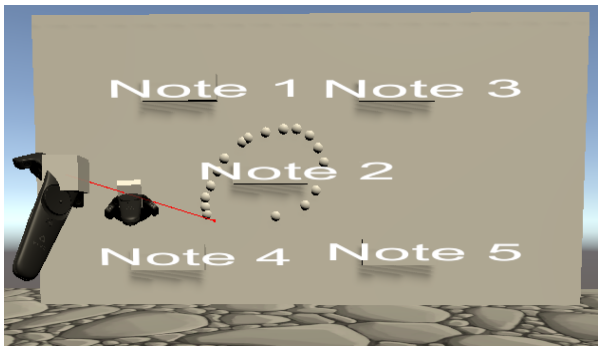
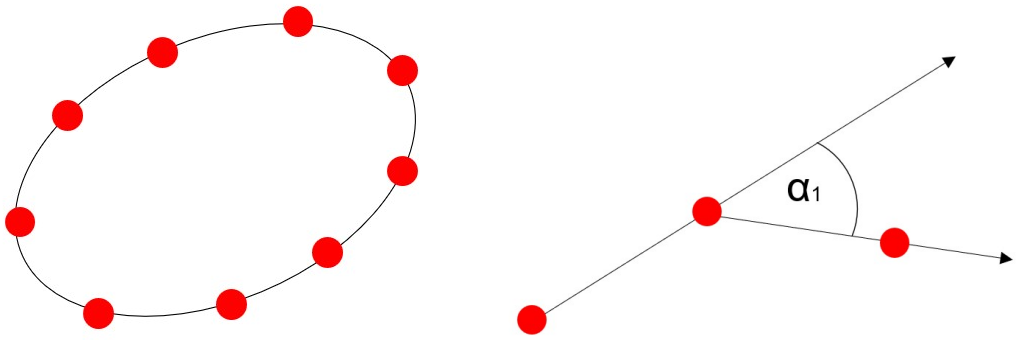



***Figure 2.5:*** *Implementation of the whiteboard and the recognition system in Unity*

### 2.3.2 Gesture Recognition Algorithm

The gesture recognition gets started, as soon as the recognition ray hits the whiteboard. If the recognition ray hits an object with the tag "note", it starts the "pointing timer" to measure how long the ray is directed at the colliding object. If the ray does not leave the note for two seconds, the timer expires and a pointing gesture is recognised. If the ray leaves the notes collider or if the pointing time has passed, the timer is reset to zero.

When a sphere collides with a second sphere, which has been activated, the general gesture recognition algorithm starts. Then, an array is created containing the position of all spheres in that frame. Next, the vector between two subsequent spheres are calculated and saved in another array. In Addition, the angles between two subsequent vectors as shown in Figure 2.6(b) are determined and saved in a third array.




*(a) Spheres trail on a elliptic gesture (black line)* *(b) Angle between two vectors through subsequent spheres*

***Figure 2.6:*** *Recognition using spheres*

Furthermore, the difference $G = \alpha_{max} - \alpha_{min}$ between the maximum and minimum angle is computed. If this so called "G-value" is higher then the threshold 100, there is a big change the movements direction. Hence, a pairing gesture is determined. If this G-value is lower then the threshold, the change of the movements direction is rather constant and a more circular movement is detected. Thus, a grouping gesture is identified. Thereafter, the maximum and minimum X- and Y-values of all sphere positions, regarding a coordinate system attached to the whiteboard, are computed. Hence, these values are set as the boundary values of the rectangular detection box. (Figure 2.7). If a pairing gesture is recognised, the system detects any paired notes on two diagonal corners of the detection box. If grouping is recognised, the system examines the positions of all notes and calculates, whether they are inside the boundaries of the detection box or not. Thus, the notes of the grouping gestures are detected.

### 2.3.3 Limitations

Some limitations of the previous project were identified shortly after the algorithm was tested for the first time. It was found, that the tools used for detecting the pointing trajectory are uncomfortable to use, and also not accurate. In the related work, it was mentioned, that using the eye-finger orientation works best for determining the direction of pointing. Thus, using the arm trajectory leads to a higher inaccuracy on the pointing target. Furthermore, since there are two controllers used, where one needs to be hold at the crook of the elbow, displacements lead to additional inaccuracies. An further issue occurs, when

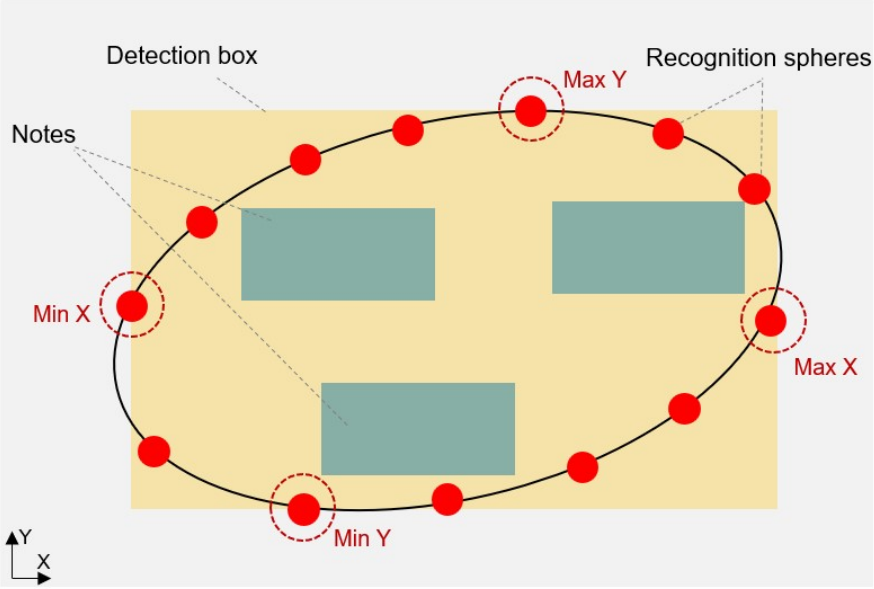
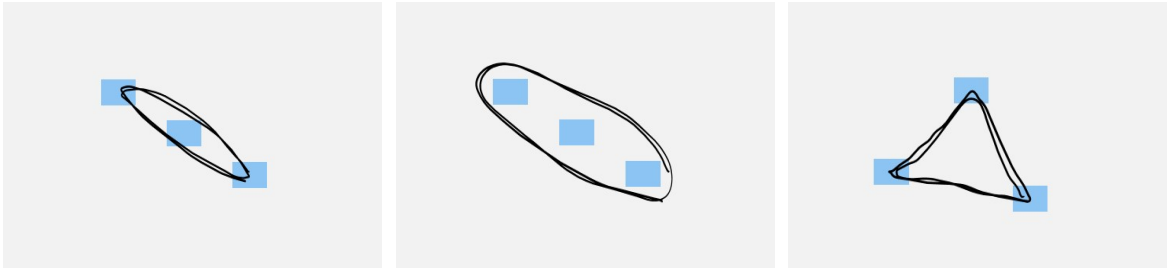



***Figure 2.7:*** *Detection Box for an Elliptic Grouping Gesture*

the recognition ray leaves the whiteboard. This happens quite often during a grouping gesture, when the movement is performed spacious around the notes. Then, the spheres trail of the recognition gets interrupted and deleted completely, which leads to no gesture detection. Additionally, there are some limitations when it comes to different kind of gestures. When a pairing gesture is performed elliptically over some notes in between, the G-Value gets smaller and it is often recognised as grouping (Figure 2.8(a)). The opposite occurs, when a grouping gesture is performed over three notes that are arranged in a line. Then, it is very likely that the G-value becomes to high and a pairing gesture was detected (Figure 2.8(b)). Besides, when a movement had a fast change in the direction, like when grouping three notes in a triangular shape, again, the G-value would be too high, such that the recognition system would determine it as a pairing gesture and subsequently detect false notes (Figure 2.8(c)).



*(a) Pairing over third note* *(b) Grouping gesture over three notes in a line* *(c) Grouping in triangular shape*

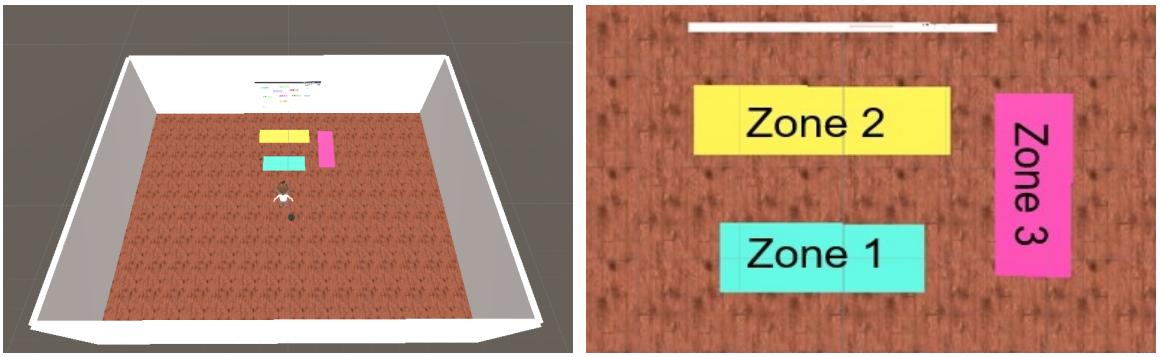
***Figure 2.8:*** *Problematic movements (black line) for the gesture recognition over notes (blue rectangles)*

## 2.4 Implementation of the Gesture Recognition

This section is about the methods that are used to implement the algorithm for the gesture recognition in the current project. First, the virtual environment in which the gesture recognition is built in and the implementation of the ray cast for the pointing direction is introduced. Afterwards follows an detailed explanation on the adjustments of the gesture recognition compared to the previous.

### 2.4.1 Virtual Environment

The previous project already included an environment in Unity which could implement the notes and the whiteboard into *Unity*. In addition, a virtual environment was modified for the user study. To create a meeting room feeling, the users see themselves inside a room with a wooden floor and white walls as depicted in Figure 2.9(a). Moreover, three different zones as depicted in Figure 2.9(b) were defined in which the participants were allowed to move. The blue "zone 1" was placed 2.5 meters away from the virtual screen and hence the furthest. The yellow "zone 2" was directly in front of the whiteboard, so that participants could touch it when they extended their arms. The pink "zone 3" was located on the side to have an oblique view of the whiteboard. It was placed on the left side of the virtual screen for left-handed participants, and on the right side for right-handed participants.




*(a) Bird's-eyes view of the virtual room* *(b) Arrangement of the three zones on the floor*

***Figure 2.9:*** *Virtual Environment for User Study*

To represent the real whiteboard, the "Brainstorming Tool" interface was implemented using an open source asset called *"uWindowCapture"*, created by *"2018hecomi"* [hec]. This asset made it possible to simply add in a window on the desktop of the computer as a canvas into the virtual environment. Including the interface of the meeting application made the virtual room more realistic. Moreover, the position of the notes in *Unity* could be compared better with the actual position on the meeting application interface. This way, offsets of the notes were easily detected.

The whiteboard has a width of 3m and a height of 1.5m. To overcome the problem, that the system is interrupted, when the recognition ray leaves the whiteboard, the height and width of its collider was magnified by an factor of 1.5. Thus, even if the ray leaves the visible part of the whiteboard, there are still spheres created and the recognition algorithm continues as depicted in Figure 2.10.

As mentioned in the related works in Section 1.2, a straight line following the trajectory from the eyes through the tip of the finger is stated to be the best approximation for the direction of pointing. Thus, two trackers are attached to the back of the users hand and the top of head. Then, the "Eye-Finger Ray" is created starting from the head tracker with a vertical offset of 11cm going through the index finger

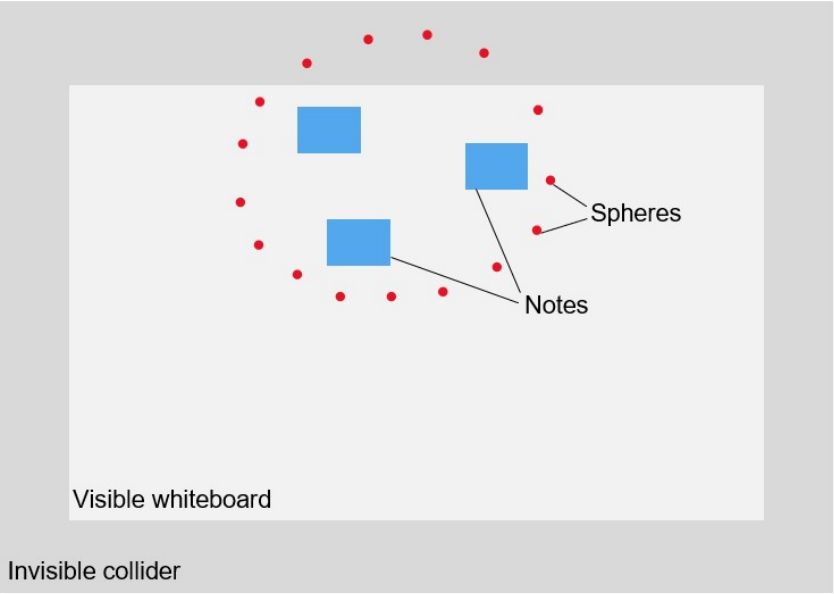



***Figure 2.10:*** *Spheres Trail on the Whiteboard with Extended Collider*

(Figure 2.11). It is assumed that a humans height is around 171cm (in average 164cm for woman and 178cm for man in Switzerland [swi19]). Additionally, according to human proportion studies, eyes are positioned at half length of a head, which again has one eighth size of a human body [Bam89]. Thus, the best value for the vertical offset of the head tracker is given by a sixteenth of the average human height (171cm) - which is approximately 11cm. For the user study, the head tracker was fixed to the HMD by a provisional 3D-printed bracket (as can be seen in the appendix in Figure A.1). The hand tracker was attached to the back of the hand using special straps and implemented with a glove model from the Steam VR asset, to visualise the hand of the user. In addition, an option for left handed persons was included, which would switch the glove to a mirrored version. The offset of the ray for the hand tracker is simply given from the fingertip of the inserted glove. To the user, only the glove is visible, as can be seen in Figure 2.12.

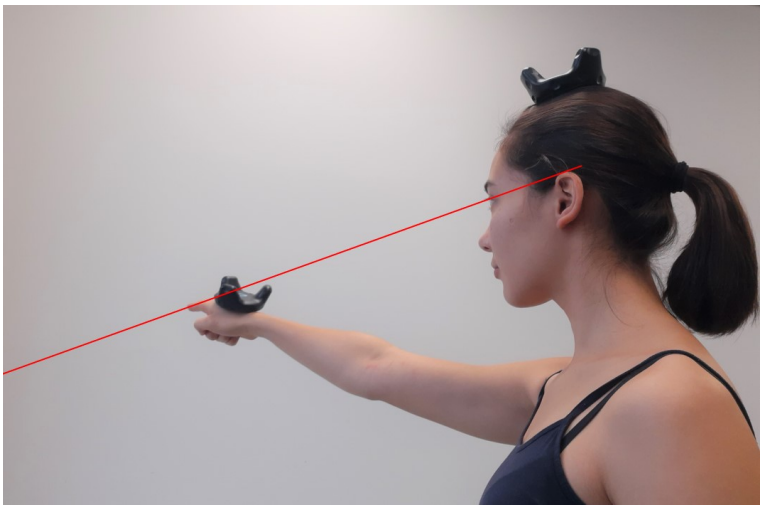


***Figure 2.11:*** *Depiction of the Recognition Ray and the Positioning of the Trackers*

Since humans adapt to feedback, for example, when their pointing direction is made visible by an laser,

***Figure 2.12:*** *User's View in the VE*

the question arose, if users change their pointing behaviour when standing closer to their target and use their hand as a feedback system. More precisely, it may be that users start to hover their hands over the note, instead of pointing in its direction, when the whiteboard is within arm's reach. This theory was backed by recordings of presentations with whiteboards: The presenter's arm was often not fully raised or straightened, when indicating something within close reach. To further test this, a perpendicular projection of the virtual fingertip was implemented as a second recognition ray. The system would automatically switch to this one, when the hand of the user came within a distance of 30cm to the whiteboard. Furthermore, for testing this theory in the user study, an option was implemented to en- or disable this so called "Near Ray". Hence, it was tested, if the recognition works better when the ray changes depending on the distance to the whiteboard or when it was left as before, only using the "Eye-Finger Ray"

### 2.4.2 Gesture Recognition Algortihm

The algorithm to start the gesture recognition was taken from the previous project. There were only some adjustments made to improve the script. For example, for calculating the angles between the spheres vectors, the last value in the array containing all spheres positions was not considered in the previous project and got added in the current. Furthermore, the script got structured and some superfluous calculations got simplified.

#### Pointing Gesture Recognition

The recognition of the pointing gesture worked quite well and was left as it was.

#### Pairing Gesture Recognition

If the pairing gesture is called, the algorithm detects all "corner spheres", after which the angle value is higher then 100° degrees as depicted inFigure 2.13. After that, collisions within an arbitrary chosen radius of 30cm around the positions of all corner spheres are detected. The name of all objects with the tag "note" which collides with the "detection circle" are then saved in a string-array. Furthermore, all notes that are repeated within the string-array are deleted from the array, so that all entries become unique. Finally, the length of this shortened array is determined and the content printed. For a length of one, it was defined as pointing, for the regular case of two, pairing. For the special case, that multiple note names were stored in the array, the system recognised the gesture as grouping.

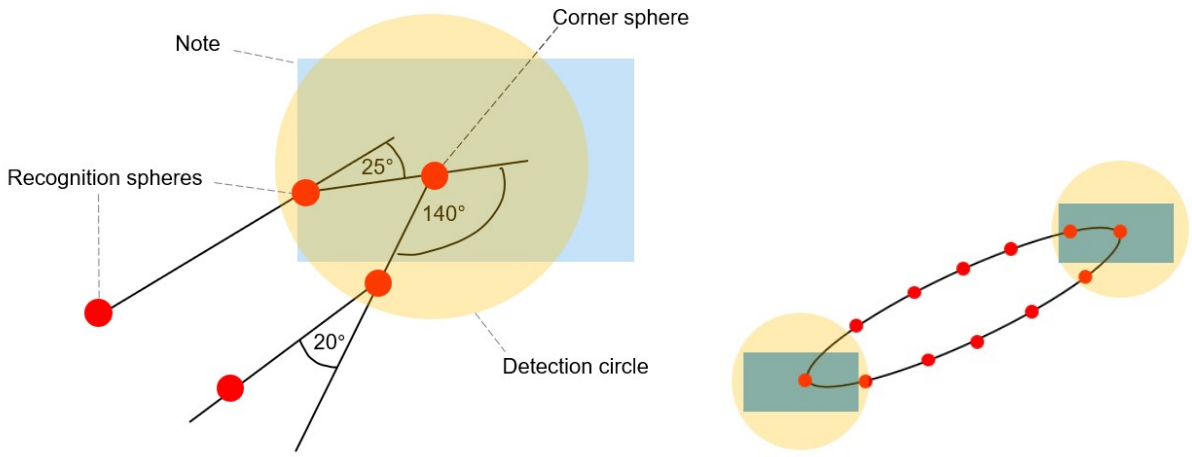



***Figure 2.13:*** *Left: Depiction of how a note is found using the corner sphere. Right: Pairing two notes (blue) with two detection circles (yellow)*

#### Grouping Gesture Recognition

For the grouping, recognition works using the same principle as the previous project, but the implementation is changed. First, all spheres positions are recorded and the maximum and minimum X- and Y-values, respective a coordinate system attached on the whiteboard, are determined. Then, a rectangular detection box with these values as boundaries is created as a collider in *Unity*. For comparison, in

the previous project the position of the notes were calculated mathematically. Hence, all colliding game objects with the tag "note" within this detection box can be identified. The system then displays these notes names including the gesture which again depends on the amount of notes that are detected. For only one note, pointing is recognised, for two, pairing and for more, grouping.

## 2.5 User Study Design

A user study was conducted to test the recognition algorithm, and also to verify improvements compared to the previous project. Therefore, the user study was divided into two parts. The first part took place in the environment of this current project using trackers, while the second part was conducted in the environment of the previous project by S.Liechti, where controllers were used to track the arm movement. To conduct the study under the same conditions, the predecessor project was modified. The desktop of the "Brainstorming Tool" was included and the zones and the black shadow point for marking the position of the user was added on the floor. Furthermore, any features to make the pointing direction visible, like a laser beam was disabled, so that the user could only see the controllers that they held in their hands.

A short summary of the structure of the user study is given in Section **??**: The participants were introduced to the study and then given the first questionnaire. Then they performed the set of gestures in the first environment using the associated tracking devices and filled in the second questionnaire to evaluate the just finished task. Afterwards, the second task was carried out and finally the last questionnaires filled out.

### 2.5.1 Participants

To recruit participants for the user study, a poster was displayed in the departmental office and an enrolment mail was sent to students and employees from the ETH Zurich via the institutions mailing list. Furthermore, a registration form was sent out to several students group chats through social media.

In the user study, there were 30 participants within the age of 19 - 31 years. Among them were 4 female and 26 male participants. All of them were students in natural, formal or applied sciences, whereof 4 were doctoral students. Furthermore, they were all right handed.

| Participants | Male | Female | Age Distr. | Age Average | Age SD |
|---|---|---|---|---|---|
| 30 | 26 | 4 | 19 – 31 | 23.3 | 2.85 |

***Table 2.1:*** *Age and gender distribution*

### 2.5.2 Task Design

The participants were wearing the VR headset and performed the gestures they were instructed to do by the experimenter. An virtual human was included into the virtual environment to represent the location of a listener.

Per type of gesture (pointing, pairing, grouping) five different variations were defined, where each included a different subset of notes. The arrangement of the notes can be seen in Figure 2.14. Pointing was

performed on notes that were spread over the whiteboard (notes [C] [A] [G] [D] [F]) to cover the largest possible range of pointing directions. For pairing, three simple pairing gestures were chosen (notes [D,G] [D,F] [C,H]), one pairing gesture ([H,B]) had to be performed over a third note ([G]) and the last two notes ([A,E]) had to be paired over the hole board. For grouping, too, three rather simple movements over three notes were chosen ([A,B,G] [D,E,F] and [B,F,G]), while one grouping gesture covered four (notes [C,D,G,H]). The last grouping gesture over the notes [H,G,B] was chosen for comparison to the very similar pairing gesture ([H,B]). The order in which the different gestures were instructed was randomised. Also, there were no timings given to the participant to ensure they did not feel pressured and were able to rest their arms in between gestures. To ensure the recognition system performs well for a range of user locations, all gestures were repeated in each of the three zones that were described in Section 2.4.1.

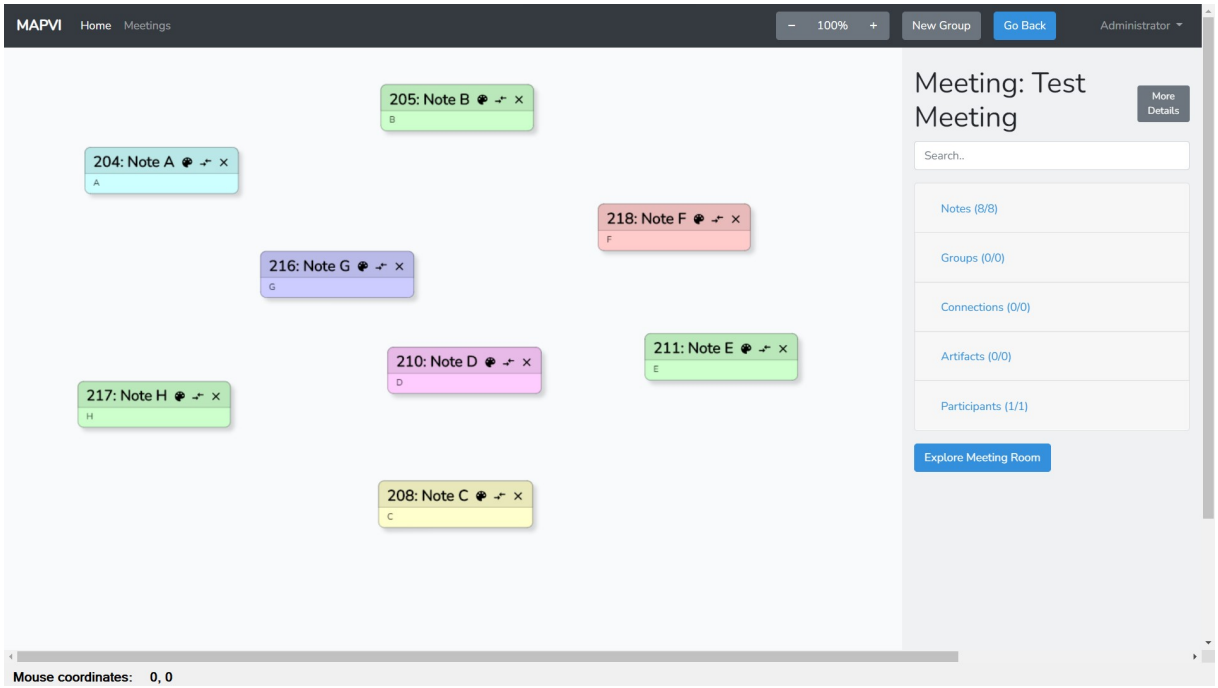



***Figure 2.14:*** *Layout a the Notes on the Whiteboard for the User Study*

As mentioned, there are two parts in the user study. One part took place in the virtual environment of the previous project and the other in the environment of the current project. For simplicity - and since the floor was the most obvious difference, the two environments were introduced to the participant as "stone floor" for the project by S.Liechti and "wooden floor" environment for the current project by C. Merkle.

In addition, to test if there were any changes in participant behaviour resulting in different pointing orientation when the user was standing close to the whiteboard, the gestures were repeated twice in the yellow zone 2 (see Figure 2.9(b))in the wooden floor environment. First, the "Near Ray" was used for a set of gestures. Then, switching to the regular "Eye-Finger Ray", the set was repeated. The participant was not aware of the change of settings.

Besides, to avoid frustration effects or any adaptions in their gesturing behaviour, the participants were not given any feedback on the recognition.

### 2.5.3 Procedure of the User Study

The participants were first given a short introduction to the study. The purpose of the three gestures were briefly explained, but no further instructions on how to perform them were given. Then, they were asked for written consent with which they agreed to the use of their data. Next followed the first questionnaire consisting of questions about the demographics, VR-experiences, the personal innovativeness of the participant, and also the first Simulator Sickness Questionnaire (SSQ). Afterwards, the participants were instructed in the use of the VR devices, and the first task began. During the task, the output of the program printed in the console of *Unity* was monitored and the recognition output was manually entered into the table. When the task was finished, the participants were instructed to take off the HMD and put down the trackers or controllers and to fill in the second questionnaire, which included the SSQ, NASA TLX and the usability scale. Subsequently followed an explanation of the usage of the devices from the second task and the same procedure was repeated in the second environment. The final questionnaire concluded with the question about the personal preference of the two environments.

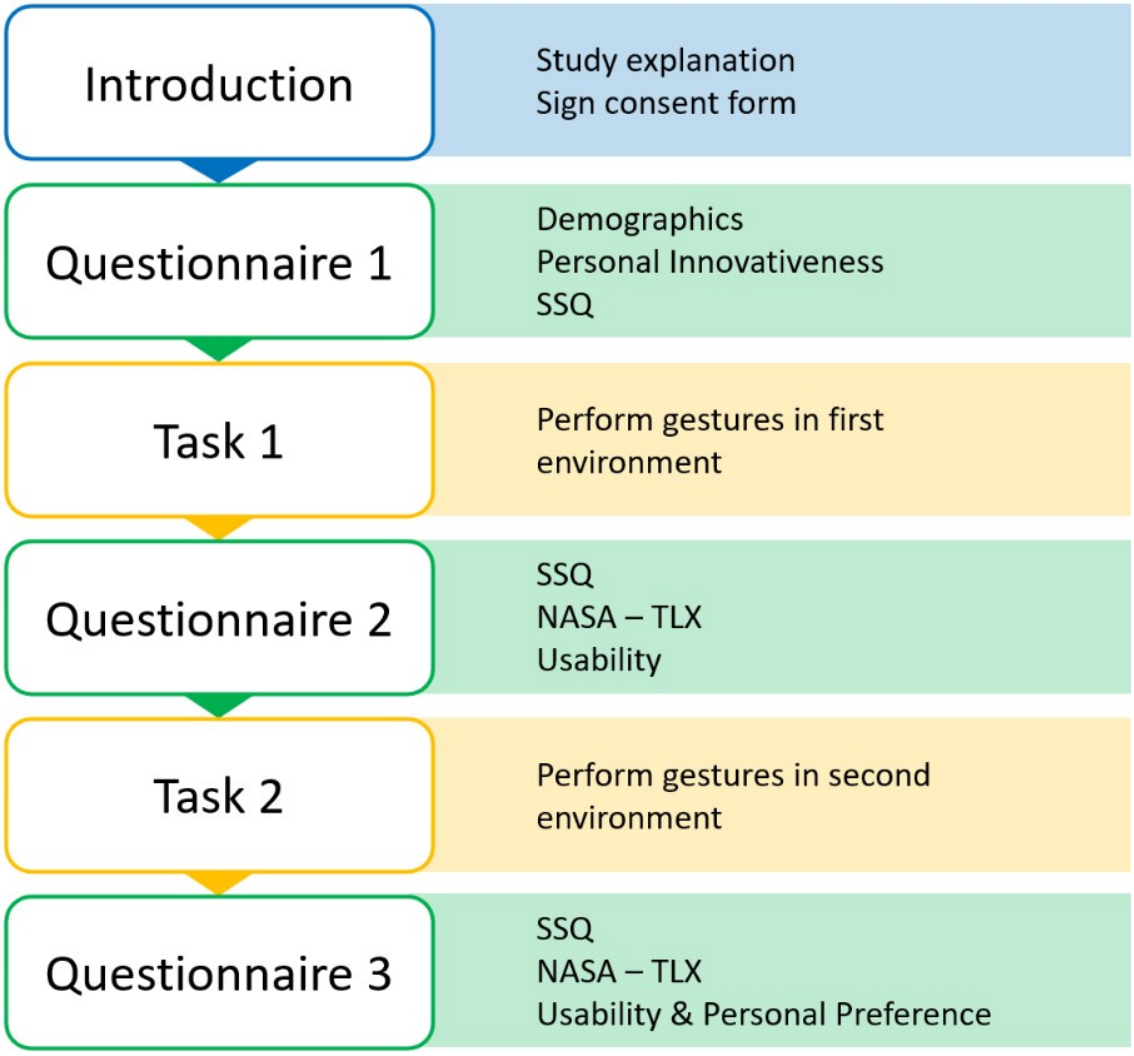



***Figure 2.15:*** *Outline of the User Study*

### 2.5.4 Questionnaires

To obtain information about the participants and receive some subjective data from them, the following questionnaires were used:

- Demographics Questionnaires
- Personal Innovativeness Questionnaire
- Simulator Sickness Questionnaire (SSQ)
- System Usability Scale Questionnaire (SUS)
- NASA Task Load Index (NASA TLX) Questionnaire
- Personal Preference

#### Demographics Questionnaire

The demographic questionnaire is used to establish an overview of the characteristics of the participants, like age, gender and profession. Importantly, one question informed the instructor about the dexterity of the participant, so that the environment could be adapted as mentioned.

#### Personal Innovativeness Questionnaire

The *Personal Innovativeness* questionnaire is used to obtain information on the participant's experiences in VR and also to investigate possible correlations between technological interests and the performance of the users in the virtual environment. It consists on one question asking for previous experience in VR and three further questions about the technological affinity of the participant in a range from one to seven.

#### Simulator Sickness Questionnaire

Simulator sickness happens similar to motion sickness, when the brain perceive different informations about the own motion from the eyes and the sense of balance in the ears. Every person has a different sensitivity to this sickness and moreover, it can be highly dependent on the used VR system or the virtual environment the person is placed in. The SSQ proposed by Kennedy [KLBL93] consists of questions on 16 symptoms which are each rated from zero (no symptoms) to three (severe symptoms). All these sub-symptoms relate to the following three distinct symptoms: oculomotor (overexertion of the eyes), nausea, disorientation, which occur most often for VR and other simulations. For this study, the SSQ was used to find if any participants needed to be excluded from the data set due to severe side-effects they might have been experiencing. Furthermore, it was used to analyse if the two different environments had a measurably different impact on the participants in general.

#### System Usability Scale

The System Usability Scale (SUS) is a questionnaire containing ten questions to define how useful a participant finds a system. Thereby, the three main aspects effectiveness, efficiency and satisfaction are

considered and rated on a scale from one to five [Bro13]. For this thesis, the SUS Questionnaire was mainly used to compare the two different systems the participant were using. Since they did no get any feedback whether the recognition succeeded or not, the effectiveness and efficiency were only rated based on the participant's intuition. For the evaluation, all scores were weighted equally.

#### NASA Task Load Index

The NASA Task Load Index (NASA TLX) questionnaire was originally developed by NASA's Ames Research Centre and allows a subjective workload assessment for operators working with any kind of human machine interfaces [HS88], [WNA]. The six questions cover the topics mental demand, physical demand, temporal demand, performance, effort and frustration. These question were asked after each of the tasks and served mainly as a comparison of how the two tasks were perceived by the participants.

#### Personal Preference

The final question was used to inquire about personal thoughts of the participants concerning the two systems. It was simply formulated by: "which environment do you prefer and why?"

### 2.5.5 Objective Data Collection

Besides the questionnaires there was also objective data collected. During the study, a table with every single performed gesture, as shown in 2.16, was filled out manually by the instructor: A check mark for correctly recognised gestures, an "X" if the recognition failed completely, and the letters of the notes that were incorrectly recognised. In addition, the output of the recognition system was printed into a text file, including the exact time when the recognition algorithm was started and when notes got detected.

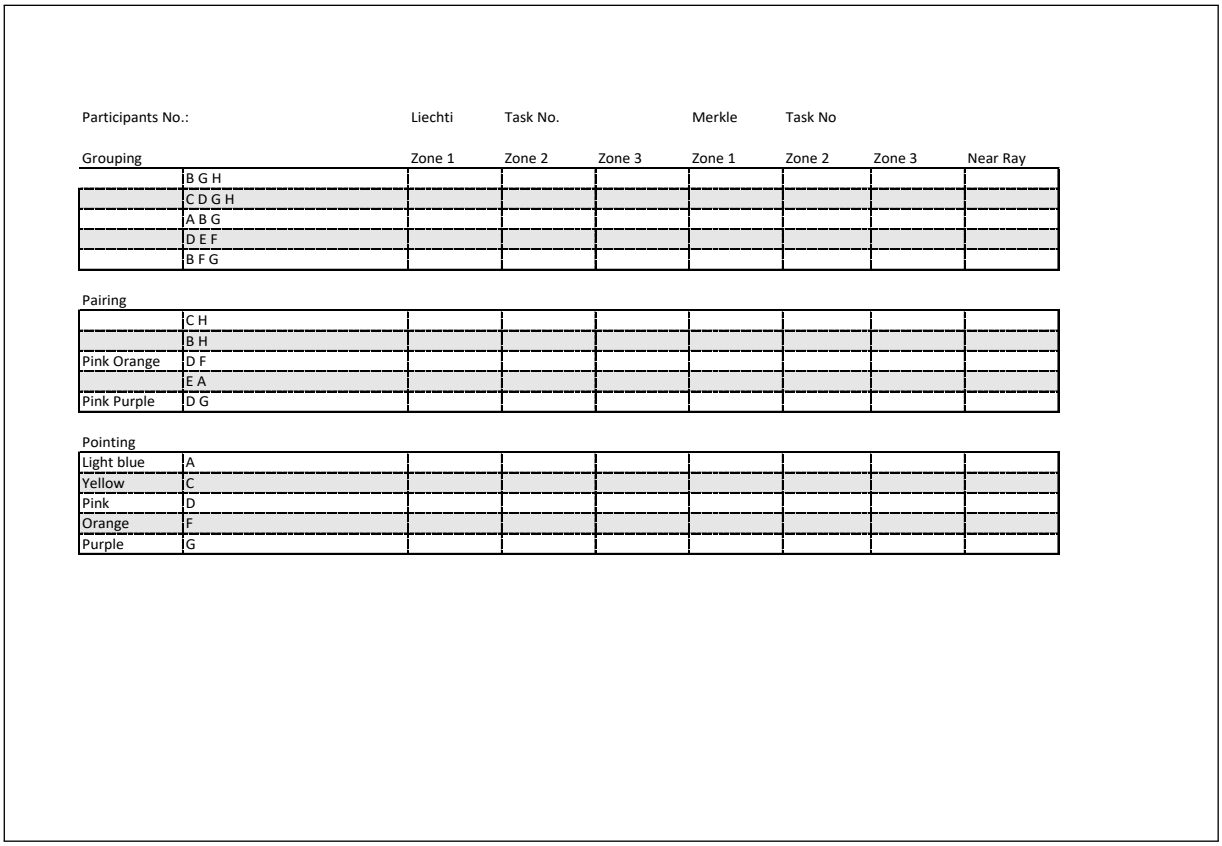
## 2.6 Data Analysis

This section is about the different analysis methods that were used for all obtained data. The first subsection explains how the different questionnaires were evaluated and the second subsection explains the analysis of the objective results.

### 2.6.1 Questionnaires Analysis

For the analysis of the questionnaires, all answers were transferred to an *excel*-sheet and sorted for each participant. The first questionnaire regarding the demographics of the participants were simply evaluated by comparing and summarising all answers. The second, personal innovativeness questionnaire was evaluated by calculating the mean value and the standard deviation of the answer scores for the general overview and also by comparing the results for each participant individually. If the participants scores are in the extremes, the following questionnaires are looked at in more detail to find correlations.

The SSQ was evaluated following the the original proposal of the SSQ [KLBL93]. Thus, each score for nausea, oculomotor and disorientation were calculated by weighting the sums of the associated subsymptoms per participant. In addition, the overall SSQ score, which is a weighted sum of the three groups of sicknesses, was computed. The scores before and after each task are then compared once per



| Participants No.: | | Liechti | Task No. | | Merkle | Task No | | |
|---|---|---|---|---|---|---|---|---|
| Grouping | | Zone 1 | Zone 2 | Zone 3 | Zone 1 | Zone 2 | Zone 3 | Near Ray |
| | B G H | | | | | | | |
| | C D G H | | | | | | | |
| | A B G | | | | | | | |
| | D E F | | | | | | | |
| | B F G | | | | | | | |
| Pairing | | | | | | | | |
| | C H | | | | | | | |
| | B H | | | | | | | |
| Pink Orange | D F | | | | | | | |
| | E A | | | | | | | |
| Pink Purple | D G | | | | | | | |
| Pointing | | | | | | | | |
| Light blue | A | | | | | | | |
| Yellow | C | | | | | | | |
| Pink | D | | | | | | | |
| Orange | F | | | | | | | |
| Purple | G | | | | | | | |

***Figure 2.16:*** *Gesture table with the chosen gestures on the left and the zones on the top*

participants and once for the group averages to find the impact the environments and the study had on the participants. The System Usability Scale and the NASA TLX were analysed similar to the personal innovativeness questionnair, by computing the average of the answer scores, the standard deviation and also the total distribution for each question. Finally, the last question about the personal preference was evaluated by comparison of the answers.

### 2.6.2 Table Analysis

For the data analysis all 30 filled in tables from the user study are summarised in an *Excel*-sheet, which can be found in the appendix. The resulting table gives an overview of how many gestures were recognised correctly (True Positive (TP)), not at all (False Negative (FN)) or with errors (e.g. wrong notes included) False Positive (FP) depending on each gesture type and zone. Then, the recognition was evaluated for different aspects. Therefore, two values are always calculated: The recognition rate and the precision. The recognition rate represents the percentage of correctly detected gestures and the precision is defined as the ratio of the number of correctly detected gestures to the number of total detected gestures (including false positives). They are calculated as follows:

$$Rate = \frac{\#TP}{\#(TP + FP + FN)} \qquad Precision = \frac{\#TP}{\#(TP + FP)}$$

For better illustration and comparison of all the data, there were graphs created using *Microsoft Office Excel*.

# Results and Discussion

## 3.1 Results

In the first subsection the subjective data is evaluated. This data refers to the questionnaires and point out how the participants of the user study felt about both environments and tracking devices. Afterwards an analysis of the objective data follows. The objective data is obtained from the forms that were filled out as part of the user study and shows how well both recognition algorithms work.

### 3.1.1 Subjective Results

The subjective results are given by evaluating the questionnaires containing the personal innovativeness, the Simulation Sickness Questionnaire (SSQ), the NASA Task Load Index (NASA TLX), the System Usability Scale (SUS), and the personal preferences.

According to the Personal Innovativeness Questionnaires, four persons experienced VR the first time during the user study, 18 participants had some minor experiences with VR (less then 5 hours), eight participants were already more familiar with being in a virtual environment (Table 3.1). Furthermore, most participants considered themselves rather curios and friendly towards new technologies. However, by comparing the results of the Personal Innovativenss Questionnaire and the results of the ratio of gestures per participants, there were no correlations found.

| | 0 – 5h | 5 – 20h | 5 – 20h | >20h |
|---|---|---|---|---|
| Number of participants | 4 | 18 | 5 | 3 |

***Table 3.1:*** *Experience in VR*

The SSQ scores showed there was only one participant who felt a significant increasing nausea (from 0 to 38.2) and disorientation (from 0 to 41.8) after both tasks in VR. But since the results for the recognition of his gestures did not differ much from the other participants, they were still taken into account. Furthermore, he stated after the task that he only started to feel more sick after taking off the HMD.

Otherwise, it was found that there were no severe changes in any participants well-being. On average, there was a minimal increase of ocular overexertion but also a decrease of nausea and disorientation for the room with the wooden floor. For the stone floor environment from the previous project, the factor of the oculomotor stayed the same as before the task, disorientation factor decreased slightly, but conversely, the nausea factor increased. The exact differences of the averages can be seen in Table 3.2, where the scores, which vary due to the different weighting for each disease, range from zero to about 200. However, the changes were small and hence considered negligible.

| | Nausea | Disorientation | Oculomotor | SSQ score |
|---|---|---|---|---|
| Before study | 10.81 | 19.02 | 16.42 | 17.45 |
| After S.Liechti's project | 11.77 | 18.56 | 16.42 | 17.70 |
| After C-Merkle's project | 9.54 | 17.17 | 18.19 | 17.33 |

***Table 3.2:*** *Average score for the main symptom*

From the SUS questionnaire it can be seen, that the participants find both systems are in general easy to use and also well integrated. Moreover, most participants feel confident to use the systems and believe other persons would be able to learn quickly how to use it as well. When comparing both systems, it is found that the system in the wooden environment and the use of the strap-on trackers are preferred in all categories. The more detailed results can be found in Table A.1.3.

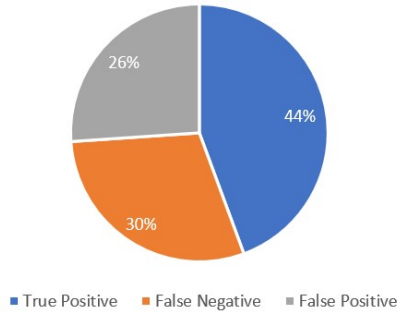
From the NASA TLX questionnaire, it can be obtained, that both tasks, using the different tracking systems, are rated very easy and neither physical nor mentally demanding. In average, the participants felt very secure in both tasks and did not feel any rush in performing their tasks. The only significant difference is, that the participants felt slightly more annoyed and irritated in the stone floor environment, using the controllers. The evaluation of the results for the NASA TLX can be found in Table A.1.3.

The last question about the personal preference of the participants shows that using trackers inside the room with the wooden floor is generally more preferred. 24 out of the 30 participants answered that the room with the wooden floor is designed more warm and feels more realistic like a meeting room. Also, it is often stated, that it is more convenient, to use the trackers and having the hands free. Furthermore, many participants wrote, that it feels more natural to be able to see the hand. From the other statements, two participants find, that the white wall behind the white board is irritating and one participant prefers holding something to point with. Another participant prefers the stone floor environment, due to his claustrophobia, since the room with wooden floor has no doors. Last but not least, there was only one participant, who had no preferences. The original answers can be found in the appendix in A.1.3.

### 3.1.2 Objective Results

The objective results of the study are composed of an analysis of the data table, which can be found in the appendix in Section A.1.3. First, a general overview of the recognition algorithm of both the task in the current project by C. Merkle and the task in the previous project by S. Liechti is given to see how well the recognition in general works in each of the projects. Then, the differences of the recognition for the three gesture types are elaborated for both projects. Afterwards, some selected gestures, which were considered extreme in the range of possible user inputs, are looked at in more detail, but only for the current project. In addition, the dependencies of the total gesture recognition on the participant's location is reviewed. Moreover, the gesture recognition for each gesture type and the relation to the three zones is evaluated, again for both projects. Finally, the recognition rate and precision of the two different rays that were tested are determined and compared to find if there was any change of habits, if a user was standing next to the whiteboard in the second, yellow zone (Section 2.9).

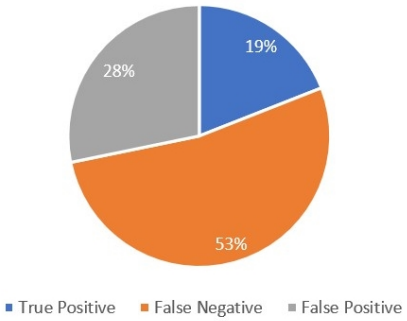
In total there are 1350 gestures performed per task (using the regular "Eye-Finger Ray" in the current project). The total recognition rate of this current project for the "Eye-Finger Ray", was 44% with a precision of 62%. 30% of the gestures were not recognised at all and in 26% of the cases some notes were falsely detected as illustrated in Table 3.3.




| | Total gestures |
|---|---|
| Rate [%] | 44 |
| Precision [%] | 62 |

***Table 3.3:*** *Recognition of all gestures in total in C. Merkles project*

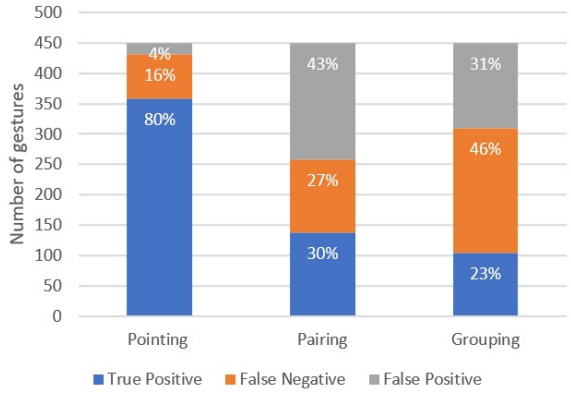
In the previous project, both the rate and the precision were significantly smaller, as can be seen in Table 3.4. The rate of the previous project is only 19% and the precision 40%. Moreover, there were 53% of the gestures not recognised at all and in 28% of the recognition cases, there were notes detected erroneously.




| | Total gestures |
|---|---|
| Rate [%] | 19 |
| Precision [%] | 40 |

***Table 3.4:*** *Recognition of all gestures in total in S. Liechti's project*

Looking at the gestures more closely, it was obtained that the pointing gesture recognition was the most accurate at a rate of 80% and a precision of 95% in the current project. The recognition of the dynamical gestures, pairing and grouping, was less successful. The pairing gesture had a recognition rate of 30% and a precision of 42%. Similar values were obtained for grouping with a total rate of 23% and a precision of 42%.




| | Pointing | Pairing | Grouping |
|---|---|---|---|
| Rate [%] | 80 | 30 | 23 |
| Precision [%] | 95 | 42 | 42 |

***Table 3.5:*** *Recognition depending on gesture type in C. Merkle's project*

Similar values for the different type of gestures were obtained in S. Liechtis work, but with recognition rates and precision's below half of what was achieved in this project, as illustrated in Table 3.6. The Pointing gesture had a total rate of 38 % and a precision of 66% and the pairing gesture a rate of 15% and a precision of 29%. Grouping was almost never recognised and had only a rate of 4% and a precision of 13%.


| | Pointing | Pairing | Grouping |
|---|---|---|---|
| Rate [%] | 38 | 15 | 4 |
| Precision [%] | 66 | 29 | 13 |

***Table 3.6:*** *Recognition depending on gesture type in the S. Liechti's project*

Considering the individual pairing and grouping gestures in the recent project by C. Merkle, the following results presented in Table 3.7 for pairing and in Table 3.8 for grouping.

| | [C, H] | [B, H] | [D, F] | [E, A] | [D, G] |
|---|---|---|---|---|---|
| Rate[%] | 36 | 18 | 42 | 14 | 42 |
| Precision[%] | 48 | 26 | 52 | 21 | 58 |

***Table 3.7:*** *Rate and precision for each pairing gesture*

| | [B, G, H] | [C, D, G, H] | [A, B, G] | [D, E, F] | [B, F, G] |
|---|---|---|---|---|---|
| Rate[%] | 3 | 22 | 34 | 17 | 39 |
| Precision[%] | 6 | 45 | 63 | 31 | 66 |

***Table 3.8:*** *Rate and precision for each grouping gesture*

For pairing [E, A] across the whiteboard, the performance can be found in Table 3.9, where the recognition rate was 14% the precision 21%. From the false positives it was observed that in 39 of 49 cases, the system recognised a pointing gesture at either note E or A , while in 9 cases a grouping of the whole whiteboard was recognised.

| Rate | Precision | TP | FN | FP |
|---|---|---|---|---|
| 14% | 21% | 13 | 28 | 49 |

***Table 3.9:*** *Analysis for pairing [E, A]*

The paring gesture [H,B] had a slightly higher recognition rate with 18% and a precision of 25% and the total amount of false positives was 46 as can be seen in Table 3.10. Taking a closer look on these, one could find that there were 28 gestures, were only B or H was recognised as Pointing and 11 of the gestures were falsely recognised as grouping. For the grouping gesture [B, G, H] there were only 3 out of 90 gestures recognised with a precision of 6%. In absolute numbers, there were 47 gestures falsely recognised and 40 not recognied at all as can be seen in Table 3.11. When the results of these mistaken gestures are examined more closely, it can be found that in 22 cases, there were too many notes recognised, in 9 cases the gesture was recognised as pairing and in 14 cases, there was only one note (either B or H) recognised.

| Rate [%] | Precision [%] | TP | FN | FP |
|---|---|---|---|---|
| 18 | 25 | 16 | 28 | 46 |

***Table 3.10:*** *Analysis for pairing [H, B]*

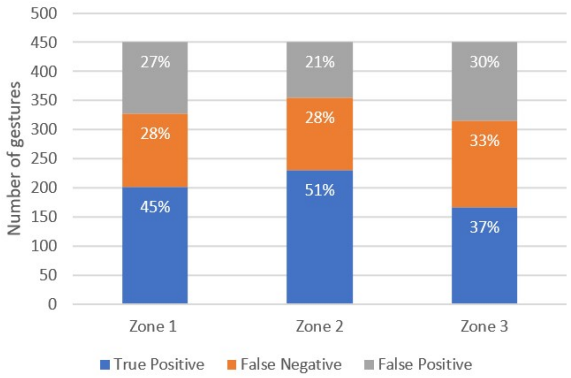
| Rate [%] | Precision [%] | TP | FN | FP |
|---|---|---|---|---|
| 3 | 6 | 3 | 40 | 47 |

***Table 3.11:*** *Analysis for grouping [G, B, H]*

Reviewing the recognition depending one the zones (as depicted in Figure 2.9(b)), it can be seen that the accuracy decreases with the distance to the whiteboard and the more to the side the person is standing. For the current project, in Zone 2, directly in front of the whiteboard, the total recognition rate was 51% and the precision 71%, followed by Zone 1, more far away with a rate of 45% and a precision of 62% and Zone 3, sideways of the whiteboard, with a rate of 37% and precision of 55%.

For S. Liechtis project, the differences between the recognition in the three zones are more significant, as depicted in Table 3.13. In zone 2 the rate and precision were the highest with 34% and 52%. In Zone 1 the rate and precision were 14% and 30% while the rate and precision were only 8% and 29% in zone 3.

Table 3.14 shows the detailed results depending on each gesture in the different zones for the current project. It can be seen that the rate and precision of the pointing gesture is always the highest and only slightly dependent on the zones. However, the results obtained for the other gestures are different: For pairing, the same rate and precision were obtained for zone 1 and zone 2, but both decreased for zone 3.




| | Zone 1 | Zone 2 | Zone 3 |
|---|---|---|---|
| Rate [%] | 45 | 51 | 37 |
| Precision [%] | 62 | 71 | 55 |

***Table 3.12:*** *Recognition depending on zone in the C. Merkle's project*


| | Zone 1 | Zone 2 | Zone 3 |
|---|---|---|---|
| Rate [%] | 14 | 34 | 8 |
| Precision [%] | 30 | 52 | 29 |

***Table 3.13:*** *Recognition depending on zone in the S. Liechti's project*

For grouping, the highest rate and precision were achieved for zone 2 with a rate of 31% and a precision of 58%, while zone 1 and 3 share nearly the same values: For grouping, zone 1 had a recognition rate of 19 % and a precision of 35, while Zone 3 was one percent higher for both.

| | Zone 1 | | | Zone 2 | | | Zone 3 | | |
|---|---|---|---|---|---|---|---|---|---|
| | Point | Pair | Group | Point | Pair | Group | Point | Pair | Group |
| C.Merkle's project | | | | | | | | | |
| Rate [%] | 81 | 35 | 19 | 89 | 35 | 31 | 69 | 22 | 20 |
| Precision [%] | 95 | 46 | 35 | 99 | 46 | 58 | 90 | 32 | 36 |
| S.Liechti's project | | | | | | | | | |
| Rate [%] | 32 | 11 | 0 | 63 | 29 | 11 | 20 | 4 | 1 |
| Precision [%] | 55 | 22 | 0 | 87 | 40 | 22 | 45 | 15 | 8 |

***Table 3.14:*** *Recognition depending on zones and gesture types in both project*

For S. Liechti's project, the rates and precision of the gestures in the different zones vary greatly. For pointing, the rates and precisions were highly dependent on the zones. For zone 2 they were highest with a recognition rate of 63% and precision of 87%, while for zone 1, the values were smaller with 32% and a precision of 55%. Finally, for zone 3 the recognition rate decreased to 20% with a precision of 45%. The pairing performance degraded similarly, with a rate of 29% and precision of 40% in zone 2, a rate of 11% and precision of 22% in zone 1 more far away, and only a rate of 4% and precision of 15% for the side view in zone 3. For grouping, non-gestures were recognised in the first zone while only 1% of the grouping gestures were recognised correctly with a precision of 8% in zone 3. The best recognised grouping gesture is found in zone 2, still with only a rate of 11% and precision of 22%.

When comparing the "Eye-Finger Ray" and the "Near Ray" in Zone 2, similar results are found, regarding the ratio of the performances for each gesture type. Nevertheless, in total the regular "Eye-Finger Ray" had a slightly higher total recognition rate with 51% compared to the "Near Ray" with 48%. The total precision with 71% was the same for both rays. A comparison of the recognition rates of both rays for each gesture can be seen in Figure 3.1. Thereby, it can be concluded that the recognition rate from the "Eye-Finger Ray" is higher then the "Near Ray" for pointing and grouping, but not for pairing. The precision of both rays also varies for the different gestures. While the precision for pointing is 99% for both rays, for pairing it is 11 % higher using the "Near Ray" but 6% lower for grouping.

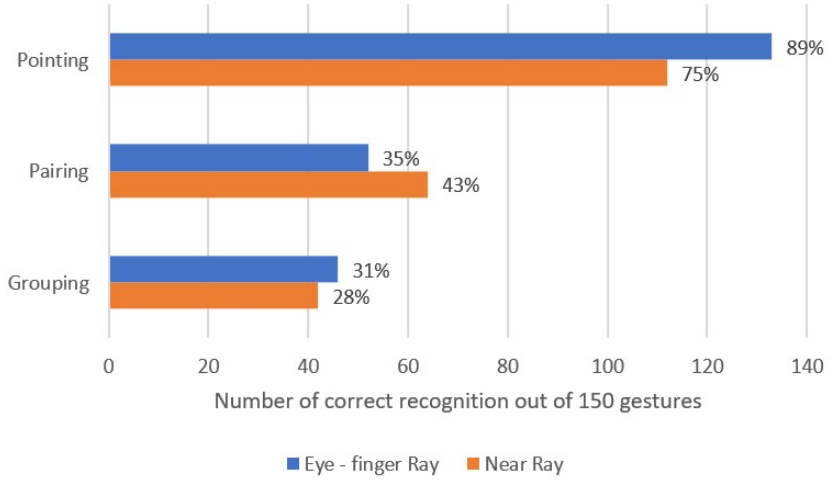



***Figure 3.1:*** *Comparison of the Rates of the "Eye-Finger Ray" and the "Near Ray"*

| | Eye-Finger Ray | | | Near Ray | | |
|---|---|---|---|---|---|---|
| | Point | Pair | Group | Point | Pair | Group |
| Rate [%] | 89 | 35 | 31 | 75 | 43 | 28 |
| Precision [%] | 99 | 46 | 58 | 99 | 57 | 52 |

***Table 3.15:*** *Recognition depending on "Eye-Finger Ray" or "Near Ray" and gesture type*

## 3.2 Discussion

In this section, the meaning of the results is elaborated. After evaluating the results obtained from the questionnaires, the objective data is examined. The discussion follows a similar structure as the results. First an overall look is taken on the total recognition and the recognition of the different gesture types. Afterwards, a comparison of the recognition in the different zones is given. Then follows a comparison of the recognition system of the current and the previous project, and finally a comparison of the two different rays, which are tested in the zone right in front of the whiteboard.

### 3.2.1 Evaluation of the Questionnaires

From the questionnaires it was found that most of the participants were educated in a similar field; mainly young students with technological affinities. But since the way a person is using gestures is very individual and not dependent on education, the study can still be seen as representative. Furthermore, in the SSQ it was found that almost non of the participant had significant issues with simulator sicknesses. Moreover, it was rather surprising to find, that many participant even felt partly better after the study. Besides, there were no correlation found for the simulator sickness and the way how a person performed the gestures. From the NASA Task Load Index and the System Usability scale, it can be obtained, that the system is considered as useful and easy to learn, even though many participants did not found an use in daily life. Since it is actually made for very special circumstances, mainly to have a recognition system for BVIPs, these are valid results.

### 3.2.2 Comparison of the Gesture Types

For the overall recognition algorithm it is obtained, that the total rate of recognition with 44% is quite low. In 30% of the cases, the recognition algorithm was not triggered at all, due to difficulties with the algorithm to start the gesture recognition, which is explained in more detail later in Section 3.2.6. The 28% of falsely detected recognition had multiple reasons. In some cases grouping and pairing are confused, there are to many or to few notes detected, or when the pointing direction was too inaccurate, sometimes there are also completely false notes detected.

The pointing recognition is working quite well with a rate of 80%, since it is the most simple movement that cannot be much varied. Furthermore, the pointing detection algorithm is not dependent on the start of the gesture recognition like the other gestures. The problem why 20% of the pointing gestures were not recognised is, that the pointing timer is set too long. Many participants raised their arm and the notes were hit precisely, but they lowered their arm, before the two seconds passed.

For pairing only 30 % of the gestures were recognised, with the precision of 42 %. The reason for the low precision is that often only one of the notes was recognised. Furthermore, while monitoring the output of the recognition system, it was observed, that the recognition system often also recognised both paired notes, but subsequently as two separate pointing gestures. This happened, when multiple pairing gestures where recognised within a repetitive movement, but for each gesture only one note was found.

On the other hand, when the pointing the back and forth movement between the two notes is not repeated at all and the pairing movement appeared to be more like pointing at two notes sequentially as depicted in Figure 3.2, the pairing gesture also remained unrecognised. Moreover, the participants were less likely to point for two seconds per note, when thinking of moving forward to the next note, so that not even pointing was recognised.

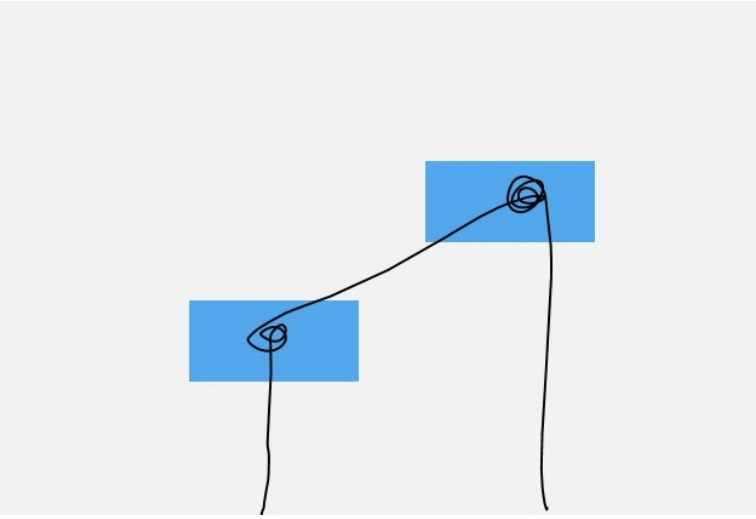


***Figure 3.2:*** *Pointing on two notes subsequently for pairing*

Another reason, why so many pairing gestures were recognised wrong is that there is no clear line for differentiating the elliptic grouping and pairing movement and thus, they often get confused.

For the grouping gesture, the rate was only 23% and the precision 42%. The main problem, besides the start of the algorithm, lies in the rectangular form of the detection field. Any notes nearby the actual grouped notes were likely to be detected additionally, since the area of the rectangle was often much larger than the area included by the movements trajectory as can be seen in Figure 3.3. Thus, it can be stated, that the rectangle is an insufficient approximation for the movement trajectory of the grouping gesture.

Furthermore, the accuracy of the gesture decreases with a faster speed of movement. It was observed that the participants, which in general moved faster, would also do for example larger circles around the notes for a grouping gesture. Consequently, too many notes are recognised.

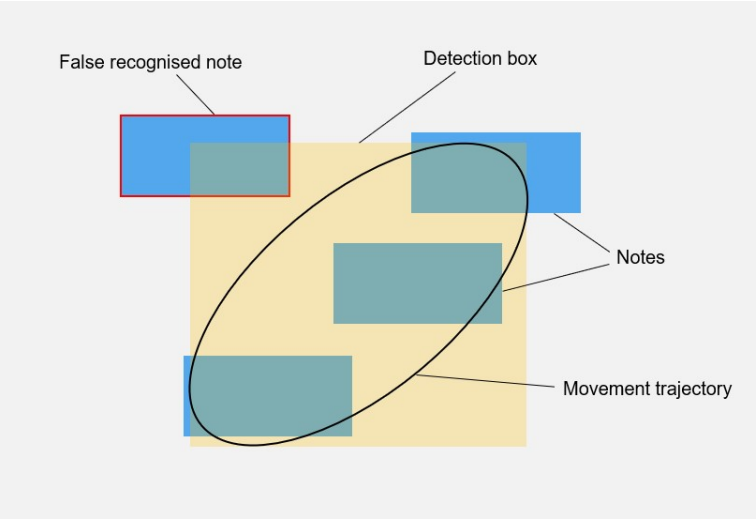



***Figure 3.3:*** *Grouping gesture with one additional false detected note*

### 3.2.3 Comparison of Different Zones

Comparing the three different zones in total, it is found that the algorithm worked quite similar for all of them. Of course there were differences in the results: Inaccuracy increased with the distance from and angle towards the whiteboard. But this can be easily explained by the interim theorem: the longer the distance to the pointing target, the heavier is the effect of the smallest movement. Thus, the inaccuracy of the pointing gesture of a person arises. Therefore it can be said, that the recognition worked quite well. Moreover, considering a group meeting in reality, it is unlikely, that the moderator is gesturing on small objects from far away, since it would also be getting more difficult for other meeting attendees to identify the selected notes. The result, that the rate and precision for the pairing recognition is lower in zone 2 can be explained by looking at each gestures. For example, pairing [E,A] has a large distance, all over the whiteboard. Since the distance is big, the moving time between each notes is rather long, thus the recognition spheres trail starts to vanish before the gesture is finished. Furthermore, when pairing a note on the side, while standing in front of the whiteboard, the ray from the head through the finger might be missing the actual target, since the finger would be pointing more parallel then towards the whiteboard and the previous explained interim theorem holds again. This would have a heavier effect on pairing then on pointing, since paring is more dynamical, and the participants would aim less exact then holding the hand still.

### 3.2.4 Comparison of Different Rays

Comparing the results for the "Near Ray" and the "Eye-Finger Ray", it was found that the ray projected perpendicularly onto the whiteboard was less accurate for the pointing and the grouping gesture, but it performed better for pairing. The reason for the imprecise pointing gesture could be easily observed: When standing close to the whiteboard, the normally projected ray was activated - even though the pointing target might have still been further away, e.g. for notes in the far corner of the board. Thus, the ray detected the note next to the finger instead of the one pointed at. The same problem occurred for grouping. the higher inaccuracy of pairing recognition for the "Eye-Finger Ray" might be due to the shallow pointing angles w.r.t. the whiteboard as they occur when standing close to it. These cause larger changes in the movement trajectories, even for small changes in the pointing angle. Thus, the "Eye-Finger Ray" can be highly error prone when used close to the whiteboard. For the

### 3.2.5 Comparison with S. Liechti's Recognition Algorithm

As the results show, the recognition algorithm of the previous project by Simon Liechti did only succeed in 19% of the cases – less then a half compared to the current project by C. Merkle. Main reason for his much lower recognition rate is the ray for determining the pointing direction, that he used. Besides, holding the controllers was not only less comfortable for the user, it was very likely that its position gets shifted from time to time. The effect of the bad accuracy can be observed best by comparing the results of the three zones. The bigger the distance and the more side-wards to the whiteboard the participant stood, the bigger is the inaccuracy. So when it came to zone 3, only 8% of the notes were detected correctly, compared to the 37% of the current project. It was observed, that the ray was often not even reaching the whiteboard, since the offset of the arm direction to the target was enormous. Further issues why the recognition of S. Liechti's project does not perform very well, was already stated in Section 2.3.3 and confirmed again in this study. Some of those limitations were resolved and thus, the recognition of the current project performed better then the previous.

### 3.2.6 Limitations

The main limitation of both systems, is the algorithm of the start of the gesture recognition, which is explained in Section 2.3.2. It is very difficult to find an optimal way to detect the start of a gesture and thus, the used method, that one sphere needs to collide into a second, activated sphere was now proven to be insufficient. The spheres do not necessary collide, when the path of the movement was crossing. This could be changed simply by reducing the spheres rate and increasing the lifetime, although other problems might occur: if the distance between the spheres is too small, the angles of their vectors will also get smaller. Thus, it gets harder to detect a pairing gesture. Similar problems arise, when the lifetime of the spheres is too long. In that case the algorithm takes too many spheres in account and therefore uses a too big detection field, when it comes to detecting notes in the grouping gesture.

Another issue is the different gesturing speed depending on the participant. If the movement was too fast, the distance between two spheres were too large, so that the spheres trail would not cross and the recognition algorithm did not get started. If on the other hand the movement was too slow, the lifetime of the sphere was to short, and if there was a gesture recognised, half of the sphere trail already vanished. In other words, the values for the spheres life- and activation time, and the spheres rate are not set ideally and would need to to be adapted for each user.

A further limitation is the similarity of elliptic pairing and grouping gestures. For those, it is barely impossible to differentiate which gesture should be given as output, and when e.g. pairing was recognised as grouping, all notes in the detection box were printed instead of only the two notes in the corners. The opposite occurred, when grouping three notes in a line was initiated (see results of grouping BGH, Section 3.11), but pairing was detected and therefore not all notes were identified. This limitation, which was already observed from the previous project, could neither be solved in this current thesis.

# Conclusion and Future Work

## 4.1 Conclusion

In this thesis a gesture recognition algorithm for pointing, pairing and grouping upon a whiteboard was presented and tested in a user study. VR-devices from *HTC's Vive Pro* are used to track the gesture movements. To determine which kind of gestures are performed, a recognition algorithm is implemented in *Unity*, whereas a VR twin of the whiteboard containing the notes is created. The algorithm is using so called recognition spheres to mark a trail of the gesture movement. These spheres can be used to determine the type of gesture and also the objects at which the gesture is directed. In a user study, each gesture were repeated multiple times in different variations and from different angles and distances. The results of this study show, that the rather static pointing gesture is much easier to determine then the more dynamic pairing and grouping movements. Explanations were given, why dynamic movements are often recognised incorrect or even not recognised at all. The main difficulties are firstly, to deal with the differences of the movements depending on the participant, and secondly, in the definition of the recognition start, which is often not triggered.

## 4.2 Future Work

### 4.2.1 Improvement of the Current Recognition Algorithm

A starting point to improve the system is to test and optimise all parameters of the algorithm. Most of these parameters were chosen based on intuition and experience, but not analysed for their impact on the system's performance. For example, the duration a pointing gesture needs to be held by a user for successful recognition (set to 2s) was observed to be too long. However, it is unclear what the optimal threshold could be. Thus, It is necessary to test which shortest time period can be used so that there are not too many false positives. Other values that needs to be adapted are the lifetime, the activation time and the creation rate of the recognition spheres, as explained in chapter 3.2.
In addition, an optimal angle threshold for the differentiation of the pairing and grouping gestures could be evaluated. Another future work could also focus on finding a better start and end of the recognition algorithm. The challenge remains to identify a distinct starting point on when to differentiate between e.g. the simple raising of an arm and the actual gesture. The same can be done to find the end point of the gesture, i.e. when the arm is lowered. The reason for this additional task is simply that at this state, there are often multiple gestures recognised for one movement. Hence, accurately identifying the end of a gesture could reduce falsely detected gestures and notes. Moreover, another part could be added to the algorithm so that only the most frequently recognised gesture is returned to determine the gesture that was most likely intended. But this could only be done after the gesture was already performed and there would be a slight time delay between the gesture and the output of the recognition.

Considering the pairing gesture, one approach to improve the system is to find the nearest notes of the spheres at the edges of the movements trajectory, instead of using a predefined radius of the corner sphere. This way, errors occurring when the spheres do not squarely hit one of the notes that is paired, is reduced. Another improvement that could be added is a function, detecting whether two pointing gestures are performed successively, for example by adding a second timer in which further pointing gestures are observed.

To improve the precision of the grouping gesture, an elliptic instead of a rectangular detection box should be used. Thereby, the "capsule collider", provided by *Unity*, could be used instead of the "box collider". This would be a more natural choice, considering the shape traced by users performing the grouping gesture is usually en ellipses.

### 4.2.2 Other Methods for the Recognition Algorithm

The pointing and pairing gestures could be improved by analysing the speed of the movement. Since, for example, when pointing the velocity of the movement is slowed down to near zero when the arm is held still. Also, in the case of pairing, the speed of motion reaches its smallest values close to the two notes that are supposed to be paired. The velocity of an object can be easily determined using the function "rigid body velocity" given in *Unity*. Hence, a little sphere could be implemented at the tip of the recognition ray without much effort and thus, when its speed is recognised to be nearly zero, its current position can be used to find any nearby notes. However, for the grouping gesture, where the speed of the motion is unlikely to have such big alterations, other changes might be necessary. For this, a "lasso-tool", like often used in photo editing software, could be used. Thereby, instead of using the spheres trail, the movement would be drawn as a line on the whiteboard, and if this line intersects with itself, the gesture recognition searches for any notes in the area surrounded by it.
A very different approach, which could be interesting for the gesture recognition, is to train machine

learning algorithms to differentiate the gestures and detect the notes. For this, a study needs to be conducted to capture the marker trajectory and intended gesture as a labelled data pair for a large number of gestures and users. If enough data can be collected and a suitable supervised learning model is chosen, such a system could fine-tune itself and yield promising results.

### 4.2.3 User Study

For the next user study, it could be beneficial not to have a virtual environment, but to actually use the system in reality, on the real whiteboard and without the HMD. Thereby, the side effects of the virtual environment may be eliminated, and the actual application can be replicated in a more realistic set up. This way, the accuracy of the virtual hand can also be tested, as many participants were using their virtual hand as a reference point and not their actual hand, which they obviously could not see. Moreover, the participants were more likely to feel like the only person in the room and tend to stand straight in front of a board, instead of sideways next to the whiteboard, as they perhaps would, when presenting something to other people.

### 4.2.4 Features for the User

This section is about improvements, not for the recognition algorithm, but for the user-facing implementation. Many parameters have to be written manually into the program. An automatic handling of these parameters could make the usage of the application less cumbersome. For example, the size and position of the whiteboard is one of the issues that could be addressed here. To make the recognition system more accurate, the real whiteboard has to be copied into the virtual space as precisely as possible. Using controllers or two trackers to monitor the positions on two diagonally aligned corners, it would be possible to facilitate the adjustment of size and position of the whiteboard. Hence, the system could be used more flexibly and it could be easier to transfer the set up to different meeting rooms.
In addition, an automatic adjustment of the pixel size of the presentation screen could be of great help. Since every desktop has a different resolution, the size and position of the notes and the screen varies. For now, this has to be adjusted manually in *Unity*. Automating this would make the system less error prone to any offset of the notes. Moreover, a solution would be needed if the zoom function in the *Brainstorming Tool* is used. The zoom factor and also the section on the screen that is displayed is not given so far. Thus, the notes in the virtual and the real room might not be aligned anymore and the recognition does not work properly.

Besides, a better design for the attachment of the trackers is needed, so that they are secured better and cannot turn or slip while being used.

# Appendix

## A.1 Evaluation Data Recognition

### A.1.1 SUS and NASA TLX Evaluation

### A.1.2 Answers to personal preferences

### A.1.3 Data Table

# System Usability Scale – Average, Standard Deviation, and Distribution of the Answers

General Question: To what extent do you agree with the following statements in a scale from 1 to 5? 1 = strongly disagree 5 = strongly agree

| | Average S. Liechti | Average C. Merkle | Standard Deviation S. Liechti | Standard Deviation C. Merkle | Maximum Score S. Liechti | Maximum Score C. Merkle | Minimum Score S.Liechti | Minimum Score C. Merkle |
|---|---|---|---|---|---|---|---|---|
| I think that I would like to use this system frequently. | 2,70 | 3,03 | 1,09 | 1,03 | 4 | 5 | 1 | 1 |
| I found the system unnecessarily complex. | 1,87 | 1,73 | 0,94 | 0,78 | 4 | 4 | 1 | 1 |
| I thought the system was easy to use. | 3,97 | 4,27 | 0,96 | 0,91 | 5 | 5 | 2 | 1 |
| I think I would need the support of a technical person to be able to use this system. | 1,53 | 1,41 | 0,73 | 0,63 | 4 | 3 | 1 | 1 |
| I found the various functions in this system were well integrated. | 3,73 | 3,87 | 0,91 | 0,78 | 5 | 5 | 1 | 2 |
| I thought there was too much inconsistency in the system. | 1,87 | 1,67 | 1,04 | 0,76 | 5 | 3 | 1 | 1 |
| I would imagine that most people would learn to use this system very quickly. | 4,33 | 4,43 | 0,76 | 0,68 | 5 | 5 | 2 | 3 |
| I found the system very much cumbersome to use. | 2,53 | 2,07 | 1,28 | 1,17 | 5 | 5 | 1 | 1 |
| I felt very confident using the system. | 3,97 | 4,13 | 0,81 | 0,68 | 5 | 5 | 2 | 3 |
| I need to learn a lot of things before I could get going with this system. | 1,43 | 1,23 | 0,63 | 0,50 | 3 | 3 | 1 | 1 |

# NASA Task Load Index – Average, Standard Deviation and Distribution of the Answers

Scale from 1 to 10  0 = very low  10 = very high

| | Average S. Liechti | Average C. Merkle | Standard Deviation S. Liechti | Standard Deviation C. Merkle | Maximum Score S. Liechti | Maximum Score C. Merkle | Minimum Score S.Liechti | Minimum Score C. Merkle |
|---|---|---|---|---|---|---|---|---|
| How mentally demanding was the task? | 2,10 | 2,07 | 1,65 | 1,66 | 6 | 6 | 0 | 0 |
| How physically demanding was the task? | 2,23 | 2,03 | 2,36 | 1,87 | 8 | 8 | 0 | 0 |
| How hurried or rushed was the pace of the task? | 1,37 | 1,47 | 1,35 | 1,38 | 5 | 5 | 0 | 0 |
| How successful were you in accomplishing what you were asked to do? | 8,17 | 7,90 | 1,42 | 1,47 | 10 | 10 | 4 | 5 |
| How hard did you have to work to accomplish your level of performance? | 2,53 | 2,60 | 2,10 | 2,31 | 8 | 8 | 0 | 0 |
| How insecure, discouraged, irritated, stressed and annoyed were you? | 1,93 | 1,40 | 1,93 | 1,43 | 7 | 6 | 0 | 0 |

# Which of the two environments do you prefer and why?

**Original Answers**

1. The second (wooden) because we see the hand, which feels more natural to point to locations. the controllers didnt feel that natural
2. Second one(wooden), hand visualisation made the task easier
3. Neither, creepy dude standing behind me
4. Wooden
5. Wood, better input mechanism
6. Second one (wooden). It is much more convenient to use, because you dont need to hold two controllars.
7. On the wooden floor. It is more convinient to handle.
8. Second (wooden), since my left hand is free and the image of a hand is more intuitive
9. Wood
10. Wooden floor
11. Wooden floor: more classroomfeeling, where you  normally held presentations like that
12. The first one(stone), cause its opan space, the 2nd is closed with no chance of escape :/
13. Wooden, because the tracker band was more comfortable to use.
14. Generelly a room but in this case the stones because in the room the background was also white like the screen.
15. Wooden
16. The wooden one since it appeared warmer to me.
17. Wood
18. wooden floor
19. The first environment (wooden), ss it feels more natural
20. Wood because it shows a more comfortable place to present
21. Stone, because then i can direct my fingers better.
22. Second (stone) because i prefer holding something to point with, but preferrably without the second controller on the elbow
23. Wooden Environment because the controller used is much more comfortable and easy to use
24. Wooden floor, because it is more intuitive to point with your hand than with the two controllers
25. Generelly a room but in this case the stones because in the room the background was also white like the screen.
26. First environment(wooden) due to the tracker and the feeling of being inside of a real meeting room
27. First (wooden), justt more appealing. Not a strong preferance though.
28. Wooden floor. It is easier to use the trackers.
29. First one (wooden) - the design just spoke out to me more
30. Wooden world

# Table of Recognition Analysis Merkle

TP = True Positive FN = False Negative FP = False Positive

| Merkle | | Zone 1 | | | Zone 2 | | | Zone 3 | | | Near Ray 1 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Grouping | | TP | FN | FP | TP | FN | FP | TP | FN | FP | TP | FN | TN |
| | B G H | 0 | 14 | 16 | 3 | 15 | 12 | 0 | 11 | 19 | 1 | 11 | 18 |
| | C D G H | 7 | 14 | 9 | 6 | 17 | 7 | 7 | 15 | 8 | 8 | 14 | 8 |
| | A B G | 8 | 14 | 8 | 13 | 12 | 5 | 10 | 15 | 5 | 11 | 16 | 3 |
| | D E F | 3 | 13 | 14 | 8 | 15 | 7 | 4 | 13 | 13 | 7 | 14 | 9 |
| | B F G | 10 | 14 | 6 | 16 | 11 | 3 | 9 | 12 | 9 | 15 | 14 | 1 |
| Grouping total | | 28 | 69 | 53 | 46 | 70 | 34 | 30 | 66 | 54 | 42 | 69 | 39 |
| Pairing | | | | | | | | | | | | | |
| | C H | 11 | 7 | 12 | 13 | 7 | 10 | 8 | 10 | 12 | 12 | 5 | 13 |
| | B H | 6 | 10 | 14 | 8 | 7 | 15 | 2 | 11 | 17 | 13 | 6 | 11 |
| | D F | 17 | 4 | 9 | 11 | 6 | 13 | 10 | 7 | 13 | 15 | 9 | 6 |
| | E A | 6 | 9 | 15 | 3 | 9 | 18 | 4 | 10 | 16 | 5 | 12 | 13 |
| | D,G | 12 | 6 | 12 | 17 | 9 | 4 | 9 | 9 | 12 | 19 | 5 | 6 |
| Pairing total | | 52 | 36 | 62 | 52 | 38 | 60 | 33 | 47 | 70 | 64 | 37 | 49 |
| Pointing | | | | | | | | | | | | | |
| Light blue | A | 26 | 3 | 1 | 28 | 2 | 0 | 17 | 11 | 2 | 20 | 9 | 1 |
| Yellow | C | 24 | 5 | 1 | 25 | 5 | 0 | 17 | 13 | 0 | 21 | 9 | 0 |
| Pink | D | 23 | 5 | 2 | 26 | 3 | 1 | 23 | 6 | 1 | 24 | 6 | 0 |
| Orange | F | 24 | 5 | 1 | 28 | 2 | 0 | 24 | 4 | 2 | 24 | 6 | 0 |
| Purple | G | 25 | 3 | 2 | 26 | 4 | 0 | 22 | 2 | 6 | 23 | 7 | 0 |
| Pointing Total | | 122 | 21 | 7 | 133 | 16 | 1 | 103 | 36 | 11 | 112 | 37 | 1 |
| All Gestures Total | | 202 | 126 | 122 | 231 | 124 | 95 | 166 | 149 | 135 | 218 | 143 | 89 |

# Table of Recognition Analysis Liechti

TP = True Positive FN = False Negative FP = False Positive

| | Zone 1 | | | Zone 2 | | | Zone 3 | | |
|---|---|---|---|---|---|---|---|---|---|
| Grouping | TP | FN | FP | TP | FN | FP | TP | FN | FP |
| BGH | 0 | 19 | 11 | 0 | 16 | 14 | 0 | 27 | 3 |
| C D G H | 0 | 18 | 12 | 6 | 14 | 10 | 0 | 27 | 3 |
| A B G | 0 | 23 | 7 | 1 | 24 | 5 | 1 | 28 | 1 |
| D E F | 0 | 17 | 13 | 4 | 5 | 21 | 0 | 17 | 13 |
| B F G | 0 | 24 | 6 | 6 | 14 | 10 | 1 | 26 | 3 |
| Pairing | 0 | 101 | 49 | 17 | 73 | 60 | 2 | 125 | 23 |
| C H | 8 | 10 | 12 | 14 | 7 | 9 | 2 | 25 | 3 |
| B H | 0 | 19 | 11 | 2 | 13 | 15 | 0 | 28 | 2 |
| D F | 5 | 12 | 13 | 12 | 7 | 11 | 2 | 18 | 10 |
| E A | 0 | 18 | 12 | 0 | 11 | 19 | 0 | 23 | 7 |
| D,G | 4 | 12 | 14 | 15 | 4 | 11 | 2 | 17 | 11 |
| Pointing | 17 | 71 | 62 | 43 | 42 | 65 | 6 | 111 | 33 |
| A | 13 | 16 | 1 | 18 | 10 | 2 | 5 | 25 | 0 |
| C | 14 | 7 | 9 | 22 | 8 | 0 | 11 | 13 | 6 |
| D | 9 | 12 | 9 | 15 | 9 | 6 | 5 | 16 | 9 |
| F | 5 | 12 | 13 | 20 | 8 | 2 | 6 | 12 | 12 |
| G | 7 | 16 | 7 | 19 | 7 | 4 | 3 | 18 | 9 |
| | 48 | 63 | 39 | 94 | 42 | 14 | 30 | 84 | 36 |
| Total | 65 | 235 | 150 | 154 | 157 | 139 | 38 | 320 | 92 |

## A.2 Tracker bracket

Since there were no mounting systems for attaching trackers on the head given, a bracket was 3D printed and filed out of wooden dowels. The bracket had two slots, so that straps could be fastened there. The screw was made of wood to not scratch the trackers internal thread since they had an non metric system, which are hardly found in simple workshops.

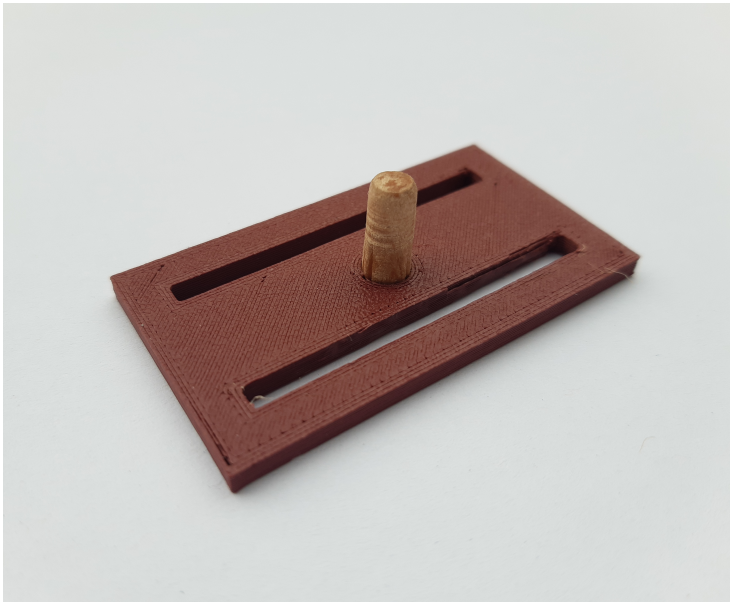


***Figure A.1:*** *Bracket for the headtracker*

# Bibliography


[AI16] Deepak Akkil and Poika Isokoski. Accuracy of interpreting pointing gestures in egocentric view. sep 2016.

[Bam89] Gottfried Bammes. *Die Gestalt des Menschen: Lehr- und Handbuch der Anatomie für Künstler*. Maier, Ravensburg, [6. aufl.], [lizenzausg.] edition, 1989.

[BGD97] John D. Bonvillian, Amanda Miller Garber, and Susan B. Dell. Language origin accounts: was the gesture in the beginning? *First Language*, 17(51):219–239, apr 1997.

[Bro13] John Brooke. Sus: a restrospective. *Journal of Usability Studies*, 8(2):29–40, February 2013.

[CV10] Hélène Cochet and Jacques Vauclair. Pointing gesture in young children. *Gesture and Multimodal Development*, 10(2-3):129–149, dec 2010.

[DJP03] Steve Giambrone Daniel J. Povinelli, Jesse M. Bering. *Pointing*, chapter 3, Chimpanzees' "Pointing": Another Error of the Argument by Analogy?, pages 35–68. Psychology Press, jun 2003.

[GKD+19] Sebastian Gunther, Reinhard Koutny, Naina Dhingra, Markus Funk, Christian Hirt, Klaus Miesenberger, Max Mühlhäuser, and Andreas Kunz. MAPVI. In *Proceedings of the 12th ACM International Conference on PErvasive Technologies Related to Assistive Environments*. ACM, jun 2019.

[GM07] Susan Goldin-Meadow. Pointing sets the stage for learning language—and creating language. *Child Development*, 78(3):741–745, 2007.

[hec] hecomi. Github Website: `https://github.com/hecomi/uWindowCapture`. Accessed 2022-02-07.

[HS88] Sandra G. Hart and Lowell E. Staveland. Development of NASA TLX (task load index): Results of empirical and theoretical research. In *Advances in Psychology*, pages 139–183. Elsevier, 1988.

[Ken97] Adam Kendon. Gesture. *Annual Review of Anthropology*, 26:109–128, 1997.

[KGD+20] Reinhard Koutny, Sebastian Günther, Naina Dhingra, Andreas Kunz, Klaus Miesenberger, and Max Mühlhäuser. Accessible Multimodal Tool Support for Brainstorming Meetings. In *Lecture Notes in Computer Science*, pages 11–20. Springer International Publishing, 2020.

[KJP+20] Seungwon Kim, Allison Jing, Hanhoon Park, Gun A. Lee, Weidong Huang, and Mark Billinghurst. Hand-in-air (HiA) and hand-on-target (HoT) style gesture cues for mixed reality collaboration. *IEEE Access*, 8:224145–224161, 2020.

[KLBL93] Robert S. Kennedy, Norman E. Lane, Kevin S. Berbaum, and Michael G. Lilienthal. Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The International Journal of Aviation Psychology*, 3(3):203–220, jul 1993.

[LDK21] Simon Liechti, Naina Dhingra, and Andreas Kunz. Detection and localisation of pointing, pairing and grouping gestures for brainstorming meeting applications. In *23rd International Conference on Human-Computer Interaction*, pages 22–29. Springer, 2021.

[MRS+20] Sven Mayer, Jens Reinhardt, Robin Schweigert, Brighten Jelke, Valentin Schwind, Katrin Wolf, and Niels Henze. Improving humans' ability to interpret deictic gestures in virtual reality. apr 2020.

[NS03] Kai Nickel and Rainer Stiefelhagen. Pointing gesture recognition based on 3d-tracking of face, hands and head orientation. 2003.

[OGGM13] SEYDA OEZCALISKAN, DEDRE GENTNER, and SUSAN GOLDIN-MEADOW. Do iconic gestures pave the way for children's early verbs? *Applied Psycholinguistics*, 35(6):1143–1162, feb 2013.

[Poi] Website: `https://www.dimensions.com/collection/people-pointing`. Accessed 2021-11-21.

[ROG78] WILLIAM T. ROGERS. THE CONTRIBUTION OF KINESIC ILLUSTRATORS TOWARD THE COMPREHENSION OF VERBAL BEHAVIOR WITHIN UTTERANCES. *Human Communication Research*, 5(1):54–62, sep 1978.

[Ste] Steam. Steam VR Website: `https://valvesoftware.github.io/steamvr_unity_plugin/`. Accessed 2022-02-07.

[swi19] Durchschnittliche Körpergrösse (in cm). Website: `https://www.bfs.admin.ch/bfs/de/home/statistiken/kataloge-datenbanken/tabellen.assetdetail.7586022.html`, 28.02.2019. Accessed 2022-02-06.

[Uni] Unity. Unity Website: `https://unity.com/`. Accessed 2022-02-06.

[Viv] HTC Vive. HTC Vive Website: `https://www.vive.com/us/product/vive-pro-full-kit/`. Accessed 2022-02-06.

[WBR02] Chadwick A. Wingrave, Doug A. Bowman, and Naren Ramakrishnan. Towards preferences in virtual environment interfaces, 2002.

[WK11] Marta Wnuczko and John M. Kennedy. Pivots for pointing: Visually-monitored pointing has higher arm elevations than pointing blindfolded. *Journal of Experimental Psychology: Human Perception and Performance*, 37(5):1485–1491, 2011.

[WNA] Website: `https://humansystems.arc.nasa.gov/groups/tlx/index.php`. Accessed

2021-11-21.