Improving Essential Interactions for Immersive Virtual Environments with Novel Hand Gesture Authoring Tools

Dissertation

Thesis approved by
the Department of Computer Science
University of Kaiserslautern-Landau
for the award of the Doctoral Degree
Doctor of Engineering (Dr.-Ing.)

to

Alexander Schäfer

Date of Defense	21.04.2023
Dean	Prof. Dr. Christoph Garth
Reviewer	Prof. Dr. Didier Stricker
	apl. Prof. Dr. Achim Ebert

DE-386

Abstract

Augmented (AR), Virtual (VR) and Mixed Reality (MR) are on their way into everyday life. The recent emergence of consumer-friendly hardware to access this technology has greatly benefited the community. Research and application examples for AR, VR and MR can be found in many fields, such as medicine, sports, the area of cultural heritage, teleworking, entertainment and gaming. Although this technology has been around for decades, immersive applications using this technology are still in their infancy. As manufacturers increase accessibility to these technologies by introducing consumer grade hardware with natural input modalities such as eye gaze or hand tracking, new opportunities but also problems and challenges arise. Researchers strive to develop and investigate new techniques for dynamic content creation or novel interaction techniques. It has yet to be found out which interactions can be made intuitively by users. A major issue is that the possibilities for easy prototyping and rapid testing of new interaction techniques are limited and largely unexplored.

In this thesis, different solutions are proposed to improve gesture-based interaction in immersive environments by introducing gesture authoring tools and developing novel applications. Specifically, hand gestures should be made more accessible to people outside this specialised domain. First, a survey which explores one of the largest and most promising application scenario for AR, VR and MR, namely remote collaboration is introduced. Based on the results of this survey, the thesis focuses on several important issues to consider when developing and creating applications. At its core, the thesis is about rapid prototyping based on panorama images and the use of hand gestures for interactions. Therefore, a technique to create immersive applications with panorama based virtual environments including hand gestures is introduced. A framework to rapidly design, prototype, implement, and create arbitrary one-handed gestures is presented. Based on a user study, the potential of the framework as well as efficacy and usability of hand gestures is investigated. Next, the potential of hand gestures for locomotion tasks in VR is investigated. Additionally, it is analysed how lay people can adapt to the use of hand tracking technology in this context. Lastly, the use of hand gestures for grasping virtual objects is explored and compared to state of the art techniques. Within this thesis, different input modalities and techniques are compared in terms of usability, effort, accuracy, task completion time, user rating, and naturalness.

Acknowledgements

Many people made it possible for me to write this thesis. I would like to start with Didier Stricker, who always motivated and encouraged me. I would especially like to thank him for the freedom of choice of topics and the relaxed, yet extremely effective management style, which allowed me to implement many ideas.

Next, I would like to thank Gerd Reis, who has always inspired and supported me. Above all, his blunt truths were always a welcome help. Through the many conversations that ultimately structured my research, I gained not only inspiration, but also a friend. Furthermore, I would like to thank Stephan Krauß and Torben Fetzer, who always had an open ear and helped me with sound advice. A lot of people helped me along the way, especially the Augmented Vision group at the German Research Center for Artificial Intelligence, who always welcomed me in a friendly and respectful way.

Thanks also to my father, for his unconditional love and support in my young years who unfortunately had to pass away way too early. A special thanks to my mother, who despite my late father, gave me the opportunity to study and always continue my education. Without her strength, this thesis would never have been written. A special thanks to Thora, my partner and mother of my children Rhoda and Berend.

May 3, 2023, Alexander Schäfer

Contents

Chapter 1

Introduction

Immersive applications with Augmented (AR), Virtual (VR), and Mixed Reality (MR) are becoming increasingly popular. Systems can be built that help or entertain users in several ways: For example, a VR scenario to train firefighters to deal with extreme situations without danger. An AR remote collaboration scenario where a local worker streams his surroundings and gets augmented information from a remote expert. Users of a MR application integrate their physical environment into an immersive application, making it a promising approach for future ways of working. The possibilities that these technologies provide are nearly endless.

A technology that pairs well with AR, VR, and MR systems is panorama imaging (or sometimes called 360° imaging). With the recent rise of affordable hardware, researchers explore ways to transmit the environment using a video feed from panorama cameras attached to a Head Mounted Display (HMD). This allows remote users to see the surroundings of a local person. A big advantage compared to cameras with less field of view is that the remote user is not limited to the viewport of the local person, but can explore the whole environment if desired. Statically captured panorama images are also very interesting for VR and MR applications. With a simple button press, a whole scene can be captured in great detail. Using an HMD to view the image, users can be fully immersed and get a sense of being there. This type of technique is often used for cultural heritage to visit extraordinary places virtually but can also be employed to experiment with other use cases. One use case discussed in this thesis is remote collaboration using panorama images.

Remote collaboration is one of the most promising application scenarios, but also one of the most challenging. It involves many areas of expertise such as creating virtual environments, user embodiment, virtual object interaction, data synchronisation of multiple clients, and more. So far, most people use either audio or audiovisual channels to coordinate their work from afar. One of the biggest drawbacks is the lack of social presence this method provides. Transmitting non-verbal communication cues via video is not sufficient in many cases and might therefore be the greatest flaw of the current audiovisual collaboration methods. For example, mutual eye contact is not possible when two people far away from each other are using webcams but it is very important during a natural conversation. Furthermore, common webcams which are used for video conferencing are usually not able to capture the whole user but only parts (e.g. face). Applications using AR or VR technology can immerse users, render them as avatars, and transmit non-verbal communication cues by animating these avatars. Furthermore, feature rich environments can ensure that users are immersed and have a sense of being in the same room with other people. Virtual object interactions can break the chains of the physical world by also allowing unnatural, but nevertheless intuitive and above all meaningful actions. For example, one could project an infinite number of screens to a virtual office space and interact with objects using natural user input such as hand gestures.

What was once fiction, can now be (virtual) reality. Hand gestures are an important part to AR, VR, and MR technology. On the one hand it can be used to convey non-verbal communication cues to others, on the other hand it is an essential tool for natural user input. In recent systems, a controller is used as the baseline input modality. It has the advantage that it can be tracked well and pressing buttons has been proven to be stable. Providing reliable hand tracking has just recently been build into consumer grade hardware, allowing exploration of various interaction techniques with this technology. It has yet to be found out which hand gestures can be integrated and intuitively performed for certain interactions, just as certain gestures have become established on tablets or smartphones.

1.1. Motivation

One of the most important input modalities for modern Human Computer Interaction (HCI) are hand gestures. Modern Head Mounted Displays (HMDs) for AR, VR, or MR include built-in hand tracking already. However, manufacturers of these HMDs include only a small subset of possible gestures for developers which greatly limits the opportunities for novel interaction techniques. A controller is still the state of the art input modality for immersive applications using AR, VR, or MR technology. One reason for this is, that a controller is still more reliable and arguably more intuitive (for experienced users) than using bare hands. Hand tracking technology is steadily improving where hands can be reliably tracked even under challenging conditions. This opens new opportunities for researchers to test and evaluate novel interaction techniques using bare hands. Thus it is important to investigate natural user interaction with bare hand in order to find intuitive solutions for tasks in applications that use either AR, VR or MR technology. Finding the right (and intuitive) interaction is a task for everyone and should be supported by technical laypeople in particular. As an example, domain experts know what a specific system should be capable of and provide useful information on how the interaction in a system could take place. However, implementing these interactions is not possible for non-experts in the field. Attempts to make hand gesture-based interactions more accessible are limited.

1.2. Problem Statement

The goal of this thesis is to investigate new techniques for easy development of immersive applications with AR, VR, and MR technology. Usually, a wide range of expert skills is required to design and implement such an application. For example, 3D modelling to create high fidelity visuals or machine learning to implement novel gestures for gesture based interactions. These two disciplines are very distinct. Both depend on very different skills, one is more artistic, the other more mathematical. In terms of the tools or programming languages used, there is often no intersection. A major obstacle to scientific progress for immersive technologies is that many people with very different expertise have to work on problems for which they do not have sufficient knowledge. An example of this is a scientist working on intuitive interactions who does not necessarily know how to develop gestures and vice versa. Therefore, the creation of gesture-based systems should be made as easy as possible to ensure further progress towards reasonable application scenarios. The contribution of this thesis aims to provide a set of useful tools and solutions to make immersive applications more accessible to non-experts in this area. To gain insights into the state of the art in one of the most relevant use cases for AR, VR, and MR technology, an initial survey was conducted. With the help of the results of this initial survey, the following questions were formulated and addressed in this thesis:

1. How can realistic VR applications be created without expert knowledge?

2. How can hand gestures be made more accessible for the design of natural user interfaces without having expert knowledge in this field?

3. How can hand gestures be used to improve interactions for AR/VR/MR systems?

With regard to these questions, several papers have been published that address the answers to these questions. The proposed solutions were evaluated in a user-centred approach to investigate how they work in direct application scenarios.

1.3. Contributions

The contributions of this thesis include improvements in the areas of visualisation and gesture-based interaction for immersive applications. Furthermore, the advances allow for a holistic approach for rapid prototyping of immersive applications with AR, VR, and MR applications. Several peer-reviewed conference and journal articles have been published which are the foundation of the following chapters. These papers and their respective importance to the thesis are briefly described in this section.

A Survey on Synchronous Augmented, Virtual and Mixed Reality Remote Collaboration Systems *Alexander Schäfer, Gerd Reis, and Didier Stricker. ACM Computing Surveys (CSUR) (2021)* [228].

A survey was conducted in order to identify areas within AR, VR, and MR in the need of improvement. More precisely, the application scenario of remote collaboration was selected. This scenario was chosen since it involves many aspects that can be applied to almost the whole area of AR, VR, and MR. The survey identified the state of the art by covering a wide range of research and commercial applications. A taxonomy of remote collaboration systems is introduced, namely the three pillars of remote collaboration: *Environment, Avatars, and Interaction.* The key findings from the survey are also the cornerstones for subsequent work aimed to address some issues raised in the survey.

Towards Collaborative Photorealistic VR Meeting Rooms *Alexander Schäfer, Gerd Reis, and Didier Stricker. 2019. In Proceedings of Mensch und Computer 2019 (MuC'19)* [237].

One of the biggest challenges when creating immersive applications is the diversity of skills that is required. In order to create applications with high visual fidelity, a number of skills is required to create realistic 3D models. The paper introduced a novel technique to implement a photorealistic prototpye for VR without the need of expert knowledge. The chosen scenario was remote collaboration as it is highly relevant to the AR, VR, and MR community. The technique involves using several panorama images for a seamless VR experience from multiple viewpoints. Each user can switch between views while also interacting with augmented virtual objects using hand gesture-based interactions.

Investigating the Sense of Presence Between Handcrafted and Panorama Based Virtual Environments *Alexander Schäfer, Gerd Reis, and Didier Stricker. 2021. In Proceedings of Mensch und Computer 2021 (MuC '21)* [234].

Prototyping with panorama based images has advantages and can save time and money. But how does it compare to a high fidelity, carefully recreated 3D modelled environment? This paper investigated the sense of presence that users feel within both environments. For this purpose, several panorama images were used as a template to carefully model a 3D scene. Users had to solve a visible search task within both environments. The results showed that panorama based images can be a substitute to carefully modelled environments under certain conditions.

AnyGesture: Arbitrary One-Handed Gestures for Augmented, Virtual, and Mixed Reality Applications *Alexander Schäfer, Gerd Reis, and Didier Stricker. 2022. MDPI Applied Sciences 2022* [229].

Hand gesture-based interactions have been on the rise since manufacturers integrated hand tracking into HMDs. A small subset of possible gestures are usually included with their respective Software Development Kits (SDK). These SDKs however, are often tied to specific hardware and have limited amount of available gestures. Defining new gestures is not possible or complicated. Existing approaches require data sets and training, involving complex procedures to be able to recognise new gestures. Furthermore, most of the existing literature covers the creation of simple static gestures and do not include dynamic gestures. This work provides simple and yet effective solutions to rapidly design, prototype, and implement arbitrary one-handed gestures. Users of the framework can create new static or dynamic gestures by simply pressing a button without the need of expert knowledge.

Controlling Teleportation-Based Locomotion in Virtual Reality with Hand Gestures: A Comparative Evaluation of Two-Handed and One-Handed Techniques *Alexander Schäfer, Gerd Reis, and Didier Stricker. 2021. MDPI Electronics 2021* [233].

This paper explored how hand gestures can be used for one of the most important tasks in VR, namely locomotion. Four different techniques were implemented and compared to each other. User preference as well as quantitative measures such as task completion time, number of teleportations, and number of hand tracking failures were considered. It was especially investigated how one-handed gestures perform compared to two-handed techniques. The results concluded that all proposed techniques can be used well for locomotion in VR.

Controlling Continuous Locomotion in VR With Bare Hands *Alexander Schäfer, Gerd Reis, and Didier Stricker. 2022. EuroXR 2022* [232].

This paper investigated how hand gestures can be used to control continuous locomotion in VR. Different hand-gesture based techniques were compared to each other in terms of user preference, usability, and task completion time.

Learning Effect of Lay people in Gesture-Based Locomotion in Virtual Reality. *Alexander Schäfer, Gerd Reis, and Didier Stricker. 2022. International Conference on Human Computer Interaction, HCII 2022* [235].

Natural user interfaces such as hand gestures are not yet widely used and are not the standard for applications. Previous results indicate that hand gestures are a viable choice to control locomotion in VR. However, non-technical people have never used hand tracking systems before, can they adapt to this technology? This is the question that this paper investigated. After the subsequent use of hand gestures for locomotion, the quantitative metrics were compared. The results showed a significant improvement over lay people who use a technique for the second time.

Comparing Controller with the Hand Gestures Pinch and Grab for Picking Up and Placing Virtual Objects *Alexander Schäfer, Gerd Reis, and Didier Stricker. 2022. IEEE Conference on Virtual Reality and 3D User Interfaces* [230].

Another essential task in immersive applications is interaction with virtual objects. This paper investigated how hand gestures compare to controller when grabbing and placing objects. The pinch gesture is currently the state of the art for bare handed interaction with objects since it is the standard gesture included by manufacturers of AR, VR, and MR HMDs. Pinching is simple and easy to use, but has several disadvantages. For example, it is an unnatural gesture to grasp objects or other gestures that should be performed with the index finger and thumb do not work properly. Therefore, another grab gesture using the hand gesture prototyping framework AnyGesture was introduced. The three techniques were compared within a user study.

The Gesture Authoring Space: Authoring Customised Hand Gestures for Grasping Virtual Objects in Immersive Virtual Environments *Alexander Schäfer, Gerd Reis, and Didier Stricker. 2022. In Proceedings of Mensch und Computer 2022 (MuC'22)* [236].

Previous studies have shown that people try to grab virtual objects as they would in real life when asked to do so. However, this only works if it has been implemented by the developers, which is often not the case. Therefore, this paper introduces the Gesture Authoring Space, a tool built on AnyGesture to create custom tailored hand gestures for grasping virtual objects. The authoring process uses a two-step mechanism to capture gestures: First, the user wraps a hand around the object as if grabbing it. Second, the user has to hold the hand still in order to capture the desired hand pose for grasping the object. A user evaluation compared the three grasping techniques Pinch, Controller, and the proposed custom tailored hand gestures. The results of the study suggest that gestures created with the proposed approach are perceived by users as a more natural input modality than the others.

1.4. Outline

This section provides a brief overview of the organisation of the thesis.

Chapter 2 Contains background information on main topics of the thesis. A basic introduction to applications in AR, VR, and MR is provided. Furthermore, it is explained how these systems are understood and differentiated in this thesis. Additionally, the common input modalities for immersive applications such as controller, hand tracking, and eye gaze are briefly described.

Chapter 3 This chapter introduces remote collaboration as an important scenario for immersive AR, VR, and MR based applications. Remote collaboration covers a wide range of topics, such as immersive environments, user representation, gesture-based interaction techniques, transmission of non-verbal communication, and more. This makes it the ideal choice for an initial overview of the subject area to identify problems that need further research. Based on the results of this survey, the subsequent chapters cover advances to the state of the art of the identified issues.

> *The content of this chapter is based on and partly adopted from work previously published in the following publication:*
>
> **A Survey on Synchronous Augmented, Virtual and Mixed Reality Remote Collaboration Systems** *Alexander Schäfer, Gerd Reis, and Didier Stricker. 2022. ACM Comput. Surv. Just Accepted (April 2022).* [228].

Chapter 4 In this chapter, an exemplary prototype of a VR remote collaboration application using panorama images is introduced. Multiple panorama images are used to provide a seamless experience for multiple users. A first introduction to hand based interactions is given. A user study comparing the panorama-based environment with a 3D modelled one rounds off the chapter.

> *The content of this chapter is based on and partly adopted from work previously published in the following publications:*
>
> **Towards Collaborative Photorealistic VR Meeting Rooms** *Alexander Schäfer, Gerd Reis, and Didier Stricker. 2019. In Proceedings of Mensch und Computer 2019 (MuC'19). Association for Computing Machinery, New York, NY, USA, 599–603.d* [237]
>
> **Investigating the Sense of Presence Between Handcrafted and Panorama Based Virtual Environments** *Alexander Schäfer, Gerd Reis, and Didier Stricker. 2021. In Mensch und Computer 2021 (MuC '21). Association for Computing Machinery, New York, NY, USA, 402–405.* [234]

Chapter 5 Focuses on hand gesture-based interactions. The implementation of a gesture framework which allows the design and implementation of arbitrary one-handed gestures is provided. The focus of the framework lies on a solution for rapid prototyping of any one-handed gesture. The proposed solution does not require data sets, training, or expert knowledge to design and implement new gestures. Gestures can be captured with a simple button press and directly assigned to actions within an immersive environment to allow rapid prototyping of gesture to interaction mapping.

> *The content of this chapter is based on and partly adopted from work previously published in the following publication:*
>
> **AnyGesture: Arbitrary One-Handed Gestures for Augmented, Virtual, and Mixed Reality Applications** *Alexander Schäfer, Gerd Reis, and Didier Stricker. 2022. MDPI Applied Sciences 2022, 12, 1888.* [229]

Chapter 6 This chapter proposes several novel techniques for locomotion in VR using hand gestures. The proposed techniques were evaluated in a user-centred manner. The ability of users to adapt to the proposed hand-based interaction is also explored.

> *The content of this chapter is based on and partly adopted from work previously published in the following publications:*
>
> **Controlling Teleportation-Based Locomotion in Virtual Reality with Hand Gestures: A Comparative Evaluation of Two-Handed and One-Handed Techniques** *Alexander Schäfer, Gerd Reis, and Didier Stricker. 2021. MDPI Electronics 2021, 10, 715.* [233]
>
> **Learning Effect of Lay people in Gesture-Based Locomotion in Virtual Reality.** *Alexander Schäfer, Gerd Reis, and Didier Stricker. Learning Effect of Lay People in Gesture-Based Locomotion in Virtual Reality. In: Chen, J.Y.C., Fragomeni, G. (eds) Virtual, Augmented and Mixed Reality: Design and Development. HCII 2022. Lecture Notes in Computer Science, vol 13317.* [235]
>
> **Controlling Continuous Locomotion in VR With Bare Hands.** *Alexander Schäfer, Gerd Reis, and Didier Stricker. EuroXR 2022. Lecture Notes in Computer Science, vol 13484. Springer, Cham.* [232]

Chapter 7 To further harness the potential of hand gesture-based interactions, an introduction to grasping virtual objects with bare hands is given. The gesture authoring tool *Gesture Authoring Space* is introduced. This tool allows users to create their own hand gestures for grasping virtual objects. These custom gestures are compared to the pinch gesture and controller for picking up virtual objects. The comparison is based on user preference, usability, and naturalness.

> **Comparing Controller with the Hand Gestures Pinch and Grab for Picking Up and Placing Virtual Objects** *Alexander Schäfer, Gerd Reis, and Didier Stricker. 2022. IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW).* [230]
>
> **The Gesture Authoring Space Authoring Customised Hand Gestures for Grasping Virtual Objects in Immersive Virtual Environments** *Alexander Schäfer, Gerd Reis, and Didier Stricker 2022 In Proceedings of Mensch und Computer 2022 (MuC'22). Association for Computing Machinery, New York, NY, USA, 1-11.* [236]

Chapter 8 A conclusion is given in the final chapter. The tasks of the thesis are summarised as well as its contributions with an outlook on future work.

Chapter 2

Background

This chapter provides an overview of the background knowledge required to facilitate understanding of the content presented in the subsequent chapters. These sub sections focus on the description of immersive applications and their delimitation, an introduction to panorama images for VR, and input modalities for immersive applications.

2.1. Immersive Applications

Applications that use AR, VR and MR technologies give users the feeling of being physically present in a non-physical world. These applications generate a "sense of being there" which is usually described as the sense of presence. This reflects the degree to which an individual feels present in a virtual environment. Immersion is tightly coupled with sense of presence and can be described as the accumulated properties within an environment which is able to create a sense of presence. Different technologies as well as hardware and software generate different degrees of immersion. This section briefly describes how AR, VR and MR is achieved and which application scenarios are important for research and industry.

2.1.1. Definition of AR, VR and MR systems

This section explains how AR, VR and MR systems are defined in this thesis. AR can be achieved with two approaches: Video see-through and optical see-through. In both cases, the real world is augmented to the user. In video see-through the world is captured with a camera and virtual objects are placed onto the captured images. In optical see-through systems, users perceive the outside world with their own eyes through a transparent projection surface which displays the AR content (Further explanation in section 2.1.2). Regarding VR, most literature couples the term VR with a Head Mounted Display (HMD) which is placed on the head of a user. In this thesis systems without HMD's are also considered as VR, independent of the specific display device, as long as it is possible to immerse users into a virtual 3D environment. While

AR and VR systems are often quite clear in their separation, the distinction between MR systems often leads to confusion. Because of an inaccurate and often contradictory definition, Speicher *et al.* [263] has written a whole paper on "What is Mixed Reality?". Based on the conclusions of this work, it can be said that there is no single definition of MR. Therefore, a clear description of how MR is to be understood in this thesis is provided. Milgram et al.'s [165] definition of the Reality-Virtuality continuum describes MR as the area where both, Augmented Reality and Augmented Virtuality (VR content augmented with the real world such as a live video feed) are encapsulated (see Figure 2.1). MR is therefore described as everything between the real world and a completely virtual world and excludes the two extremes. However, with the recent emergence of applications that identify themselves as MR, this continuum is no longer strictly applicable. This conclusion comes from the fact that there are systems where a user who is in pure virtual reality can communicate with people in the real world. For example, a hybrid system where a VR user is in a collaborative environment with an AR user. A shared coordinate system provides the same content of the environment but what is seen is clearly different for both (the VR user only sees an artificially created world). Derived from many systems and publications that refer to themselves as MR, the following is considered to be a MR system in this thesis:

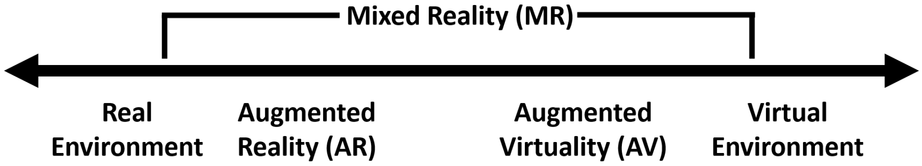

Figure 2.1.: *The Reality-Virtuality continuum according to Milgram et al. [165].*

1. There is a mix of AR and VR hardware (this also includes projector based systems).
2. Real world objects are used for interaction with either AR or VR hardware.

In addition, AR, VR and MR systems are collectively referred to as Extended Reality (XR) in the rest of the thesis such as described by Memmesheimer and Ebert [157]. XR can therefore be seen as a superset that includes the whole Reality-Virtuality continuum from Milgram *et al.* [165]. Devices which enable XR such as VR or AR HMDs and hand held devices are therefore referred to as XR devices.

2.1.2. Augmented Reality

Augmented Reality is a technique to augment a users natural view with virtual objects. For example, showing the price of a product by just looking at it or guiding a mechanic to the next construction step with virtual arrows. AR is usually achieved with one of the two approaches: optical or video see-through (See Figure 2.2). The main difference for an end user is the hardware which is used to display the augmented content. Optical see-through uses HMDs which can be put on. Prominent examples are Google Glass, Microsoft Hololens and the Magic Leap. These use reflective mirrors to project virtual content onto glasses which are worn by the user. By using a helmet or glasses-like object that the user has to put on, the augmented objects will be displayed as realistically as possible to the user and allow natural interactions.

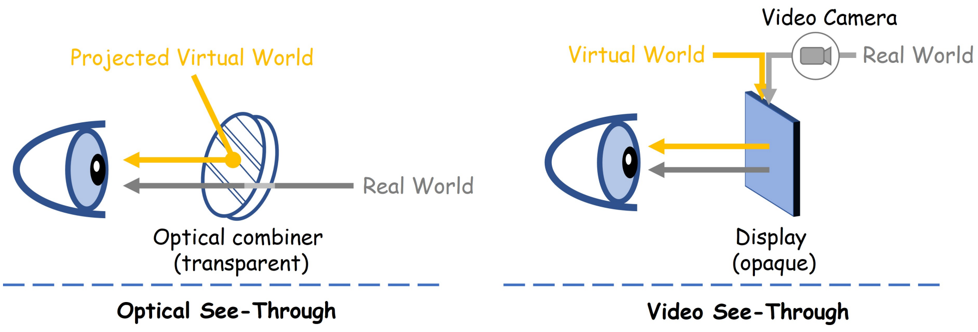

Figure 2.2.: *Two AR display techniques: Optical see-through where the user perceives the real world through his own eyes and video see-through where a camera feed is augmented.*

Video see-through uses commonly available devices such as smartphones or tablets to display the virtual content. Users are able to perceive the real world with augmented content through the screen of a device. A camera feed is displayed on a screen and virtual objects are augmented on top of the video stream. Recently, some HMDs built to fully immerse its users, use video streams to show camera output on a stereoscopic display.

Popular areas in which AR technology is applied are remote collaboration, cultural heritage, staff training, entertainment, and more. Remote collaboration uses mostly a scenario where a local person receives help and information from a remote expert. AR for cultural heritage can be used to augment exhibitions in museums. An example of video see-through AR, augmenting an old sewing machine is shown in Figure 2.3. In this case, AR is used to provide the user with additional information such as design plans or a handbook of the sewing machine. Additionally, a video how the sewing machine was used in the past can be overlaid. Examples for AR in the entertainment sector are the games Google Ingress and Pokemon Go. In general, users can walk around with their smartphones and watch their display for augmented content. In Pokemon Go, users have to go to different physical places in order to catch different fictional creatures. At its peak, the application had over 1 billion downloads and more than 232 million active users.

Figure 2.3.: *A user using video see-through AR with an old sewing machine. By looking at the sewing machine with a tablet, the user gets additional information such as design plans, handbook, or videos showing how it was used in the past.*

2.1.3. Virtual Reality

Unlike AR, Virtual Reality occludes the physical surroundings of a user. VR usually deals with a completely artificial world to immerse the user. Strictly speaking, all virtual worlds, regardless of the display medium, are in the VR category. However, this section describes how immersion is achieved when users wear an HMD.

In order to fully immerse a user, the HMD blocks the view to the natural surroundings and transmits movement and interaction intentions to the virtual environment. Today, two different approaches are generally used to transmit the movement of the user: Outside-in and Inside-out tracking. The Outside-in approach uses external cameras which are placed in the physical place of the user. HMD, controllers, and sensors attached to the user and physical objects can be tracked by those cameras. The position of each tracked object is then passed to the virtual environment. Unlike Outside-in tracking, Inside-out tracking does not use external cameras to track movement (See Figure 2.4). Instead, multiple cameras that are built into the HMD are used to understand the surroundings of a user and can estimate HMD and controller positions. These cameras can be used for motion tracking (especially hand tracking) as well.

Panorama images (often called 360 degree images) and recorded videos can also be displayed well with a VR HMD. These applications are especially important for the cultural heritage sector or industrial maintenance. Environ-

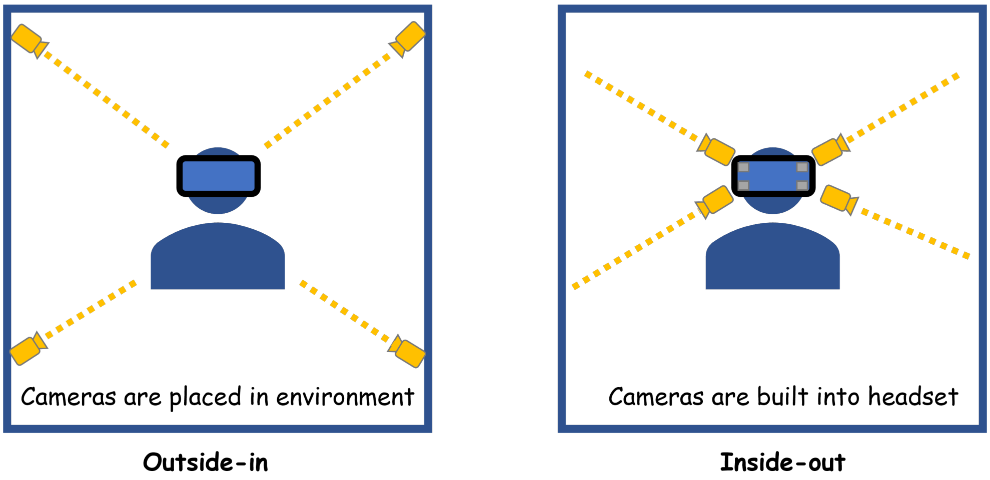

Figure 2.4.: *Two different types of tracking: Outside-in and Inside-out. With Outside-in, cameras are placed in the physical environment of a user. Inside-out uses built-in cameras of a HMD.*

ments can be captured with great detail and allow users to experience places that are far away and probably not accessible.

VR has a wide range of application scenarios. It can be used for training firefighters, the police, or the military to prepare for extreme situations. A high degree of immersion can prepare for realistic situations without real danger. Manufacturers can discuss and demonstrate products without having a real prototype. VR can be used for rehabilitation purposes or as medical treatment device (claustrophobia, fear of height, motion sickness and more) where users wear an HMD to do exercises in a playful environment. Museums can provide visitors with immersive and interactive scenarios, complementing their exhibitions. The application possibilities are endless.

2.1.4. Mixed and Extended Reality

The term Mixed Reality (MR) is often also mentioned with Extended Reality (XR). In this thesis, XR is used as an umbrella term including AR, VR, and MR. Applications which are described as MR usually integrate the natural/real world in a seamless experience. For example, a virtual ball that is responsive to objects in the real world. Another example would be using a stick-like object in the real world, which is visualised as a torch in the virtual world. The term MR is also often used for applications which use AR and VR together. A virtual world could be shown on a projector to users, while different users experience the same virtual world through an immersive HMD. With regular body movements, users in the real world can interact with the virtual world. The VR user can manipulate the virtual world as well. This scenario is depicted in Figure 2.5.

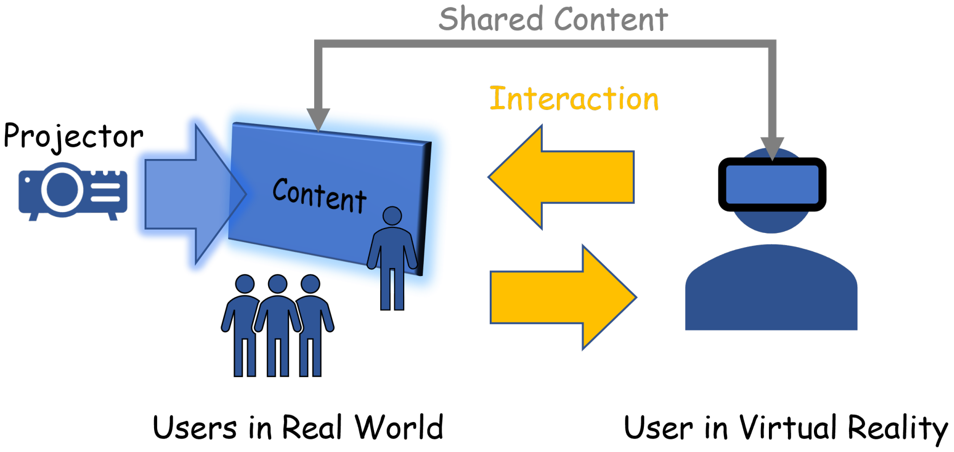

Figure 2.5.: *A Mixed Reality application scenario where one user is in VR and other users perceive the virtual world through a projector. Using different input modalities, users are able to interact with the same space.*

2.2. Panorama Imaging for Creating Immersive Environments

A fast and practical way to create immersive virtual environments for VR is panorama imaging. A panorama image is a picture taken from a camera that can capture a wide area of a scene. Usually, multiple pictures are taken which are then stitched together for a seamless experience. Figure 2.6 depicts two 180° cameras built into one device which will result in a 360° picture. Not only pictures but also videos can be recorded with such cameras. Such cameras are especially useful to capture a whole scene in great detail with minimal effort. There are many usage examples such as entertainment, cultural heritage, virtual tours, and more. These images can be viewed on a normal screen but using a VR HMD has great benefits. Viewing a panorama image through a VR HMD can create a sense of “being there”. Creating an immersive application with panorama images generally follows a few simple steps. A straightforward way to implement a panorama viewer and create the illusion of being on the scene consists of three steps (depicted in Figure 2.6):

1. Take a picture of the desired scene with a panorama camera.
2. Use the picture as texture on a 3D object (either sphere or cube).
3. Place the virtual camera of the user in the centre of the 3D object.

Taking a picture with a panorama camera will usually result in an equirectangular image. This is suited to texture a 3D sphere. Inverting the normals of this 3D sphere and placing the virtual camera (viewpoint of the user) in the centre of the sphere creates the illusion of being at the place the picture

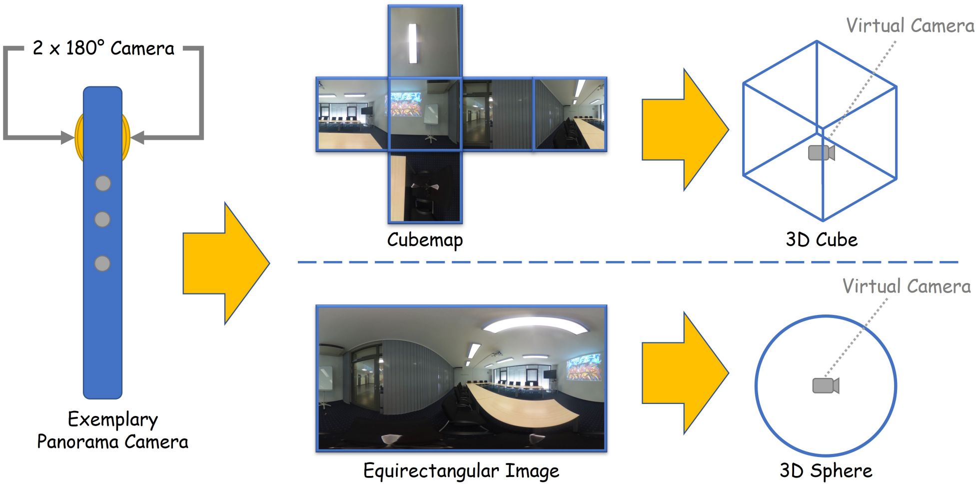

Figure 2.6.: *Simple workflow for creating an immersive 3D viewer using panorama images.*

was taken. It is also possible to unwrap the equirectangular image to a cubemap and repeat the same procedure with a 3D cube. A cubemap divides a panorama image into 6 regions which are placed on each side of the cube: top, bottom, left, right, front, and back. This can provide a performance boost since a cube requires less vertices than a sphere. From the user's point of view, both rendering types are hardly distinguishable.

2.3. User Input for Immersive Virtual Environments

This section briefly describes how current systems with XR technology connect humans with machines. Fundamentally, interaction and immersion is achieved by coupling the senses (e.g. vision, smelling) and the motor actions (e.g. body movement) of the human with the technology. Ideally, XR strives to process all sensory inputs and motor actions of a human being to provide the most immersive experience imaginable.

2.3.1. Controller

The current standard input method for commercially available VR systems is controller-based. This is achieved with a device that has sensors attached to it which can be tracked and whose position and rotation is transmitted to the virtual world. The tracking is performed the same way as the HMD is tracked (Outside-In or Inside-Out) as described in Section 2.1.3. Some of the buttons on these controllers are standardised but the majority of the controllers are tailored for specific HMDs. However, a VR controller has always a Trigger and a Grip button which should follow a similar implementation standard. While using a controller has its benefits, it is not optimal for social interaction. This is because within a virtual world, other users will only see an abstraction of the other users' hands. A lot of social information such as gesturing with the hands is therefore not transferred to the virtual world. Additionally, they are an auxiliary device (such as mouse/keyboard for PC) which does not enable natural interaction with the environment.

2.3.2. Speech

Today's systems increasingly support voice commands as input. Whether for accessibility or simple convenience, this type of input offers advantages that other input modalities cannot: It enables touch-free and hands-free operation of devices and systems and does not require any physical movement. The history of speech recognition goes back to the year 1952 where Bell laboratories designed the automatic digit recogniser Audrey which could recognise a single voice speaking digits aloud. In the early 1960s, IBM created a system called Shoebox which could calculate numbers via voice commands [1]. In the science fiction series Star Trek (1966), voice recognition was introduced for the common people, whereby the captain and crew could give voice commands to the ship's computer. Before graphical user interfaces became popular, it was thought that voice recognition would be the main input modality of the next decades. In today's world, voice systems are used almost everywhere. For example in smartphones, cars, desktop computers, smart homes and especially in assistance systems.

Voice commands do not necessarily have to consist of words. Blowclick [321], [325], for example, uses blowing into a microphone as the input method for clicking. A compilation of work regarding voice input is given by Monteiro *et al.* [170] where it is stated that the most usage is found for system control, symbolic input, selection, manipulation, and creation. XR devices usually have a built-in microphone and use speech in combination with other input modalities such as controllers or hand tracking. Although speech recognition gets increasingly robust, it is still used in combination with other modalities and systems exclusively relying on voice without other options are rare.

2.3.3. Eye Gaze

There are many application scenarios where eye gaze is used in XR applications. Accurate eye gaze can be retrieved by cameras built into HMDs. These cameras are usually located around the displays (for VR HMDs) or the optical combiner (for AR HMDs). A straightforward usage for eye gaze would be selecting/interacting with virtual objects that are looked at. According to the survey of Monteiro *et al.* [170], eye gaze is mostly used for system control and selection tasks. Other ways to use eye tracking can also be found in the literature. For example, Nguyen and Kunz [176] exploited eye blink for a more seamless redirected walking in VR experience. It can be used to create heatmaps of the users viewing activity, which can be used to examine where the user looks most. This is useful for conversation analysis or product placement within XR applications. It can also be used to animate eyes of an

---

[1] `https://www.ibm.com/ibm/history/exhibits/specialprod1/specialprod1_7.html`
Last accessed 31.08.2022

avatar to further increase immersion in multi user collaborative environments. Furthermore, eye tracking enables foveated rendering such as introduced by Patney *et al.* [193]. It is a rendering technique to essentially increase visual quality while reducing the required processing power. It is achieved by rendering less detail outside the eye fixation region. Since HMDs integrate eye gaze as standard, it is to be expected that future XR systems will use eye gaze in one form or another.

2.3.4. Hand Tracking

Hands are the main input mechanisms to influence the world around us. Hand tracking allows users to interact with a virtual environment with their bare hands. Sensors capture pose data of a users hands. Software uses this data to render the hands and to allow various interactions. To achieve hand tracking, many different techniques can be used. There are invasive techniques such as motion capture gloves and wristbands or non invasive techniques such as an optical motion capture system to acquire finger movements. The history of hand gesture recognition starts with glove-based interfaces. One of the first glove prototypes was the Digital Entry Data Glove from 1977 as described in the work of Dipietro *et al.* [53]. It used flexible tubes with a light source to detect finger bending. The light amount through a tube was measured and decreased by bending a finger (i.e. also bending the tube), resulting in values that could be interpreted and a 3D hand could be rendered accordingly. Since then, far more flexible, versatile, and wireless gloves are available. For more information regarding this technique, the reader is referred to Dipietro *et al.* [53] and Premaratne [204] to learn more about hand tracking achieved with gloves. Vision based systems use RGB, RGB-Depth, or infrared cameras to extract features from a video stream. In essence, all techniques try to estimate hand joint positions as accurately as possible by observing hands with cameras. Hand tracking in this thesis is achieved by using vision based tracking systems and the hardware is described in each Apparatus section of the respective experiment.

Hand gestures are an important part to human body language. Important examples are pointing at objects or emphasise at certain elements during talking. Kelly *et al.* [118] and Holler *et al.* [97] found out that gestures make people pay more attention to the speaker and essentially making them understand more if hand gestures are used. They can also give clues to the emotional state of a person. Sign language using hands is also the most common way for auditory impaired people to communicate in everyday life. It should be apparent at this point that hand gestures are very important for communication. In this thesis, hand gestures are mainly investigated with regard to their suitability for interaction within immersive virtual environments. Regarding hand gestures and interaction, advantages such as contactless and more natural and easier input are expected.

2.3.5. Summary

From the thoughts and examples written above, it can be deduced that there is no best input modality for XR systems. However, there are many advantages and disadvantages of different input modalities. One example would be that speech ensures operability without physical movement, but is still relatively inaccurate in recognition. The input modalities of this thesis are primarily confined to hands and controllers. The reader of this thesis will learn why the controller is usually more accurate and faster, but also why hands are more intuitive to use. Many of today's interactions in the field of immersive technologies are insufficiently researched in the field of natural user input. While a variety of different hand gesture recognition techniques have been published, there are almost no tools that can easily record and recognise a hand gesture that one has in mind. Various gesture recognition techniques are still being investigated, but their practical use in relation to immersive environments is almost unexplored. An example would be the movement within a virtual world. It is an essential interaction for VR applications, but there is currently no standard solution for moving around without a controller. Even picking up objects naturally is currently only possible if there are carefully created gestures for certain objects. This work fills a gap in the area by making gesture input more accessible and easier to integrate, while comparing it to other modalities in a user-centred way.

Chapter 3

Remote Collaboration using XR technology

As mentioned earlier, XR technology has a wide range of application scenarios. This chapter deals with one of the most promising and at the same time most versatile scenario, namely remote collaboration. This was chosen because of its multifaceted character and the many areas of possible interactions it covers. In particular, it was investigated how interaction takes place between multiple users and what types of interaction are important. Furthermore, remote collaboration systems have become increasingly important in today's society, especially during times where physical distancing is advised. Industry, research and individuals face the challenging task of collaborating and networking over long distances. While video and teleconferencing are already widespread, collaboration systems using XR are still a niche technology. An overview of recent developments of synchronous remote collaboration systems and a taxonomy by dividing them into three main components that form such systems is proposed in this chapter: *Environment*, *Avatars*, and *Interaction*. A thorough overview of existing systems is given, categorising their main contributions in different fields by providing concise information about specific topics such as avatars, virtual environment, visualisation styles and interaction. The focus is clearly on synchronised collaboration from a distance. A total of 87 unique systems for remote collaboration are discussed, including more than 100 publications and 25 commercial systems.

3.1. Introduction to Immersive Remote Collaboration

XR technologies are becoming more mature and open new ways for remote collaboration. Video and teleconferencing systems are already in extensive use in today's society, enabling a focus on more novel alternatives which utilise virtual, augmented and mixed reality technologies. Systems for remote collaboration in XR are developing slowly but steadily, yet they are not well established. An early example for remote collaboration using a virtual environment is Second Life. Almost two decades ago, the Second Life project [214], [218] allowed universities, corporations, cities, artists and individuals to create and share virtual spaces with many people [128]. Such systems consist of many different facets and requires expertise in various fields such as 3D modelling, animation, development of interactive systems, creation of avatars and dynamic content, multi-user communication and more. In addition, mixed reality systems are often implemented using a combination of AR and VR hardware which requires a certain expertise in both technologies.

In this chapter, a thorough summary and discussion of synchronous remote collaboration systems which utilise XR technology is given. The importance of such systems was emphasised during the COVID-19 outbreak in December 2019. People all over the world were put into quarantine, cities and local communities forbid travelling and even stepping outside. During this time, the scientific community was forced to find novel ways to network and communicate. Most scientific conferences where either cancelled or held completely virtual. Some events could be attended with VR HMDs as the organisers implemented immersive 3D experiences. The IEEE VR 2020 conference as an example, used virtual rooms where conference participants could join and interact with each other. Paper and poster presentations where done within a virtual environment which was streamed online for a broader audience. Although video and teleconferencing systems in particular experienced a significantly increased use as a result of this global crisis, the XR community received a major awareness push as well. A sophisticated XR system could help to reduce travel costs, office space, time, and carbon emissions by creating shared immersive spaces with believable person embodiment and interaction. To compete with each other in this crisis, many companies and researchers have recently invested in creating novel systems, which makes a recent review of existing systems and research even more interesting. Remote collaboration systems utilising XR technology are used in many different fields such as human computer interaction, computer graphics, medicine, training, cognitive sciences and many more.

Three components have been identified that any remote collaboration system needs to implement, namely *Environment*, *Avatars* and *Interaction*. A detailed explanation about this taxonomy is described in section 3.2.3. By providing condensed information on certain key topics, researchers can assess the state of the art in a particular subject. As an example, a researcher focused on novel environments for remote collaboration can use Table 3.1 which lists the discussed work with respect to their technology, use case, visualisation style and the stimulated sensory inputs. The representation of other users, in regard to their visualisation style is shown in Table 3.2. A researcher focused on interaction in multi-user collaborative environments will be interested in inspecting Table 3.3 which categorises important works in regard to interaction types which where found during the survey.

3.2. Related Work and Survey Procedure

Related work regarding the proposed literature review in this chapter was conducted by Phon *et al.* [195] which reviewed the state of the art in collaborative AR systems with focus on education in 2014. This work has a clear focus on collaborative learning in AR and does not differentiate between remote and local collaboration experiences. Another survey was done by Wang *et al.* [296] with focus on AR and MR based collaborative design in manufacturing. Ens *et al.* [60] review published work in collaboration through mixed reality up to the year 2018. Although the focus of mentioned work is not remote collaboration explicitly, the authors differentiate between remote and physically co-located systems. Another Survey was conducted by Belen *et al.* [8] in which the authors provide a systematic review of collaborative mixed reality technologies.

3.2.1. Inclusion and Exclusion of Systems

A concise focus on synchronous remote collaboration systems is provided and the results are categorised to assist scientists in different fields to cover their specific research interest. In this survey, the term **remote** is emphasised, which means that physical co-location of users is not required and the term **synchronous** which allows users to collaborate in real-time. The focus is on virtual, augmented and mixed reality systems. Traditional video and teleconferencing systems are omitted. A remote collaboration system is defined as a way of communicating, interacting and sharing a space beyond the boundaries of physical space exclusively through technological channels with distributed users. Work which uses or implements a combination of AR/VR/MR technology and synchronous remote collaboration is included. Systems which allow multiple users in a system but require users to be in physical co-location are excluded. Exceptions are systems which required physical co-location and could easily be extended for remote collaboration purposes. Asynchronous systems that do not allow real-time communication between users are also excluded.

3.2.2. Methodology

The survey was conducted through an iterative process by integrating the most relevant papers first, identifying specific similarities and differences with subsequent categorisation. By incrementally adding new relevant research work the categorisation process evolved and therefore separated the relevant work into three main contribution categories: *Environment*, *Interaction* and *Avatars*. With this approach it is possible to filter the relevant papers (over 2.000 unique papers) by applying the constraints mentioned in section 3.2.1 and then fitting the remaining papers into categories of the specific research interest for researchers.

An extensive search was performed by using search queries in different data sources. The used data sources include Scopus [1], Google Scholar [2], ACM Digital Library [3], IEEE Xplore [4], Springer Link [5] and PubMed [6]. Additionally the proceedings of multiple leading AR and VR conferences such as IEEE VR, ISMAR and EuroVR with focus on collaboration related topics were taken into account. The search was performed by concatenating AR/VR/MR with keywords such as remote, collaboration, social and more (as shown in Figure 3.1).

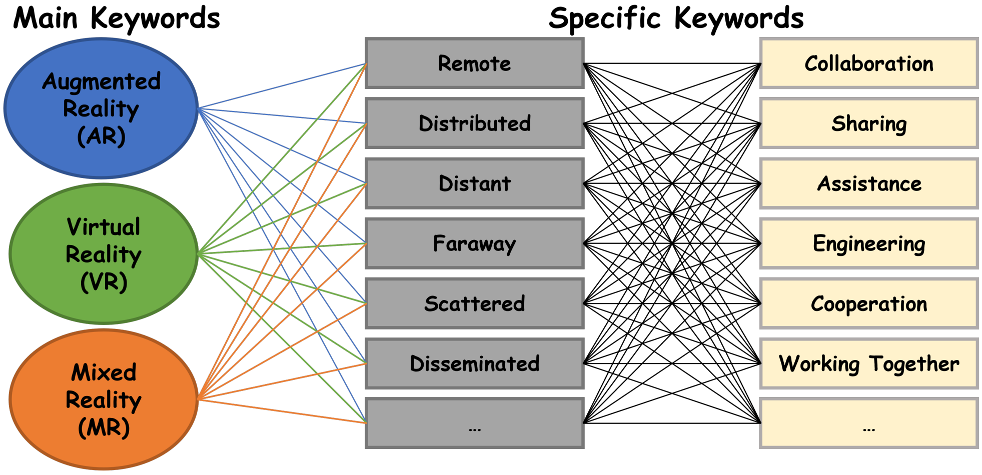

Figure 3.1.: *The used search methodology.*

The tables in the following sections summarise works from the same author if it is a continuation or extension of the previous work. Furthermore, the focus is on general implementation details of the proposed systems rather than on their specific research questions.

3.2.3. The Three Pillars for Remote Collaboration Systems

A taxonomy and categorisation of the relevant work was designed in a logical manner. Many systems created for remote collaboration have different aspects of novelty which cannot be described by simply assigning them to a specific category. For example, one system might excel in the novelty of avatars while another introduced a new kind of interaction technique for remote collaboration. One goal of this survey was to help researchers from a wide range of fields who are interested in the area of remote collaboration systems which utilise XR technology. To illustrate this: A researcher who is interested in the topic remote collaboration using XR might ask "How are users represented in virtual environments?", "What kind of interaction is possible in a shared

---

[1] `https://www.scopus.com` Last accessed 10.10.2022
[2] `https://scholar.google.com/` Last accessed 10.10.2022
[3] `https://dl.acm.org/` Last accessed 10.10.2022
[4] `https://ieeexplore.ieee.org/` Last accessed 10.10.2022
[5] `https://link.springer.com/` Last accessed 10.10.2022
[6] `https://pubmed.ncbi.nlm.nih.gov/` Last accessed 10.10.2022

virtual space?" or "Are there collaboration systems which enable shared gaze awareness?". With this survey, condensed information to different research questions such as virtual representation of users, different types of interaction and the virtual environment is provided. To achieve this, a concept was elaborated that enables the possibility to view each of these systems from different viewpoints: *Environment*, *Avatars* and *Interaction* (see Figure 3.2) which are called **the three pillars of remote collaboration systems**. In the next sections of this chapter, each of these components are explained in more detail while important and highly cited publications are presented in each category. Additionally, tables are provided for each category to allow quick access to the desired work: Table 3.1 is summarising remote collaboration systems with focus on virtual environments, Table 3.2 focuses on user representation and Table 3.3 identifies and categorises different interaction possibilities.

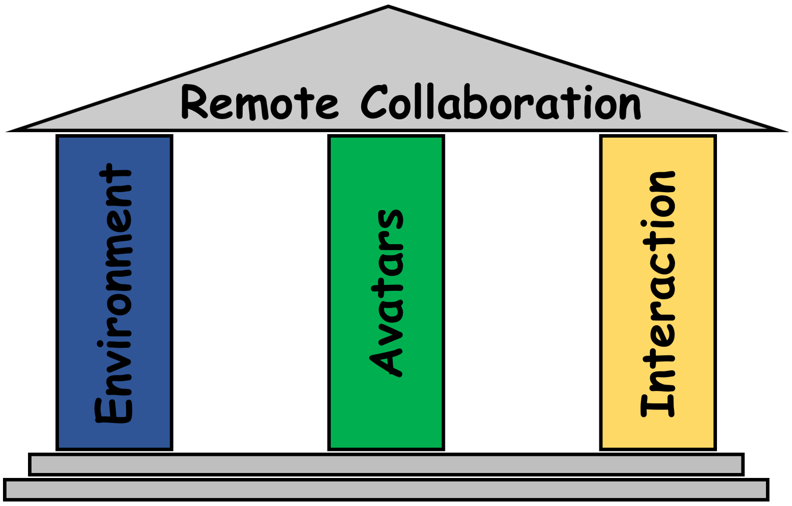

Figure 3.2.: *The three pillars for remote collaboration systems.*

3.3. Essential Components of Remote Collaboration Systems

3.3.1. Environment

Virtual environment refers to a simulated environment that stimulates the sensory impressions of a user. One of the first virtual environments was *Sensorama*, created by Morton Heilig in the early sixties [93]. It featured a simulated motorcycle ride with 3D visuals, stereo sound, olfactory cues (aromas) and tactile cues (seat vibration and wind from fans). In recent literature, most virtual environments are not as comprehensive and complete as the prototype created by Heilig, but rather focus on specific areas that are mostly visual or acoustic stimuli. Exceptions are augmented reality systems which utilise markers, where tangible interfaces with haptic feedback are still popular. As an example, Wang et al. used tangible interfaces such as a regular table [300] or tabletop [302], [303]. Other marker based systems used turntables, such as

Shen et al. [245], [247] or additional interaction tools such as a pen in [248]. With increasing maturity of AR technology, marker based systems became obsolete and such systems are not further developed.

In case of VR, there is usually a 3D modelled scene which is rendered, while in AR, the virtual environment refers to the augmented virtual objects superimposed onto the real world. Some AR systems do not include any 3D object rendering but use shared annotations and virtual pointers instead [74]–[76], [81]–[83], [145], [219], [220]. Sense of presence, often called telepresence, is highly affected by the quality and consistency of the virtual environment [313]. In early work, studies suggested that the overall sense of presence is increased by adding tactile and auditory cues [52]. The more sensory impressions are added, the greater the feeling of presence according to the studies of Dinh et al. [52]. In VR as example, telepresence is not only achieved by highly realistic 3D environments but also with consistency i.e. avatars should blend in with the environment and interaction methods should be adequate [313]. The work of Yoon et al. [313] compares different types of avatars with different styles. The authors' findings include that a stylised virtual environment should use avatars with similar visualisation style to achieve a higher sense of presence for the users. In Table 3.1 research works are presented and summarised by their respective virtual environment properties and ordered according to their respective technology (AR/VR/MR). Furthermore, remote collaboration systems are categorised into three main use cases, as these were the most common and consistent in the literature: *Meeting, Design* and *Remote Expert* . The category *Meeting* can also be seen as a means of sharing a workspace with other users. Some systems which are used for training or socialising fit also in this category. Note that the category *Event* has been added for VR-based systems, as there were three systems that could not otherwise be meaningfully categorised. These systems are used solely for event purposes. Bigscreen [106] focuses on virtual cinemas, allowing people to buy tickets and then watch movies together in a collaborative virtual environment. Sansar [48] and Wave [305] focus on virtual live events such as concerts.

Category Meeting

This category is for remote collaboration systems where users share a common workspace or environment for collaboration. These systems usually support media sharing, involve avatars to increase the sense of co-presence, and have interactive elements such as drawing on a whiteboard. In addition, use cases with knowledge transfer i.e. educational and learning scenarios are included in this category.

A focus on transferring and obtaining knowledge through augmented and virtual reality remote collaboration systems is shown by Monahan et al. [169], where a web-based VR system for managing and providing educational content was implemented. The system features an immersive 3D environment, allowing the lecturer to add media and virtual objects. Avatars are able to use gestures e.g. they can raise hands to indicate a question.

Chen et al. [42] created *ImmerTai*, a system which is designed for remote motion training. The participants are able to learn Chinese Taichi in an immersive collaborative environment. Student and a teacher are physically separated and resembled as a full body avatar in the virtual environment. This system includes a motion capturing module utilising a Microsoft Kinect, transferring the real world motion to their avatars. A motion assessment module is used to rate the movements of the student and give hints for improvement during and after a Taichi session.

Wang et al. [295] use a combination of camera, projector, VR HMD and hand tracker to create a remote collaboration system for knowledge transfer in a manufacturing scenario. A local worker assembles a water pump while a camera is recording and transmitting video footage of the worker's assembling progress to a remote expert. The remote expert views the video material through a VR HMD and transmits visual cues back to the local worker. A projector on the side of the local worker projects the hand movements of the remote expert onto his working surface.

In chapter 4 a technique using panorama images to create a shared photorealistic virtual environment in a meeting scenario is proposed. Users are able to hold virtual presentations with media sharing and hand gestures for interacting with the augmented virtual objects.

Weissker et al. [308] investigated group navigation in virtual immersive environments. The authors implemented a system which allowed users to navigate inside virtual environments together as a group or as individual. In their system, users can attach themselves to others and then organise teleportation movement through the virtual world together. Their results showed advantages in collaborative work when a switch between individual and group navigation is implemented.

Category Design

The category design combines remote collaboration systems for product design [245], [246], [248] or architectural design [45], [98], [100]. One of the earlier works was done by Lehner and DeFanti [137], who used a CAVE system in 1997. CAVE is a 10-foot by 10-foot by 9-foot surround screen which uses projections on the walls and floor. The authors implemented a system which enabled multiple users to share the same environment and discuss vehicle design remotely. The visual representation of other users was achieved by streaming 2D video inside the virtual environment.

Hsu et al. [100] developed an architectural design discussion system with interactive and immersive elements. It features voice communication, object manipulation, mid-air sketching and on-surface sketching. Overall, this tool was implemented to help architects to better understand architecture modelling and to discuss design decisions, even changing models during a remote collaboration session. The work of Chowdhury et al. [45] implements a collaboration system with an immersive virtual environment specifically created for urban design ideation and generation. The work is intended to be used by non-experts and concludes that even laypeople can take part in the design process of early stage urban design. While one user uses a VR HMD to view, interact and change virtual objects, other participants are able to perceive the changes on a display screen while giving feedback to the VR user. Hong et al. [98] utilises the multi-user virtual environment Second Life [214] for creative collaboration with focus on architectural design. They compared the effectiveness of collaborative architectural design between multi user virtual environments and a commercial architectural design software which allows 2D sketching and communication through audio. The authors argue that a remote collaboration system with avatars is more effective than two-dimensional approaches due to shared spatial informations. Ibayashi et al. [103] created a MR collaboration system which connects users on a tabletop device with a user wearing a VR HMD. The authors use a role based system with designers and occupants. Designers are able to view a 3D environment using a tabletop device. The environment can be changed with a touch interface provided to the designers. The occupant is immersed in this shared 3D environment with a VR HMD and is able to see the changes made by the designers in real-time. A see-through ceiling allows the occupant to see the designers by looking at the ceiling of the 3D environment while the designers are able to see the VR user moving around from the top-view.

A petroleum well planning application was developed by Nittala et al. [181], using hand held devices to augment the surroundings of a remote worker. A local user used a 3D printout, stylus and tablet as an interface to communicate with a remote worker who is on-site coordinating drilling operations. The 3D printout was combined with AR visualisations to provide the local user with an overview of earth's composition near the remote worker. The remote worker is able to see AR annotations made by the local user to plan drilling operations.

Category Remote Expert

Several systems for remote collaboration using a scenario with local and remote users were identified. In such systems, the local user typically executes a predefined task, being physically present at the target location while the remote user is generally far away and provides support with instructions or hints. In this type of collaboration scenario, the remote user is often called the remote expert. Many systems are based on a combination of AR and VR [73], [76], [135], [279], while the remote expert typically uses a VR HMD or 2D screen and the local user transmits his surroundings with the help of an AR HMD or an additionally mounted camera.

A mixed realiy collaboration system was developed by Piumsomboon *et al.* [196]. The system enables an AR user to share his local environment with a remote user. It provides collaborative, natural interaction with gaze and hand gesture data transmitted over a network to each user.

Another MR collaboration system was developed by Lee *et al.* [135]. The authors developed a system in which a host works with an AR HMD mounted with a 360° camera and a guest with a VR HMD. Nonverbal communication cues are transmitted via hand tracking and view awareness. Both users have visual feedback where the counterpart is currently looking at and are able to exchange hand gestures.

Teo *et al.* [279] introduced a MR remote collaboration system which combines reconstructed scenes obtained through an AR HMD with 360° panorama images. A remote user who receives footage from the AR user can move through the transmitted visual information, without relying on the local user to move. The system has been extended with the functionality that the remote user can trigger a 360° camera with the help of his VR controller to save spherical images that can be accessed independently [281]. The authors propose to add more functionality such as mid-air drawing in 3D to improve the usability of the prototype.

Mixed reality remote collaboration systems supporting local and remote users are especially useful in repairing tasks as Gauglitz *et al.* [76] suggest. In their work, a remote user is able to see the local user's current view and to annotate the view which is then visible in AR.

An interesting remote collaboration approach is *360Anwywher*e, a framework by Speicher *et al.* [262]. It allows ad-hoc remote collaboration in AR via 360° live input. Users are able to add digital annotations by drawing on a 360° video stream, either by means of a normal desktop application or mobile devices. The annotations made by remote participants are then visualised at the local physical space through a projector.

Gao *et al.* [73] implements a mixed reality collaboration system by mounting an RGB and RGB-Depth camera on top of a local user's VR HMD. The VR HMD is used as a video see through device while it captures and transmits its view to the remote counterpart. A RGB-Depth camera is used to obtain a point cloud which is streamed to the remote user. This point cloud is then stitched together, enabling an independent view control of the local workspace.

Bai *et al.* [7] developed a system which supports real-time 3D reconstruction by assembling eight RGB-Depth cameras into one sensor cluster. The remote VR user is able to see the local users surroundings through the transmission of the aligned pointclouds obtained through the RBG-Depth cameras at the local users space. The authors research focus is a shared virtual environment which supports gaze and gesture as visual cues in a remote expert scenario. Although the system supports one-way transmission of natural communication cues only, the results demonstrate advantages by providing natural gaze and gesture cues during collaboration.

Overall, it was found that the virtual environment of remote collaboration systems emphasises audio-visual stimuli. Some work did not even implement audio, focusing completely on visual feedback [197], [198], [219], [220]. While other work included some tactile feedback, these systems where mostly marker based AR systems and rely on physical markers attached to real objects [221], [249], [299], [302].

Table 3.1: Remote collaboration systems sorted by their respective technology.

System / Authors	Techn.	Use Case	Visualisatio Style	Sensory Inputs
Orts et al. [188]	AR	Meeting	Realistic	Audio, Visual
Regenbrecht et al. [213]	AR	Meeting	Realistic	Audio, Visual
Shen et al. [245]–[248]	AR	Design	Stylised	Audio, Visual, Tactile
Poppe et al. [201], [202]	AR	Design	Stylised	Audio, Visual
Sodhi et al. [259]	AR	Remote Expert	Stylised	Visual
Gurevich et al. [82], [83]	AR	Remote Expert	Annotations	Audio, Visual
Masai and Lee et al. [136], [150]	AR	Remote Expert	Stylised	Audio, Visual
Tait et al. [273]	AR	Remote Expert	Stylised	Audio, Visual
Kurata et al. [129]	AR	Remote Expert	Stylised	Audio, Visual
Ou et al. [189]	AR	Remote Expert	Stylised	Audio, Visual
Izadi et al. [109]	AR	Remote Expert	Stylised	Audio, Visual
Lukosch et al. [145]	AR	Remote Expert	Annotations	Audio, Visual
Gauglitz et al. [74]–[76]	AR	Remote Expert	Annotations	Audio, Visual
Gupta et al. [81]	AR	Remote Expert	Annotations	Audio, Visual
Zillner et al. [326]	AR	Remote Expert	Annotations	Visual
Utzig et al. [286]	AR	Remote Expert	Annotations	Audio, Visual
Zenati et al. [315]–[317]	AR	Remote Expert	Annotations	Audio, Visual

Continued on next page

Table 3.1: Remote collaboration systems sorted by their respective technology. (Continued)

System / Authors	**Techn.**	**Use Case**	**Visualisatic Style**	**Sensory Inputs**
Calandra et al. [33]	AR	Remote Expert	Annotations	Audio, Visual
Breakroom [252]	VR	Meeting	Stylised	Audio, Visual
EngageVR [200]	VR	Meeting	Realistic	Audio, Visual
Glue Collab [99]	VR	Meeting	Realistic	Audio, Visual
MeetInVR [156]	VR	Meeting	Realistic	Audio, Visual
Mozilla Hubs [171]	VR	Meeting	Stylised	Audio, Visual
Nvidia Holodeck [183]	VR	Meeting	Realistic	Audio, Visual
Stage VR [187]	VR	Meeting	Realistic	Audio, Visual
TechViz VR [276]	VR	Meeting	Realistic	Audio, Visual
TheWild [282]	VR	Meeting	Realistic	Audio, Visual
Vive Sync [271]	VR	Meeting	Stylised	Audio, Visual
WorldViz [312]	VR	Meeting	Stylised	Audio, Visual
Regenbrecht et al. [212]	VR	Meeting	Realistic	Audio, Visual
Gu et al. [80]	VR	Meeting	Realistic	Audio, Visual
Schäfer et al. [237]	VR	Meeting	Realistic	Audio, Visual
VRChat [38]	VR	Meeting	Stylised	Audio, Visual
NeosVR [161]	VR	Meeting	Stylised	Audio, Visual

Continued on next page

Table 3.1: Remote collaboration systems sorted by their respective technology. (Continued)

System / Authors	Techn.	Use Case	Visualisatic Style	Sensory Inputs
Acadicus [292]	VR	Meeting	Stylised	Audio, Visual
Rumii [251]	VR	Meeting	Realistic	Audio, Visual
VirBELA [291]	VR	Meeting	Realistic	Audio, Visual
Garou [107]	VR	Meeting	Realistic	Audio, Visual
MeetingRoom [155]	VR	Meeting	Stylised	Audio, Visual
Facebook Horizon [141]	VR	Meeting	Stylised	Audio, Visual
Second Life [214]	VR	Meeting	Realistic	Audio, Visual
Tan et al. [274]	VR	Meeting	Stylised	Audio, Visual
Weissker et al. [308]	VR	Meeting	Stylised	Audio, Visual
Calandra et al. [49]	VR	Meeting	Realistic	Audio, Visual
CollaboVR [92]	VR	Meeting	Realistic	Audio, Visual
IrisVR [108]	VR	Design	Stylised	Audio, Visual
Hsu et al. [100]	VR	Design	Realistic	Audio, Visual
Lehner et al. [137]	VR	Design	Realistic	Audio, Visual
BigScreen [106]	VR	Event	Realistic	Audio, Visual
Wave [305]	VR	Event	Realistic	Audio, Visual
Sansar [48]	VR	Event	Realistic	Audio, Visual

Continued on next page

Table 3.1: Remote collaboration systems sorted by their respective technology. (Continued)

System / Authors	**Techn.**	**Use Case**	**Visualisatic Style**	**Sensory Inputs**
Higuchi et al. [95]	MR	Meeting	Realistic	Audio, Visual
Speicher et al. [262]	MR	Meeting	Annotations	Audio, Visual
Ryskeldiev et al. [219], [220]	MR	Meeting	Annotations	Visual
Spatial [272]	MR	Meeting	Realistic	Audio, Visual
Regenbrecht et al. [210]	MR	Meeting	Realistic	Audio, Visual, Tactile
Haller et al. [85]	MR	Meeting	Realistic	Audio, Visual, Tactile
Norman et al. [182]	MR	Meeting	Annotations	Audio, Visual
Matthes et al. [153]	MR	Meeting	Realistic	Audio, Visual
Galambos et al. [67], [68]	MR	Meeting	Realistic	Audio, Visual
Bai et al. [7]	MR	Meeting	Stylised	Audio, Visual
Luxenburger et al. [146]	MR	Meeting	Realistic	Audio, Visual
vTime [294]	MR	Meeting	Realistic	Audio, Visual
PoseMMR [191]	MR	Meeting	Annotations	Audio, Visual
Gamelin et al. [70]	MR	Meeting	Realistic	Audio, Visual
Geollery [56], [57]	MR	Meeting	Stylised	Audio, Visual
Grønbæk et al. [79]	MR	Design	Realistic	Audio, Visual

Continued on next page

Table 3.1: Remote collaboration systems sorted by their respective technology. (Continued)

System / Authors	Techn.	Use Case	Visualisatic Style	Sensory Inputs
TeleAR [303]	MR	Design	Stylised	Audio, Visual, Tactile
Wang et al. [299]–[302]	MR	Design	Realistic	Audio, Visual, Tactile
Sakong et al. [221]	MR	Design	Realistic	Audio, Visual, Tactile
Sidharta et al. [249]	MR	Design	Realistic	Audio, Visual, Tactile
Ibayashi et al. [103]	MR	Design	Realistic	Audio, Visual, Tactile
Sasikumar et al. [224]	MR	Remote Expert	Stylised	Audio, Visual
Lee et al. [134], [135]	MR	Remote Expert	Realistic	Audio, Visual
Teo et al. [278]–[281]	MR	Remote Expert	Realistic	Audio, Visual
Kim et al. [121], [122]	MR	Remote Expert	Annotations	Audio, Visual
Rae et al. [205]	MR	Remote Expert	Realistic	Audio, Visual
Piumsomboon et al. [197], [198]	MR	Remote Expert	Stylised	Visual
Gao et al. [71], [72]	MR	Remote Expert	Realistic	Audio, Visual
Wang et al. [297]	MR	Remote Expert	Annotations	Audio, Visual
Elvezio et al. [58]	MR	Remote Expert	Realistic	Visual
Pouliquen-Lardy et al. [203]	MR	Remote Expert	Realistic	Audio, Visual

Continued on next page

Table 3.1: Remote collaboration systems sorted by their respective technology. (Continued)

System / Authors	Techn.	Use Case	Visualisatic Style	Sensory Inputs
Alem et al. [3]	MR	Remote Expert	Stylised	Audio, Visual
Higuch et al. [94]	MR	Remote Expert	Realistic	Audio, Visual
Chen et al. [40]	MR	Remote Expert	Annotations	Audio, Visual
Nittala et al. [181]	MR	Remote Expert	Annotations	Audio, Visual, Tactile
Sun et al. [266], [267]	MR	Remote Expert	Stylised	Audio, Visual
De Pace et al. [50]	MR	Remote Expert	Realistic	Audio, Visual

3.3.2. Avatars

Avatars represent entities in virtual environments. The most commonly used avatars in scientific literature as well as in commercial XR software are classified in categories and differentiated by descriptive terms (see also Table 3.2):

- **Realistic and Stylised Graphics**

  Avatars can be distinguished through visualisation style, i.e. how the 3D model of the avatar is rendered (stylised or realistic style). One of the reasons the rendering style is used as a descriptor is to help researches who are focused on the appearance of avatars. Additionally, there is existing work which addresses certain research questions concerned with avatar visualisation styles and collaboration. As an example, according to Yoon et al. [313] there is no statistical difference in regards to social presence with different visualisation styles but that user perception differs between stylised and realistic avatars. A stylised avatar allows for a more playful atmosphere, whereas realistic avatars tend to represent a professional environment.

- **Avatar Type** Avatar types are divided in subcategories: Full Body, Upper Body, Head & Hands and Hands only. Full body avatar refers to humanoid avatars where all limbs are attached to it (e.g. hands, arms, legs etc.). An Upper Body avatar consists of a head, hands and torso but no legs. The Head & Hands type of avatar is composed of a floating head combined with detached hands. Hands only means that a user is only represented by virtual hands.

- **Reconstructed Model Avatar** If a system is capable of creating an avatar that resembles the respective user it is categorised as a system that uses *Reconstructed Model* avatars. This category includes avatars that are created from face reconstruction in any form and excludes avatars that do not have a realistic face (e.g. reconstructed/personalised hands only is excluded). Non-reconstructed avatar means in general choosing from existing 3D models.

- **Video Avatar** Some systems implement avatars as video projections, similar to typical videoconferencing systems. Systems fall under this category if a user is seen as a video feed in an immersive virtual environment. Additionally, systems which use multiple cameras to reconstruct 3D video avatars are put into this category. As an example, the work of Matthes *et al.* [153] implemented such a system based on multiple depth cameras.

- **Audio Avatar** Although avatars are often represented as a humanoid 3D model, the term avatar is in general used for any kind of user representation in virtual worlds, even including invisible forms. In this work, *Audio Avatar* represents an entity in a system which enables communication with other users, regardless of visual appearance. Systems with this type do not rely on any visual form for users in remote collaboration systems. Users in such systems use audio for communicating with each other.

- **AR Annotations** Systems which use no 3D model for other users but have annotations instead are in this category. This differs from *Audio Avatar* in the sense that *Audio Avatar* uses audio communication only, whereas in *AR Annotations* the remote expert communicates with the local user with annotations, i.e. the other users presence is perceived through visual annotations. Unique avatars are usually not necessary in this scenario, because the roles are clearly separated and the users can distinguish each other by actions. In many systems using *AR Annotation* avatars, audio communication is not implemented. As an example, the work of Zillner *et al.* [326] uses visual annotations such as text, pictures and freehand drawings to give precise instructions to the local worker without relying on audio communication.

A typical avatar configuration for VR based systems consists of the head (representing the VR HMD) and hands (representing the controllers) and is usually the most minimalistic avatar in VR scenarios. An avatar consisting solely of virtual hands tends to be used in AR based systems which use hand tracking or gesture techniques [71], [259]. A majority of *Audio Avatars* can be found in AR systems (see Table 3.2). While VR systems seem to always rely on 3D model representation of other users, AR based approaches often omit visual representation when using remote expert scenarios. In such cases, the local and remote user communicate via audio with each other and share their view [74]–[76]. Furthermore, some systems rely mainly on visual annotations [82], [259], which are marked as *AR Annotations* in Table 3.2.

Piumsomboon *et al.* [199] developed a system with an adaptive avatar *Mini-Me* which uses redirected gaze and gestures to enhance remote collaboration with improved social presence. To assess the usefulness of the avatar, a scenario where a remote expert in VR assists a local worker in AR was used. The remote expert was shown to the AR user as a miniature avatar which was able to sucessfully transmit nonverbal communication cues according to the authors. Although focusing on the novelty of the proposed avatar, the system proved to be useful for overall remote collaboration.

Elvezio *et al.* [58] developed a system with virtual twins of physical objects. A remote expert uses virtual replicas of physically existing objects to guide a local user performing certain tasks with such objects. In this case, communication with both users only takes place by transmitting the pose of the mentioned objects. In the work of Luxenburger *et al.* [146] the communication between users takes place through media sharing. A user is filling out a report on a mobile device which is then visible to a remote user by means of a VR HMD. In the commercial VR remote collaboration system EngageVR [200], users can create their own full body avatar with reconstructed face, by uploading a single picture. Machine learning techniques in the backend of the system reconstruct a fully textured 3D mesh of the head and attaches it automatically to a predefined body models. Some other commercial systems are not as sophisticated and use stylised like avatars [99], [156], [252]. Other popular systems such as VRChat [38] or NeosVR [161] allow users to create and upload their own avatars. By means of an SDK, they can upload fully animated humanoid avatars regardless of their appearance. The seamless integration of these arbitrary avatars is achieved by applying a specific skeletal structure to the model.

Table 3.2: Remote Collaboration Systems classified in avatar categories.

Avatar Type	**References**
Stylised	[252] [99] [156] [171] [187] [271] [201], [202] [237] [161] [108] [292] [251] [217] [291] [107] [155] [106] [305] [141] [294] [281] [224] [38] [67], [68] [7] [308] [203] [3] [286]
Realistic	[200] [183] [272] [276] [188] [95] [48] [214] [274] [38] [161]
Full Body	[252] [200] [183] [187] [271] [38] [161] [48] [305] [294] [214] [274]
Head & Hands	[99] [156] [276] [312] [197] [237] [108] [292] [251] [155] [106] [281] [67], [68] [7] [203]

Continued on next page

Table 3.2: Remote Collaboration Systems classified in avatar categories. (Continued)

Avatar Type	**References**
Upper Body	[171] [272] [282] [201], [202] [217] [107] [141] [308]
Reconstructed Model	[200] [272] [276] [188]
Video	[210] [212] [80] [219], [220] [153] [137] [40] [103]
AR annotations	[259] [82], [83] [300] [81] [121] [297] [182] [326] [94] [286] [181] [315], [316]
Hands	[303] [259] [221] [109] [71] [134], [135] [224] [3] [122] [94] [103] [266], [267]
Audio Avatar	[245]–[248] [249] [74]–[76] [136], [150] [273] [129] [189] [145] [79] [191] [58] [146]

3.3.3. Interaction

In this section, common interactive elements in remote collaboration systems are identified which can be found in various works and literature. Table 3.3 provides an overview of literature and work which is categorised in multiple different interaction categories. It is to note that a category is not mutually exclusive to another, e.g. a system which uses media sharing might also use hand gestures. This section explains each interaction technique with a few examples. The common features which were found are the following:

1. Shared 3D Object Manipulation
2. Media Sharing
3. AR Annotations
4. 2D Drawing
5. AR Viewport Sharing
6. Mid-Air Drawing in 3D
7. Hand Gestures
8. Shared Gaze Awareness
9. Convey Facial Expression

Table 3.3 guides the reader to interesting and major publications which use the mentioned interaction techniques.

Shared 3D object Manipulation

The most commonly shared feature between remote collaboration systems is the possibility to interact and manipulate shared 3D objects in a virtual space. The type of interaction differs between systems, but the focus is on manipulating one or many 3D objects. AR technology is used by Shen *et al.* [246]–[248] where multiple users interact with 3D objects in a collaborative AR environment. A stylus with two markers attached is used as additional interaction tool which enables feature highlighting and 3D object manipulation. More recent work uses hand tracking/gestures to interact with objects [272]. Schlünsen *et al.* [238] compared free-hand-manipulation with widget-based manipulation techniques. Their study shows that free-hand interaction is preferred over widget-based interaction by users.

Media Sharing

Systems which are able to share documents, images, videos and other form of media are categorised here. Haller *et al.* [85] created a system with a tangible interface, a table with touchscreen for media sharing. It additionally featured sharing media from desktop applications to the tabletop. A web-based VR solution was developed by Monahan *et al.* [169] which implements media sharing such as videos and images in an educational context.

AR Annotations

One of the most commonly used tool for communication and interaction in AR based systems is annotations. Annotation types include 2D drawing, text, or simple pointers. The work of Speicher *et al.* [262] utilises a 360° camera to capture the surroundings of one user, while other users are able to draw and annotate on the input stream. The annotations and drawings are then visualised by a projector to the local user's physical space. Kurata *et al.* [129] present a wearable HMD which receives remotely annotated input in form of drawings. A special feature of this system is a laser pointer that enables the wearer of an HMD to draw the attention of remote users towards a certain object by pointing on it.

2D Drawing

Especially in remote collaboration systems with focus on replicating a virtual meeting scenario, drawing on surfaces is a widely used feature. In more sophisticated systems such as Glue Collaboration Platform [99], users are able to place a virtual whiteboard. This whiteboard can be re-positioned and resized allowing multiple users to draw with virtual pens in many sizes and colors. Mimicking real world objects, it is also possible to use an eraser.

AR viewport sharing

This category includes systems which implemented sharing a user's view perspective. Tait and Billinghurst [273] created a system which reconstructs a local user's environment by using depth sensors attached to an HMD. Reconstructing the environment from a local scene, remote users can move independently through the virtual environment. The local user is then represented as a frustum in the reconstructed scene, allowing the remote user to see where the local counterpart is looking at. The study of Tait and Billinghurst [273] suggests that implementing view independence between local and remote user improves task completion time. Sasikumar *et al.* [224] combined AR and VR users together and enabled view frustum sharing which is visible to the local user as a grey cuboid. The goal of their work was to convey nonverbal communication cues such as eye gaze and hand gestures.

Mid-Air Drawing in 3D

Mid-Air drawing allows users to create 3D paintings, which can then be observed from multiple users in different angles. This interaction method seems to be mostly available in commercial systems such as Glue Collab [99] or Hubs [171]. A user is able to draw in the air of the virtual world by utilising a VR controller as virtual pen. Other systems such as the one proposed by Zillner *et al.* [326] implement a remote expert scenario, where one user is streaming his surroundings with an RGB-D camera and a remote expert is observing and annotating for assistance. The remote expert is able to segment objects, to create animations, to draw on geometry and to place annotations which can be viewed by the AR user.

Hand Gestures

Systems which utilise hand gestures and convey hand movements through the remote collaborative space are included in this category. Sophistated systems such as Spatial [272] use AR HMD's to enable a full interaction with the 3D environment via a hand tracker. Tan *et al.* [274] implemented a VR telepresence system which allowed multiple users to interact with objects and watch videos together. The authors used motion capture gloves to animate arms, hands and fingers of a VR avatar. Kim *et al.* [122] implemented a MR collaboration system to evaluate combinations of visual communication cues using gestures. The authors found that certain combinations of communication cues such as hand visualisation together with finger pointing direction does not provide any significant benefit for remote collaboration.

Shared Gaze Awareness

Systems which allow users to share gaze awareness belong into this category. Systems that allow precise tracking and transmission of gaze awareness are included and systems that indicate gaze perception by head rotation only excluded. For example, the work of Galambos *et al.* [68] is not included in this category since the gaze directon of a user is only indicated by the direction the avatar is facing. As an example, Speicher *et al.* [262] created a system that allows the participants to show exactly which position they are looking at in a 360° video feed. The work of Poppe *et al.* [201], [202] uses avatars around a virtually augmented table and positions them according to the gaze information of the corresponding user. Billinghurst et al. developed Emapthy Glasses [136], [150] which is an HMD that enables streaming a live video feed with accurate gaze information. Norman *et al.* [182] implemented a system which gives direct visual feedback of other user's gaze behavior. Using a system that combines multiple AR HMD's with a desktop PC, participants are asked to place virtual furniture on a regular table in a collaborative manner.

Convey Facial Expression

This category addresses work which is able to transmit facial expressions to other users. Systems which use video transmission of other user's faces are not included. Lee *et al.* [136] and Masai *et al.* [150] use Empathy Glasses to transmit facial expressions bidirectionally. A local user's face is analysed by the built in modules of the glasses, while a remote user's expression is tracked via webcam. The authors of [274] integrated lip syncing into their VR remote telepresence system in order to enable a more immersive communication experience.

Table 3.3: Common interaction types in remote collaboration systems.

Interactive Feature	References
Shared 3D Object Manipulation	[252] [200] [99] [156] [171] [183] [272] [187] [276] [282] [271] [312] [210] [212] [85] [303] [300] [221] [249] [248] [247] [213] [188] [38] [161] [108] [292] [251] [107] [155] [106] [294] [214] [191] [182] [274] [67], [68] [58] [137] [203] [103]
Media Sharing	[252] [200] [99] [156] [171] [183][272] [187] [276] [282] [271] [312] [210] [212] [85] [303] [109] [79] [262] [237] [38] [161] [251] [291] [107] [155] [106] [305] [294] [214] [274] [67], [68] [146]
2D Drawing	[200] [99] [156] [171] [272] [187] [312] [210] [85] [80] [303] [189] [109] [79] [262] [95] [161] [251] [155] [67], [68]
Mid-Air Drawing in 3D	[99] [156] [171] [272] [187] [312] [161] [108] [326]

Continued on next page

Table 3.3: Common interaction types in remote collaboration systems. (Continued)

Interactive Feature	References
Shared Gaze Awareness	[201], [202] [303] [150] [81] [197], [198] [262] [95] [297] [182] [7] [94] [286]
Convey Facial Expression	[150] [136] [161] [274]
Hand Gestures	[272] [259] [201] [303] [300] [221] [109] [145] [197], [198] [71], [72] [134], [135] [278] [237] [188] [161] [224] [274] [7] [3] [122] [94] [103] [266], [267]
AR annotations	[272] [259] [201] [202] [85] [303] [249] [246]–[248] [136], [150] [273] [129] [189] [145] [213] [262] [219] [134], [135] [278] [121], [122] [224] [3] [326] [94] [40] [181] [315], [316]
AR viewport sharing	[273] [129] [197], [198] [71], [72] [262] [219], [220] [134], [135] [278], [281] [121] [224] [181] [266], [267] [315], [316]

3.4. Professional Meetings Through XR Remote Collaboration

Currently, there are already many tools and collaboration systems available which utilise virtual or augmented reality. In this section, commercial and professional systems which support more than 10 users simultaneously are considered [99], [156], [171], [183], [187], [200], [252], [272], [276], [282], [312]. During the COVID 19 outbreak, one of the largest scientific conferences for virtual reality research took place entirely online and virtually in early 2020. To this date, it was the first major conference held completely virtually. During this time, the organising committee was faced with the difficult task of providing an enjoyable conference experience for all participants without the need for physical presence. In this precedent, livestreams were offered for each track throughout the conference, where each presenter had the option to either give a presentation with recorded video, live video broadcast or with a VR HMD. Additionally, there where multiple virtual meeting rooms in which participants could join and then interact and network with other users using VR HMD's. The virtual meeting rooms utilised during this conference were built on Hubs [171]. It features a web-based meeting room creation software, which enables users to develop and maintain their own meeting experience. Utilising a stylised graphics style, this system works with a desktop application, internet browser and even mobile devices, covering a broad possible user audience. Avatars are chosen from existing, pre-defined models. They use a mix between an abstract representation and Upper Body avatars by using a robot-like representation for users.

While this solution uses an open source approach, there are several commercial products available [99], [156], [187], [252], [282] which are similar in terms of remote collaboration with VR technology. Glue Platform [99] is a system for business professionals offering stylised 3D graphics. It is built to be used with VR HMD's and claims to be an extension to the everyday working life. Main features include spatial audio, 3D avatars, interactive and persistent objects. The avatars use a simple Head & Hands approach. Another commercial software available is called Breakroom [252]. It supports VR HMD's, is available for multiple platforms, and features full body avatars.

3.4.1. Comparison of Commercial Systems

Several VR remote collaboration systems can be found which are used in a professional context [99], [183], [200], [272]. These systems have many common aspects, such as allowing many users to participate, or the collaboration tools available in the virtual environment. Commercial systems seem to differ mostly in terms of avatars, environments and visualisation styles. Especially in interaction, the systems share many common collaboration tools inside the virtual world. A typical way of collaborating in these systems is to draw together either on a virtual surface or in the air in 3D space. Copying real world interaction methods, users can sketch on virtual black- or whiteboards and share their results in real time with each other. Usually there is also a name indicator that displays the name of a user. This is also necessary in systems with completely reconstructed or personalised avatars such as EngageVR [200] and Spatial [272]. A common use case in the aforementioned systems is sharing and observing virtual objects together. Systems such as EngageVR [200] or Neos VR [161] allow users to place any 3D object previously added to a catalogue. In some cases it is also possible to show other users information by floating markers i.e. annotations. The virtual environment in these systems usually have a table and multiple seats to copy the physical space of real world meetings. A common scenario involves users to sit on virtual chairs and to present on a virtual tv or projector. Some more advanced systems [161], [171] allow screen sharing to the virtual world. Even more sophisticated systems (such as EngageVR [200]) extends screen sharing functionality with full control over normal desktop applications in the virtual world. Additionally, many developers are implementing platform independence and support multiple devices such as desktop, VR HMD's, tablets and smartphones. Some systems such as Hubs [171] are fully available in a web browser.

3.4.2. Exploring the Strengths and Weaknesses of Commercial Remote Collaboration Systems

Different strengths and weaknesses of popular commercial virtual meeting systems were identified. The combined strength of these systems include:

1. Many users are able to join and participate in virtual meetings simultaneously [156], [171], [252], usually allowing about 20-50 people to share a virtual space simultaneously
2. Intuitive interaction possibilities such as mid-air drawing [171], drawing on white- or blackboards [156], media and screen sharing [200]
3. Spatial audio which enables localisation of an audio source during meetings more naturally [99]
4. Persistent virtual objects which exist through multiple sessions (for example, a drawing from a session before is still present in the next) [99]
5. Placing arbitrary 3D objects in a shared virtual environment [161], [200]
6. Reconstructed, personalised avatars [200] and user created avatars through an API which is provided by the developers [38], [161]
7. Availability on multiple platforms: desktop, mobile devices and web browser [99], [171]

Some systems are enterprise solutions [99], [183], tailored to the specific needs of companies, what renders them unattractive or even inaccessible to the general public. One major issue with these systems is the missing transmission of nonverbal communication cues to the virtual environment by means of an avatar, which is an important feature of traditional face-to-face collaboration.

Another weak point of the current systems is dynamic content creation for the virtual environment. These systems are limited to the choice of virtual environments provided by the developer or need expert knowledge to create them [99], [156], [171], [200].

Some systems implement a virtual space for many scenarios such as training, collaboration and other social activities inside a virtual world. NeosVR [161] and VRChat [38] allow experienced users to create custom environments, avatars and interactive objects by providing an API within the game engine in which the platform is implemented.

Many systems excel in certain aspects but lack novelty in others. As an example, the immersive remote collaboration software VirBELA [291] allows hundreds of users to participate in a virtual world simultaneously but the avatars lack personalisation.

3.5. Discussion and Survey Result

This section provides an overview of the insights and statistical data which was gathered throughout the survey. The taxonomy consisting of *Environment*, *Avatars* and *Interaction* is emphasised and important findings are presented for each category. It is to note that professional and commercial systems were included in this survey (24 VR based systems and one MR based system). This implies that some of the discussed applications are not published in scientific articles and the implementation of interactive features, virtual environment and the audiovisual representation of users is subject to change in the future. For example. a discussed system does not implement full body avatars at the time of writing, but could implement it later on.

3.5.1. Environment

The remote collaboration systems were categorised according to their technology: AR/VR/MR, use case, visualisation styles and sensory inputs. The technology distribution of included systems is shown in Figure 3.3.

Technology

In the category of VR based synchronous remote collaboration systems, 24 commercial and 7 research oriented systems were included. The majority of systems with VR technology are commercial systems, which could be an indicator that VR based systems are currently more under development in the industrial sector rather than the research community and therefore could be placed on the plateau of productivity. 17 purely AR based systems were found in which users are able to communicate and collaborate in real-time by means of video or optical see through AR or a combination of both. To the best of knowledge, there was no commercial system at the time of writing this thesis which is solely based on AR allowing real-time synchronous remote collaboration. MR based systems form the majority of the discussed systems. 33 research oriented and one commercial system are included in this category.

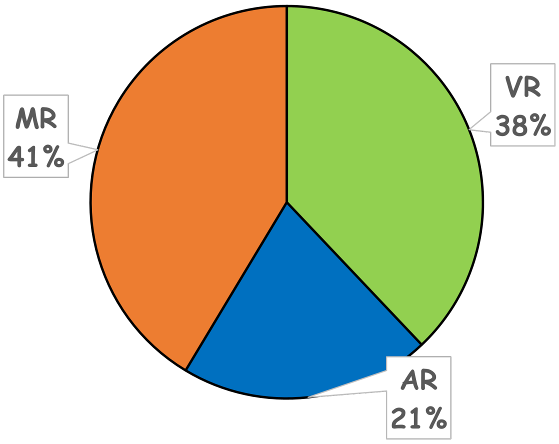

Figure 3.3.: *Distribution of used technology in the discussed systems.*

Use cases

The systems were divided into three different main use cases: *Meeting, Design* and *Remote Expert* as they are most popular in the literature. The distribution of mentioned use cases with respect to their technology is shown as a graph in Figure 3.4.

Systems based on VR technlogy often involve many users (more than two), with a focus on Meeting and Design use cases. The virtual environments used in VR technology tend to involve all participants equally, i.e. they can see the same things and have the same input modalities, whereas MR based systems often have asymmetric inputs. For example, an asymmetric input method would be when two users share a virtual environment and one has a keyboard and the other has a VR headset with controllers as input device. There was no purely VR based system which implemented a *Remote Expert* scenario although it is by far the most popular use case in AR based systems. While VR systems have a focus on *Meeting* scenarios and AR systems a focus on *Remote Expert* scenarios, MR systems are more distributed throughout use cases. This indicates a strong correlation between the hardware and its respective benefits in certain use cases. For example, VR HMD's are more beneficial in meeting scenarios with an immersive, shared virtual environment, while AR HMDs have more advantages in supporting a user with a remote expert. Since MR is a mix between AR and VR, the distribution of cases is more equally.

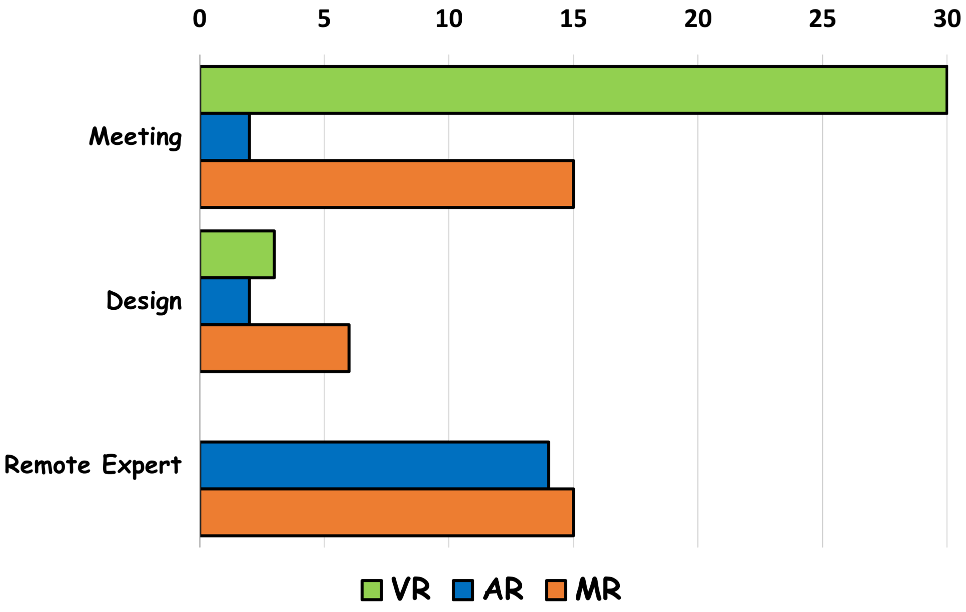

Figure 3.4.: *Distribution of use cases in the discussed systems.*

Visualisation Styles

No hints were found that indicates a clear deviation in respect to visualisation styles of synchronous remote collaboration systems. It was differentiated between stylised, realistic and annotation style. A system was labelled stylised when there was an obvious simplification of 3D objects with cartoonish visualisation. Older systems with realistic visualisation are also counted here, as long as no stylised visualisation technique was used. Many AR based systems use annotated video only while some use rendered 3D objects with a Stylised style.

Sensory Impressions

The recent literature seems to focus more on audiovisual systems and does not support other sensory impressions such as olfactory and tactile cues. Although tangible interfaces where popular especially in AR technology, the focus is drifting towards audiovisual systems. One of the reasons might be that markers are no longer required to be placed on physical objects which often already implied a tangible system, if markers are placed on non-stationary objects. Additionally, the tracking accuracy of AR systems is constantly being improved. An example is Rambach *et al.* [206], [207], where the authors use Simultaneous Localization And Mapping (SLAM) technology for accurate object tracking without markers.

3.5.2. Interaction

Examining the different types of interaction in remote collaboration systems, a general majority of the interaction type *Shared Object Manipulation* is found. A possible explanation for this is the ease of implementation and general versatility of a task involving the manipulation of 3D objects. Another popular feature is *Media Sharing*, i.e. possibility to share images, videos and other form of media. Some interaction types are more popular in VR technology, such as *Shared Object Manipulation, Media Sharing, 2D Drawing* and *Mid-Air Drawing in 3D.* The interaction types *Viewport Sharing, AR annotations* and *Shared Gaze Awareness* had no implementations at all in a pure VR scenario. Transmitting facial expressions by using avatars is by far the least prominent feature in remote collaboration systems, even though it is an important step towards more natural conversations over distance. A comparative graph about the interaction types and their presence in the discussed work is shown in Figure 3.5.

Additionally, it was found that most of the commercial remote collaboration systems rely solely on VR technology. Since the purchase of commercially available VR HMD's and a reasonably powerful PC is required to operate these systems, their actual use is still limited. Therefore, most professional and commercial systems tend to implement a desktop and mobile version of the VR application. Only a few companies, such as Spatial [272] focus on integrating AR, VR, desktop and mobile together. Deploying the same application on

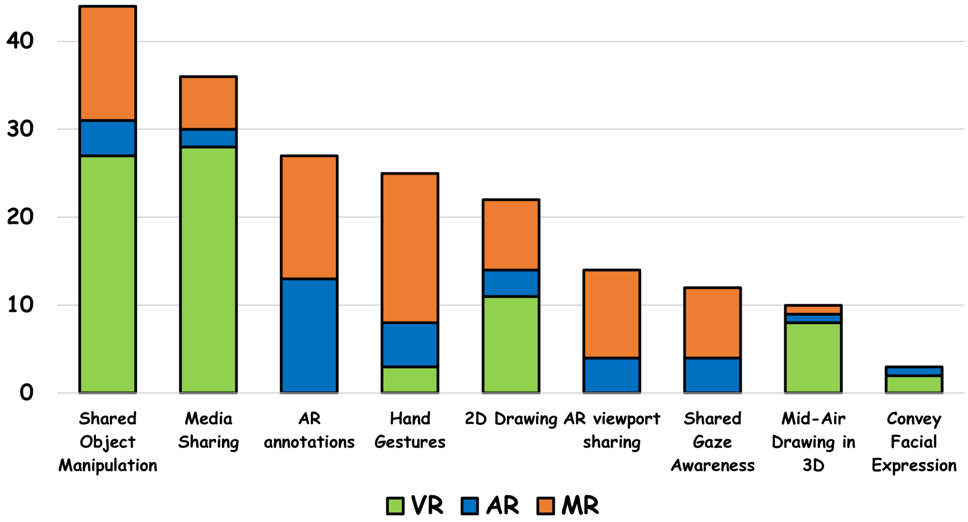

Figure 3.5.: *Interaction Types in respect to the used technology.*

multiple hardware devices, each with different input modalities, raises new issues such as asymmetric input. For example, a mobile device user can most likely collaborate only by voice or minimal interaction within a virtual environment, compared to a user with a VR HMD and full body tracking. In an attempt to solve the asymmetric input problem, Fleury *et al.* [64] investigated how the same interaction possibilities can be realised in virtual space, in a CAVE environment, and a high-resolution 2D display in wall format. Another approach is taken by Pouliquen-Lardy *et al.* [203] who implemented a multi-user remote collaboration MR system to study the asymmetric effects of different input modalities. The authors used an approach with two different roles, a guide who could observe and communicate via audio, and a manipulator who was able to manipulate a 3D object. The overall result of their study suggests that it is not necessary to develop symmetric interaction for all users during remote collaboration, but rather the same interaction possibilities for roles. For example, a user in the role of a guide should be able to observe and communicate via audio regardless of the hardware they are using.

3.5.3. Avatars

The systems were analysed with respect to their specific avatar implementation. An explanation of the avatar categorisation is given in section 3.3.2. It was found that there is not a single most used avatar type for XR systems. However, it could be observed that certain types of avatars are not used in combination with particular technologies. An overall distribution of avatars in the discussed systems is shown in Figure 3.6. Filtering the avatar types by technology gives interesting insights into the spectrum of personal embodiment in synchronous remote collaboration systems. Looking at Figure 3.7 it was found that *Full Body, Head & Hands* and *Upper Body* are most promi-

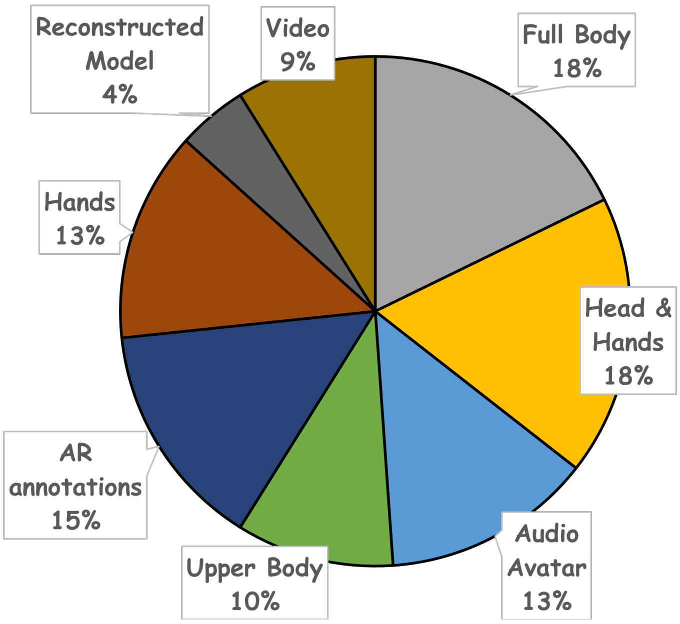

Figure 3.6.: *Different avatar types in presented sytems.*

nent in VR based systems. To the best of knowledge, there is no AR system that uses a full body 3D model for representation of other users at the time of writing this thesis. The closest approach to a full body 3D model representation in an AR system is Holoportation [188], but since several RGB-Depth cameras are involved that transmit video in real-time, it was classified as a video-based avatar system. The Hands approach for avatars is mostly used in AR and MR systems. This is due to the fact that hand gestures provide a natural and easyily understood way to convey visual communication cues in a *Remote Expert* scenario, which is used in AR and MR based systems. Overall, a lack of systems which utilises Reconstructed Models was found, i.e. an avatar that is created through face reconstruction in the virtual environment. A reason for this is that the technology for easy reconstruction of humans is not yet widely available for researchers and industry implementing such systems. Not surprisingly, *Audio Avatars* are mostly used in AR and MR based systems. One of the reasons is that the shared virtual environment is formed through video transmission and the communication with other users happens through audio. VR systems use a rendered virtual 3D scene that simplifies finding other users in a virtual environment, allowing for a more sophisticated visualisation approach such as *Full Body* or other types of visual avatars.

3.6. Summary

During this survey, different kinds of virtual environments, various types of avatars and many shared interactive elements for synchronous remote collaboration systems were identified. To help researchers in various research fields a taxonomy consisting of *Environment, Interaction* and *Avatars* was created.

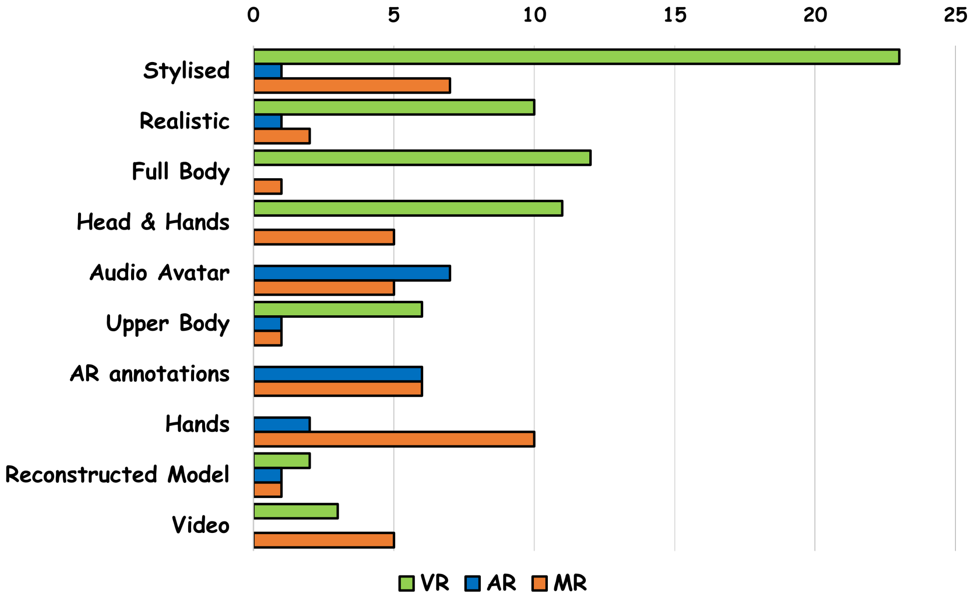

Figure 3.7.: *Different avatar types in respect to their used technology.*

This approach aims to provide condensed information to specific topics which needs to be addressed while designing and implementing a synchronous remote collaboration system.

In the category *Environment* it is found that the most prominent scenario for VR based systems is *Meeting*. While the *Design* scenario is rather equally distributed among the different technologies, *Remote Expert* is dominated by the use uf AR and MR technology. Regarding *Avatars*, most VR systems use a stylised visualisation with representations ranging from full body to head & hands equally distributed. Contrary, AR and MR systems tend to have user representations with hands only, use annotations, or rely on audio communication and omit visual avatars. Regarding the different interaction types, the most prominent features are shared object manipulation followed by media sharing.

Overall it can be said that congresses, conferences, conventions, and other events get reasonable alternatives to in person meetings with synchronous XR remote collaboration systems. However, systems based on XR technology rely on different strengths which are reflected in this overview. For example, it appears that AR based systems focus on sharing a person's surroundings with emphasis on stimulating the audiovisual senses. This has recently led to a focus on remote and local user scenarios in AR/MR systems. In such scenarios, typically a user is physically present at a certain location and shares his/her environment, which is then perceived by other users as a virtual world that can be augmented with virtual objects. This means however, that each type of user (local user or remote expert) has different input and output options in their respective virtual environment. These asynchronous input modalities often pose problems and therefore this subject is being actively researched by the scientific community. As an example, a local user streams the surroundings with an AR HMD, while a remote user can add annotations by using a VR HMD or tablet device which defines separate roles during remote collaboration.

In contrast, VR based systems tend to involve all participants equally, allowing each user to participate in the same virtual environment and the same communication and collaboration possibilities. VR systems are also commonly used to bring large events online. A trend in VR technology to focus more on design and meetings was found while remote expert scenarios are mainly the use case for AR and MR systems. Independently between AR, VR and MR the research focus in remote collaboration software drifts towards integrating non-verbal communication cues. Some researchers focus on developing solutions for non-verbal communication transmission, but this has not yet been integrated in professional and commercial remote collaboration systems. This research includes Wang *et al.* [303] focusing on eye-contact, Masai *et al.* [150] conveying facial expressions and Gao *et al.* [73] implementing mutual hand gestures to name a few.

Since novel remote collaboration systems are being developed on a continuos basis, it can be expected that systems based on XR technology will be integrated into our daily life. However, until this is the case, more research is required. VR-based systems require more comfortable hardware to allow a daily usage. Avatars within such systems must transmit the non-verbal communication signals of the participants in order to get closer to a face to face meeting. AR-based systems are still in need of better tracking algorithms and cheaper hardware so that these systems can be used by as many people as possible in everyday life.

This chapter is intended to give an impression of the wide variety and usefulness of XR systems. One of the biggest gaps in the existing concepts is the lack of tools to facilitate content creation. For example, it is difficult to find an appropriate compromise between the desired level of visualisation and cost. Furthermore, one of the biggest weaknesses of existing systems is the lack of transmission of non-verbal communication and the ease of creating gestures to allow natural interaction. This can be deduced from the fact that there is a number of works investigating hand gestures, but so far there is not a single commercial system for remote collaboration that uses them. The lack of simple design tools could be the biggest obstacle for XR technology that stands in the way of mass adoption. Therefore, the remaining chapters of the thesis focus on useful tools to support visualisation and natural gesture control with the hope of overcoming this barrier.

Chapter 4

Rapid Prototyping of VR Applications With Panorama Images

The previous chapter focused on different aspects regarding remote collaboration applications using XR technology. This chapter aims to address a specific shortcoming which was identified during this survey. Many applications excel in only one area of expertise. Some systems look very realistic, but are not able to integrate new input modalities or to transfer gestures into the virtual world. Therefore, a project to rapidly prototype immersive applications using panorama images is proposed. The aim of this work is to present a technique to allow developers a quick way to test and evaluate novel interaction techniques while still maintaining high visual fidelity with minimal effort. When designing 3D applications it is necessary to find a compromise between cost (e.g. money, time) and achievable realism of the virtual environment. Reusing existing assets has an impact on the uniqueness of the application and creating high quality 3D assets is very time consuming and expensive. A low cost, high quality and minimal time effort solution to create virtual environments is proposed here. The main contribution is a novel way of creating a virtual meeting application by utilising augmented spherical images for photo realistic virtual environments in combination with novel hand gesture interaction.

4.1. Panorama Images for Virtual Meetings

Meetings that require physical presence in other countries are often a time consuming overhead in addition to be costly. A great deal of money is spent on travel expenses while also reducing the working hours of employees by having them fly several hours from one end of the earth to the other. In today's world, where everyone is connected via the internet and where it is possible to communicate very quickly with each other, such meetings should rarely be necessary. This is often not the case in reality. Back in 2008, IBM realised the potential of virtual conferences and saved approx. 320,000 $ by having one of the biggest virtual conferences with over 200 participants according to Kantonen *et al.* [114]. Andres [4] came to the conclusion that a physical face-to-face meeting is a far superior approach than having a video conference (e.g. Skype) in terms of productivity. The loss of social presence while having a phone or video call is an important point of their conclusion. The proposedd system focuses on a virtual meeting scenario which provides tools for collaboration over long distances while maintaining a fully immersed experience. The system provides an immersive experience to be created in mere minutes by using spherical images for virtual environments. Virtual objects are placed within those environments which can be viewed and interacted with by users in a virtually enriched experience (see Figure 4.1). Multiple spherical images from different view points are used to create an illusion which represents an authentic and coherent environment. Hand tracking is used as input and interaction method. During development this approach has proven itself to be successful in different setups while maintaining a high quality virtual environment with almost no effort.

The contribution of this work is as follows:

- A novel VR application approach by utilising multiple spherical images to provide a photo realistic enriched virtual world.
- Utilising hand gestures for interactive 3D elements in enriched virtual worlds in the context of virtual conferences.

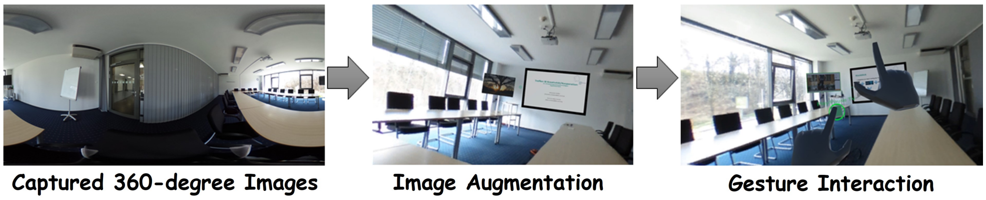

Figure 4.1.: *A virtual reality meeting room using the proposed method.*

4.1.1. Similar Approaches

A notably earlier work is from Kantonen *et al.* [114] where they propose a system for teleconferencing by combining AR mechanics with the game environment Second Life [214], which was a very popular online social game. Jo *et al.* [111] used 3D constructed environments and real environments (video background) in combination with different types of avatars. They showed that video based backgrounds are much more immersive and provide higher sense of co-presence than 3D replicated environments. A collaborative AR experience by using scene matching between two separate rooms was also done by Jo *et al.* [110]. Their main component was a Kinect sensor and a visual marker to match scenes and retarget avatar motions to fit in the physical place where each participant is located.

The presentation of the following works is focused on the virtual environments used to create applications. Lovreglio *et al.* [143] used a complex 3D scene for prototyping a virtual earthquake simulation. While it allows a more fine grained control of the virtual environment, a modelled 3D scene lacks realism. Matsas *et al.* [152] created an application to analyse human robot collaboration in VR. They state that they used a virtual scene with at least 42 new 3D models created and image textures from real industrial workplaces for a more realistic environment while also adding several auxiliary parts and objects to the scene. While Matsas *et al.* [152] did not mention how much time it took to create this scene, creating this specific virtual environment must have been a large overhead. The work of Hilfert and König [96] tries to address this overhead issue while Dinechin and Paljic [51] mention 360° images as an easy-to-use and low-cost alternative to acquire a 3D scene. Virtual service journeys where evaluated by Boletsis [14], where one evaluation method used 360° images from Google Street View for a prototyping phase. They mentioned that this approach was inexpensive in terms of man-hours and equipment costs compared to the other used prototype, a real world touring scenario. They conclude that there was no statistical difference in both prototypes which elevates the VR simulation prototype as a useful tool with significantly less expenses. A VR driving simulator for UI prototyping was done by Schroeter and Gerber [240]. They used 180° videos and a high quality car model in an automated driving scenario for rapid prototyping of UI inside the car.

Regarding hand gesture interaction techniques, there are several options available. A Microsoft Kinect was used by Oikonomidis *et al.* [185] for a marker less tracking of the hands. Hand tracking utilising machine learning was done by Malik *et al.* [148]. A colored glove was used by Wang and Popović [298] to estimate hand pose from single RGB images. For this prototype, the leap motion controller (LMC) was used. The LMC provides a tracking accuracy between 0.01mm and 0.5mm of the fingertip according to Smeragliuolo *et al.* [258]. The LMC was chosen for the development of the system because it is easy to integrate and, unlike other approaches, allows for egocentric tracking of the hands.

4.1.2. Essential Components for a Virtual Meeting Room

To create a virtual meeting room experience, the focus was on the three core aspects and their separate roles (illustrated in Figure 4.2): virtual environment, users, and shared interactive elements. According to the principle of the three pillars of remote collaboration presented in chapter 2.

Virtual Environments are essential to immerse a user in virtual worlds. In this case it is desired to create a realistic experience that includes all important aspects of a common meeting room. A room with multiple seats, a big table and projector can be identified as a meeting room. The typical way of creating such a room is to model all necessary 3D assets and properly align them. Depending on the desired amount of realism this process can be very costly and time consuming. Another option is to take photos of a room and then

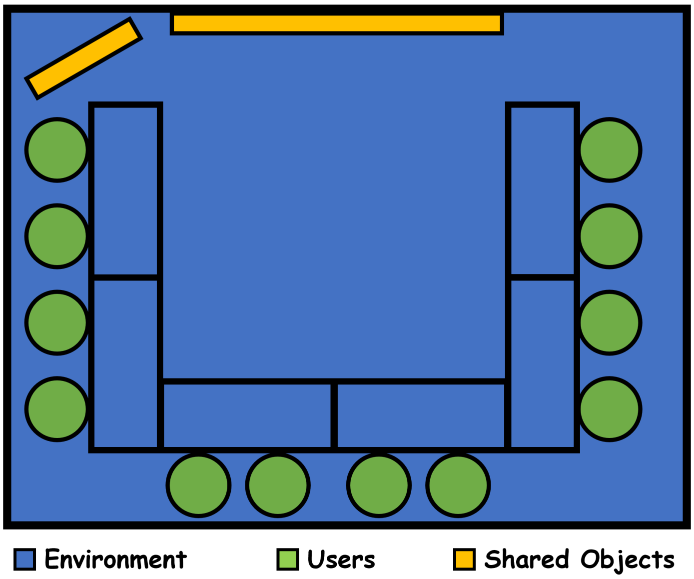

Figure 4.2.: *Schematic overview of a virtual meeting room. Three different roles are separated for a virtual meeting scenario.*

recreate it with 3D assets. It was found that this approach is too expensive and created a solution which is independent of the complexity of a room and still provide very high realism. This is why spherical images were chosen as a source to create photo realistic virtual worlds. The visual advantage compared to traditional artificially created 3D environments is enormous, they can capture the appearance of a room with every detail in a fracture of a second. Additionally it is possible to use High Dynamic Range (HDR) imaging techniques for even better visual impressions.

Users need visual representation and the ability to interact with the virtual environment. Since the used visualisation method depends on real environments, it was desired not to use any controllers and instead relied on hand gestures as input modality, since it is a natural way of interacting in the real world. Users are represented by a virtual avatar in order to be perceived by others in the virtual environment. Seeing the hands and gestures of other users is a great way to enhance the overall collaboration experience [196], [259]. The hands are primarily used for interaction with virtual objects shared across all participants in the virtual meeting.

Shared Interactive Elements are used to establish an information exchange between all persons present in the same meeting room. Those objects can be viewed, changed and interacted with by each participant. It is critical for a virtual meeting room experience to identify necessary components that enable a true meeting experience. In this case, the components are restricted to a projector and a TV to be those shared interactive elements, but this can be extended to any other element.

4.1.3. Implementation of the Prototype

The spherical images used for the prototype had a resolution of 5376 x 2688 pixels. The Unity game engine was used for visualisation and coupling of the hand tracking to the virtual simulation. The system supports all headsets supported by the OpenVR SDK (HTC Vive, Oculus Rift, ...).

Several spherical images inside a meeting room were taken, each centred at a spot where normally a participant would sit during a meeting. This enables the user to move to different positions in the room. Figure 4.2 sketches our approach while Figure 4.3 shows the result of the applied approach.

Figure 4.3.: *Multiple spherical images share a virtual TV and projector.*

For collaboration, two tools were implemented inside the meeting room. A TV and a surface where a projector could project images on the wall. Those two tools can be interpreted as shared interactive elements to be seen by several users at the same time during a meeting. The identified objects share the same content for each participant but are placed at different positions inside the spherical viewer (see Figure 4.3). The implementation uses manual placement of the shared elements. This can be extended to an autonomous process by using camera pose estimation methods. Multiple users can interact and observe the actions of each other in the meeting room. Users see each other as a minimalistic avatar with detached hands (see Figure 4.4). The fully rigged hands animated with tracking data from the leap motion controller are streamed over network to enable users to see each others exact hand movements. For some interactions eye gaze was simulated by using a forward vector of the VR headset position in virtual space. This is used for certain gestures e.g. a swiping gesture towards the projector will change slides only when the projector is being looked at. It is also possible to open a menu when the left palm is facing the VR headset. The user can grab objects from the menu and place them in the virtual world, where they expand and allow interaction (See Figure 4.4). The interaction possibilities include: Moving to a different seat, changing projector slides and showing a preview for upcoming slides.

Developing a Multi User Experience To implement a multi user experience, it is necessary to distinguish between the local and remote (other) users. The

Figure 4.4.: *Virtual objects can be placed by the user in the virtual environment to enable interaction.*

local user is always placed at position (x,y,z) = (0,0,0) in the world coordinate system. A sphere which has the spherical image as texture applied to it is placed on that position to achieve a spherical viewer experience for the local user. In order to plausibly position the other users, it is important to know the camera pose of the other viewing points i.e. photos taken at other positions. Gava and Stricker [77] provides an overview of the used method for automated camera position detection. Without proper camera alignment the user positions will not be consistent. Therefore it is required to calibrate all user spheres into a common coordinate system. Each possible viewing position requires the following information: Image texture, pose of shared interactive elements and pose of the other viewing points (user positions). The network code synchronises the position of players and the states of interactive elements as well as the fully rigged hands. Showing both hands with full motion of each of the other users is an improvement for the overall user experience of the system.

Steps to Create a VR Meeting Room Based on Panorama Images Creating a VR meeting room experience for a new room has few prerequirements:

- Take images on desired seating positions.
- Pose calibration of each seating position.
- Calibrate the pose of shared interactive elements.

Taking images of the desired seating positions is straightforward but it has proven to be better if the physical rotation of the camera stays similar. Also the height of the camera should be at eye level in seating position. Identifying the pose of each seating position can be done with an automated camera calibration framework. Identifying the pose of shared interactive elements has to be done manually at this time but can be extended to an automated process by utilising camera pose estimation methods as well as sparse 3D reconstruction of the environment. Necessary components like gesture interaction, networking, visual representation of shared objects, spherical viewer and user avatars can be reused easily.

4.1.4. Conclusion and Future Work

In this section, a method for fast and realistic creation of immersive virtual environments is presented. To avoid creating complex 3D assets which are expensive, spherical images are used to create a photo realistic environment. The idea to use multiple spherical images showing the same scene but from different viewpoints while adding shared virtual objects between viewpoints is presented. This approach enables realistic and believable virtual environments which can be created in a short amount of time. Additionally, the proposed approach allows to create virtual meeting experiences in almost any meeting room by taking multiple spherical images. Since a game engine was used, it is easily possible to reuse all of the critical components like spherical image viewer, shared virtual objects, hand based gesture interaction and networking code. The presented system provides the opportunity for all basic interactions necessary to give a virtual presentation, including shared virtual objects (projector, TV) between viewpoints and hand/gesture interaction. This approach is easily extended for prototyping and proof-of-concept in other scenarios like novel car interfaces or training scenarios. In the current state it is possible for multiple users to collaborate together in the room but there is only a minimal visual representation of each user. The addition of more suitable avatars as well as sparse reconstruction to realise faithful occlusion of more complex avatars would round off the collaboration experience.

4.2. Comparing the Panorama Based Environments with a 3D Modelled Environment

As mentioned earlier, VR applications are becoming increasingly mature and therefore the requirements and complexity of such systems is steadily increasing. Realistic and detailed environments are often omitted in order to concentrate on the interaction possibilities within the application. Creating an accurate and realistic virtual environment is not a task for laypeople, but for experts in 3D design and modelling. To save costs and avoid hiring experts, panorama images are often used to create realistic looking virtual environments. These images can be captured and provided by non-experts and are an alternative to handcrafted 3D models in many cases because they can offer immersion and a scene can be captured in great detail with the touch of a button.

An example of such applications is the previously presented prototype in section 4.1 which used panorama images to create a photorealistic VR meeting room. Furthermore, Sayyad *et al.* [227] implemented a system where panorama images are blended together with 3D objects via texture inpainting. Rhee *et al.* [215] implemented a system that seamlessly composites 3D virtual objects into a 360° panoramic video. The number of applications which use panorama images and videos is steadily increasing and new applications are constantly being developed.

However, to the best of knowledge, no study has been conducted at the time of writing this thesis which compares user perception between a panorama-based environment and a handcrafted 3D counterpart. Therefore, it should be investigated whether it is advisable to recreate an environment in detail by hand or whether it is recommended to use panorama images for virtual environments in certain scenarios. It should be discovered if it is worthwhile to create a virtual replica of an environment or if it is enough to capture the desired scene in panorama images and then present it to the user. For this purpose, an interactive virtual environment was created in which a handmade 3D environment is almost indistinguishable from one created with panorama images. In addition, hand tracking and virtual objects were included to both, the 3D and panorama-based environment. Therefore, an interactive VR scenario was created which is used for evaluation. In a user study, participants completed a visual search task and then filled out a questionnaire to allow comparison between the two environments. The goal of the study is to answer the following question: *Is the sense of presence in VR through panorama images on par with an environment modelled in 3D?*

4.2.1. Immersion and Sense of Presence with Panorama Imaging

Immersion is often described as the objective properties of the virtual environment that create the feeling of presence [19], [255], [257]. Presence and immersion are two distinct concepts which are directly related [254] as the sense of presence is the result of immersion [241]. Immersion, by its technical definition, is able to create a sensation of presence as described by Mestre *et al.* [160]. Presence is the sense of an individual within an immersive environment and immersion stands for what the technology provides from an objective point of view. As immersion is at the core of VR applications, it is an important area in research.

More examples of panorama images to create immersive experiences in VR are found in the literature. Škola *et al.* [253] combines an interactive VR experience with 360° storytelling experience in the context of cultural heritage. Using an underwater VR scenario, the authors report high levels of immersion using 360° videos. Ben Ghida [9] uses panorama images to lecture a history class and argues that such a system could be used in the future of teaching. Metsis *et al.* [162] use panorama images to study and treat psychological disorders such as social anxiety. The authors use 360° videos for rapid prototyping and create therapeutic VR environments.

The use of 360°/panorama images and videos is steadily increasing, but the sense of presence in a panorama-based compared to a similarly modelled 3D environment has yet to be explored.

4.2.2. Experiment Implementation

Several panorama images within a real meeting room were taken. The images were captured with a tripod on a chair at the height of a person in seating position. The images have a resolution of 5376 x 2688 pixels. These images are used to create a spherical image viewer to be experienced with a VR HMD. Users are able to "sit" on each chair by switching between captured images. The selected real room has a television and projector screen. In the panorama-based virtual environment, these screens are simulated by superimposing virtual objects on the panorama images (See Figure 4.5**C**). Images and videos can be shown in a way that appears to the user as if it were displayed by a real TV or projector.

A virtual replica of this room was carefully handcrafted by an experienced 3D artist (See Figure 4.5**A-B**). It took about 120 working hours to recreate the room in full detail. Nearly each aspect of the real space was recreated, from the texture of the carpet to the plastic bag in the room's trash can. Furthermore, realistic lighting conditions were recreated. Behind the blinds of the windows is a high resolution HDR texture that is used to realistically illuminate the room. Figure 4.5 shows both scenes.

For the experiment, interactive elements are placed in the near field of the user to answer questions shown on the projector screen. To answer these questions, the user has to enter digits on a calculator-like object with their hands.

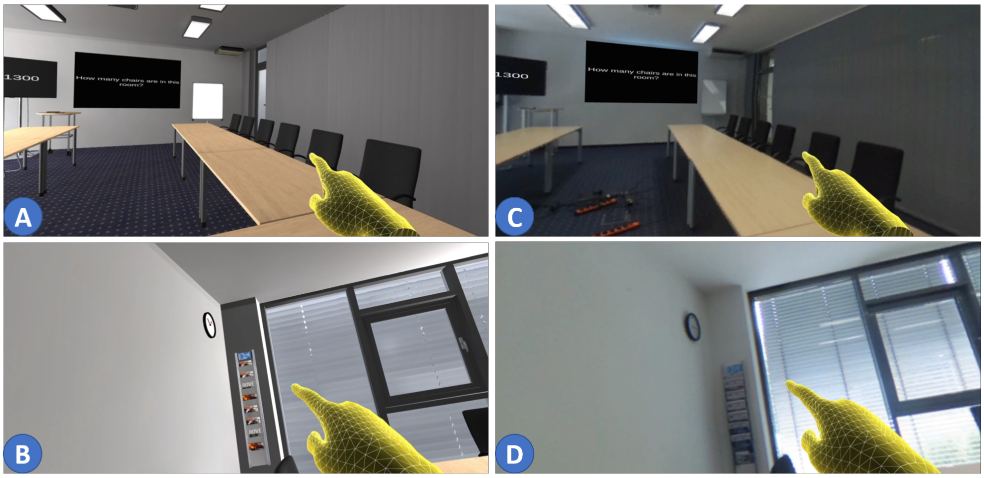

Figure 4.5.: *Image (**A**) and (**B**) show the handcrafted virtual environment with 3D geometry, (**C**) and (**D**) show the virtual environment based on panorama images. The text shown in image (**C**) is superimposed onto the panorama image and shows questions to the participants.*

4.2.3. Conducting the Experiment

Objectives In this experiment, the perceived realism of a handcrafted 3D environment compared to a virtual environment based on panorama images within an interactive VR scenario is investigated. In order to achieve this, selected questions of the igroup presence questionnaire[1] (IPQ) [211], [241], [256] are used. Only a subset of the complete questionnaire was used since some questions were not applicable to our scenario and in order to shorten the time for the experiment. The list of questions is shown in section 4.2.4. Subjects answered the questionnaires within the virtual environment as the work of Schwind *et al.* [243] suggests.

Participants For this study, 8 volunteers were recruited (4 Male, 4 Female). The age of the participants ranged from 31 to 60 years. All participants had no prior VR experience.

Apparatus The experiment was conducted using an Oculus Quest 2 VR HMD connected to a gaming laptop. The resolution of the HMD is 1832 x 1920 and it has a 95 degree field of view. Hand tracking and interaction with virtual objects was implemented using the Oculus SDK in the Unity game engine.

Experimental Task The participants sit on a chair in the virtual environment. They have to perform a visual search task in which the participant is shown questions. The questions include counting the number of chairs, tables, windows, or coat hooks on the wall. There are a total of eight questions to be answered. These questions encouraged the user to fully look around in the environment. For example, to answer how many coat hooks are on the wall, the user had to turn 180° in order to solve this question. The virtual environment changes to the handcrafted or panorama-based environment after each question is answered correctly. The order of the environments was counterbalanced.

To answer the questions, a virtual object similar to a calculator is displayed to the user which can be operated hands free. After all questions are answered, the environment changes to the 3D or panorama environment respectively and the selected questions of the IPQ are answered in the end. For this questionnaire, an object with 7 buttons is displayed that represents the answer options ranging from "fully disagree" to "fully agree".

Procedure Participants were told to put on the VR HMD and look around to familiarise themselves with the environment. Immediately after putting on the VR HMD, the first question is visible on the projector screen. After all questions of the visual search task are completed, the selected IPQ questions are displayed. These are answered twice, once in the 3D modelled environment

---

[1] http://www.igroup.org/pq/ipq

and once in the panorama-based environment. Each session lasted about six minutes, where participants spent three minutes in each environment.

4.2.4. Results and Observations

The participants filled out a questionnaire for qualitative measurement for sense of presence in the proposed environments. Five questions from the IPQ are chosen and answered with a 7 item Likert scale (ranging from 1 to 7):

Q1: In the computer generated world I had a sense of "being there".
Q2: Somehow I felt that the virtual world surrounded me.
Q3: How real did the virtual world seem to you?
Q4: I felt present in the virtual space.
Q5: I felt like I was just perceiving pictures.

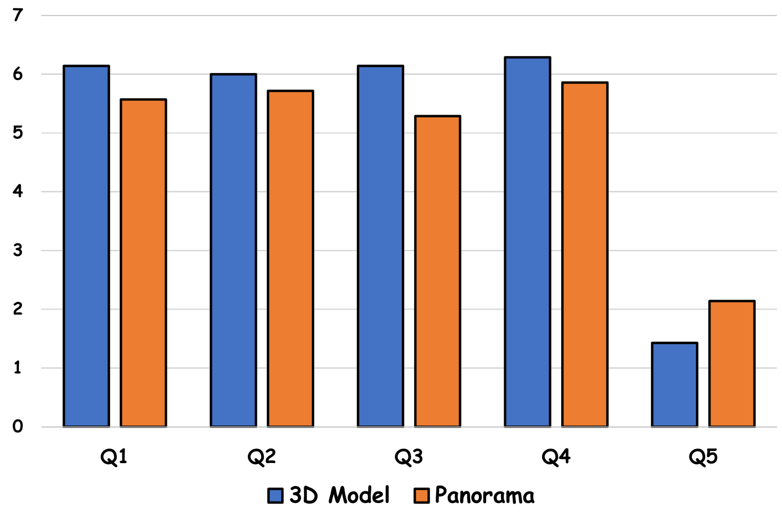

Figure 4.6.: *Raw average scores from questionnaire results. A score of 1 means fully disagree and 7 means fully agree.*

The average score for each question is shown in Figure 4.6. Interestingly, some participants did not notice that they observed an image instead of the 3D environment. One participant asked why the questions repeated, not knowing that the environment had changed. Some participants described the panorama based environment as pixelated compared to the handcrafted environment. Additionally, some users felt elevated in the panorama based environment. Statistical tests were performed for each question to find significant differences between the two groups 3D and panorama. Levene's test assured the homogeneity of the input data for each question with $p > 0.05$ and therefore the data was analysed using one-way ANOVA. No significant differences between the group scores were found. The obtained p values from post-hoc analysis are the following: **Q1:** $F(1,14) = 2.13$, $p = 0.17$; **Q2:** $F(1,14) = 0.07$, $p = 0.78$; **Q3:** $F(1,14) = 1.17$, $p = 0.29$; **Q4:** $F(1,14) = 0.29$, $p = 0.59$; **Q5:** $F(1,14) = 3.26$, $p = 0.09$.

Answering the Research Question *Is the sense of presence in VR through panorama images on par with an environment modeled in 3D?*

To answer this question, the user responses as well as the questionnaire results are used. Using the five questions mentioned in section 4.2.4, no significant different scores between the 3D modelled and the panorama-based environment were found. One participant did not notice that the environment had changed at all. This data and the observations lead to the conclusion that a panorama-based environment could successfully substitute a handcrafted 3D environment. However, further studies with more subjects should be conducted in order to make a conclusive assessment.

Limitations The interactions in a panorama-based environment are limited but virtual objects can be superimposed to enable interaction. In this work, users had two screens and virtual objects with buttons (to answer the questions) as interaction possibilities. For the chosen scenario, panorama images can be a successful substitute, but further investigation for different and more complex scenarios is necessary to draw more comprehensive conclusions.

Furthermore, panorama images should not be used in VR scenarios where the user is allowed to freely move around but rather in situations where he is allowed to teleport to specified points in the virtual environment. However, panorama images should be considered as a low-cost alternative in cases were users are not allowed to freely move around. For example, if scenarios involve roles such as observers or referees.

In future work, images with increased resolution should be used as some participants perceived the panorama environment as blurry which was caused by the comparatively low resolution of the images.

4.2.5. Conclusion and Future Work

The proposed experiment suggests that panorama images can be a viable alternative to handcrafted virtual environments. Depending on the context, panorama images can be an affordable and effective substitute for carefully crafted virtual environments. According to the study, users did not experience a significant difference in the sense of presence within the proposed environments. Although the chosen experiment is limited to a visual search task, there are many scenarios to which the results of this work are applicable. While the findings of this study are a starting point, a more complex study with a larger number of subjects will be conducted in the future. Furthermore, different scenarios should be evaluated. This work aims to encourage researchers to further investigate in this area.

Chapter 5

Building a Framework for Arbitrary One-Handed Gestures

The prototypes in chapter 4 already used hands for interaction, but the existing solutions showed that they are limited, rather complex, and not adaptable. The search for appropriate gestures for specific interactions is a rather tedious task. The aim of this chapter is to provide an easy way to create any gestures one has in mind. More precisely, the content of this chapter aims to support filling the research gap towards the acceptance of hand gestures in immersive applications, like those mentioned in section 3.6. Natural user interfaces based on hand gestures are becoming increasingly popular. The need for expensive hardware left a wide range of interaction possibilities that hand tracking enables largely unexplored. Recently, hand tracking has been built into inexpensive and widely available hardware, allowing more and more people access to this technology. The previous chapter used hand gestures provided by SDKs from manufacturers. The number of supported gestures is limited and creating new gestures is a complex task. This chapter provides researchers and users with a simple yet effective way to implement various one-handed gestures to enable deeper exploration of gesture-based interactions and interfaces. To this end, the presented work provides a framework for design, prototyping, testing, and implementation of one-handed gestures. The proposed framework was implemented with two main goals: First, it should be able to recognise any one-handed gesture. Secondly, the design and implementation of gestures should be as simple as performing the gesture and pressing a button to record it. The presented approach was evaluated in a user study with 33 participants and gestures received high accuracy and user acceptance.

5.1. Introduction to Hand Tracking for Interaction

Especially for Augmented and Virtual Reality research and its applications, gesture recognition and hand gesture based interfaces are becoming increasingly important. In the earlier years of AR, the interaction was mostly based on physical objects with markers attached to them [11] while more recent applications are using hand tracking for interaction [208], [225], [268]. Hand tracking technology is becoming more reliable and is built into various types of HMDs allowing hand interaction out of the box.

The importance of hand-based interaction is steadily growing and hand gestures are more frequently used in various application and research scenarios. It is utilised for design and engineering by Murugappan *et al.* [173] to visualise new concepts and ideas through the use of a hand gesture-driven 3D shape modelling tool for creative expression. Furthermore, hand gestures are also used in vehicles to access various functions while driving without averting one's gaze, as described by Riener *et al.* [216] or Verma and Choudhary [290]. Medical applications use touchless hand gesture based interfaces to guarantee safety or sterility during operation [90]. Hand gestures are also used to improve remote collaboration [224], [237]. Grasping virtual objects is investigated by Vosinakis and Koutsabasis [293], and virtual object manipulation by Song *et al.* [260]. Locomotion solely based on hand gestures is discussed in chapter 6. New applications based on hand gestures are constantly being developed. Commonly available hardware to detect and track human body parts contributed significantly to development of gesture based interactions and interfaces.

Frameworks from HMD manufacturers usually allow an easy integration of hand gestures for AR and VR applications. These frameworks however, are often tied to specific hardware and have limited amount of available gestures. Defining new gestures is not possible or complicated. Additionally, dynamic gestures such as drawing a sign in the air, concatenation of gestures, or gestures composed of complex hand movements are not supported.

Researchers implement various gestures with rather complex methodology and require many training samples in order to implement a reliable gesture. One of the most used approaches is the Hidden Markov Model (HMM) which requires many samples to complete graph optimisation and the process of training is relatively complicated. Many existing frameworks are therefore either not generalisable, complex to use, or tied to specific hardware.

This chapter proposes a solution to that with a simple yet effective way to record and recognise any one-handed gesture. The gestures which can be implemented with the proposed framework are ready to be used for many different applications. This contribution has the intention to support a variety of hardware and should be seen as a complementary solution and not as a replacement to existing frameworks and toolkits. Part of this work was inspired by the \$1 recogniser from Wobbrock et al. [310] in the sense that defining and recognising gestures should be cheap, easy, and flexible to design. In order to evaluate the proposed methodology, a user study was conducted.

The contributions of this chapter are:

- The possibility to capture arbitrary one-handed static and dynamic gestures via button press, enabling rapid design, prototyping, testing, and implementation of one-handed gestures. This includes dynamic gestures with changing hand shape

- A simple, yet unique way to record and recognise dynamic one-handed gestures
- A comprehensive user study to evaluate the proposed method

5.1.1. Popular SDKs for Hand Gestures

Devices built to support hand tracking usually come with a SDK that provides visualisation and simple interactions with virtual objects using hands within immersive virtual environments. The Mixed Reality Toolkit [164] (MRTK) is a popular choice for AR and VR applications, as there are several sophisticated devices which are supported. It allows users to manipulate virtual objects by pinching and other rather simple hand gestures. The MRTK relies mostly on virtual objects such as menus, buttons, and sliders for interaction. This SDK should not be considered as a framework for hand gestures but rather than a whole toolkit to enable interaction with virtual objects by using natural input such as hands or eye gaze. The Leap Motion SDK [285] can be used with specific hardware to enable hand gesture based interaction in AR, VR, MR, or even desktop based systems without HMDs. The SDK provides various options for hand gestures such as pinching or grabbing. Other gestures can be defined, but the SDK as it stands is limited to static gestures. Recently, the Oculus portfolio was extended with HMDs that have built-in hand tracking support. The SDK [142] to support this hardware enables hand based interaction with objects by using simple hand gestures such as pinch, unpinch, and pinch & hold.

In contrast to these SDKs, the framework proposed in this paper allows for fast and easy design for arbitrary one-handed gestures. Complex dynamic gestures can be recorded and recognised for usage within an interactive system which is not possible with the aforementioned SDKs.

5.1.2. Dynamic Gesture Recognition

Several techniques and frameworks for dynamic hand gesture recognition are proposed by researchers [138], [140], [174], [244], [275], [287], [319]. Unlike existing solutions, the framework presented in this paper does not require data sets, network training, or expert knowledge to create new gestures.

Studies have been conducted on gesture recognition systems and frameworks in the context of sign language. Galván-Ruiz *et al.* [69] provides an overview of systems ranging from 1963 to 2020. The discussed papers include a wide variety of different types of devices used to recognize hand signs e.g. Wi-Fi, RFID, Vision, and Electromyogram data based recognition systems. Cheok *et al.* [44] provides an overview for systems between 1995 - 2016 and states that Hidden Markov Model (HMM) appears as promising approach towards dynamic gesture recognition while Support Vector Machine (SVM) is the most popular method for static gestures. The survey also highlights the accuracy and sample size of the individual systems. For additional background regarding hand gesture recognition systems the reader is referred to surveys such as

[39], [41], [104], [119], [167], [222]

5.1.3. Gesture Design Tools with Similar Objectives

This section covers the closest gesture-based interface prototyping tools to AnyGesture (name of the proposed framework). These works have similar objectives such as creating gestures without expert knowledge and rapid prototyping of gestures. However, AnyGesture is clearly distinguishable from existing work. A comparison is depicted in Table 5.1. As an example, Ashbrook and Starner [5] introduced the tool MAGIC which allows to create and prototype gestures with a 3-axis accelerometer. Unlike the system presented in this paper, it cannot recognise hand shapes. Speicher and Nebeling [264] proposes GestureWiz, a system which is capable of recording and recognising gestures from video input data. The system allows rapid prototyping for gestures with a consumer grade webcam. Compared to this system, AnyGesture works with hand skeleton data and is mainly focused on egocentric vision, which makes it more attractive for AR, VR and MR applications.

Table 5.1: Comparison and differentiation with similar systems in existing literature.

	Easy Design of New Gestures	**Hand Movement Recognition**	**Hand Shape Recognition**	**Individual Finger Movement**	**No Training Data Required**
MAGIC [5]	✓	✓	✗	✗	✗
GestureWiz [264]	✓	✓	✓	✗	✗
Gesture Knitter [168]	✓	✓	✓	✗	✗
AnyGesture	✓	✓	✓	✓	✓

The system Mogeste is proposed by Parnami *et al.* [192] which allows prototyping and testing of gestures realised with inertial sensors on commodity wearable and mobile devices. Lü and Li [144] propose a system for rapidly prototyping multi-touch gestures. Their system is able to generate gestures which can then be used for multi-touch gesture based interfaces.

Nebeling *et al.* [175] proposes Kinect Analysis to inspect and annotate motion recordings obtained from depth cameras which could then potentially be reused to create gestures. Mo *et al.* [168] introduced Gesture Knitter, a tool which allows users to combine fine and gross primitives with a visual scripting language. As an example, users are able to combine a static fist gesture (fine primitive) with a forward movement (gross primitive) to create a “fist forward” gesture. The authors of [168] train HMMs to recognise gestures which require training samples as opposed to AnyGesture. Furthermore, Gesture Knitter uses the centre of mass of the hand to track and detect the movements. This allows only static hand shapes to be formed and recognised while AnyGesture additionally allows movement of individual fingers for forming a gesture. This allows AnyGesture to record and recognise a wider range of gestures, including microgestures where only small subtle finger movements are involved. Another advantage of AnyGesture is that it is intended for use in an XR application.

5.2. Framework Implementation

5.2.1. Possible Gestures

A distinction between two essential types of gestures is made within the proposed framework: Static and dynamic gestures (similar to the work of Li *et al.* [138]):

Static Gestures. A gesture that is detected solely by recognising a specific hand shape is considered a static gesture. It can be either rotation invariant or tied to a specific hand orientation. Temporal and spatial information, such as the path of the hand, is not considered while recognising this type of gesture.

Dynamic Gestures. A dynamic gesture uses spatial information, such as the path of the hand or individual joints. This type of gesture is recognised by detecting patterns in movement. Thus, previously defined behaviour can be recognised and used. It is to note that stroke gesture recognition, such as that conducted in [126], [288], [310], is a subset of this category. Dynamic gestures in the proposed system also allow for individual fingers to be moved and recognised as gesture, which is a unique feature compared to similar work in the literature.

A typical static gesture would be a “thumbs up” while drawing a circle in the air with the index finger would be considered a dynamic gesture. Performing a pistol gesture by only moving the thumb down is a dynamic gesture with an individual finger movement.

5.2.2. Framework Architecture

This section provides a general overview of the implementation of the proposed framework. A crucial component of the framework is the hand data provider which was implemented to decouple the framework from specific hand tracking hardware and SDKs. It is designed as a wrapper to allow all relevant framework components access to hand tracking data without relying on a specific implementation for hand tracking. In essence, it allows access to hand joints, hand scale, and has support functions to provide data to other components. It generalises incoming hand tracking data by relying only on joint positions as input. Thus, there is no dependency to specific implementations by manufacturers, and users of the framework only have to make a few adjustments to support a particular device.

The shape of the hand, which is composed of the positions of each individual hand joint, can be stored for recognition of a gesture. Common hand tracking devices are able to track the Distal, Middle, and Proximal Phalanges of the fingers and provide information about palm and wrist position.

While a static gesture is recognised with a previously stored hand shape, dynamic hand gestures need a subset of these points. As an example, a dynamic gesture can be recorded by tracking the position of a single or multiple chosen joints. In the process of implementing the framework, it was found that the most important joints are the finger tips, palm, and wrist. Invisible 3D objects are attached to these points in order to allow more interaction possibilities. These invisible objects are basically spheres attached to key positions which can be used to record a spatial path or to perform collision detection (e.g. detect if something was touched by a joint). Collision detection within the hand was found to be particularly useful for some gestures, e.g. the pinch gesture, since a pinch can be reliably detected when the thumb and index finger collide.

To enhance robustness, a subsystem was implemented that observes the movement state of the key positions. It can be considered as a post processing step to the provided hand tracking data. While developing the proposed framework, it was found that such a component is crucial to reliably recognise gestures since it can prevent many undesired hand gestures. Such undesired hand gestures often occur if too many possible gestures are stored in the system, the user is moving the hands, and no preventive measures are implemented. The position of a hand joint is observed over time and previously chosen criteria determine whether a specific joint is currently moving or not. These criteria define the following behavior:

- At which threshold a joint position can be considered still. This is necessary to eliminate noisy data from the hand tracking device.

- How long the joint should be still until it is considered to be not moving.

As mentioned earlier, the motion path of joints is stored. The sampling rate as well as the amount of stored information can be configured. Depending on desired behavior, a gesture can only be recognised if one or multiple joints are not moving. Each joint is individually observed but a higher-level component is used to monitor the state of the whole hand.

A rough overview of the main parts that make up the framework is shown in Figure 5.1. The framework supports an easy swap of the *Hand Tracker* component, allowing many hand tracking solutions to be used with the framework. The *Gesture Recorder* allows to record and save gestures which are then stored in a set of known gestures. These gestures are then recognised by the *Gesture recogniser* by comparing stored data with live data from the hand tracker. A *Gesture Interpreter* is used to communicate with the desired application by using events that inform when gestures are performed.

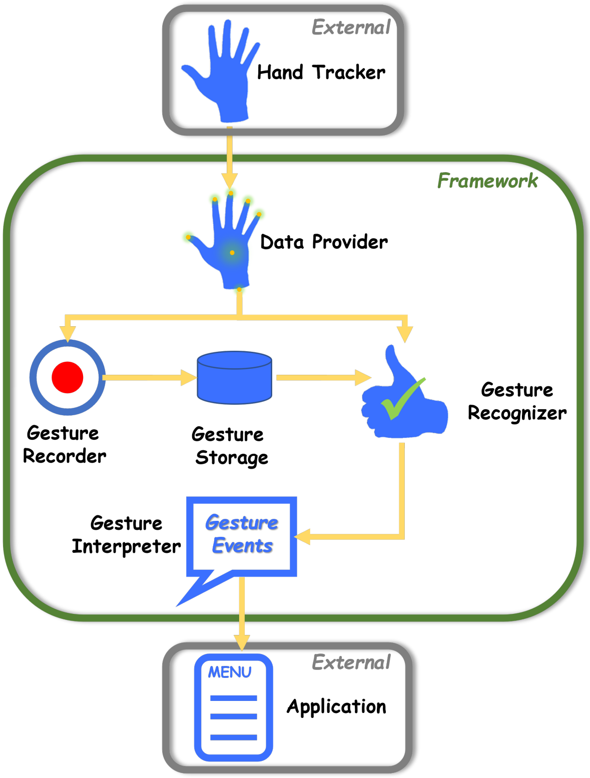

Figure 5.1.: *Overview of the proposed framework.*

5.2.3. Tracking

The hand detection in the proposed framework is realised using a hand tracker built into an HMD. The tracking part is crucial and serves as an entry point to the gesture recognition framework as it is responsible for accurate pose matching. The hand tracker primarily used in the development of the framework provides 23 points for each hand (See Figure 5.2). Other configurations are supported as well. The more points a tracking device provides, the more fine-grained is the gesture recognition.

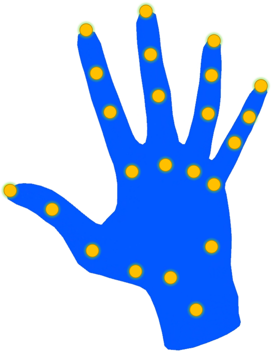

Figure 5.2.: *Hand points used to capture static gestures.*

5.2.4. Feature Extraction

The proposed framework requires two primary features for recognising one-handed gestures:

- **Joint positions** for static hand shape detection and matching
- **Finger tip positions** for recording spatial information required to perform dynamic gestures

Hand poses and shapes can be stored for later recognition of static hand gestures, e.g. while a user is performing hand movement it can be matched against a predefined set of hand shapes. In order to increase the recognition performance for users with hands below or above the average human hand size, the hand positions are adjusted by a scaling factor. This normalisation is necessary to increase the recognition accuracy for gestures that were recorded by a different user than the user performing the gestures. To achieve this, each joint position is divided by the hand scaling factor.

The raw hand shape does not take hand rotation or orientation into account and therefore has rotation invariance. Performing a hand shape that resembles a “thumbs up” will therefore be indistinguishable from a “thumbs down”, but it should likely have a different meaning. To solve this problem, the orientation of the hand in relation to the head of the user is utilised. The direction the user is looking at is calculated by using the forward vector of the HMD. Then, the angles between head direction and the x, y, z directions perpendicular to the hand at its current position are calculated. This feature can clearly identify the orientation of a hand relative to the VR HMD. This allows for gestures that should be activated when the hand is facing the user or the palm should be upwards etc. A static gesture is then formed by storing joint positions provided by the hand tracker and the hand orientation relative to the users facing direction. Static gestures can be captured by pressing a single button on the keyboard. To be more precise, the user shapes his hand as the gesture should be defined, presses a key, and this shape is then added to the set of predefined gestures that can be recognised by the system (see Algorithm 1).

Dynamic gestures on the other hand use spatial information for specific points on the hand, e.g. finger tip positions or the palm. The proposed framework allows the recording of spatial information for any joint with arbitrary movements that can be stored for later recognition. This process is depicted in Figure 5.3. Each dynamic gesture requires a “start” shape, i.e. the system needs to know when the user intends to perform a dynamic gesture. For this reason, each dynamic gesture has a static gesture attached to it which is later matched to the hand shape from the tracking device during run time. A dynamic gesture can be considered as a more complex variant of a static gesture. While a static gesture is considered as performed as soon as it is recognised, it can be used as a cue to activate a dynamic gesture.

In order to record and store a dynamic gesture, the user shapes his hand as the gesture should be defined and presses a key to start the recording process. The gesture recording component attaches itself to predefined points on the

Algorithm 1 Capturing Static Gestures

```
Desired hand shape is formed by the user
Gesture capture event is triggered via keyboard press
Extract joint positions from current hand pose
Create new gesture object
for Each joint on the hand do
    Transform joint position from world space to local space
    Adjust joint position with hand scaling factor
    Store joint position in gesture object
    Calculate the orientation angles of the hand relative to the head direction of the user
end for
Store calculated hand orientation relative to the users' facing direction in gesture object
Add gesture object to list of known gestures
```

hand such as the index finger tip and follows the path of it until the key is pressed again. These predefined points should be chosen by the user before a gesture is being recorded. A gesture that is primarily done with the index finger should record the index finger, a swipe gesture might be recorded best if the palm is observed, etc. Which joints should be tracked for which gesture can be decided by the user of the framework. Recording will save a path consisting of 3D points for chosen joints. These 3D points are stored by computing the spatial differences between subsequent points.

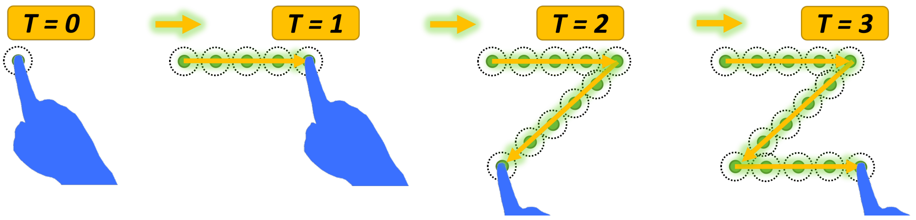

Figure 5.3.: *The proposed framework allows to record gestures by storing the spatial path of desired hand joints as 3D points and represents them as spheres. To help the user redo the desired gesture, a visual path is shown (can be customised or disabled) which the hand joints should follow.*

The sampling rate of the gesture recording can be freely adjusted (i.e. the number of 3D points stored for a gesture). A high sampling rate means a fine-grained gesture with many points stored, while a lower rate might be more inaccurate but stores less information. It should be noted that a hand tracking device with a low frame rate may cause difficulties in imitating a gesture. This is because the hand tracking might introduce stuttering which can not reproduce a fine-grained spatial path. Therefore, a low sampling rate is advisable if the target device has a low frame rate. Multiple hand joints can be recorded at the same time. The spatial path is stored as a doubly linked list for each individual joint, i.e. each point knows it's predecessor and successor. The list is always initialised with a point at the local coordinates (X,Y,Z) = (0,0,0) set as first element. This ensures that each stored gesture path always starts at the initial position of its associated joint during the recognition phase.

5.2.5. Gesture Recognition

Static Gestures

Static gestures are recognised similar to how they were recorded, e.g. current joint positions are retrieved from the tracking module, hand scaling is applied, and hand orientation is calculated. The values from each hand frame are then compared to the stored values i.e. gestures. A threshold T_{Shape} can be used to either relax or increase the constraint to detect a gesture. To recognise a gesture, the mean difference for each joint position from the live data and the positions stored inside a gesture are calculated and then compared with the threshold (Equation 5.1).

$$|(\sum_{n=1}^{joints} CurrentPos_n - StoredPos_n)| < T_{Shape} \tag{5.1}$$

The same is done with the orientation where OC_x, OC_y, OC_z represents the current data from the tracker and OS_x, OS_y, OS_z the stored angles between HMD facing and hand direction (Equation 5.2).

$$|(OC_x - OC_y - OC_z) - (OS_x - OS_y - OS_z)| < T_{Orientation} \tag{5.2}$$

The thresholds T_{Shape} and $T_{Orientation}$ can be freely adjusted for each gesture individually within the framework and therefore allow various options for the design of an interactive gesture based system. For example a $T_{Orientation} = 360°$ value causes a gesture to be rotation invariant and the gesture is then detected by shape only regardless of the hand rotation. It was found that the threshold values $T_{Shape} = 0.05$ and $T_{Orientation} = 15°$ are a good compromise for recognition accuracy of many gestures. A simple algorithm for detecting a single static gesture among a predefined set of static gestures is described in Algorithm 2.

This algorithm is suitable for recognising the closest matching rotation invariant hand shape in a set of known gestures. In case of similar gestures inside the gesture storage, it is a design choice whether the system should take the gesture with the closest matching shape or the closest matching orientation of the hand. The closest matching hand orientation as the final decision to decide for a single gesture out of multiple similar gestures was chosen for the framework. Ultimately, the decision as to which factor (hand shape or orientation) is decisive is rather negligible. This is because the threshold for both factors can be chosen more strictly for each individual gesture. If a gesture is designed more strictly it will less likely have competing gestures with similar hand shape.

Algorithm 2 Recognising Static Gestures

```
New hand frame arrives
Get current hand frame $H_{current}$from hand tracker
for Each known gesture in the gesture storage do
    Set $D_{minimum}$ to Float.Maximum value
    for Each joint on the hand do
        Transform joint position from world space to local space
        Adjust joint position with hand scaling factor
        Calculate distance $D$ between stored joint position and current position
        if $D > Threshold$ then
            Discard current gesture (hand shape is not matching)
        end if
        Add $D$ to $D_{Sum}$
    end for
    if $D_{Sum} < D_{minimum}$ then
        Set $D_{minimum}$ to $D_{Sum}$ (store the smallest distance)
    end if
end for
if $G$ not discarded AND $D_{Sum}$ is the lowest between gestures then
    Current gesture is detected
end if
```

Dynamic Gestures

In the proposed framework a dynamic gesture consists of a static gesture and a predefined spatial path for hand joints. A gesture always starts by recognising a hand shape (the static gesture). After that, the spatial path of the joint positions must be reproduced in order to complete the gesture. A visualisation provides the user with information on how to move the hand accordingly. As mentioned in Section 5.2.4, the spatial path of hand joints such as finger tip positions can be recorded which must be imitated in order to complete a dynamic gesture. This path is reconstructed by placing invisible 3D points in the virtual environment once the attached static gesture is detected. These points are surrounded by a collider which allows them to be touched. These colliders are freely adjustable, which means that a gesture path must either be followed quite strictly or the user is allowed to deviate from it. Each 3D point of the path has a specific joint which it must collide with. Recording the path of the index finger will result in 3D points which must be touched by the index finger and not with any other joint such as thumb, pinky etc. In order to achieve this, each finger tip on the hand has an object attached to it (invisible to the user) which allows to distinguish which finger has touched which point.

As mentioned before, the points are implemented as a doubly linked list, i.e. each point knows its predecessor and successor. Furthermore, the first point in this list is always set to the world position of the joint it is supposed to be touched with. This will be done once the start gesture for the dynamic gesture is found, allowing the dynamic gesture to be performed. After a point is touched, it is marked as such, deactivated, and its successor is activated. If there is no successor left and the current point has been touched, the dynamic gesture was reproduced successful. This implementation provides a simple yet efficient method for a gesture to be replicated in the correct order. Other constraints are applied in order to ensure that the user imitates the desired gesture correctly. For example, once a dynamic gesture is detected, the first point of the spatial path will be enabled. The system knows which point should be touched next and therefore calculates the distance between the joint it should be touched with and the next point which should be touched. If the distance is too large to the next point, the gesture is considered failed. This test is done for each joint which was used to record the gesture.

Handling Similar Gestures

In a system with arbitrary one-handed gestures, the algorithm to recognise static gestures described in Algorithm 1 is not sufficient, since it can only recognise the closest matching gesture. In case a similar hand shape is required for multiple gestures, this algorithm will search for the best match and therefore only one gesture will be recognised. In a system with arbitrary hand gestures however, there could be many similar gestures which should be interpreted differently. A Swipe gesture for example can be performed by swiping left or right with the same hand shape. However, swiping left and right might be performed for different desired actions. In the proposed framework, each swipe with a different motion direction would be considered as a separate dynamic gesture, e.g. “Swipe Left” and “Swipe Right” would be recorded separately. To achieve this, the algorithm described in Section 5.2.5 is adjusted to allow multiple gestures to be activated at the same time:

- Instead of choosing the gesture with the lowest joint distance (closest match to the stored gestures), all gestures are marked as found that fit within the chosen thresholds.
- If there is more than one gesture found, check if there are gestures marked as ”Allow Similar Gestures”. If yes then allow all gestures marked as such to be performed. If visualisation is enabled, the user gets visual cues for the different motion directions the hand joints can follow.
- If there is more than one gesture found but no gesture is marked as ”Allow Similar Gestures” then only allow the closest match to be performed.

Further Refinement of Dynamic Gestures

To this point, gesture recognition can be refined by comparing stored information with information obtained from current frames. This includes hand shape, orientation, and the recorded dynamic hand path. This is not sufficient to cover arbitrary one-handed gestures. A ”pistol” gesture for example would require the hand to be still and the thumb should be moved in a specific way as shown in Figure 5.4 **A-C**. This can be leveraged by the user by following the expected thumb path with the whole hand instead of moving only the thumb (Figure 5.4 **D-F**). The framework allows for an option that forces the user to hold the hand still if a specific gesture should be conducted. If a dynamic gesture is detected and this option is enabled, the position of the hand where the gesture was first found is stored. If the distance of the hand is too far from its original point, the gesture will be cancelled. This refines the recognition in a way that undesired hand movements no longer fulfill a dynamic gesture path as depicted in Figure 5.4 **D-F**. The various refinement options provided by the framework are shown in Table 5.2

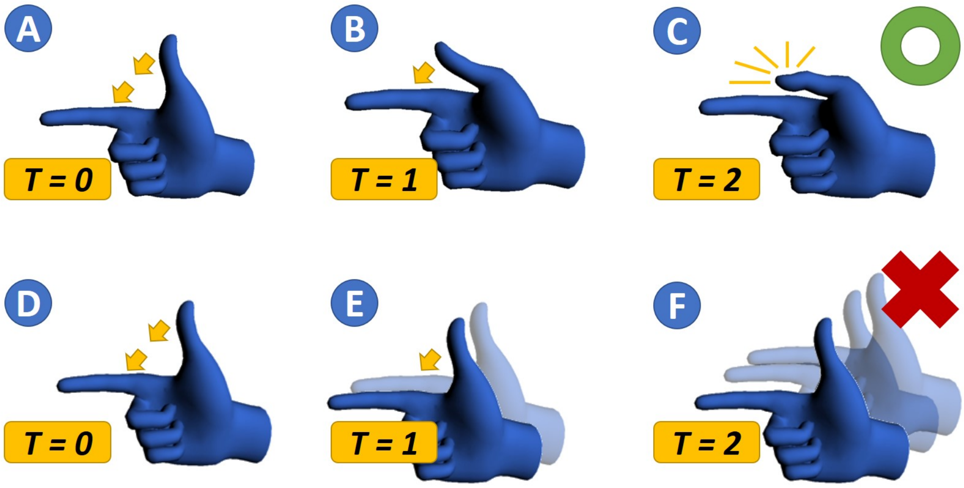

Figure 5.4.: *Images A-C show the desired behavior for doing a "pistol" gesture. D-F shows wrong movement which is still accepted by the system if no proper constraints are applied. In this case, the user follows the gesture path by moving the whole hand instead of only the thumb. This can be solved by applying constraints to the palm position of the hand.*

Table 5.2: Various options can be applied to refine gestures in order to implement desired behavior.

Refinement	Effect
Spatial Path	Requires the user to follow a predefined path with specific joints.
Hand Lock	In order to correctly perform the gesture the user is only allowed to move fingers (See Figure 5.4).
Allow Similar Gestures	Allows multiple dynamic gestures to be started with the same hand shape.

Continued on next page

Table 5.2: Various options can be applied to refine gestures in order to implement desired behavior. (Continued)

Refinement	Effect
Pose Threshold	A value that represents how close the current hand shape must match to a stored shape.
Orientation Threshold	A value that represents how strict the hand direction should be with respect to the stored values. A high chosen value causes a gesture to be rotation invariant and it will be detected regardless of the hand rotation.
Visualisation	This option enables a visual path which is shown once a dynamic gesture starts in order to help the user perform the gesture correctly.

5.2.6. Gesture Interpretation

A gesture in the proposed framework has various events which enables customisable interpretation. An event can be considered as a callback function that is called when certain conditions are met. These conditions include completing the gesture, cancelling the gesture, gesture was not executed correctly, and gesture progress.

Gesture Completed

This event is called once a gesture is considered as performed. For static gestures this simply means that the current hand frame data matched the stored static hand shape and orientation and therefore the gesture was recognised. Dynamic gestures are completed if all constraints attached to the gesture are met. These constraints include following the stored spatial path and others which were mentioned in Table 5.2.

Gesture Failed

A dynamic gesture can fail due to various reasons such as not following the spatial path, moving the hand too far away, performing a different static hand shape etc. Once the system detects that the user violated one of these constraints while a dynamic gesture is active, the framework triggers a callback which informs other components of the failed gesture and the reason. In the proposed framework a static gesture can not fail. This is because it is either being performed or not. Once the hand shape and orientation matches, a static gesture is already considered complete.

Gesture Progress

Dynamic gestures can notify other components about the percentage how far the gesture has been progressed. This value is coupled to the number of points which were stored during recording the gesture. More points in the stored spatial path will result in finer grained progress reporting by the gesture. Static gesture do not support progress.

Gesture Activated/Deactivated

Static gestures by default will trigger a Completed event for each hand frame it was found. If this behaviour is not desired, additional framework components can be attached to implement custom behavior. These components will listen to the Completed event and will trigger an Activation event once. As long as the Completed event is being triggered, no further Activation event will be send out by the component. If there was no Completed event for a certain amount of time (can be freely changed and the default value is 0.5 s) a Deactivation event is triggered. This is particularly useful to use static gestures to enable and disable certain elements in a virtual environment such as showing menus or enabling tools.

5.2.7. Combining Gestures

Gestures can be combined in order to further reflect desired behavior. As an example, a static gesture can be used to activate and deactivate a hand interaction tool while a dynamic gesture allows to trigger interactions while this tool is available. In Figure 5.5 a prototypical implementation of a selection tool is shown. The tool is activated by performing a static variant of the "pistol" gesture shown in Figure 5.4**A**. Once activated, a visible ray will be shown from finger tip to the direction the finger is pointing. Enabling this ray also enables the option to perform the dynamic variant of the "pistol" gesture (Figure 5.4 **A-C**) which will select an object the ray intersects with.

Besides this example, there are various other implementations possible with the proposed framework to combine and concatenate gestures which allows for truly arbitrary single hand gestures.

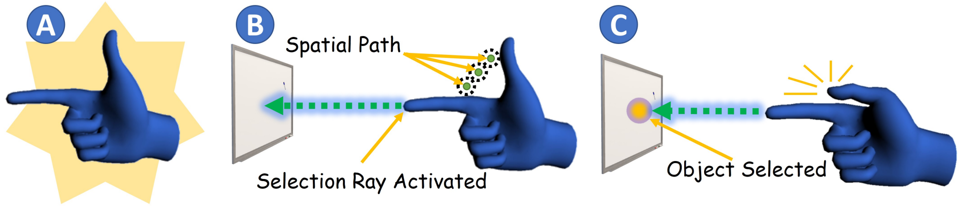

Figure 5.5.: *Exemplary implementation of a selection tool using the framework. A recognised static gesture (**A**) activates a visible ray and a dynamic gesture (**B**). If the dynamic gesture is performed by moving the thumbs down, an object is selected which intersects with the ray (**C**).*

5.3. Evaluating the Framework

5.3.1. Choosing Relevant Gestures for the Evaluation

Since the framework supports an arbitrary number of possible gestures, a subset of gestures must be chosen in order to evaluate the framework. It was decided to include a total number of 25 gestures for the evaluation. An overview of all gestures is given in Figure 5.6. The number of gestures lines up with some existing work that evaluates recognition of the American Sign Language alphabet which is usually about 24 distinct gestures. It is to note that the gestures which denote Z (D3) and J (D4) in the American Sign Language alphabet are usually not evaluated since the proposed solutions from authors are not able to recognise dynamic gestures. It was decided to include some gestures from the sign language alphabet (S1, S5, S15, D3, D4). Some gestures were added due to their complex hand configuration (S3, S12, S16, S17). Furthermore, some gestures were included because they are almost universally understood gestures such as swiping left and right (D5, D6) and thumbs up and down (S18, S19). Additionally, existing SDKs from hardware vendors usually provide a small set of gestures which were also included in the evaluation (S7, D1, D2, etc.).

Figure 5.6.: *The 25 gestures which were used in the evaluation. S1 - S19 are static gestures and D1-D6 are dynamic gestures.*

5.3.2. Choosing a Relevant Assessment

One difficulty in selecting an appropriate assessment is that the existing literature mostly uses labeled data sets which allows to easily calculate the accuracy of a proposed system within a certain data set. Thus it can be directly compared to other approaches. Since the framework proposed in this article does not use data sets and ground truth data for recognition, calculating the accuracy of gestures within a certain data set is not possible. During a first pilot study, it was observed that users often made a wrong gesture before actually conducting the desired gesture. For example, swiping left instead of right or a thumbs up instead of thumbs down. Usual quantitative analysis which can be represented in confusion matrices would be not accurate and therefore not applicable in this case.

For an appropriate assessment of the framework, it is assumed that gestures that are well recognised can also be recognised quickly. Therefore, it was decided to include a task that shows gestures and users should try to perform them as quickly as possible. If a gesture is not well recognised by the framework, it is reflected in the time taken between showing and recognising the gesture. To further investigate the robustness of the proposed approach, it was decided to include a second part to the experiment. The aim of this task is to investigate how the framework performs when multiple different gestures should be performed in a row. This is interesting for interfaces that require successive gestures to achieve a certain task. For example, a pointing gesture to select something followed by a thumbs up to confirm the selection. For that reason, it was decided to include an additional experiment which involves the user performing multiple successive gestures, namely gesture sets. All gestures from the previous experiment are active in the background (See Figure 5.6). The number of falsely recognised gestures within a gesture set is recorded. This metric gives insight how the proposed approach performs when many different gestures should be recognised.

5.3.3. Experiment Overview

Objectives

To evaluate the framework, a user study was conducted. The aim was to investigate the reliability of the gestures and whether the gestures can be imitated without difficulty. This is especially important since the gestures were recorded and designed from a single user which did not participate in the study. Many different gestures are performed by multiple users. Therefore, it can be investigated if gestures recorded from one user can be reproduced reliably by any other user. Furthermore, it shall be investigated how users feel about the gestures produced by the framework. This is a particularly useful insight as end users are often the decisive factor when it comes to selecting a gesture for a task. Subjective questionnaires have also been used in relevant previous work such as [168], [264].

Users

33 participants were recruited (18 Male, 15 Female). The participants age ranged between 19 and 61 years old ($\mu = 28.6$). The technology affinity of the users was assessed with the Affinity for Technology Interaction (ATI) questionnaire [65]. The mean score for affinity for technology interaction of the participants is 4.30.

Apparatus

The evaluation was performed by using a gaming notebook with an Intel Core I7-7820HK, 32 GB DDR4 RAM, Nvidia Geforce GTX 1080 running a 64 bit Windows 10. Hand tracking was realised using the Oculus Quest 2 VR HMD. No controllers were used.

Experimental Task

Participants had to perform gestures in front of a virtual screen while wearing a VR HMD (See Figure 5.7). After the shown gesture was recognised by the system, a visual confirmation was displayed for one second. Experiment part one required the user to imitate 25 different hand gestures which were shown one after another. The different gestures are shown in Figure 5.6. This procedure is repeated five times, i.e. each gesture was performed five times from each participant. The order of gestures was randomised for each repetition and participant. After five repetitions, experiment part two started. For that, ten gesture sets are to be performed by the user. A gesture set is a concatenation of 2-3 gestures which have to be performed in specific order. Gestures which were recognised but are not part of the current gesture set which needs to be performed are recorded as a false positive gesture. All 25 gestures from experiment part one are enabled in the system during this task.

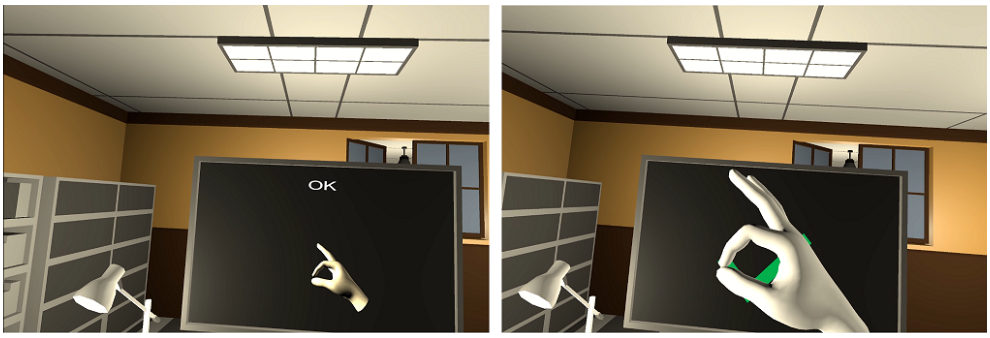

Figure 5.7.: *Experiment part one. A gesture is displayed to the participant (left) which should then be imitated (right). This test is to ensure that gestures can be recognised correctly.*

Procedure

Each trial session was conducted individually with the subject. Subjects were naive to the purpose of the experiment. The experimenter explained the trial session procedure, followed with handing out the informed consent. After that, the subjects put on the VR HMD and performed the experimental task. After the task was completed, a questionnaire for the perceived usefulness of the gestures is handed to the participant as well as the ATI questionnaire. The total execution time for one user session was about 30 minutes.

5.3.4. Results

125 gestures from experiment part one and 28 from experiment part two were performed by each participant. With 33 participants, this corresponds to a total number of 5049 gestures performed for the evaluation.

Quantitative Evaluation

Experiment Part One. As a first assessment of whether the gestures produced by the framework are useful, the time required to perform a gesture is measured. The time between displaying the gesture and recognising it is measured. The mean time to complete the gestures for all participants is depicted in Figure 5.8. The time until recognition is particularly interesting to find out which gestures were not easy to imitate by users. For example, gesture S17 took longer than most of the other gestures. This might be an indication that gesture recognition configuration was too strict for this particular gesture and could be loosened in order to improve recognition time. A trend that gestures are recognised more quickly with the fifth repetition is noticeable as depicted in Figure 5.8. The figure shows graphs for the first, last, and average time for five repetitions. Overall, most static gestures were recognised about one second after being displayed to the user in the fifth repetition.

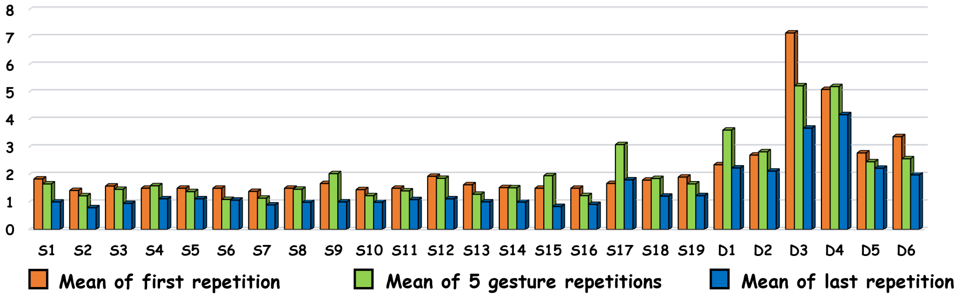

Figure 5.8.: *Mean time required by the subjects to perform the individual gestures. The time given refers to the moment when the gesture was recognised by the system after it was displayed to the participant. Mean time for the first time a gesture was performed in orange, five repetitions of a single gesture is shown in green, and the last repetition in blue.*

Experiment Part Two. In a second evaluation phase, the number of times a gesture was falsely recognised is assessed. Participants had to perform gesture sets consisting of two to three gestures which should be performed in a row (See Figure 5.9). The results are useful in two ways: First, it can be assessed how many false positive gestures are in between each gesture set. Secondly, the frameworks capability to produce gestures which can be concatenated. Ten gesture sets are shown consecutively to the user. They are displayed on a virtual screen with visual feedback once a gesture was recognised.

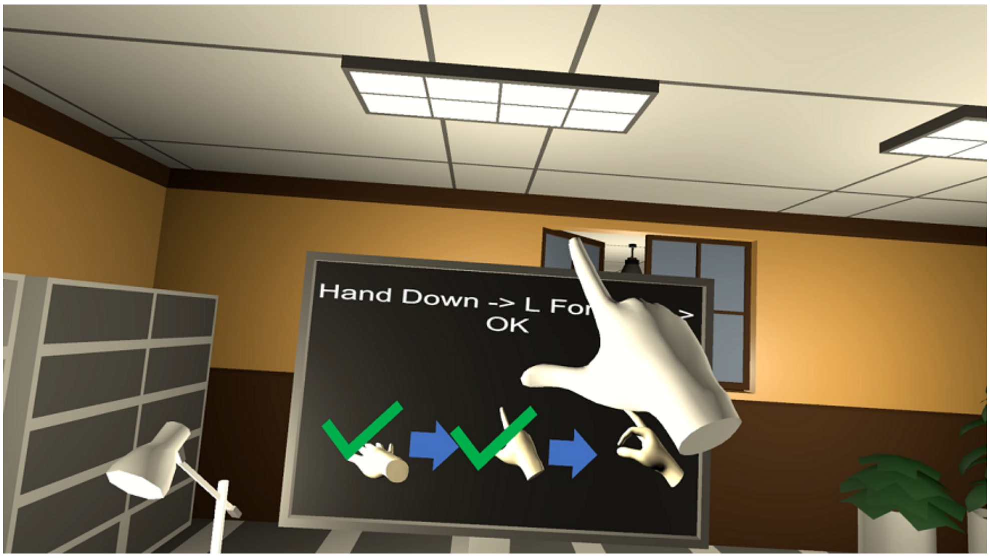

Figure 5.9.: *Experiment part two. A gesture set is displayed to the participant which should then be imitated. The aim is to test whether gestures can be concatenated easily and if false positives are detected.*

For the ten gesture sets, a total of 28 gestures are to be performed by each participant. This corresponds to a total of 924 performed gestures by participants in this phase. A total number of 51 false positive gestures were recorded during the second phase. This leads to 5.52% of falsely recognised gestures between gesture sets. The results in the second evaluation phase should be treated with caution. During the evaluation it was found that the test persons often made a wrong gesture because they tried to solve the task as quickly as possible. For example, a “Swipe Left” was often performed until the test persons realised that a “Swipe Right” should actually be performed. Nevertheless, there are some false positive gestures, for example gesture S1 and S12, because they are quite similar. After subtracting the supposedly correctly recognised but incorrectly performed gestures, about 2% false positive gestures remain. Out of 330 gesture sets performed, seven gesture sets were interrupted with falsely recognised gestures in between.

Qualitative Evaluation

A questionnaire was handed out to the participants to ask for their subjective opinion of the system. The results of the questionnaire are shown in Figure 5.10. Three questions were asked:

- **Q1** The system was able to recognise the gestures I made.
- **Q2** Dynamic gestures like Z or SWIPE were easy to perform.
- **Q3** I had the feeling that the gestures I made were immediately recognised.

Participants were overall very satisfied according to the questionnaire. Most users gave either the best possible rating or the second best.

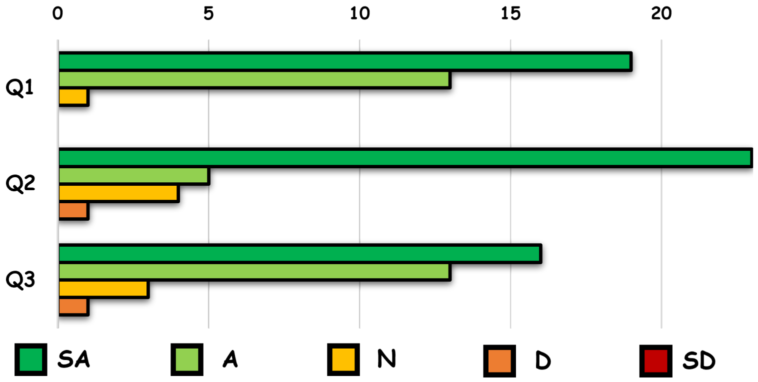

Figure 5.10.: *Answers from participants to the given questionnaire. SA = Strongly Agree, A = Agree, N = Neutral, D = Disagree, SD = Strongly Disagree.*

5.4. Experiment Discussion

The system implemented with the proposed framework got positive ratings from the participants. The lowest scores in the questionnaire also match with the highest time required to recognise some gestures. For example, a participant gave neutral rating for Q2 because he had difficulties replicating the Z gesture. Another participant had difficulties performing S9 and gave neutral rating to Q1. As shown in Figure 5.8, most static gestures were detected about one second after being displayed to the participant. Dynamic gestures require longer time to complete than static gestures which is due to the fact that there is hand motion involved. Furthermore, in order for the participant to understand which gesture to perform, a short video is displayed which needs to be understood first.

Additionally, some recognition times are impacted by the hand configuration itself. For example, some subjects had problems shaping the hand properly due to either congenital characteristics of the hand or previous injuries (especially S3, S10, and S17). Along with the settings that a gesture had to correspond quite closely to the recorded gesture, this resulted in overall longer recognition time. However, all participants were able to perform the desired gestures and the questionnaire results are overall very positive. The measurements from the first part of the experiment helped to reveal some potential problems with certain gestures. S1 for example took a surprisingly long time to be recognised. The participants had problems turning the hand exactly as it was shown to them in the picture.

The proposed implementation shows promising results for recognising one-handed gestures. To the best of our knowledge, there is no one-handed gesture that cannot be designed, performed, and recognised with the framework. In addition, recording static and dynamic gestures is as simple as performing the desired gesture and subsequently pressing a key. The simplicity of the framework was confirmed when the gestures for the experiment were defined. It took no more than five minutes to design and implement the 25 gestures used in the evaluation. The framework can be used to implement various applications and is not bound to specific hardware. In order to support other hand tracking devices, the hand data provider has to be adjusted to support the target hand tracking SDK.

5.4.1. Limitations and Future Work

Although both hands can be used to recognise gestures, each hand has it's own gesture recognition system attached to it. This means that each hand can perform gestures individually but do not take the other hand into account. Allowing gestures that involves monitoring both hands simultaneously is planned future work. Furthermore, the proposed framework is heavily dependent on the accuracy of the hand tracking data which is fed into the feature extraction component. A highly accurate hand tracking will result in better gesture recording/recognition while a poorly performing hand tracking device leads to corresponding results. In addition to that, a gesture recognition system should not only use the hands but other signals of the human body such as eye gaze, limb movement, and face mimic. Integrating more input modalities is part of future work. A subset of 25 hand gestures was evaluated. The gestures were chosen in order to include some similar and some rather complex hand configurations. Many more gestures are possible with the proposed technique, but the experiments were conducted in order to ensure that the framework works as intended.

As part of future work it is planned to implement various applications and scenarios which use one-handed gestures. These applications are then evaluated with the focus on the usefulness of specific gestures. Furthermore, automatic detection of similar gestures should be explored in order to eliminate gestures with a high false positive rate. Additionally, automatic classification of recorded gestures would further help gesture designers to record gestures. In the current system, the gesture designer has to manually choose whether to record a static or dynamic gesture. It is also necessary to select which joints of the hand are to be tracked to form a dynamic gesture.

5.5. Summary

In this chapter, a framework for design and implementation of one-handed gestures is proposed. It can be used to implement natural user interactions within a wide variety of immersive XR and desktop applications. Existing hand tracking solutions serve as an entry point to the framework and the necessary data is redistributed to internal components. These components are responsible for recording, storing, recognition, and interpretation of gestures. The detected gestures can be integrated into applications by using events such as gesture completion, activation, deactivation, failed, and progress.

The goals of the framework are a simple gesture creation process and the recognition of any one-handed gesture. The evaluation confirms that the main objectives of the framework have been achieved. Static and dynamic gestures can be recorded by performing the desired gesture and subsequently pressing a key. For static gestures, hand shape and orientation is stored. Dynamic gestures can be recorded by monitoring joint positions and storing the spatial differences into a doubly linked list. To allow a wide range of complex gestures, several solutions are proposed to enhance and improve gesture recognition. Furthermore, a system to evaluate the framework was built. A user study with 33 participants was conducted and gestures designed with the framework were evaluated. Overall, the system showed promising results. Gestures designed and implemented by a single person can be reliably reproduced by others without the need for elaborate data sets to define new gestures.

The core of this work is the ease of adding new gestures which are reliably recognised and directly applicable. By not requiring data sets, training, or expert knowledge to implement new gestures, it allows integration into a variety of applications. Even complicated gestures can be easily recorded and tested for usability. The multitude of possible gestures is only limited by the creativity of the users. The framework aims to encourage researchers to create gesture-based applications and then examine them for usability, feasibility, and other measures. It paves the way for systems to design, prototype, evaluate, and support arbitrary gestures.

Chapter 6

Use of Hand Gestures for Locomotion in Immersive Applications

VR technology offers users the possibility to immerse and freely navigate through virtual worlds. An important component to achieve a high degree of immersion in VR is locomotion. Often discussed in the literature, a natural and effective way of controlling locomotion is still a general problem which needs to be solved. As HMD manufacturers have integrated more and more sensors, hand or eye tracking is possible without additional equipment. This allows a wide range of application scenarios with natural interaction techniques. This chapter focuses on techniques to control locomotion with hand gestures, where users are able to move around in VR using their hands only.

In general, there are two ways of moving around in virtual reality: Instantaneous and continuous movement. Instantaneous movement is often called teleportation and allows the user to move from point A to B instantly without movement in-between. On the other hand there is continuous movement which lets the user to move gradually from point A to B (See Figure 6.1). This chapter explores how hand gestures can be used for both methods. Section 6.1 introduces four bare handed techniques to teleport while section 6.3 proposes three techniques to move gradually in VR. The techniques were evaluated in a user centric manner.

6.1. Teleportation-Based Locomotion using Bare Hands

It is possible that natural user interfaces like freehand gestures in VR will be a built-in standard. Future VR software should therefore be usable just with an HMD and without additional hardware. Controlling locomotion is an essential activity in virtual environments and it should thus be possible to perform it not only with controllers but also with hand gestures. In this section the term locomotion technique is used to describe a way of controlling

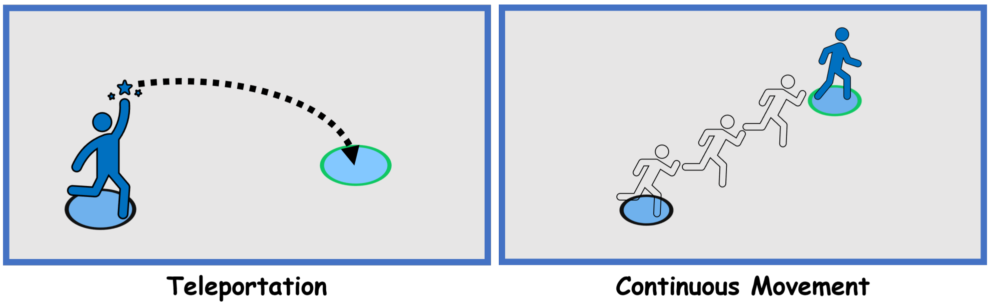

Figure 6.1.: *There are two ways of movement within VR: Teleportation (instant movement) and continuous (gradual movement).*

movement in VR. Locomotion using hand gestures should be easy to use and accessible to many users. Techniques for controlling locomotion with freehand gestures are found in several varieties in the literature, including techniques which use jumping, body leaning, hand gestures, and more. The proposed locomotion techniques in this section focus purely on hand gestures. One objective of this work is to show that hand gestures can provide a natural and effective way to control movement in VR. Moreover, locomotion based on hand gestures has the advantage that it can be performed while the user is sitting or standing with little physical activity. This work investigates whether two hands are required for control or if a single hand is sufficient. In addition, the advantages and disadvantages which are associated with different types of hand gestures should be investigated. The aforementioned points lead to the following research questions:

- **RQ1:** Can users easily navigate through a virtual environment by using hand gestures?
- **RQ2:** Can users move through virtual environments efficiently and effectively with just one hand?
- **RQ3:** Do users prefer controlling locomotion with one hand rather than both hands?

A unique system to help uncover the answers to those questions is implemented. The system is able to recognise a wide range of previously captured static hand gestures. It is used as a tool to rapidly design, implement, and evaluate locomotion techniques in VR. Four hand gesture based locomotion techniques which were designed and implemented using this system are proposed. Two two-handed and two one-handed approaches are evaluated in a user study with 21 participants. Since the existing literature already suggests that hand gesture based locomotion techniques are efficient and useful [21], [22], [30], it was decided to not include an equipment-based method to the evaluation in order to focus on purely hand gesture based application scenarios. In addition, newer VR HMDs have already incorporated hand tracking technologies in their hardware [61], [163], [184] turning controllers into an optional accessory.

6.1.1. Input Modalities for Moving in VR

Several techniques for moving in VR have been proposed by researchers. Some of these techniques involve rather large and demanding body movements such as the well established technique Walking in place (WIP). To move virtually with this technique, users perform footsteps on a fixed spatial position in the real world. This technique is already widely explored and a large body of existing work can be found in the literature. Templeman *et al.* [277] attached sensors to knees and the soles of the feet to detect the movements, which are then transmitted to the virtual world. Bruno *et al.* [26] created a variant for WIP, Speed-Amplitude-Supported WIP which allows users to control their virtual speed by footstep amplitude and speed metrics.

Another technique for moving without controllers is leaning. This technique uses leaning forward for acceleration of the virtual avatar. Buttussi and Chittaro [29] compared continuous movement with controller, teleportation with controller, and continuous movement with leaning. Leaning performed slightly worse compared to the other techniques. Langbehn *et al.* [130] combined WIP with leaning where the movement speed of the WIP technique is controlled by the leaning angle of the user. Different techniques for leaning and controller based locomotion was evaluated by Zielasko *et al.* [324]. The authors suggest that torso-directed leaning performs better than gaze-directed or virtual-body-directed leaning. In another work, Zielasko *et al.* [323] compared leaning, seated WIP, head shaking, accelerator pedal, and gamepad to each other. A major finding of their study is that WIP is not recommended for seated locomotion. A method that pairs well with WIP is redirected walking. With this method, the virtual space is changed so that the user needs as little physical space as possible. Different techniques exist to achieve this, for example manipulating the rotation gains of the VR HMD [78] or foldable spaces by Han *et al.* [86]. More redirected walking techniques are found in the survey from Nilsson *et al.* [180]. Since a large body of research work exists around locomotion in virtual reality, the reader is referred to surveys such as [1], [12], [36], [149], [179], [242] to gain more information about different locomotion techniques and taxonomies. Physical movement coupled with virtual movement offers more immersion, but hand gesture-based locomotion is expected to be a less strenuous and demanding form of locomotion than those mentioned above. Furthermore it is a technique that requires minimal physical space and can be used in seated position as well as standing.

Motion of the human body can be transformed and transmitted to the virtual environment by using sensors attached to the body or camera based tracking devices. With these approaches, users move in VR by physically walking around, moving limbs, leaning forward, etc. Other approaches rely on specific interaction devices to enable locomotion in VR such as the well known VR controller. More complex devices can be found in the literature such as treadmills, gloves, or special chairs. These approaches can be summarised in two major categories:

Moving with real motion:

- Physical Walking: Users walk around in physical space as they move in VR with usually just an HMD as input to the system. The user's physical movements are typically redirected by the virtual environment to compensate for limited physical space. These techniques are usually known as redirected walking [6], [15], [34], [43], [55], [127], [132], [133], [159], [166], [239], [250], [265], [283], [314].
- Stationary Movement: Such methods utilise a system where users can move in VR by moving their limbs in restricted physical space (the user is stationary in sitting or standing position). The walking in place method, in which users stand or sit in place and move their legs, falls into this category [54], [87], [91], [112], [309]. Wolf *et al.* [311] implemented a locomotion technique where users have to perform real jumps to move in VR. The jump is recognised by monitoring the inertia of the VR HMD. The work of Zielasko *et al.* [322], [324] included leaning as effective locomotion technique. This approach uses a VR HMD and a tracking device attached to the torso to detect body leaning and move accordingly in the virtual environment.
- Hand Gestures: This type of input modality relies solely on hand gestures to control movement in VR [21], [22], [30]. The proposed techniques in this chapter fall under this category.

For extensive research regarding locomotion based on real motion the reader is referred to the survey conducted in 2019 by Cardoso et. al [37].

Moving with interaction device:

- VR Controller/Gamepad: Using hand-held hardware for locomotion is the common standard solution. The user moves by pressing a button (usually the grip button) on a VR controller [10], [84], [117], [154], [308]. Some systems use a non VR controller such as a gamepad or joystick and users can move with the thumb sticks or pressing a button [35].
- Special Input Devices: Prototypes and other devices such as treadmills [304], specialized shoes [115] or chairs are used as locomotion input in this method. Some work also uses touch devices such as smartphones or tablets in combination with VR HMDs [158]. Englmeier *et al.* [59] used a handheld spherical device for locomotion. The rotation of the device is translated to first-person movement in VR.

A survey conducted in 2017 by Boletsis [13] reviewed the literature on locomotion techniques from 2014 to 2017. This survey included only seven gesture-based locomotion techniques, with the vast majority of Walking In Place (17 papers), Redirected Walking (17 papers), and Controller/Joystick (15 papers) based techniques being found in the literature.

To assess the current state of research on hand gesture based locomotion, a superficial search on locomotion techniques published in 2018–2020 was conducted. The search was performed by using related search queries in different data sources such as Google Scholar[1], ACM Digital Library[2] and IEEE Xplore[3]. 39 papers were included that had a main contribution on proposing new locomotion techniques. In Figure 6.2 the distribution of the individual works is shown, which is based on the previously mentioned categories. An overwhelming amount of research was conducted in physical walking techniques. Although some works mention walking as the gold standard for navigation in VR research [178], a full-body based locomotion approach is often infeasible and space-demanding. Hand gesture based locomotion has the advantage that no additional hardware is required, it can be used in limited spaces, and still provides a natural method of interaction.

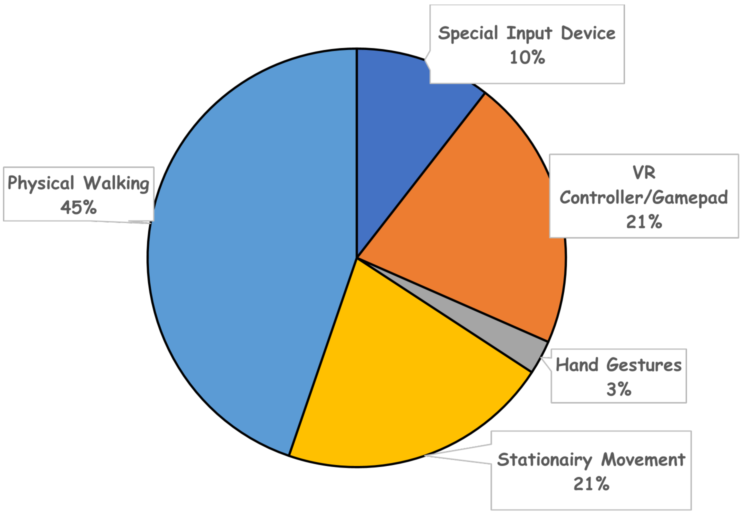

Figure 6.2.: *Locomotion techniques used in research papers in the years 2018–2020.*

Controlling Locomotion in VR through Hand Gestures

Early work on how hand gestures can be used for virtual locomotion was conducted by Kim *et al.* [123], [124]. The authors presented Finger Walking in Place (FWIP), which enables virtual locomotion through the metaphor of walking triggered by finger movements. Huang *et al.* [102] used finger gestures to control movement within virtual environments. The gestures are used to control the velocity of moving forward and backwards. Four different locomotion techniques are proposed by Ferracani *et al.* [62]. The techniques are WIP, Arm Swing, Tap, and Push. Tap uses index finger pointing and Push involves closing and opening the hand. The authors conclude that the bare

---

[1] `https://scholar.google.com/`
[2] `https://dl.acm.org/`
[3] `https://ieeexplore.ieee.org/`

handed technique Tap even outperformed the well established WIP technique. Zhang *et al.* [318] proposes a technique to use both hands for locomotion. The left hand is used to start and stop movement while the right hand uses the thumb to turn left and right. Cardoso [35] used hand gestures with both hands as well for a locomotion task. Movement was controlled by opening/closing both hands, speed was controlled by the number of stretched fingers, and the rotation of the avatar was mapped to the tilt angle of the right hand. The authors concluded that the hand-tracking based technique outperformed an eye gaze based technique but was inferior to a gamepad. Hand gestures were also used by Caggianese *et al.* [31] in combination with a navigation widget. Users had to press a button to move through a virtual environment whereas with the proposed technique, users can move by performing a hand gesture. In subsequent work, Caggianese *et al.* [30] compared three freehand and a controller based locomotion technique. In their experiment, participants had to follow a predefined path. The authors show that freehand steering techniques using hand gestures have comparable results to controller. While Caggianese *et al.* [30] uses hand gestures to start/stop movement, this work compares two techniques with a 3D graphical user interface, a one-handed gesture to start movement. The direction of movement was also tied to the direction of the hand, whereas in this work the direction of movement is tied to the direction of the user's VR HMD. Bozgeyikli *et al.* [20], [23] compared Joystick, Point and Teleport, and WIP. The results showed that the hand gesture based teleportation technique is intuitive, easy to use, and fun.

6.1.2. Teleport System and Locomotion Gesture Design

General Overview

To evaluate hand gestures for controlling locomotion, a dedicated system was implemented using inexpensive and widely available hardware. By implementing this system, all techniques use the same internal algorithms for locomotion and gesture recognition, allowing a reliable comparison. Teleportation-based movement was chosen since it provides less motion sickness as compared to other movements such as steering/driving as Christou and Aristidou [46] found in their study. One of the goals is to enable locomotion with limited physical space and therefore the system was built with the intent of being used in a seated position. A seated position has the advantage of being more accessible to the elderly or people with walking disabilities. The locomotion techniques can be used regardless of the user's posture, but have been designed and tested for use in a seated position. Another goal is to rely solely on hand gestures to control locomotion in order to explore its usefulness. Different gesture types should be investigated and then compared to find out which gestures will work best for the user. To find suitable gestures, the system should be able to easily recognise different types of gestures and then enable testing directly in VR. To summarise the aforementioned points, the following requirements are formed to build the proposed system:

- Locomotion through virtual environments by means of teleportation.

- No controller or additional hand-held hardware should be used to interact and navigate through virtual environments.
- The system should provide easy access to underlying data in user sessions such as task completion time, number of teleportations, number of times the hand tracking failed and overall reduce the time necessary to analyse and evaluate trial sessions.
- Researchers should be able to define and replace their own locomotion gestures during runtime and assess how they perform

The proposed implementation was built using a hand tracking device which provides accurate hand pose estimations [306]. The general concept and implementation can be transferred to other hardware and frameworks that provide hand pose information. The system consists of four major components: gesture capturing, teleport pointers, VR user interface, and evaluation system.

In the gesture capturing component, hand gestures can be extracted during runtime of an application. These gestures can then be used to realise any form of interaction. However, in this section the focus is on teleportation-based locomotion and therefore gestures are used for activating a teleportation mode and to perform the teleportation itself.

Teleport pointers are used to provide users visual feedback for selecting a teleport destination. In the proposed system either a straight or a curved ray is provided which can be aimed via hand movements. The origin of the ray can be adjusted with the VR user interface and supports either the index finger tip or palm position. The VR user interface provides the possibility to change different teleportation options during runtime of the application. The ability to choose between teleport pointer appearance, origin, and capturing gestures is provided in this interface. This part of the system is only used by researchers to find suitable gestures for locomotion and is not part of the evaluation.

The evaluation system consists of a VR parkour environment which can be used to evaluate gestures. A detailed explanation of the VR parkour environment used for evaluation is shown in section 6.1.3. A second part of the evaluation system is not visible to the user but gives researchers the ability to easily record and log the decisions made by users during a trial session such as teleport frequency, location, distance, number of times the hand tracking was lost, the time a virtual object was touched etc. This system is used for quantitative evaluation of the proposed gestures (see section 6.1.4).

Activation, Teleportation, Pointer Ray

The teleportation system considers three possible stages a user is currently performing:

1. No teleport desired, user is currently interacting with the environment.
2. User wants to teleport and is deciding on a new position in the virtual environment.

3. The user chose a position to teleport and wants to teleport (activate teleport).

Differentiating between those stages plays an important role in the implementation of a system that is intended to handle many different hand gestures. During previous pilot testing, many participants teleported accidentally. While interacting with buttons or looking around, hand gestures were accidentally made which triggered the teleport activation conditions. Since the system allows arbitrary gestures, a button press can be similar to a teleport or activation gesture. This led to the development of a “safety system” which prevents accidental movement. The first improvement deactivated the ability to teleport if the hand is near interactable objects. Next, a timer which will start upon successful teleport is implemented to prevent undesired fast successive teleportations. Furthermore, teleportation is only possible if the teleportation mode is active. This feature will visually highlight the hands, show the teleportation ray, and enable navigation through the virtual environment.

Capturing Static Hand Gestures

A virtual hand representation usually consists of 21 points (such as provided by the software development kits [101] or [284]), 16 of which represent joint positions and 5 represent the finger tips (i.e., end joints). The presented system builds upon a hand tracking device which uses a hand skeleton with 21 points, each having its position relative to the hand tracking device. A gesture capture system is proposed that relies on finger state and palm direction as gesture descriptors which can reliably recognise gestures.

The system monitors the state of the fingers, distinguishing two states: stretched or curled. The states of each individual finger of the hand thus result in a clear descriptor for a gesture. Fore example, a fist is recognised when all fingers are curled. A pointing gesture is detected when the index finger is in the stretched state while the remaining fingers are in the curled state. With the finger state as descriptor it is possible to detect certain hand postures, but it is not yet possible to detect a variety of more complex gestures.

A “thumbs up” gesture for example, depends not only on the thumb being stretched out, but also on the direction in which the thumb as well as the hand is facing. Therefore the finger state descriptor alone is not sufficient. The orientation of the hand provides information about which direction the hand is facing as well as the implicit information in which the individual fingers point. For this reason hand direction was added as a descriptor to the gesture recognition system and found that it is well suited for this purpose. Since the raw directional value of the hand is too restrictive, a tolerance value is added which allows the system to activate the gesture even if the hand direction is not identical (but very similar) to the previously captured.

By combining the two descriptors, the system is able to recognise a wide range of static hand gestures. Researchers are able to perform gesture capture events during runtime and can rapidly prototype any combination of hand gestures to control locomotion. The gesture capturing process is shown in

Figure 6.3. The gestures can be used to activate the teleportation mode or directly as teleportation gesture.

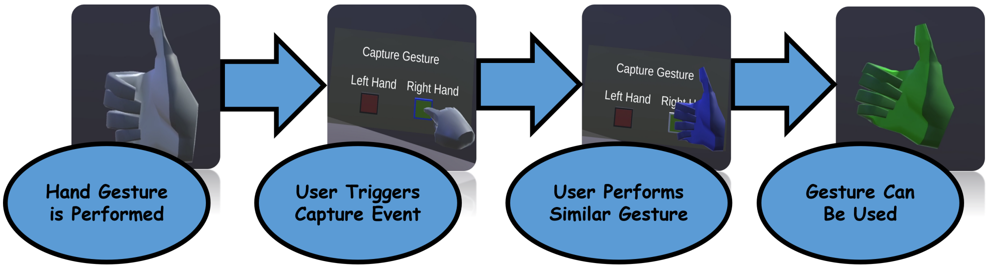

Figure 6.3.: *The user of the system can capture and test gestures for later use. In the depicted case, a button is pressed to start gesture capturing. The user then has a certain amount of time to perform the desired gesture. After capturing, the gesture can be used within the virtual environment.*

Example Gestures

Using the described system, four locomotion techniques were implemented which are evaluated in this work. The perhaps most natural hand gesture to show where you want to move is the pointing finger gesture. For this reason, pointing as a hand gesture to control locomotion was included. In addition, palm gestures are studied to select a location for locomotion in VR. The palm as a navigation gesture had previously been shown to be effective in an empirical evaluation during the development of the system. Furthermore, two-handed and one-handed gestures are investigated. While two-handed techniques require both hands, one-handed techniques can be performed with either the left or right hand. The two-handed techniques use the right hand for navigation and the left hand to perform the teleport. In this study, a pointing gesture is used to perform the teleport. This means that the user selects the position with the right hand and points with the left hand to confirm the movement. Moreover, the two-handed approaches require a dedicated activation gesture that enables the teleport functionality. This activation gesture will color the virtual hands green to give visual feedback to the user. This technique is intended to provide the user with precise and accurate control while moving through the virtual world. On the other hand, one-handed methods use an algorithm that allows the user to move with only one hand. These techniques are detailed in the following sub sections.

TwoHandIndex: Two-Handed Approach Using Index Finger Navigation with Active Teleportation Gesture In this method, the right hand is used to choose the position a user wants to teleport to by casting out a visible ray from the right index finger (see Figure 6.4). The left hand performs a pointing gesture to conduct the teleport. The users can choose a location with the right hand, while the left hand can repeat the gesture by curling and stretching the index finger. This allows rapid teleportation with minimal physical effort. This method requires a dedicated gesture to activate the teleportation mode. In this case, the right hand needs to be turned upside down with all five fingers stretched (opening the hand). Once activated, locomotion control with this method is enabled. If a hand leaves the field of view (FOV) of the hand tracking sensor, the teleportation mode is deactivated until both hands are visible again.

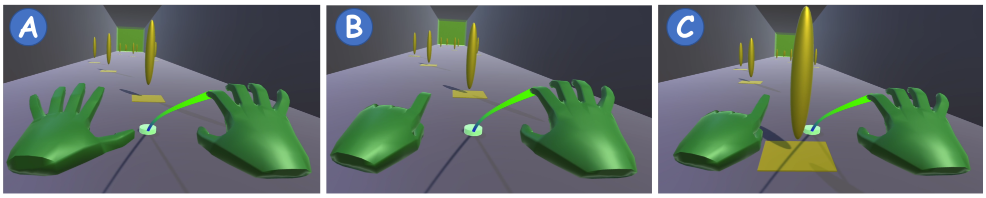

Figure 6.4.: *(**A**) Index finger of right hand points to desired position (**B**) Left hand performs teleportation gesture (**C**) User moved to position.*

TwoHandPalm: Two-Handed Approach Using Palm Navigation with Active Teleportation Gesture Similar to TwoHandIndex, the right hand is used for choosing the desired position and the left activates the teleport (see Figure 6.5). The only difference to TwoHandIndex is the ray origin, which is casted out of the palm instead of the index finger. The teleport activation gesture is the same.

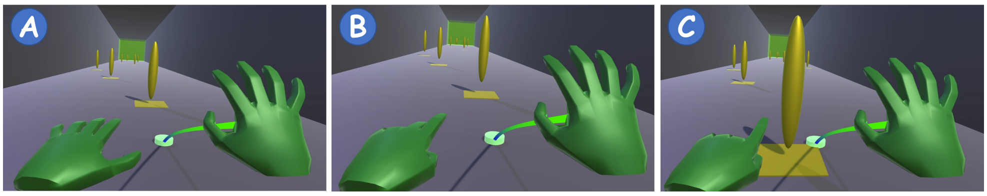

Figure 6.5.: *(**A**) Palm normal of right hand points to desired position (**B**) Left hand performs teleportation gesture (**C**) User moved to position.*

OneHandIndex: One-Handed Approach Using Index Finger Navigation with Passive Teleportation Gesture This teleportation method is a one-handed approach which can be used with either left or right hand. Unlike TwoHandIndex, this technique requires only the dominant hand to be in the FOV of the hand tracking sensor. The index finger needs to be stretched and all other fingers curled (pointing gesture) as seen in Figure 6.6. The direction of the index finger tip must point forward. For this method we implement a subsystem called velocity teleport. With velocity teleport, the velocity of the gesture performing hand should not go above a certain threshold. A timer with $n = 1.5$ s is started, once the gesture is detected. If the velocity of the hand does not go above a certain threshold during the timer interval, a teleport will be activated. If the gesture is no longer performed during the timer interval, the locomotion attempt will be cancelled. After a teleport was performed, or the hand velocity goes above the threshold while the gesture is being performed, the timer is restarted. This means that the user points to a location and then tries to hold the position of the hand for a certain amount of time, which will then perform a teleport to the pointed position. While the hand performs a gesture and is held still, the user receives visible feedback in the form of a change in color (shown in Figure 6.6B) of the hand to indicate that a teleport is imminent.

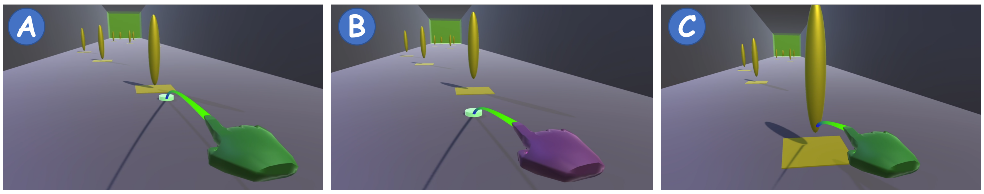

Figure 6.6.: *(**A**) Index finger of right hand points to desired position (**B**) Hand is kept still for* $n = 1.5$ *s (**C**) User moved to position.*

OneHandPalm: One-Handed Approach Using Palm Navigation with Passive Teleportation Gesture Similar to TwoHandIndex, only one hand is used to navigate through the virtual environment. Instead of a pointing gesture with the index finger, this method uses the palm for choosing a teleport position (see Figure 6.7). After the hand is held still for 1.5 s, a teleport is performed.

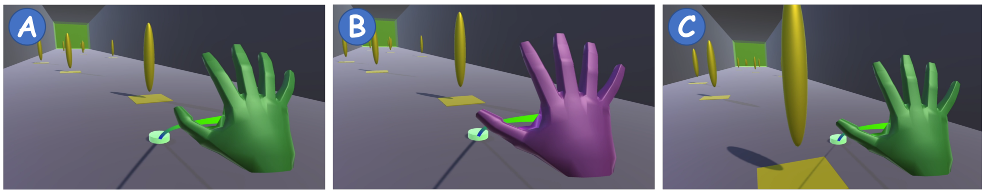

Figure 6.7.: *(**A**) Palm normal of right hand points to desired position (**B**) Hand is kept still for* $n = 1.5$ *s (**C**) User moved to position.*

6.1.3. Evaluation

Objectives

Four different methods of controlling locomotion in VR are examined for their applicability. One-handed methods were included to determine if they offer a viable way of controlling movement. The main objective of the evaluation is to answer the research questions stated in section 6.1.

A comprehensive testbed evaluation as described by Bowman et al. [17], [18] was conducted. The efficiency of the different methods is measured by the task completion time and the effectiveness is measured by the required number of teleportations and the amount of hand tracking failures. Additionally, well known evaluation questionnaires are used such as the System Usability Scale (SUS) [24], [25] which provides subjectively perceived usability of a system and the NASA Task Load Index (NASA-TLX) [88], [89] which allows a measurement of the perceived workload. By combining the quantitative measures (efficiency + effectivity) with the perceived usability (SUS) and workload (NASA-TLX) comprehensible conclusions about the overall usefulness of the proposed techniques can be drawn.

Participants

For the study 21 volunteers (15 Male, 6 Female) were recruited. The participants' age ranged between 25 and 60 years old (M = 35.4, Median = 31). All participants were right handed. Using a 5-point Likert-scale, where 1 denotes less knowledge and 5 expert knowledge, 81% of users think they have good general knowledge in software and computer (they answered with 4 or 5 in the questionnaire). Using the same procedure, asking for the VR experience, about 86% of users have never worn a VR HMD before and the remaining 14% used a VR headset at least once.

Apparatus

The evaluation was performed by using a gaming notebook with an Intel Core I7-7820HK, 32 GB DDR4 RAM, Nvidia Geforce GTX 1080 running a 64 bit Windows 10. The hand tracking is realised by using the Leap Motion Controller. The hand tracker uses two infrared cameras in combination with infrared LEDs to detect and trace the user's hands. The device performs a short-distance tracking with a range of about 25 to 600 mm and has 150° FOV. The Samsung Odyssey+ was used as the VR HMD, which has a a 1440 × 1600 pixel resolution per eye with 90 Hz refresh rate and 110° as FOV. The rotational and positional tracking on 6 degrees of freedom (DOF) is realised by using inside-out tracking. Inside-out tracking requires no additional external sensors and the tracking algorithms use two cameras built into the headset. For the evaluation, only the VR HMD and no controllers are used.

Experimental Task

The task of the participants was to touch ten pillar-like objects in a virtual environment (VE). The VE was kept as minimal as possible, using a primitive graphics style with no complex objects to reduce the "wow-effect" for users who never had used a VR HMD before. Users had to navigate through a large corridor, 10 m high, 10 m broad and 100 m long with no additional obstacles other than the touchable pillar-like objects (see Figure 6.8).

The pillars were placed with a distance of about 10 m to each other, ten in total. The user's locomotion was limited to about 6 m per teleport, requiring them to make at least two locomotion attempts to reach from one pillar to the next. The pillars are placed in a way that forces a redirection of locomotion. To avoid accidental movement into the virtual objects, a visible plane was placed under each pillar which does not allow moving on top of it. Users are allowed to move through pillars, but the pillar itself blocks the teleportation ray, thus requiring users to steer around it if one is too close.

The task is completed once a user touched all pillars in the VE. Touching the objects will give visual feedback to the user by changing its color to green. If a pillar is missed during the task, the user is required to go backwards and touch it. Participants were shown a video for each locomotion technique in form of a flying billboard. This billboard could be activated or deactivated at any time with a button that appeared when the left palm was facing the face. The experiment was conducted in seating position.

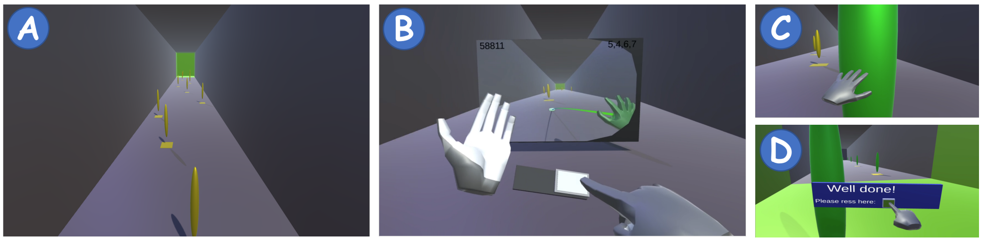

Figure 6.8.: *The virtual environment used for the evaluation employs a minimalist rendering style to avoid unwanted changes in subjects' attention. (**A**) Overview of the environment (**B**) Tutorial video shown to the user. The video can be closed by pushing a button attached to the left hand (**C**) User touching a pillar and change its color. (**D**) User gets notification that the level is completed after touching all pillars.*

Procedure

The study was conducted with 21 participants, each session was performed individually with the subject. The following steps were repeated for each subject: (i) Explanation of the trial session procedure and the locomotion techniques, (ii) training phase where all four methods could be learned and practiced, (iii) filling out a background questionnaire, (iv) task execution and completion of the task level questionnaires, and (v) filling out a questionnaire for a final subjective rating of the locomotion techniques.

In step (i), the experiment supervisor explained each individual the overview of the experiment and how the hardware is used. Furthermore, a brief introduction about moving in VR was given, since 86% of the participants never wore a HMD before. In about ten minutes, the subjects were told how to wear a VR HMD, got a rough overview of the questionnaire procedure, and to know the limitations of the hand tracking sensor. The range of the hand tracker (25–600 mm) was explained and it was highlighted that subjects should try to stay in the FOV of the sensor.

In the next step (ii), the subjects put on the VR HMD. The participants could familiarise themselves with the virtual environment and prepare for the experimental task. A background questionnaire was handed over in step (iii), asking the subjects about their age, gender, previous experiences with VR, and general confidence using computer and software.

After the background questionnaire has been fulfilled, step (iv) was performed where each sample run used a different locomotion technique. The order was defined by using an implementation of the Fisher-Yates shuffle algorithm, randomising the order of locomotion techniques for each participant. The randomisation was performed to reduce the learning effect and biases in the subsequent completion of the questionnaires. Each task run had a video placed in the VE, explaining the locomotion technique which can be used during this walkthrough. Following each task run, subjects filled out the SUS followed by the NASA-TLX questionnaire. After all locomotion techniques were performed and both the SUS and NASA TLX questionnaires were completed, the subjects were given a final questionnaire (step (v)). This questionnaire allowed the participants to grade each locomotion technique on a scale from 1 (poor) to 10 (good), select one technique as personal preference, and finally a text field to add a comment why this technique is preferred. The time allocation for each trial can be summarised as follows:

- Step (i): Explanation of the trial session procedure and brief explanation of locomotion in VR about - 10 minutes
- Step (ii): Training phase with learning and practicing four locomotion techniques - 15 minutes
- Step (iii): Filling out background questionnaire - 1 minute
- Step (iv): Task execution and filling out task level questionnaires-16 min divided as follows: 2 minutes performing the task + 2 minutes filling out questionnaires, repeated four times.
- Step (v): Filling out a questionnaire for final method comparison - 1 minute

Combined, the total execution time for one user session was about 43 minutes.

6.1.4. Results

Quantitative Evaluation

The performance of the proposed methods was measured by using the variables Task Completion Time (TC), number of times the hand tracking was lost HTL and the number of required teleportations to reach the goal NT. Table 6.1 shows the mean data gathered during the user study. For statistical analysis, significant results are reported at the 0.05 level.

Table 6.1.: *Data collected during the evaluation. Task Completion Time TC, number of times the hand tracking was lost during a session HTL, and number of teleportations required to reach the goal NT. Lower values are considered better, best values are marked in bold.*

Variable	Task	Mean	Median	sd	MIN	MAX
TC	TwoHandIndex	85.14	76	30.79	51	166
(in seconds)	TwoHandPalm	66.19	63	26.38	40	155
	OneHandIndex	94.95	91	23.03	58	142
	OneHandPalm	**63.76**	**57**	**19.73**	**36**	**116**
HTL	TwoHandIndex	12.90	11	5.21	7	29
	TwoHandPalm	13.28	11	5.55	6	28
	OneHandIndex	9.19	**7**	**4.40**	**3**	**18**
	OneHandPalm	**8.95**	**7**	6.0	**3**	29
NT	TwoHandIndex	25.76	24	4.21	18	37
	TwoHandPalm	25.52	23	8.07	**16**	48
	OneHandIndex	23.04	22	5.21	18	44
	OneHandPalm	**22.33**	**21**	**4.62**	17	**35**

Task Completion Time (TC) The task completion time measures the time it takes a participant to reach the task goal, i.e. to touch all the pillars. Although not visible to the participant, the precise time when each pillar is touched is recorded. For TC the time gap between touching the first and last pillar is measured. It is important to use the first pillar as an indicator when the trial started, since the user receives input from the instructor and a video tutorial at the beginning of each task. The gathered TC data is visualised in Figure 6.9

Levene's test assured the homogeneity of the input data ($p > 0.05$) and therefore the data was analysed using a one-way ANOVA. The ANOVA result $F(3, 80) = 7.42$, $p = 0.0001$ showed a statistically significant difference in task completion time between the techniques. For further investigation, Tukey's honest significant difference (TukeyHSD) was used as post hoc analysis of the data. TukeyHSD did not reveal a significantly different TC between the pairs TwoHandPalm-OneHandPalm ($p = 0.9895$) and TwoHandPalm-TwoHandIndex ($p = 0.0804$). However, the pairs OneHandPalm-TwoHandIndex ($p = 0.03$), OneHandIndex-TwoHandPalm ($p = 0.002$), and OneHandPalm-OneHandIndex ($p = 0.0008$) showed significant differences. These results indicate that there is a significant variance between palm based and index finger based techniques, whereby the palm-based techniques are significantly faster.

Number of Teleportations (NT) The number of teleportations each participant required to achieve the goal is recorded. This measure is particularly

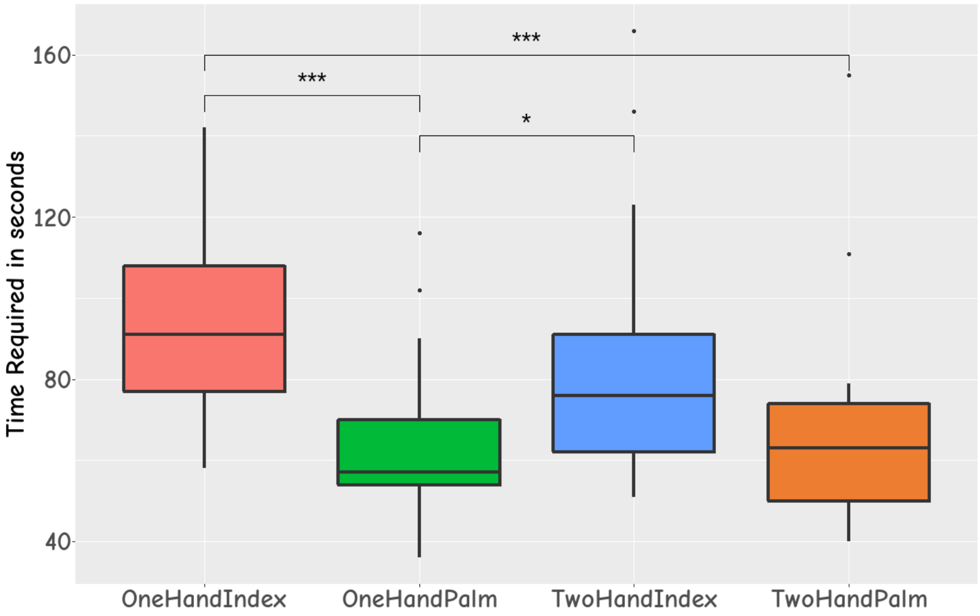

Figure 6.9.: *Box plot of task completion time of the proposed navigation methods. Significance Levels: *** = 0.001; ** = 0.01; * = 0.05.*

interesting to evaluate the effectiveness of a method, where lower number means a more effective locomotion technique.

Levene's test showed a violation for homogeneity of variances ($p < 0.05$) and therefore Welch's ANOVA was used for further analysis. The results of the ANOVA $F(3, 43.7) = 2.530$, $p = 0.0694$ revealed that there is no significant difference in terms of teleportation count between the proposed locomotion techniques.

See Figure 6.10 for a visualisation of the NT data. Using NT we found no evidence that a particular locomotion technique requires the user to do significantly more teleportations. An initial thought was that the two-handed techniques might require fewer jumps because of the direct control to when a jump happens but this seems to be not the case.

Number of Times the Hand Tracking Was Lost (HTL) This measure indicates how often the hand tracking has failed because the device failed to track or the subjects moved a hand out of the sensor's FOV. Two variants of this measure are distinguished: For two-handed methods, both hands must always remain visible to the sensor. For the one-handed methods, only the dominant hand must remain visible to the sensor. HTL as a measure is useful to emphasise how subjects are performing the task given the sensor's limited FOV while controlling locomotion. A lower number of tracking lost can indicate a better overall usability of the system. The gathered data is visualised in Figure 6.11.

Levene's test assured the homogeneity of variances ($p > 0.05$) and a one-

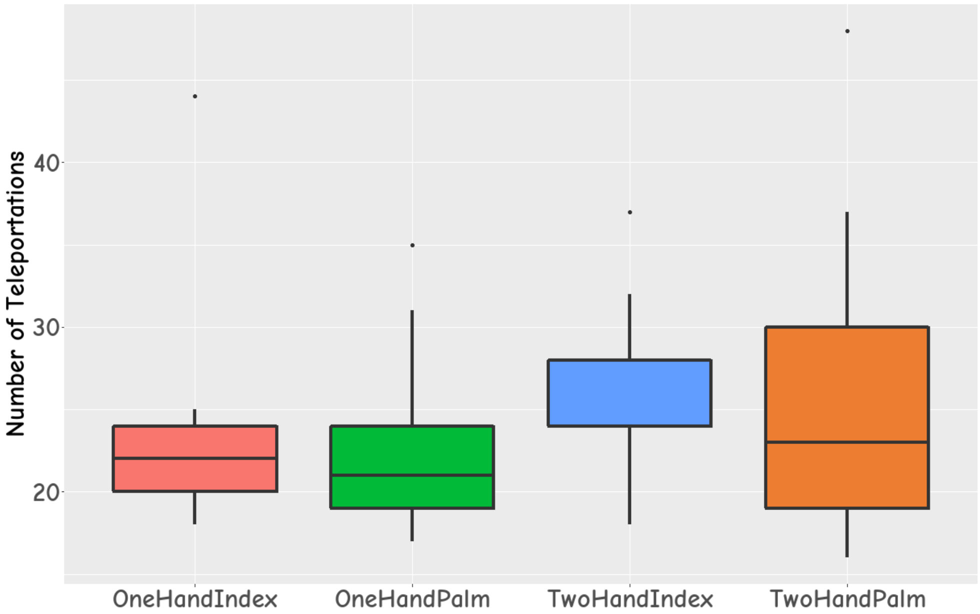

Figure 6.10.: *Box plot for the number of teleportations required to complete the experimental task using the proposed navigation methods. No significant difference between the techniques was found.*

way ANOVA for further analysis was used. The ANOVA result $F(3, 80) = 4.022, p = 0.0102$ showed a significant difference of HTL between the methods. The additional post hoc analysis using TukeyHSD revealed that there is a significant difference between TwoHandPalm and OneHandPalm ($p = 0.0483$) but the other methods did not significantly vary between each other.

Qualitative Evaluation

System Usability Scale A 5-point Likert-scale questionnaire has been used to measure the subjects perception of the usability of the proposed locomotion techniques. Participants completed the System Usability (SUS) questionnaire [24], [25], answering questions from a scale 1 (very low) to 5 (very high). In order to avoid response biases, the 10 questions alternate with positive and negative statements. In general, the SUS allows a rapid usability evaluation of techniques with a single number from 0 to 100. Sauro [226] conducted a meta-analysis from over 500 studies with more than 5000 scores and came to the conclusion that a total SUS score above 68 is considered above average. Albert and Tullis [2] state that a score above 70 can be interpreted as acceptable usability. It can be observed that all proposed techniques are above those thresholds. The locomotion techniques achieve the following SUS scores in ascending order: TwoHandIndex (M = 77.4 and sd = 13.7) with the lowest score, TwoHandIndex following with a marginally higher score (M = 78.1 and sd = 17.0), TwoHandPalm with a slightly higher (M = 83.6 and sd = 14.3), and finally OneHandPalm with the highest score (M = 89.6 and sd = 10.0).

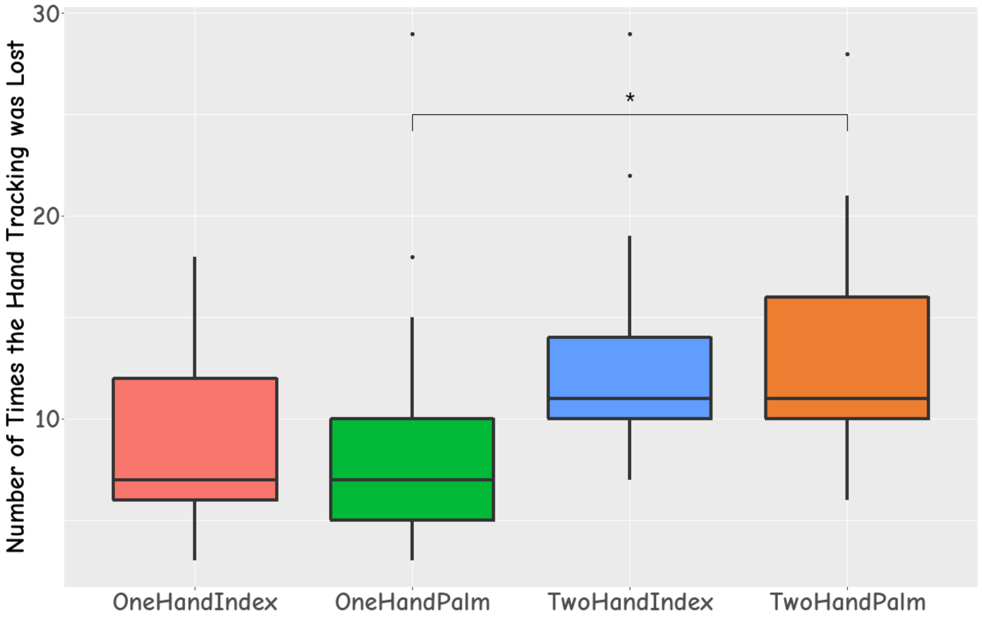

Figure 6.11.: *Box plot for the number of times the hand tracking failed during the experimental task. Significance Levels: *** = 0.001; ** = 0.01; * = 0.05.*

The SUS scores are shown in Figure 6.12.

Both methods involving the palm for navigation scored higher in the SUS than index navigation. The reason for this rating is found in the different gesture used to choose the teleport position and the ray origin. TwoHandIndex and TwoHandIndex use index navigation, meaning that a ray is shot out of the index finger of the subject which is used to choose the teleportation point. TwoHandPalm and OneHandPalm use palm navigation, where the ray is shot out of the subjects' palm. While observing the subjects, the palm proved to provide better stabilization than the index due to a stronger directional noise while pointing. Additionally, users tend to stretch their arms fully during a pointing gesture, thus reducing the tracking accuracy by moving out of the reliable tracking zone of the hand tracking device. Participants had to be reminded several times, that moving the hand closer to the face (and therefore closer to the sensor) will provide better tracking accuracy. Sometimes participants narrowly missed their target and leaned forward to touch the pillar. Smaller movements with the chair were also observed which is similar to a step to correct in standing position. Therefore, it can be assumed that this inaccuracy occurs not only when sitting, but also when standing.

NASA-TLX After a method was performed in the evaluation task, subjects had to fill out a NASA-TLX questionnaire [89]. The NASA-TLX questionnaire indicates the overall subjective perceived workload for each locomotion technique proposed. It consists of a set of six subscales measuring mental

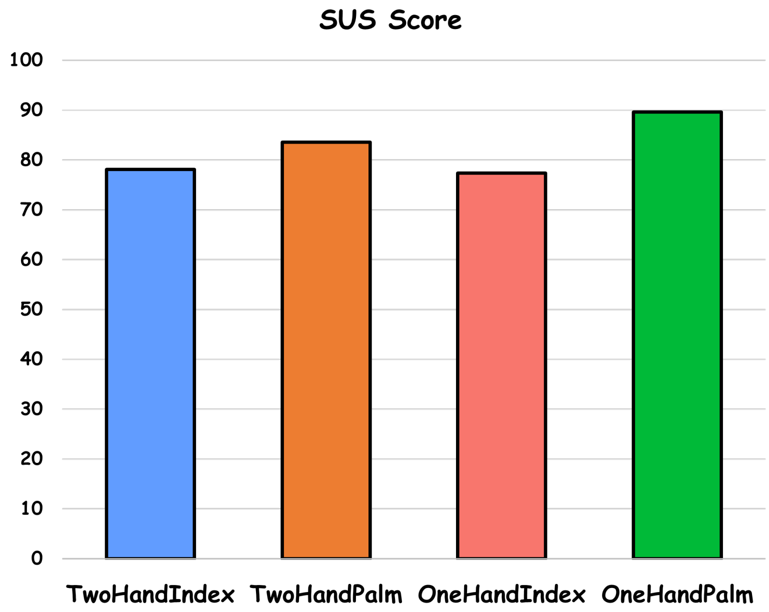

Figure 6.12.: *Average System Usability Scale (SUS) scores for each method, indicating the subjects' perceived usability. The line shows threshold of 68, which indicates an above average evaluation.*

demand, physical demand, temporal demand, performance, effort, and frustration. Each measure is rated on a scale from 0 to 100 divided into 20 grades. Each subscale is graded along a low-high continuum. During a NASA TLX evaluation, the subjects weight the subscales they feel are more important. To achieve this weighting, 15 single questions are asked which compare two subscales (e.g., mental demand against physical demand, mental demand against performance, etc.) and the participants choose the measure which seems more important to them. In order to shorten the evaluation procedure we omitted the weighting process, which is known as Raw-TLX (RTLX) [88]. Bustamante and Spain [28] compared NASA-TLX with RTLX and came to the conclusion that RTLX is a valid alternative.

Figure 6.13 shows the results after each locomotion technique was performed. The task order for each locomotion technique was counterbalanced in order to enable more comparable results. It is to note that each task execution took about two minutes and filling out the questionnaires enabled a break of about two minutes between each consecutive task. The perceived workload of proposed techniques is calculated by averaging the six subscales. The overall score in order from high to low: The highest perceived workload with TwoHandIndex (M = 23.78 and sd = 16.66), followed by a slightly lower workload with TwoHandIndex (M = 18.61 and sd = 12.14), with no great difference to TwoHandPalm (M = 18.38 and sd = 15.17), and finally OneHandPalm (M = 12.96 and sd = 11.02). The results indicate that TwoHandIndex is generally more demanding than other techniques. Furthermore, OneHandPalm seems to be less demanding than other techniques. ANOVA revealed no statistically significant difference between techniques regarding each subscale.

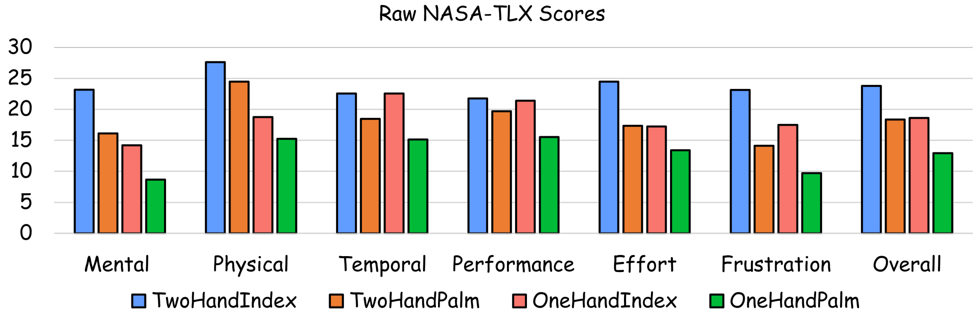

Figure 6.13.: *Raw NASA Task Load Index (NASA-TLX) scores for each proposed locomotion technique. Lower numbers are better and the maximum value of a subscale is 100.*

User Rating of Proposed Methods The evaluation process was concluded with a final questionnaire, where the subjects could rate each method on a scale of 1 (poor) to 10 (good). Furthermore, subjects had to choose one locomotion technique as their personal preference and had a text box to explain why. The user rating for each technique are as follows: The lowest score has TwoHandIndex (M = 6.6), TwoHandIndex (M = 6.8), TwoHandPalm (M = 8.0), and OneHandPalm (M = 8.5). OneHandPalm was preferred by most users, slightly followed by TwoHandPalm while TwoHandIndex and TwoHandIndex got the least supporters. ANOVA $F(3, 80) = 3.822$ showed significant differences across methods. The additional post hoc analysis using TukeyHSD revealed that TwoHandIndex received significantly lower scores than OneHandPalm ($p = 0.0287$). The overall preferences (see Figure 6.14) matches with the SUS score (see Figure 6.12). Palm navigation was generally preferred and a trend in favor of OneHandPalm is recognisable, but there is no obvious winner between those two techniques. Most subjects who varied between TwoHandPalm and OneHandPalm gave both methods the highest scores. Many who preferred TwoHandPalm said that they would like to have control over when to teleport. If OneHandPalm was preferred, the subjects said that it was more relaxing to use only one hand.

6.1.5. Discussion

The findings in quantitative and qualitative evaluation show promising results for all presented locomotion techniques. No significant difference was found in the analysis between the age groups 25–37 (17 participants) and 57–60 (4 participants). An overview of the findings is presented in Table 6.2. The locomotion techniques designed and implemented using the proposed system scored high in the SUS, indicating a general good usability for all techniques. Further examining the results of the four presented locomotion techniques, there are two clear winners of this experimental study. OneHandPalm (one-handed,

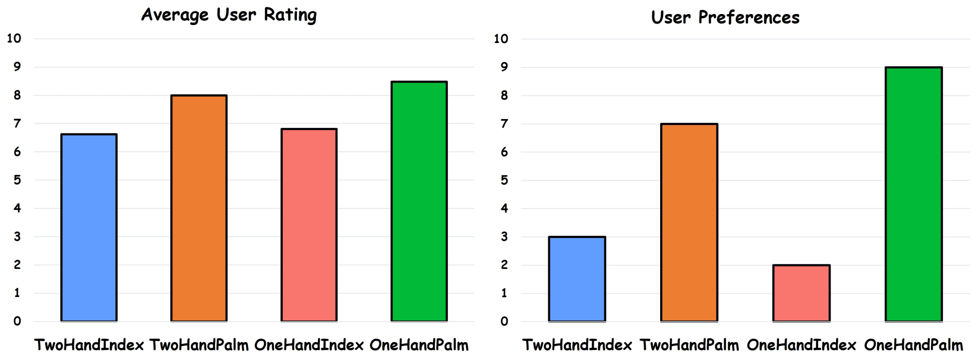

Figure 6.14.: *(**left**) User rating for each technique in a scale from 1 (poor) and 10 (good). (**right**) User preferences for the proposed locomotion techniques. Subjects were allowed to choose only one technique as their favourite.*

palm navigation) and TwoHandPalm (two-handed, palm navigation) have a much higher performance than TwoHandIndex (two-handed, index navigation) and TwoHandIndex (one-handed, index navigation). The SUS scores are higher and the general subject preference is clear in favor of TwoHandPalm and OneHandPalm. Furthermore, TwoHandPalm and OneHandPalm have higher efficiency (significantly lower task execution time) than TwoHandIndex and OneHandIndex. However, a significant difference in the proposed techniques in terms of effectiveness could not be found. The number of teleportations NT does not vary much between techniques. The number of tracking interruptions HTL is not drastically affected by the proposed techniques either.

Table 6.2: Summary of findings in the user study.

	Findings
Quantitative Evaluation	TwoHandIndex had the slowest task completion time.
	OneHandPalm had the fastest task completion time.
	Task completion time did not significantly vary between a two-handed technique and their one-handed alternative, i.e. TwoHandPalm-OneHandPalm $p > 0.05$ and TwoHandIndex-TwoHandIndex $p > 0.05$.
	No evidence that a particular locomotion technique requires the user to do significantly more teleportations was found.
	TwoHandPalm had significantly more number of tracking interruptions compared to OneHandPalm in our evaluation scenario ($p = 0.0483$).

Continued on next page

Table 6.2: Summary of findings in the user study. (Continued)

	Findings
	In general it can be concluded that the proposed techniques do not have a strong impact on HTL.
Qualitative Evaluation	All proposed techniques rank above the threshold of 68. The lowest SUS score of M = 77.4 is from TwoHandIndex.
	OneHandPalm has the best perceived usability with a SUS score of M = 89.6.
	OneHandPalm has consistently performed better in the NASA-TLX scores than all other techniques (lower perceived workload).
	TwoHandIndex has consistently performed worse than all other techniques (higher perceived workload).

Answering the Research Questions

Using the different quantitative and qualitative measures the questions initially formed in section 6.1 can be answered.

RQ1: Can users easily navigate through a virtual environment by using hand gestures?

To answer this question, the SUS as well as the NASA-TLX scores are considered. The proposed gesture based locomotion techniques score high in the SUS, indicating a generally high level of usability and ease of use. The lowest score of 77.4 scored by TwoHandIndex is being considered above average as according to Sauro [226] and thus imply above average usability. Looking at the NASA-TLX scores, there is an indication that the TwoHandIndex technique requires a higher perceived workload compared to other techniques, but overall it can be concluded that the techniques are adequate in terms of workload. Especially when subjects used the OneHandPalm technique low perceived workload can be observed. During the study no evidence that subjects struggle to control movement in VR with hand gestures was found. The quantitative and qualitative data suggests an easy navigation through the VE by using hand gesture based movement control.

RQ2: Can users move through virtual environments efficiently and effectively with just one hand?

This question is answered by analysing the quantitative data collected during the user study. For efficiency the number of teleportations required during the experimental task is compared. It is observed that there is no significant difference between proposed techniques as determined by an ANOVA $p > 0.05$. This result suggests that the proposed two-handed and one-handed techniques are equally efficient regarding number of teleportations. Furthermore, the hand tracking interruptions between two-handed and one-handed techniques are considered. An ANOVA with $p < 0.05$ indicates a significant difference regarding the number of hand tracking interruptions across all techniques. A post hoc analysis revealed that TwoHandPalm had significantly more interruptions compared to OneHandPalm with $p = 0.0483$. Other techniques however, did not significantly vary in terms of HTL. Looking at the raw data, TwoHandPalm had the highest number of tracking interruptions with M = 13.28, followed by TwoHandIndex (M = 12.90). The one-handed techniques seems to have less tracking interruptions where TwoHandIndex had M = 9.19 and OneHandPalm M = 8.95 but the statistical analysis showed only significance between TwoHandPalm and OneHandPalm. In general it can be concluded that there is no drastic effect on hand tracking interruptions within the proposed techniques. Most likely this is due to the fact that some users are more comfortable with keeping only one hand in the sensor's field of view rather than both hands. However, the difference in HTL is not sufficient to draw conclusions about which method performed better.

Analysing the task completion time as measurement for effectivity, ANOVA revealed significant differences between the proposed techniques ($p < 0.05$). The difference is found mostly in the palm-based techniques being faster than index-based techniques as shown in Section 6.1.4. With the given data it can be determined that one-handed techniques are generally not slower compared to two-handed ones. With the aforementioned points it can be concluded that one-handed techniques are at least equal to their two-handed counterparts in terms of efficiency and effectiveness.

RQ3: Do users prefer controlling locomotion with one hand rather than both hands?

To answer this question the user rating of the proposed techniques is considered. The locomotion techniques TwoHandPalm and OneHandPalm were liked the most by participants as shown in Figure 6.14. OneHandPalm got significantly better user rating than TwoHandIndex ($p < 0.05$). If the participants were undecided, they usually gave two methods an equal score. An additional question was included in the questionnaire, asking people to choose one technique over the other. The questionnaire also asked why people chose one method over the other. When a two-handed technique was chosen, the most common response was that it offered more control than a one-handed technique. If a one-handed technique was chosen, the most frequent answer was that it was simply more comfortable and even perceived as faster. Analysing the user preferences show that there is no clear winner between two-handed and one-handed locomotion techniques but rather between index and palm navigation. The results of the study suggests that using a one-handed technique is a viable alternative to a two-handed technique and vice versa.

Limitations and Future Work

The proposed locomotion techniques are tested with a specific hand tracking sensor. During the user study, some users mentioned that it was annoying to keep both hands in the sensor's field of view. A more sophisticated hand tracking solution with a wider FOV could be a major improvement to the proposed techniques.

Furthermore, the user study itself had some limitations. A simple evaluation task was used where users had to touch virtual objects. There where no obstacles besides the touchable objects and there was no other type of interaction. Especially the one-handed techniques might require a deeper investigation. Future work should find out if one-handed techniques are still applicable in scenarios where many objects need to be grabbed, touched, pressed, etc. A strong indication that palm navigation is preferred over index navigation was observed, but a user study with more participants would help to strengthen this conclusion.

6.1.6. Conclusion

This section presents and compares four hand gesture based techniques for controlling locomotion in VR. The design and implementation of the proposed locomotion techniques was achieved by a gesture capturing system which is able to capture static hand gestures for use in a virtual environment. With the help of 21 volunteers, the four locomotion techniques presented in this section are evaluated. A user study was conducted which aimed to provide useful insights into hand-based locomotion control techniques, especially involving one-handed techniques. Two two-handed methods which require an activation gesture to move around, and two one-handed methods-which activate passively after a certain time are presented. These methods use either an index finger or the palm to navigate through the virtual world. The techniques were evaluated by utilising quantitative and qualitative measurements. For quantitative measurements, the task completion time (TC), number of times the hand tracking was lost during a session (HTL), and number of teleportations required to reach the task goal (NT) was used. As qualitative measures, the participants were given questionnaires, including the System Usability Scale and the NASA-TLX questionnaire. In addition, the user study participants evaluated each method subjectively and allowed them to choose their favourite method. One of the results is that navigation with the palm is preferred over navigation with the index finger. The evaluation results indicate that all the proposed techniques are a viable choice for moving in VR. Moreover, there was no clear winner between two-handed and one-handed techniques. The results of the study show that one-handed methods can be used well for locomotion in VR. The presented techniques which require only one hand did not show much difference compared to the two-handed alternative, both qualitatively and quantitatively. With regard to the proposed techniques, it is suggested to use either TwoHandPalm or OneHandPalm in VR. If possible, the system should allow users to choose which locomotion technique to use or to choose the locomotion technique based on the possible interaction features of the virtual environment.

6.2. Learning Effect of Laypeople using Hand Gestures for Locomotion

Relevant work regarding locomotion in VR focuses mostly on user preference and performance of the different locomotion techniques. This ignores the learning effect that users go through while new methods are being explored. In this section, it is investigated whether and how quickly users can adapt to a hand gesture-based locomotion system in VR. The four different locomotion techniques from the previous section are examined with regard to the learning effect. The goal of this is twofold: First, it aims to encourage researchers to consider the learning effect in their studies. Second, this study aims to provide insight into the learning effect of users in gesture-based systems.

6.2.1. Formulating the Research Questions

Usually a controller is used for virtual locomotion [66], [209], [307], but more recent work uses other techniques such as vision-based tracking [233] or sensors that are attached to the body [190], [223]. The controller-based methods are extensively researched and already in commercial use. Gesture-based locomotion is getting more attention lately and requires more research to find out which methods are adequate. Many studies focus entirely on subjective user preference in researching and developing such locomotion systems. Moreover, studies related to locomotion in VR usually use a single "ease-to-learn" question for participants to find out if the technique is easy to learn. The analysis in this section addresses another important factor that is often neglected. Namely, how quickly people can learn such a system. Complicated methods lead to longer times to achieve objectives and ultimately to user frustration. This is one of the hurdles that stand in the way of the mass adoption of hand gestures as an interface. Therefore, attention should be paid to whether or not users can quickly adapt to implemented techniques.

To investigate the learning effect, the user study from the previous section is further examined for new findings, focusing on the learning effect. People with minor and no background in VR were recruited in order to remove additional bias or knowledge regarding VR systems. Emphasis is put on the learning effect that can be observed by these lay people. More concrete, this analysis answers the following research questions:

Q1: *Do lay people significantly improve their task completion time when using hand gesture based locomotion during a second session compared to the first?*

Q2: *How much do lay people adapt to the limitations of a hand tracking device after a first trial session?*

Q3: *Will lay people significantly improve their efficiency (less number of* *teleportations) when using hand gesture based locomotion during a second trial session?*

6.2.2. Subjective Learning Experience for Locomotion in VR

To the best of knowledge, no paper was published that focuses exclusively on investigating the learning effect of lay people for hand-gesture based VR locomotion. However, some work mentions learning effect during their studies.

Zhao *et al.* [320] investigates different techniques to control locomotion speed. The gestures include Finger Distance, Finger Number, and Finger Tapping. Users were asked within a questionnaire to subjectively evaluate the ease-to-learn. According to the results, users find the proposed techniques easy to learn. However, no quantitative analysis was performed to gain insights into the learning effect of participants. Zielasko *et al.* [322] implemented and evaluated five different hands-free navigation methods for VR. The techniques include Walking In Place, Accelerator Pedal, Leaning, Shake Your Head, and Gamepad. Using a questionnaire, the authors come to the conclusion that the introduced techniques are easy to learn. Kitson *et al.* [125] introduced NaviChair, a chair based locomotion technique for virtual environments. Users are required to move within the chair to get different locomotion effects. The authors compare this technique with a technique based on a normal joystick. During exit interviews, it was revelead that the joystick variant is preferred by users because it is more accurate and easier to learn. The work of Keil *et al.* [116] uses VR locomotion techniques to measure users' learning effect in distance estimations. The authors found a significant decrease in distance estimation errors after a subsequent task. The proposed analysis does not rely on a subjective questionnaire to answer whether techniques are easy to learn. Instead, gathered data is quantitatively analysed to measure the learning effect of users.

6.2.3. Answering the Research Questions

The data collected during the experiment (described in section 6.1.3) is used to answer the research questions. The experiment was split into two phases: Learning phase and evaluation phase. In both phases, the participants had to move through the environment and perform the given task. In the learning, as well as the evaluation phase, all four techniques were used by the participants. Both phases are identical in terms of task and VE. In the learning phase, participants have performed each technique for the first time. Therefore, data is collected where users performed each technique for the first and the second time. One session lasted about 43 minutes.

No questionnaire regarding learning experience was given to the participants and our findings are based solely on quantitative results. Task completion time measures the time a participant requires to complete the task. More precisely, it represents the time from touching the first pillar to the last inside the VE. Additionally, the number of times the hand tracking was lost is collected. This value represents the tracking failures of the chosen device. The tracking fails if the participant moves a hand out of the reliable tracking range of the device. For two-handed locomotion techniques, both hands have to be tracked by the sensor and for one-handed techniques only the dominant hand needs to be tracked. This measure is particularly useful to measure the adaption of the user to overcome the limitations of the tracking device. Furthermore, the number of teleportations a user required to reach the goal is collected. The raw values such as mean, median, standard deviation, minimum, and maximum are shown in Table 6.3. One-way ANOVA is used for statistical analysis and throughout the section significance at the 0.05 level is reported.

Table 6.3: Raw data collected during the experiment. TC (Task Completion Time), HTL (number of times the hand tracking was lost during a session), and NT (number of teleportations required to complete the task)

Variable	Task	Mean	Mdn	sd	MIN	MAX
TC	M1 - Learning	158.33	136	90.23	79	452
	M2 - Learning	92.85	92	33.21	42	168
	M3 - Learning	108.71	96	37.66	71	217
	M4 - Learning	66.09	61	20.14	45	110
	M1 - Evaluation	85.14	76	30.79	51	166
	M2 - Evaluation	66.19	63	26.38	40	155
	M3 - Evaluation	94.95	91	23.03	58	142
	M4 - Evaluation	63.76	57	19.73	36	116
HTL	M1 - Learning	26.14	22	16.21	8	78
	M2 - Learning	19.57	15	10.68	7	46
	M3 - Learning	12.14	11	8.48	2	42
	M4 - Learning	6.80	6	3.84	2	14
	M1 - Evaluation	12.90	11	5.21	7	29
	M2 - Evaluation	13.28	11	5.55	6	28
	M3 - Evaluation	9.19	7	4.40	3	18
	M4 - Evaluation	8.95	7	6.0	3	29
NT	M1 - Learning	33.19	31	18.51	17	102
	M2 - Learning	25.61	23	8.78	15	47
	M3 - Learning	23.00	22	4.38	18	36
	M4 - Learning	22.19	20	5.87	17	41
	M1 - Evaluation	25.76	24	4.21	18	37

Continued on next page

Table 6.3: Raw data collected during the experiment. TC (Task Completion Time), HTL (number of times the hand tracking was lost during a session), and NT (number of teleportations required to complete the task) (Continued)

Variable	Task	Mean	Mdn	sd	MIN	MAX
	M2 - Evaluation	25.52	23	8.07	16	48
	M3 - Evaluation	23.04	22	5.21	18	44
	M4 - Evaluation	22.33	21	4.62	17	35

Do lay people significantly improve their task completion time when using hand gesture based locomotion during a second session compared to the first?

To answer this question, the task completion time is taken into account. Levene's test was conducted in order to ensure homogeneity of the input data ($p > 0.05$). One-way ANOVA was used in order to answer whether users are faster at completing the given task after performing a training. A comparison between each technique's learning phase with it's corresponding evaluation phase was conducted. The average values are depicted in Figure 6.15. The results of the ANOVAs are: TwoHandIndex: $F(1,40) = 12.38, p = 0.001$; TwoHandPalm: $F(1,40) = 8.298, p = 0.006$; OneHandIndex: $F(1,40) = 2.04, p = 0.161$; OneHandPalm: $F(1,40) = 0.144, p = 0.707$.

The results show significant difference in the task completion time for the techniques TwoHandIndex and TwoHandPalm with $p < 0.05$. The techniques OneHandIndex and OneHandPalm do not show significance with $p > 0.05$. Therefore, it can be concluded that users performed significantly faster after conducting a learning phase for the two-handed techniques. The one-handed techniques however did not show significant improvements.

How much do lay people adapt to the limitations of a hand tracking device after a first trial session?

Today, hand tracking devices have several limitations such as occlusion, low field of view, and tracking range. Scientists working in this field know these limitations and already avoid them unconsciously. Non-experts who have never come into contact with such technology will discover many of these limitations. This inevitably leads to many tracking errors until the user becomes aware of why the system has problems. For this reason, the number of times the hand tracking failed during a session is used as an indicator to answer this research question. Once the users hands are no longer tracked, it counts as hand tracking lost. It can be said that users unconsciously and unintentionally move their hands out of the sensor's FOV because they are not accustomed to the technology. Therefore, this metric is used as an indicator of lay peoples learning effect of the chosen techniques. The average values are depicted in Figure 6.16. One-way ANOVA was used to find significant improvements

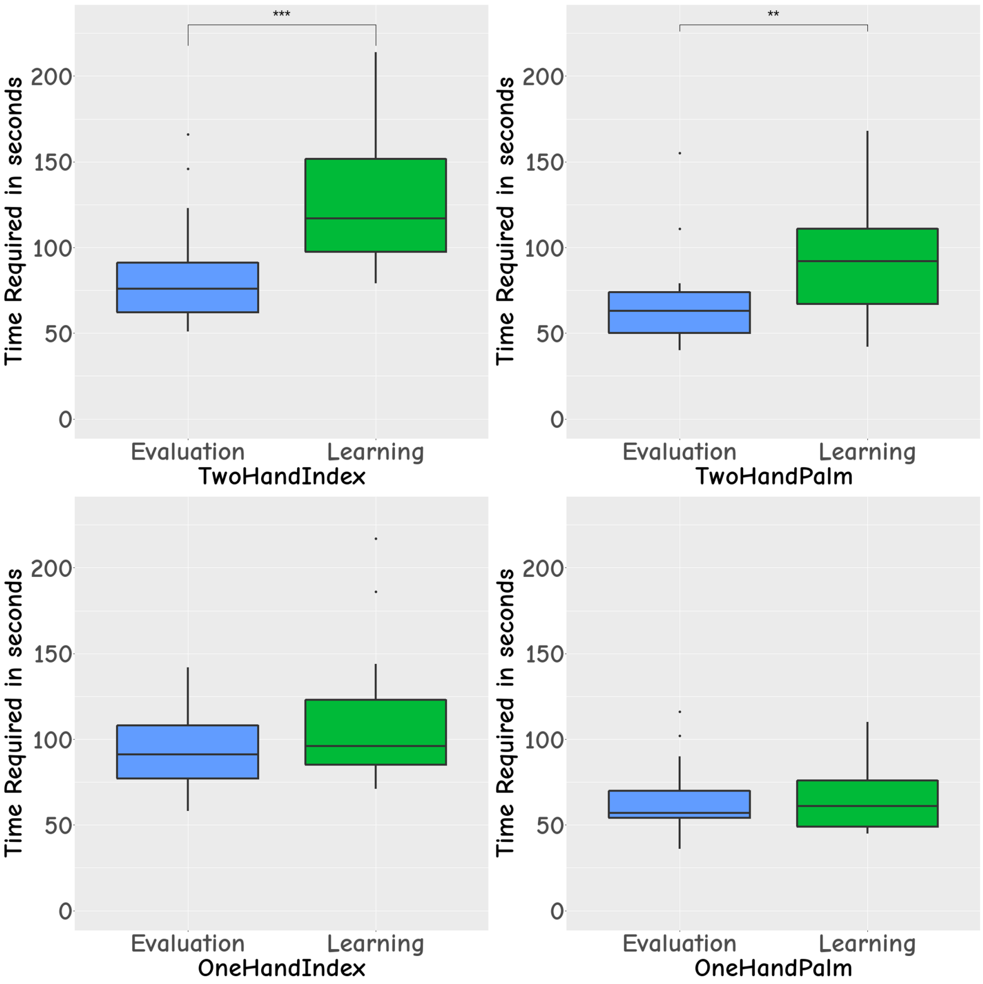

Figure 6.15.: *Box plots comparing task completion time between learning and evaluation phase. The values represent the time users required to fulfill the given task in seconds.*

between the learning and evaluation phase. Levene's test assured homogeneity of the input data. The result of the ANVOAs are: TwoHandIndex: $F(1, 40) = 12.69, p = 0.001$; TwoHandPalm: $F(1, 40) = 5.727, p = 0.021$; OneHandIndex: $F(1, 40) = 2.003, p = 0.165$; OneHandPalm: $F(1, 40) = 1.898, p = 0.176$.

These results indicate, that the two-handed techniques showed significant improvements between the learning phase and the evaluation phase. The one-handed techniques do not show significance. The two-handed techniques show overall increased tracking errors compared to the one-handed techniques. Therefore, it can be concluded that people perform better when using one-handed techniques. However, users are also able to significantly improve with the two-handed techniques by only doing one prior session with the technique.

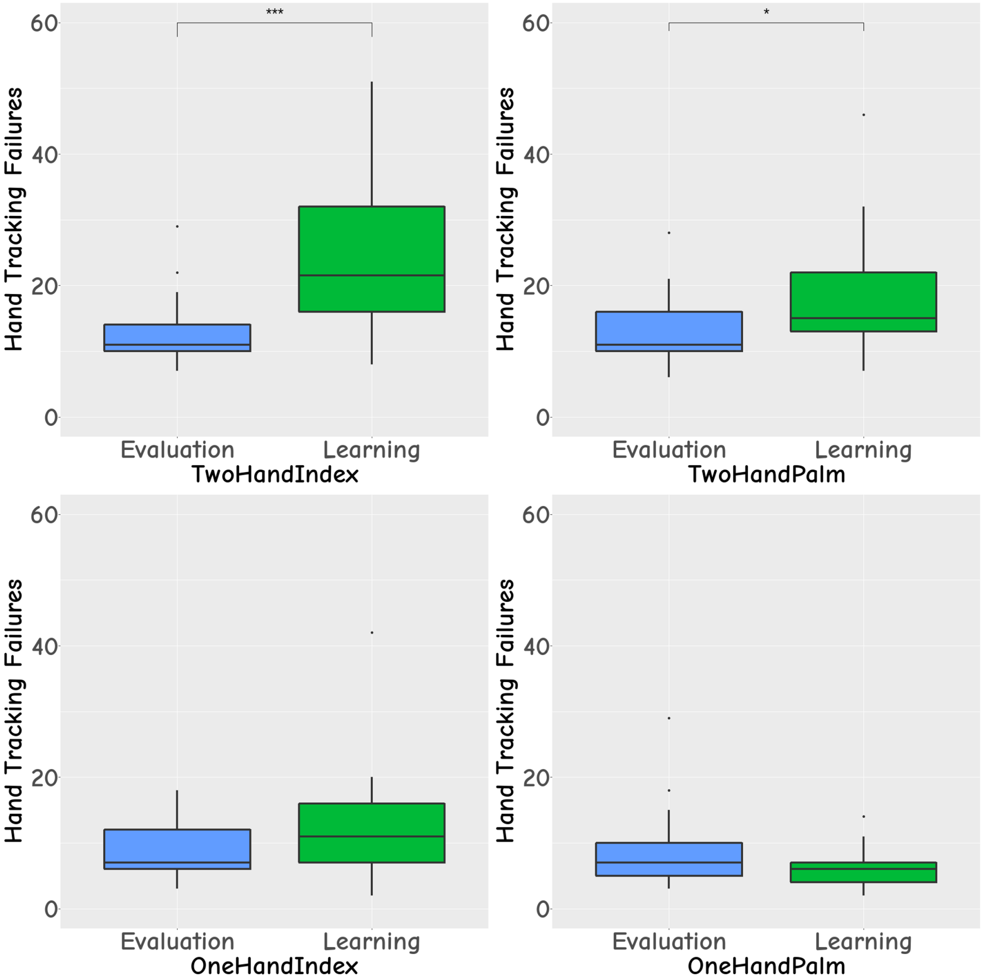

Figure 6.16.: *Box plots comparing the number of times the tracking has failed between learning and evaluation phase for each technique.*

]

Will lay people significantly improve their efficiency (less number of teleportations) when using hand gesture based locomotion during a second trial session?

To answer this question, the number of teleportations is considered. Levene's test was conducted in order to ensure homogeneity of the input data. One-way ANOVA was used to identify significant differences between the learning and evaluation phase of the experiment. Each technique's learning phase was compared to it's corresponding evaluation phase. The average values are depicted in Figure 6.17. The result of the ANOVAs are: TwoHandIndex: $F(1,40) = 3.214, p = 0.08$; TwoHandPalm: $F(1,40) = 0.001, p = 0.971$; OneHandIndex: $F(1,40) = 0.001, p = 0.975$; OneHandPalm: $F(1,40) = 0.008, p = 0.931$.

According to the one-way ANOVAs, there was no significant improvement observed for individual techniques between learning and evaluation phase ($p > 0.05$). Therefore, no individual learning effect could be observed in teleportation behavior.

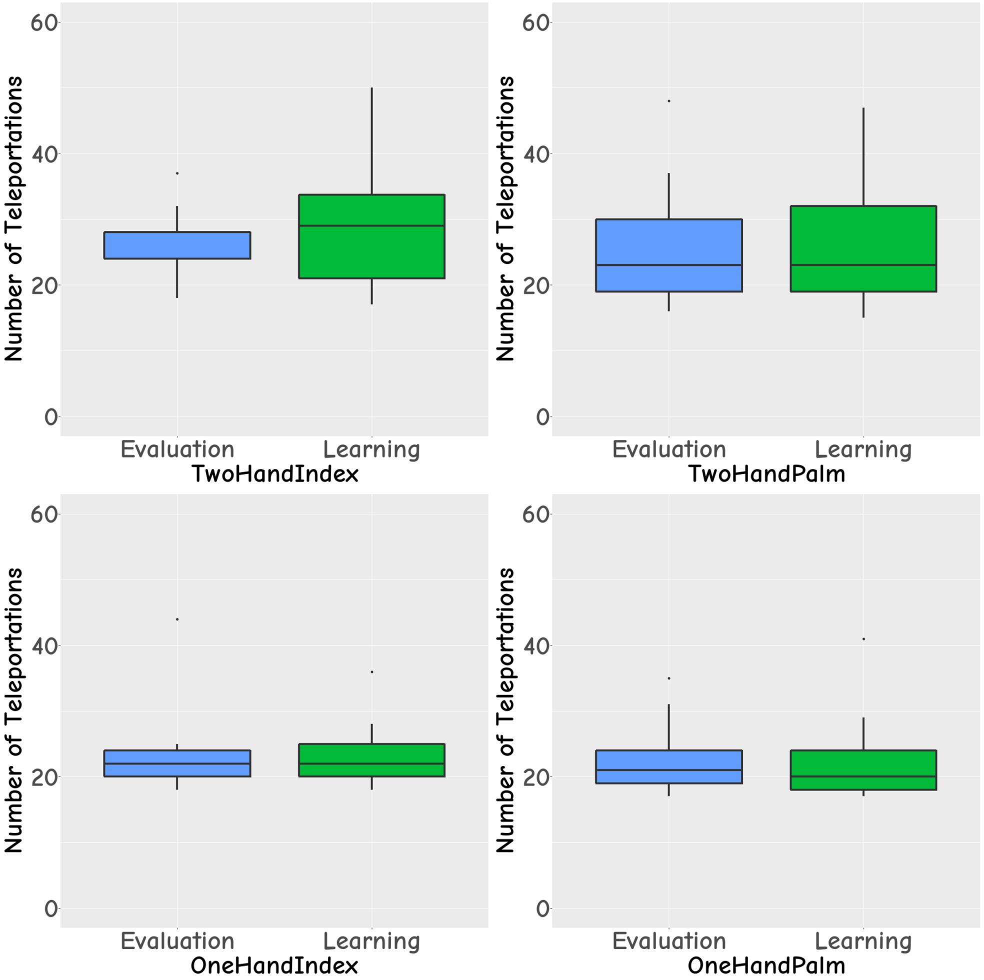

Figure 6.17.: *Box plots comparing the number of teleportations required to reach the goal between learning and evaluation phase.*

6.2.4. Discussion of Experiment Results for Learning Effect

The participants showed a significant improvement in task completion time for the two-handed techniques. The one-handed techniques did not significantly improve between the first and second phase of the experiment. During the experiment, subjects already expressed that one-handed techniques seem to be more intuitive and henceforth would explain these results. However, after performing the two-handed techniques for a second time, there is already significant improvement noticeable. This result is also backed by the fact that users significantly improved in regards to hand tracking failures. In the first phase, the users were uneasy because they first had to understand the limitations of the hand tracker. In the second phase, a clear improvement was noticeable. The number of teleportations required to reach the goal did not significantly vary between learning and evaluation phase. This could mean that the users understood how to achieve the goal and the virtual environment in combination with the task was straightforward to understand.

6.2.5. Experiment Summary for Learning Effect

This section investigated the learning effect of lay people performing hand gesture-based locomotion. A user study with 21 participants was conducted. In this study, four locomotion methods were utilised and the experiment was divided into a learning phase and an evaluation phase. All four methods were carried out twice by the subjects. The first time a method was performed was referred to as the learning phase and the second time as the evaluation phase. The study showed significant improvements for the subjects while using two-handed techniques. The participants were considerably faster and significantly improved at using the hand tracking device. Therefore, it can be said that users struggle at first and then, with just one more trial run, they can significantly adapt to gesture-based systems with two hands. Furthermore, no significant learning effect was observed using one-handed techniques.

6.3. Continuous Locomotion Using Bare Hands

As mentioned in the beginning of this chapter, there are two ways to move in virtual environments: Teleportation based locomotion and continuous locomotion. Teleportation locomotion instantly changes the position of the user. Continuous locomotion on the other hand is more like a walk, where the user gradually moves in the desired direction. Teleportation based locomotion is known to cause less motion sickness compared to continuous locomotion, but the latter is more immersive [10], [47], [131]. It is a trade off between immersion and motion sickness. Therefore, if the application scenario permits, care should be taken to allow the user to choose between the two methods. Games and other commercial applications using a controller usually allow for an option to choose which locomotion method is desired. In the following experiment, three novel locomotion techniques using bare hands for continuous locomotion are proposed and evaluated. A technique which uses index finger pointing as metaphor was implemented. Steering is performed by moving the index finger into the desired direction. A similar technique using the hand palm for steering was implemented. The third bare handed technique utilises a thumbs up gesture to indicate movement. Compared to other freehand locomotion techniques which involve rather demanding body movements, locomotion using hand gestures could be a less stressful and demanding technique. This assumption arises from the fact that only finger and hand movements are required for locomotion, whereas other techniques require large parts of the body to be moved. The three techniques are compared to each other and to the current standard for moving in VR, the controller. The following study aims to provide more insights into hand gesture based locomotion and whether it is applicable and easy to use by users. In particular, the research gap of continuous locomotion with hand gestures should be addressed, as most existing techniques use teleportation. In addition, it is not yet clear which hand gestures are suitable for the locomotion task in VR, and further research should be conducted to find suitable techniques. The contributions of this section are as follows:

- Introducing three novel locomotion techniques for VR using bare hands
- A comprehensive evaluation of these techniques

6.3.1. Proposed Locomotion Techniques

Four different locomotion techniques were developed: Controller, FingerUI, HandUI, and ThumbGesture. The proposed locomotion techniques are depicted in Figure 6.18. The implementation of each technique is briefly explained in this section.

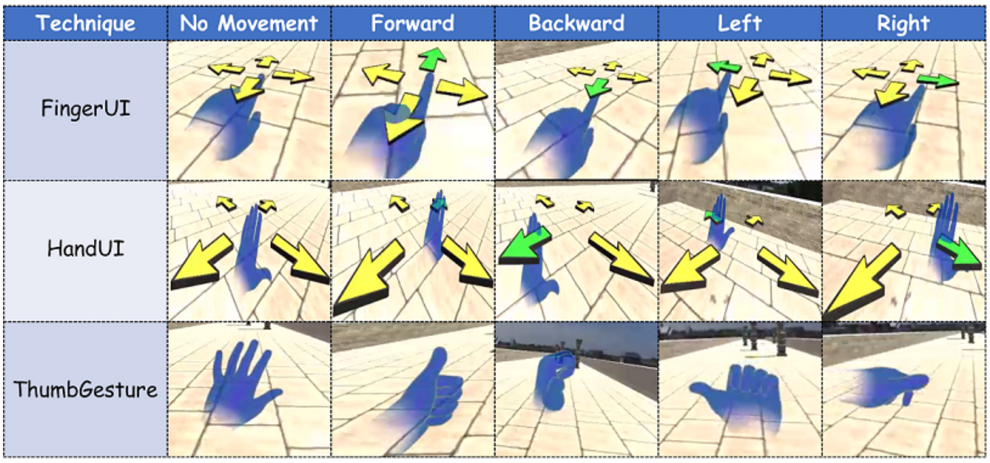

Figure 6.18.: *The proposed one-handed locomotion techniques. Users could move by moving the hand to a designated zone for moving forward, backward and rotating left and right. The arrows to control movement all had the same size.*

Controller. This technique uses the standard implementation for continuous locomotion with the Software Development Kit (SDK) of the chosen VR HMD. The thumbstick on the left controller is used for acceleration and the thumbstick on the right controller can be used to rotate the user. Using the right thumbstick is optional since the user can turn normally by just moving the head.

FingerUI. If the user points the index finger forward, a 3D graphical user interface will be shown. A 3D arrow for the four different directions Forward, Backward, Left, and Right are shown. While the user is maintaining the index finger forward pose with the hand, locomotion is achieved by moving the hand to one of the arrows depending on which movement is desired. The arrows are only for visualisation purposes. The actual movement is triggered when the index finger enters invisible zones which are placed around the 3D arrows indicating the movement direction. Only touching the arrows would be too strict, whereas the introduction of movement zones allows more room for user error. For this reason, zones are actually larger than the arrows shown to the user. This is depicted in Figure 6.19. Furthermore, the zones for moving left and right are generally bigger than for moving forward and backward. The reason for this is that during first pilot testing it was found that users generally made wide movements to the left and right. If the hand moves out of a zone, movement will unintentionally stop. Moving the hand forward was restricted due to arm length and moving backwards was restricted because the own body was in the way. Furthermore, with the design showed in Figure 6.19, users could move forward by putting the hand forward and then swiped to the left/right to rotate instead of moving the hand to the center and then to the left/right. Once the UI is shown, the zones are activated for all movement directions and the center can be used to indicate that no movement is desired.

HandUI. This technique is similar to FingerUI. The difference is the hand pose to enable the user interface. A”stop” gesture, i.e. palm facing away from the face and all fingers are up, is used to show the user interface. Instead of the index finger, the palm center needs to enter a zone to enable movement. The size of the zones is also adjusted (bigger and more space in the center for no movement).

ThumbGesture. A thumbs up gesture is used to activate movement. The four movement directions are mapped to different gestures. Thumb pointing up = Forward; Thumb pointing towards face = Backward; Thumb left = Left; Thumb right = Right.

All hand based locomotion techniques used a static gesture to activate locomotion and no individual finger movement was necessary. Furthermore, while the gestures and controller had a dedicated option to rotate the virtual avatar, users could also rotate by looking around with the VR HMD. Users can not change the locomotion speed but once the user enters a zone with their hand

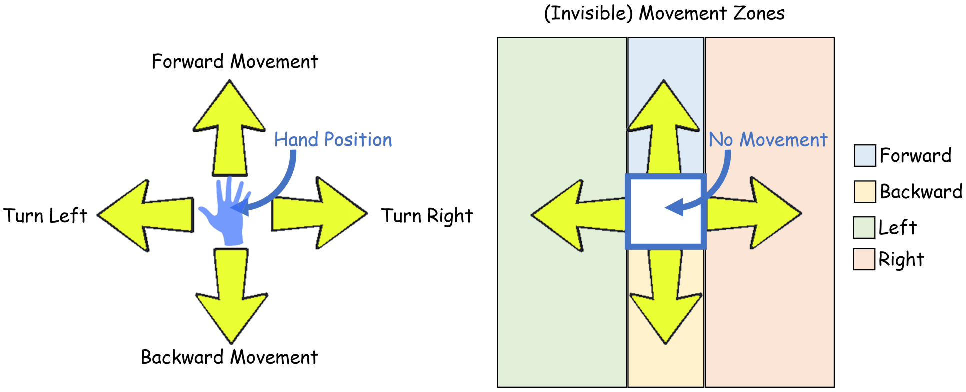

Figure 6.19.: *The 3D graphical user interface which is visible once a specific hand gesture is detected. The interface will be shown around the hand of the user. The left image shows the possible movement actions. The right image shows zones which are invisible to the user. If the hand/index finger enters one of these zones, the respective movement is triggered. The two techniques FingerUI and HandUI use different sizes for the zones (smaller zones for the IndexUI).*

to enable movement, the users locomotion speed increases over the first second up to a maximum of 28.8 km/h (8 m/s). The time it takes a user to rotate their body about 90° using hand gestures or the controller is 1.5 seconds. Movement is immediately stopped if the users' hand is no longer in a movement zone.

6.3.2. Explanation of Chosen Techniques

Techniques with different input modalities such as controller can be adapted or serve as metaphor to implement bare handed techniques for locomotion. With this in consideration, the proposed techniques were implemented. ThumbGesture was implemented since it is quite similar to rotating a thumbstick into the desired direction as it uses the direction of the thumb to indicate the movement direction. Furthermore, ThumbGesture can be seen as a variation of the locomotion technique introduced by Zhang *et al.* [318]. ThumbGesture however uses only one hand instead of two. FingerUI was developed to use the metaphor of pointing forward to enable movement. The shown 3D graphical user interface is similar to a digital pad on common controllers that allow movement of virtual characters. Previous studies suggest that the gesture for pointing forward could be error-prone due to tracking failures since the index finger is often obscured for the cameras by the rest of the hand [231]. For this purpose, HandUI was implemented which should be easy to track by the hand tracking device since no finger is occluded. Controller was added as a baseline and serves as the current gold standard for locomotion in VR. Only one-handed techniques were implemented, as one hand should be free for interaction tasks.

6.3.3. Evaluation

Objectives

The goal of this study was to compare the three locomotion techniques using bare hands. Controller was added as a baseline, to generally compare hand gesture locomotion with the gold standard. It was anticipated that a controller will outperform the bare handed techniques. However, the main objective was to find out which of the three bare handed techniques is best in terms of efficiency, usability, perceived workload, and subjective user rating. The efficiency of the different techniques was measured by the task completion time. The well known System Usability Scale (SUS) [24], [25] was used as usability measure. The perceived workload was measured by the NASA Task Load Index (NASA-TLX) [88], [89]. Since hand tracking is still a maturing technology and tracking errors are expected, NASA-TLX should give interesting insights into possible frustration and other measures. It was deliberately decided not to use more questionnaires to keep the experiment short. This was because it was expected that some participants would suffer from motion sickness and might decide to abort the experiment if it takes too long. It was also decided not to include any questionnaire for motion sickness as it can be expected that the proposed techniques are similar in this regard.

Participants

A total number of 16 participants participated in the study and 12 completed the experiment. Four participants cancelled the experiment due to increased motion sickness during the experiment. The participants' age ranged between 18 and 63 years old (Age $\mu = 33.38$). Six females participated in the study. All participants were laypeople to VR technology and wore a VR HMD less than five times.

Apparatus

The evaluation was performed by using a gaming notebook with an Intel Core I7-7820HK, 32 GB DDR4 RAM, Nvidia Geforce GTX 1080 running a 64 bit Windows 10. Meta Quest 2 was used as the VR HMD and the hand tracking was realized using version 38 of the Oculus Integration Plugin in Unity.

Experimental Task

The participants had to move through a minimalistic, corridor-like virtual environment and touch virtual pillars. The environment is 10m wide and 110m long. A total of ten pillars are placed in the environment about 10m apart from each other. The pillars are arranged in a way that users had to move left and right to reach the pillars (See Figure 6.20). After a pillar was touched, its color changed to green, indicating that it was touched. Once ten pillars were touched, a trial was completed.

Figure 6.20.: *The virtual environment used for the experiment. Users had to move in a large corridor-like environment, touching 10 pillars. After all pillars are touched once, the experiment continues with the next step.*

Procedure

The experiment had a within-subject design. Each participant had to move twice through the virtual environment with each technique. This allowed the subjects to understand and learn the technique in one trial and the latter trial can be used more reliable as measure for task completion time. A short video clip was shown to the participant to inform them how to move with the current technique. The experiment was conducted in seating position and users could rotate their body with a swivel chair. The order of locomotion techniques was counterbalanced using the balanced latin square algorithm. After a participant touched all ten pillars in the virtual environment twice, the participant was teleported to an area where questionnaires should be answered. Participants first filled in the NASA-TLX and then the SUS. The answers could be filled in with either the controller or using bare hands in VR. This was repeated for each locomotion technique. After the last, a final questionnaire was shown to the participant were they could rate each technique on a scale from 1 (bad) to 10 (good). One user session took about 30 minutes.

6.3.4. Results

Task Completion Time

For the task completion time, the time between touching the first and the last pillar is measured. The average time to touch all ten pillars in a trial is depicted in Figure 6.21. Levene's test assured the homogeneity of variances of the input data and therefore one-way ANOVA was used. The result F(3,47) = 8.817 with

p value < 0.01 showed significant differences between the techniques. The post-hoc test TukeyHSD revealed the following statistically significant differences between technique pairs: Controller-FingerUI $p < 0.001$; Controller-HandUI $p < 0.05$; ThumbGesture-FingerUI $p < 0.01$.

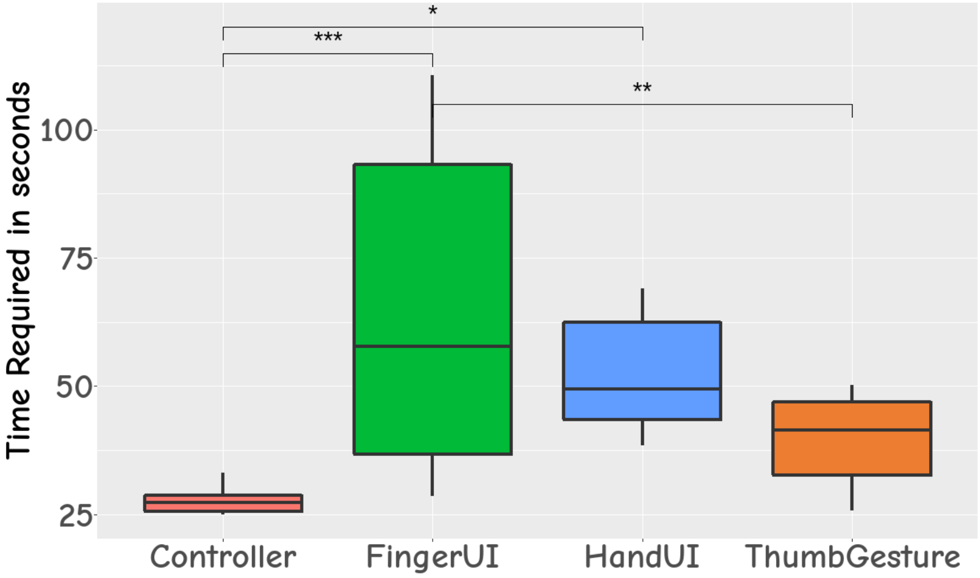

Figure 6.21.: *Average time taken by users to touch all ten pillars. Significance Levels: *** = 0.001; ** = 0.01; * = 0.05.*

NASA Task Load Index (NASA-TLX)

The NASA-TLX questionnaire was answered after performing the experimental task with a technique. A task took about two minutes to complete and the completion of the questionnaires allowed a break of about two minutes between each successive task. The raw data of the NASA-TLX is used without additional subscale weighting in order to further reduce the amount of time required by participants to spend in VR (Questionnaires were answered within the virtual environment). Using the raw NASA-TLX data without weighting is common in similar literature [30], [231]. The questionnaire measures the perceived mental and physical workload, temporal demand, performance, effort, and frustration of participants. The overall workload of the proposed techniques is calculated by the mean of the six subscales. The overall score for each technique in order from high to low: The highest perceived workload was using HandUI (M = 53.72), followed by FingerUI (M = 46.13), a slightly lower workload by using ThumbGesture (M = 41.55), and finally Controller (M = 37.92).

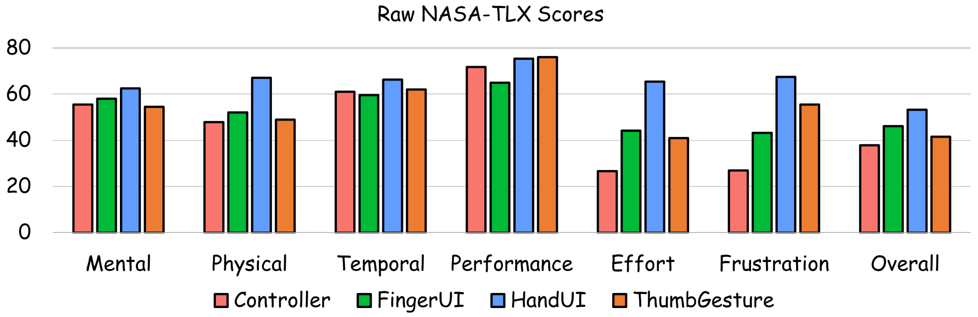

Figure 6.22.: *The raw NASA-TLX scores. Perceived mental and physical workload, temporal demand, performance, effort, and frustration are measured by using the questionnaire. The overall perceived workload is shown on the far right of the bar charts.*

System Usability Scale (SUS)

The SUS gives insight into the subjective perceived usability for the different techniques. Generally, a higher value means better perceived usability and a value above 69 can be considered as above average according to Sauro [226]. It is to note that the SUS scores of this evaluation are only meaningful within this experiment and should not be compared to SUS scores of techniques within other research work. The following SUS scores were achieved: Controller 66.1; FingerUI 62.9; HandUI 61.4; ThumbGesture 76.8. The scores are depicted in Figure 6.23.

6.3.5. Subjective Ranking of Techniques

Participants were asked to rate each technique on a scale from 1 (bad) to 10 (good). The techniques got the following average rating from users: Controller 8.5; FingerUI 6.42; HandUI 5.21; ThumbGesture 7.57. The scores are depicted in Figure 6.23.

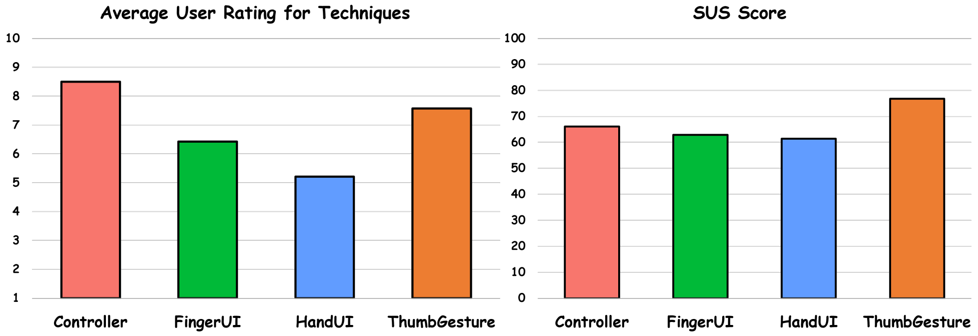

Figure 6.23.: *Left: Average user rating for the proposed techniques. Users could rate each technique on scale from 1(bad) to 10(good). Right: Results of the System Usability Scale.*

6.3.6. Discussion of the Study and Future Work

It was anticipated that the controller outperforms the hand gesture based techniques in task completion time. However, no statistically significant difference was found between Controller and ThumbGesture. Another noteworthy observation is that ThumbGesture received a better SUS score than Controller. This could be explained by the fact that all participants were laypeople to VR and therefore have minimal experience with using a controller which lead to a better usability rating.

No significant differences were found in the overall scores regarding the perceived workload of the techniques. However, it can be observed in Figure 6.22 that Controller required less effort and led to lower frustration by the participants.

Ranking of techniques was also in favor of Controller but ThumbGesture received similar results. Overall it can be said that ThumbGesture was the winner out of the three proposed one-handed locomotion techniques as it got the best SUS scores, highest user rating, and fastest task completion time. This leads to the conclusion that a one-handed technique for continuous locomotion should use a simple gesture for moving without an additional user interface.

Interestingly, some participants exploited the fact that turning the head also rotated the virtual character. Thus, only the gesture for moving forward was necessary to achieve the goal. A follow-up study could investigate whether gestures to change the direction of movement offer added value or if they are unnecessary. It was also interesting that three out of four subjects who stopped the experiment, stopped during the controller condition (the last participant interrupted at HandUI). This could be a hint that the controller actually causes more motion sickness than gesture-based locomotion. However, more data is required to support this hypothesis.

6.3.7. Study Limitations

Little research has been been performed on how bare hands can be used to move in virtual environments. Therefore, it is not yet clear which bare handed technique is performing well enough to compare it to other freehand techniques which are widely researched and acknowledged such as WIP. In that regard, once suitable bare handed locomotion techniques have been found, they should be compared to sophisticated techniques such as WIP. Only then can a well-founded insight be gained into whether hand gestures are a valid alternative.

The robustness of the bare handed techniques is highly dependent on the quality of the hand tracking solution. Some participants had problems with the gestures, even though they were quite simple. This was particularly noticeable with the ThumbGesture technique, where the virtual hand sometimes had an index finger pointing outwards, even though the physical hand was correctly shaped. Similar false hand configurations occurred once the index finger pointed outwards because the finger was covered by the cameras. Furthermore, no questionnaire for motion sickness was used. The experiment was designed without a questionnaire on motion sickness in order to keep it as short as possible, also so that subjects would not have to spend much time in VR. However, since some subjects dropped out due to motion sickness, an evaluation in this regard would have been useful.

Another limitation is the number of participants. Only a limited number of participants could be recruited due to the COVID-19 pandemic. More participants would be required in order to be able to draw stronger conclusions about the proposed techniques.

6.3.8. Conclusion of the Study

This work presents three one-handed techniques for continuous locomotion in VR. The techniques are compared with a standard controller implementation and the respective other techniques. The techniques are compared with respect to task completion time, usability, perceived workload, and got ranked by the participants. Controller was fastest in task completion time and got the highest rating from participants. In the other measurements, however, there is no clear winner between the use of a controller and one of the presented one-handed techniques for continuous locomotion. ThumbGesture even got a higher SUS score than Controller. Overall, it can be said that out of the three one-handed techniques, ThumbGesture was the winner in this experiment. This technique received the highest scores in the SUS and ranking by participants. Furthermore, it got lowest perceived workload out of the three one-handed techniques. It was also the fastest in task completion time among the bare handed techniques. The experiment conducted aimed towards using natural hand gestures for moving around in VR. The techniques presented show promising results overall, but further techniques should be evaluated to find potential suitable hand gestures for the locomotion task. This is especially important if physical controllers are to be replaced by hand tracking in the future or if controllers are not desired for an application.

Chapter 7

Hand Gestures for Virtual Object Manipulation

Besides locomotion, interaction with virtual objects is another essential type of interaction in immersive virtual environments. Using a controller to pick up virtual objects is unnatural because we use our hands to interact with objects in real life. When developing applications with immersive virtual environments that are intended to be as realistic as possible, it should be explored how hand gestures can be used to enhance this interaction. Even though recent hardware allows hand tracking out of the box, creating new gestures is not straightforward. Especially for people with little technical background, it is almost impossible to define their own gestures. For this reason, the framework proposed in chapter 5 was developed. However, the framework itself is in a raw state and allows diverse application of hand gestures. This chapter proposes a hand gesture authoring tool for object specific grab gestures allowing virtual objects to be grabbed as in the real world (see Figure 7.1). The presented solution uses template matching for gesture recognition and requires no technical knowledge to design and create custom tailored hand gestures. In a user study, the proposed approach is compared with the pinch gesture and the controller for grasping virtual objects. The different grasping techniques are compared in terms of accuracy, task completion time, usability, and naturalness. The study showed that gestures created with the proposed approach are perceived by users as a more natural input modality than the others.

Surveys about applications with such immersive technologies show the importance of this interaction. In section 3.3.3 it is shown that virtual object manipulation is ranked as the most common interaction type for remote collaboration scenarios. The most common way to pick up virtual objects is to press a button on a VR controller. With the recent rise of available hand tracking in affordable head mounted displays (HMDs) for XR applications, researchers and practitioners are exploring different ways to pick up objects. Without using controllers, the pinch gesture is the most common gesture to pick up objects. The software development kits for HMDs with hand tracking usually include a pinch gesture as the default gesture to pick up objects, as

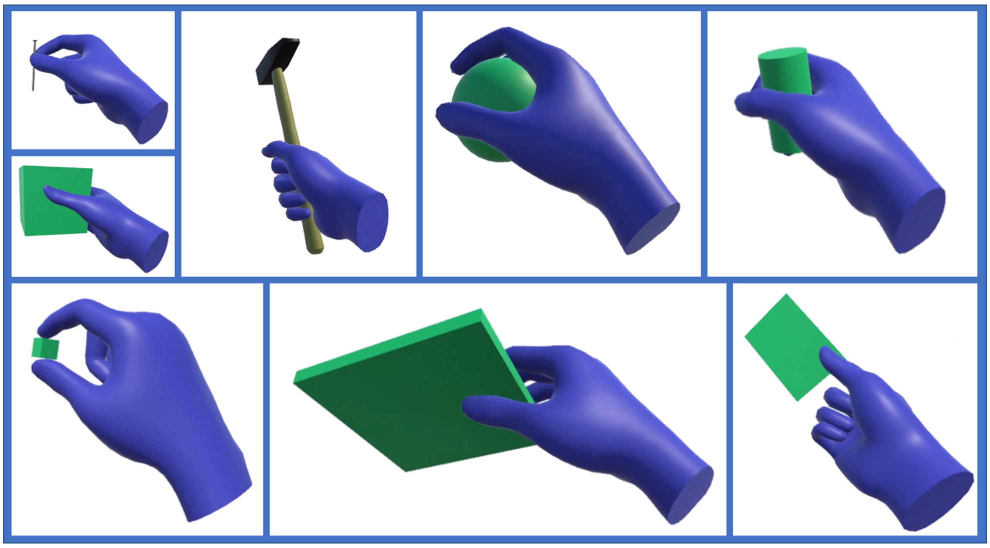

Figure 7.1.: *The proposed gesture authoring tool allows users to create their own custom tailored hand gestures for grasping virtual objects. The tool was evaluated with a user study and the grasping is compared to picking up with a controller and the pinch gesture.*

for example with the Hololens 2[1], Meta Quest[2], or the Leap Motion[3] to name the most common examples. An early example which used the pinch gesture to grasp virtual objects was introduced by Buchmann *et al.* [27]. As the pinch gesture is easy to recognise and can be reliably performed by users, it is the preferred method for vendors to showcase the capability of hand tracking technology. However, the pinch gesture is not optimal for many use cases. First, it is an unnatural gesture to pick up many objects because of their geometric properties. Second, the thumb and index finger need to be close to each other for a pinch gesture. This prevents some other gestures which require the thumb and index finger being close to each other from being recognised.

Previous studies show that users, and especially lay people to this technology, often try to grab virtual objects as if they were picking up objects in the real world such as observed by Kang *et al.* [113] and Schäfer *et al.* [230]. It is investigated how the current state of the art for picking up virtual objects with bare hands, the pinch gesture, can be replaced by more intuitive hand motions for users. The implementation and evaluation design of the proposed solution was guided by the formulation of three research questions:

- **RQ1:** Is a template-based gesture matching approach for picking up virtual objects reasonable?
- **RQ2:** Can users define and use their own gestures without help and technical knowledge?

---

[1] https://docs.microsoft.com/windows/mixed-reality/mrtk-unity/, last accessed at 20.10.2022

[2] https://developer.oculus.com/, last accessed at 20.10.2022

[3] https://developer.leapmotion.com/, last accessed at 20.10.2022

- **RQ3:** How do custom gestures compare to the state of the art for picking up virtual objects in terms of accuracy, task completion time, and perceived naturalness?

A user study was designed and conducted in order to answer each of these questions. The contributions of this chapter can be summarised as follows:

- A comprehensive user study consisting of two experiments that compare three techniques to pick up objects:
  - Controller as a baseline for comparison.
  - The pinch gesture representing the current state of the art for picking up virtual objects with bare hands
  - The proposed technique to pick up objects with customised hand gestures.
- A system for design, implementation, and prototyping of object tailored hand gestures to pick up virtual objects

7.1. Virtual Object Manipulation

Previous work on how to move virtual objects with bare hands in immersive environments has been conducted. Suzuki *et al.* [270] introduced an AR system to grab a virtual object with bare hands using the pinch gesture. The authors generate composite images to achieve occlusion of the virtual object by the real hand. Boonbrahm and Kaewrat [16] used the pinch gesture as well in an AR system for assembling small virtual models. Furthermore, the pinch gesture was used by Sorli *et al.* [261] to compare different hand visualisation techniques. Participants had to grab and place big and small cubes. Another example of using the pinch gesture for grabbing virtual objects is the work of Mu and Sourin [172]. In their work, two different implementations of the pinch gesture are compared. The work of Nguyen [177] uses hand features to detect a grasping gesture to overlay tools such as a hammer or screwdriver over the real hands using an XR HMD. Kang *et al.* [113] investigated how the interaction techniques Gaze and Pinch, Direct Touch and Grab, and their novel technique Worlds-in-Miniature compare to each other. It has been found that all techniques have advantages as well as disadvantages. One of the most important findings of the study conducted by Kang et al. is that users prefer a visual guide to the possible interactions regardless of the interaction method used. Grasping virtual objects using bare hands was investigated by Vosinakis and Koutsabasis [293]. Specifically, the authors investigated if different visual feedback techniques such as highlighting an object had an impact on usability. The virtual objects could be grabbed and released by closing and opening the hand. Song *et al.* [260] used a handle bar as metaphor to manipulate virtual objects with two hands. Khundam *et al.* [120] compared hand tracking and controller in a medical training scenario. Users had to pick up virtual objects and interact with virtual elements by using either real hands or controllers. The authors did not find any significant difference in terms of usability. Masurovsky *et al.* [151] compared controller and different pick up techniques with the hand tracking device Leap Motion. An important finding was that controller outperformed the other techniques and that controller was not perceived as more natural than hand gesture grasping. Similar results were proposed by Caggianese *et al.* [32] where grabbing objects with a controller was compared to a bare handed technique. Olin *et al.* [186] proposed a system for cross device collaboration in VR where users could pick up virtual objects with their hands. Another interesting approach was introduced by Pei *et al.* [194] where hand gestures are not used to pick up objects but to imitate an object. For example, instead of picking up a virtual scissor, the hand is shaped as a scissor to cut a paper. Hand gestures in the work of Pei *et al.* [194] are defined in a similar template matching approach than the proposed gesture capture technique.

The main difference with the existing works compared to the presented approach is that the gestures used in the aforementioned works to grasp virtual objects are not tailored to the shape of the objects.

7.2. Gesture Authoring in the Existing Literature

This section briefly presents relevant work on the simple creation of gestures that are not necessarily related to the grasping of virtual objects. Mo *et al.* [168] introduced Gesture Knitter, a system to design and implement hand gestures. Users are able to create their own gestures with a visual declarative script. Fine and gross primitives can be combined to create dynamic hand gestures. A system to create and prototype multi-touch gestures was introduced by Lü and Li [144]. Kinect Analysis was proposed by Nebeling *et al.* [175] where motion recordings captured from a RGB-D camera can be inspected and annotated. These recordings can then be used as gestures. Ashbrook and Starner [5] introduced MAGIC to create and prototype gestures with a three-axis accelerometer. Speicher and Nebeling [264] introduced GestureWiz which uses video input data to record and recognise gestures with a consumer-grade webcam. GestuRING from Vatavu and Bilius [289] introduced a web based tool to create hand gestures with a finger-worn device. Gestures from a database could be mapped to certain actions. For example, the gestures could be used to navigate a menu when a ring was rotated. Another web based tool is introduced by Magrofuoco *et al.* [147] for the creation of stroke gestures using a 3D touchpad. A tool to evaluate and create micro gestures was introduced by Li *et al.* [139]. The systems presented in this section allow for easy creation of new gestures, but none are designed to make virtual objects naturally tangible by allowing one to create custom hand gestures.

7.3. Gesture Authoring to Pick up Objects With Custom Tailored Hand Gestures

This section describes how the gesture authoring process was implemented. First, the semi-automatic steps for authoring a gesture are described. Next, the architecture of the proposed solution is presented.

7.3.1. The Two-Step Gesture Authoring Process

The Gesture Authoring Space uses a simple two-step mechanism in order to allow custom tailored hand object interaction.

1. The user needs to place the desired hand near a virtual object, imitating a grab interaction.
2. After the hand is kept still for three seconds, the gesture is captured and coupled to the object.

This process is depicted in Figure 7.2. Two necessary conditions were identified in order to enable this two-step mechanism: First, the user has to place the hand near an object. The distance between the virtual object and the hand is computed which is then used to recognise if the hand is close enough to the desired object. Secondly, the user needs to keep the position of the hand for a certain amount of time. The user gets visual feedback how long the hand must be maintained in form of a percentage going up from 0% to 100%. Once 100% is reached, i.e. the user kept the hand still for a certain amount of time, the object is attached to the hand and can be released by changing the hand shape to a different pose than the captured gesture (e.g. opening the hand). The first condition is required to store the actual hand shape for the grab gesture. The second condition is necessary in order to capture the gesture when the user finished placing the hand around the object. The desired virtual object is placed on a table within the virtual environment. This table provides information about the two conditions so that the user is informed about the current status of the gesture capturing process. Virtual buttons are placed in front of the user which allows to switch the virtual object to a different one as well as resetting the position and all attached gestures on the current object. In the current implementation only one custom gesture can be attached to an object but this can easily extended to multiple gestures for a single object.

7.3.2. System Architecture

The gesture authoring process requires four principal components: A hand tracking provider, gesture capture, gesture recognition, and the object interaction logic. The interaction capabilities such as attaching a virtual object to the hand were implemented by using a built-in framework of the game engine Unity[4].

---

[4] https://unity.com/de, last accessed 31.08.2022

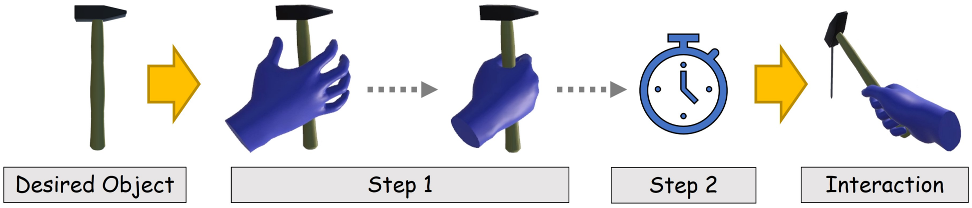

Figure 7.2.: *The gesture authoring workflow. A chosen 3D object is placed inside the virtual environment. Then, the user has to imitate the grab of the object by shaping the hand as desired. Finally, the hand needs to be in position for three seconds in order to capture the desired grab gesture. In the end, the object can be grabbed and used for more interaction.*

The proposed gesture capture and recognition technique was developed through customisation of the framework proposed in chapter 5. The recording process will store the currently formed hand shape while the recognising component will compare these stored values with the live hand tracking data. By calculating the Euclidean distance between hand joints of the stored and live data, gestures can be recognised. The similarity S_g of a gesture is recognised by comparing each hand joint (25 provided by the hand tracker) to a stored template by calculating the spatial differences of joints. With j_c being the current joint position provided from the tracker and j_{gt} the joint position in the gesture template as shown in equation 7.1.

$$S_g = \sum_{i=1}^{25} \sqrt{(j_c - j_{gt})^2}, \text{ for g } = 1{,}2 \ldots, \text{ N} \tag{7.1}$$

The hand shape recognition can be done strictly or loosely by adjusting a threshold which will mark when a hand gesture should be detected. The capture process is depicted in Algorithm 3 and the recognition is explained in Algorithm 4. A threshold of 5 cm was used in order to recognise a gesture, i.e. the combined euclidean distance of hand joints (S_g) in the current hand tracking frame should not exceed 5 cm to the stored template. This is an empirical value that proved suitable in initial pilot tests, as it was neither too strict nor too loose in recognising certain gestures.

Hand gestures for grabbing virtual objects in the Gesture Authoring Space are attached to specific objects rather than a global storage for gestures. Virtual objects enter a hover state if a users hand is near them (hover radius is about 10 cm) and gets ”unhovered” if the hand is too far away (depicted in Figure 7.3). Once a tangible object is hovered, all attached gestures for grabbing it are registered to the gesture recogniser. The gesture recogniser will then search for hand shapes associated to the gestures in each hand tracking frame that arrives. Unhovering the object will unregister all gestures attached to the tangible object. This allows many different gestures to be recognised without worrying about falsely recognised gestures since the gestures can only be activated when they are actually desired. If two similar gestures are detected, the gesture with the smallest spatial difference will be triggered.

Algorithm 3 Capturing Gestures

```
Desired hand shape is formed around desired object by the user
Gesture capture event is triggered by holding hand and fingers still (3 seconds in the user study)
Extract joint positions from current hand pose
Create new gesture object
for Each joint on the hand do
    Transform joint position from world space to local space
    Adjust joint position with hand scaling factor
    Store joint position in gesture object
end for
Fill new gesture with the stored values and attach it to the virtual object
```

Algorithm 4 recognising Gestures

```
New hand frame arrives
if Hand is near an object with attached hand gestures then
    Get current hand frame $H_{current}$ from hand tracker
    for Each registered gesture in the nearby object do
        Set $D_{minimum}$ to Float.Maximum value
        for Each joint on the hand do
            Transform joint position from world space to local space
            Adjust joint position with hand scaling factor
            Calculate distance $D$ between stored joint position and current position
            if $D > Threshold$ then
                Discard current gesture $G$ (hand shape is not matching)
            end if
            Add $D$ to $D_{Sum}$
        end for
        if $D_{Sum} < D_{minimum}$ then
            Set $D_{minimum}$ to $D_{Sum}$ (store the smallest distance)
        end if
    end for
    if $G$ is not discarded AND $D_{Sum}$ is the lowest between gestures then
        Current gesture is detected and virtual object is grabbed
    end if
end if
```

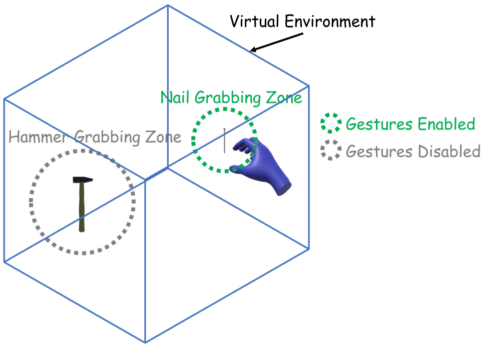

Figure 7.3.: *Example of context aware gestures: While being near a virtual object, certain hand gestures are enabled. If the hand is too far away from a virtual object, hand gestures for this object are disabled.*

7.4. Evaluation

The Gesture Authoring Space (See Figure 7.4) was evaluated with two experiments. The main objective of the evaluation is to compare the created gestures with the pinch gesture. The pinch gesture is considered because it is widely used and the standard solution for many hardware manufacturers regarding bare handed interaction. This includes the Microsoft Hololens 1 and 2, Leap Motion, Meta Quest 2. Controller was added as a control variable as it is the current gold standard for interaction in VR. When grasping virtual objects, user preference, accuracy and task completion time are the most important metrics found in the literature. Furthermore, the research questions mentioned in the beginning of this chapter should be answered. Therefore, the following evaluation was considered:

First, since the Gesture Authoring Space uses template matching as gesture recognition, it should be investigated how a template-based gesture compares in relation to the pinch gesture and the controller. Accuracy and task completion time are used as metrics in this evaluation step. The accuracy is determined by how close a virtual object was placed to its target. The task completion time represents the speed a task was considered completed. It should be mentioned that there was no gesture authoring in this step and the used gestures were recorded/created by one of the authors.

Secondly, the gesture authoring process should be evaluated in terms of usability to gain insights if it can be reliably used by participants. More precisely, it should be investigated if users, especially lay people, can create and use custom tailored hand gestures for grabbing virtual objects. Furthermore, it should be explored how the custom tailored gestures compare to the pinch gesture and controller in terms of accuracy, task completion time, usability, and naturalness.

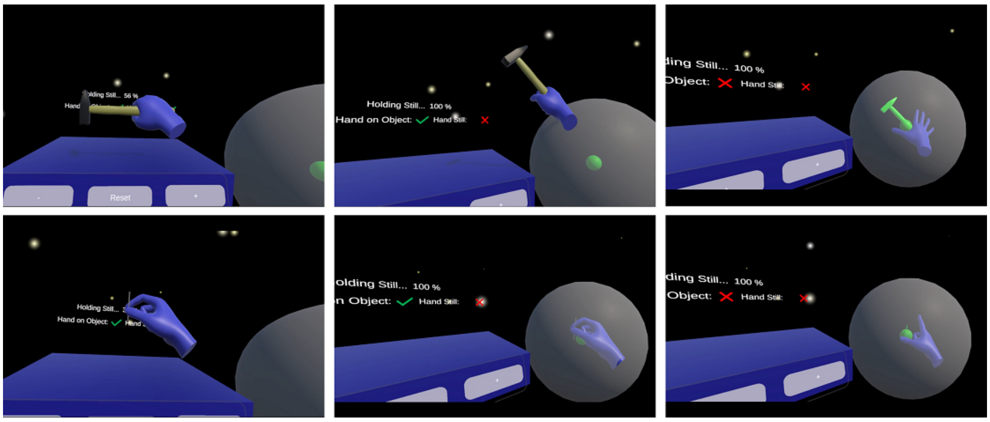

Figure 7.4.: *The Gesture Authoring Space. A user wraps the hand around a virtual object (left). After three seconds, the object can be grabbed and moved (middle). The object is then placed inside a transparent sphere (right) for the experimental task to measure accuracy and task completion time. The depicted objects are Hammer (top) and Nail (bottom).*

7.4.1. Apparatus

The evaluation was performed using a gaming notebook with an Intel Core I7-7820HK, 32 GB DDR4 RAM, Nvidia Geforce GTX 1080 running a 64-bit Windows 10. Hand tracking was realised using the Oculus Quest 2 VR HMD. The game engine Unity was used to develop the system.

7.4.2. First Experiment: Compare Template-Based Gestures and Pinch Gesture

Implementation of Grasping Techniques

As grasping virtual objects with controllers or the implementation of a pinch gesture can vary, it is briefly described here for reproducibility.
Pinch. The Pinch gesture was implemented by utilising thumb and index tip positions provided from the chosen hand tracking solution. The distance between those two points is measured for each frame. A threshold determines if a user is currently pinching. If thumb and index finger are closer than 3 cm to each other, pinching is activated (this is an empirical value). It was found that there was a critical area where noise affected recognition. For example, if the fingers were held about 3 cm apart (2.9 - 3.1 cm), the system would jump between pinch detected and no pinch detected. Therefore, it was introduced that the state of the gesture would only change if the same value was reported for 100 ms which resulted in an overall much smoother user experience.
Grab. The Grab gesture was implemented by using the aforementioned gesture capturing process. However, it is a generic hand gesture not tailored to a specific object, no virtual object was used to capture it (See Grab gesture in Figure 7.5). Two different static gestures are required for this approach: One for initiating the grab and one for releasing the object. Therefore, a

static gesture resembling a closed hand and a static gesture with a relaxed hand are stored. The gestures are rotation invariant. Detecting a closed hand will initiate a grab event to nearby objects while detecting a relaxed hand will release the currently grabbed object. It was decided to include two static gestures for the release state: One with the hand partially open and one with the hand fully open. Ideally, the gesture with the hand partially open releases the object.

Controller. Grabbing and releasing an object with the controller is performed by pressing the grip button on the VR controller. This is also depicted in Figure 7.5.

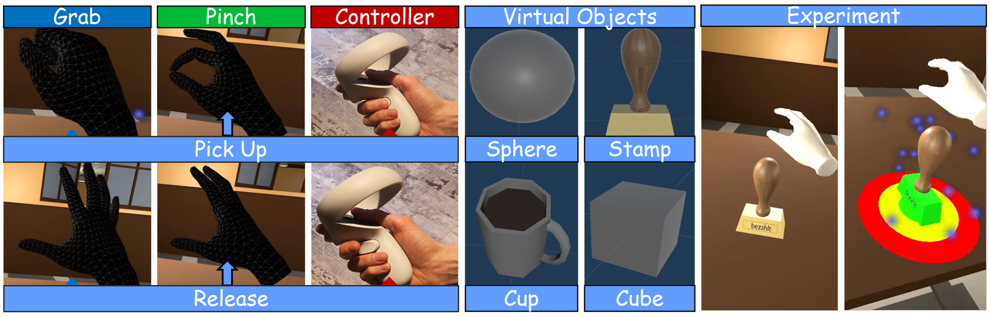

Figure 7.5.: *The first experiment compares three different pickup techniques for grabbing and placing virtual objects. The gestures Grab and Pinch as well as Controller are compared. Participants are required to place the four virtual objects Sphere, Stamp, Cup, and Cube on the centre of a target. The Grab gesture was implemented using a template matching approach.*

Objectives

The main objective of this evaluation was to ensure that template-based gestures can reliably be used. Especially lay people who never wore an HMD should be able to perform the given tasks and pick up virtual objects. Additionally, the template-based gesture should perform at least similar to the pinch gesture, which is usually the standard gesture for hand-gesture interfaces. Accuracy and task completion time are used as metrics.

Users

A total of 18 users (6 female) participated in the experiment (Age $\mu = 33.5$). A 5-Point Likert scale (ranging from 1 to 5) was used to assess the users subjective VR experience which resulted in an average of $\mu = 2.1$ (higher value means more experience).

Task

The experiment was conducted as within-subject design. Users sat in front of a virtual desk and had to place a virtual object which appeared in front of them. A target appeared on the table were users had to first grab the virtual object and then place it on the target. Depending on how close the users placed the object, the object gave visual feedback on how close it was to the center. Red for far, yellow for near, and green for being very close to the center of the target. The participants were instructed to place the virtual object as fast and as close as possible to the center of the target. The three interaction techniques Grab, Pinch, and Controller were performed in succession. Users got a quick introduction how to grasp an object with each technique and had time to practice each technique for a short time. Figure 7.5 depicts this experiment.

Procedure

The order of techniques was counterbalanced using the Balanced Latin Square algorithm. Each technique could be practiced for a short time by participants. Four objects had to be placed 10 times respectively, resulting in a total of 120 placed objects for each participant.

Quantitative Results

In total, 2.160 virtual objects have been placed in this first experiment of the evaluation (720 per technique). The dependent variables *Accuracy* and *Task Completion Time* are shown in Figure 7.6. Levene's test assured the homogeneity of the input data ($p > 0.05$) and therefore one-way ANOVA was used for statistical analysis. The ANOVA result ($F(2, 51) = 16.95; p < 0.001$) showed significant differences between the techniques in *Task Completion Time*. *Accuaracy* showed significant differences as well with the ANOVA

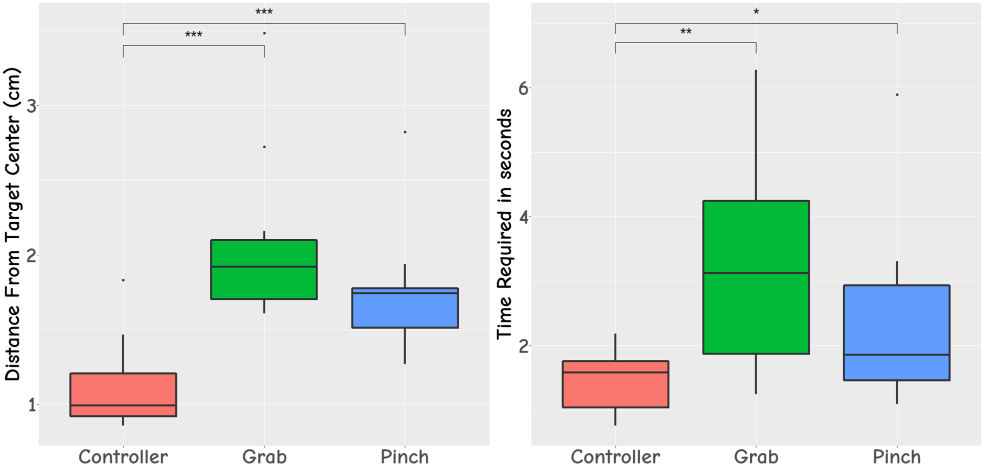

Figure 7.6.: *First Experiment: Three interaction techniques are compared to each other by grabbing and placing virtual objects. Accuracy is shown left and the task completion time on the right. Significance Levels: *** = 0.001; ** = 0.01; * = 0.05.*

result ($F(2, 51) = 7.108; p < 0.01$). Tukey's Honest Significant Difference (TukeyHSD) was used as post hoc analysis of the data. Controller was significantly faster in picking up and subsequently placing an object as compared to the other techniques. Controller was also significantly more accurate than Grab and Pinch. Grab and Pinch are not significantly faster or more accurate compared to each other.

Qualitative Results

To evaluate the subjective user experience, participants had to answer the System usability scale (SUS) [24], [25]. Each time a task was completed with a technique by placing 40 virtual objects, the SUS questionnaire appeared in the virtual reality. After all three techniques were performed, a final questionnaire was shown to the user. This questionnaire allowed users to rate each technique from 1 (bad) to good (10). The SUS scores are the following: Controller 90; Grab 75; Pinch 86. The subjective user rating ranked Grab the worst with an average of 3.75, followed by Pinch with an average of 8 and Controller was rated best by users with an average rating of 9.5. The results are shown in image 7.7.

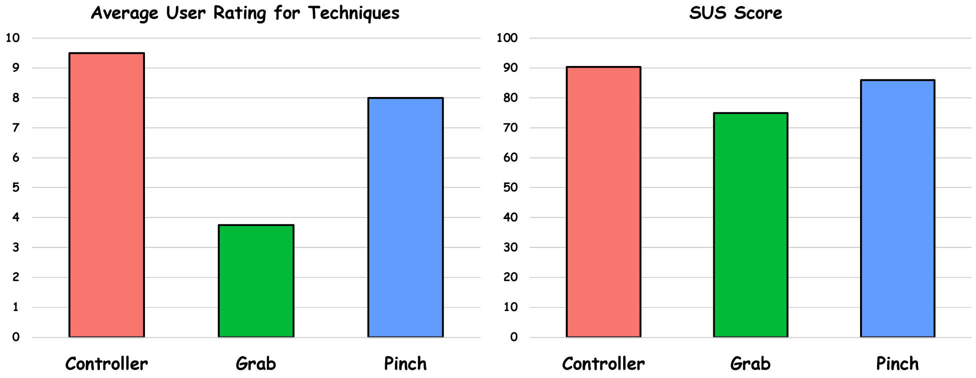

Figure 7.7.: *First Experiment: Average user rating for each technique on a scale of 1 - 10 (left) and SUS scores (right).*

7.4.3. Second Experiment (Part 1): Gesture Authoring

The second experiment was performed three months after the first and consisted of two parts. The first part focuses on evaluating the gesture authoring process and the second part on the performance of the gestures itself. Five participants from the first study also participated in this experiment.

Objectives

The main objective of the first part of this experiment was to investigate how users can create custom tailored gestures for virtual objects. Specifically, it was investigated if users can create and reuse hand gestures without expert knowledge.

Users

A total of 18 users (10 female) with age ranging between 18 and 62 participated in the experiment (Age $\mu = 29.7$). A 5-point Likert scale (ranging from 1 to 5) was used to assess the users subjective VR experience which resulted in an average of 1.83 (higher value means more experience). Only three participants rated their VR experience higher than two.

Task

The experiment was conducted as within-subject design. Users sat in a virtual space with minimal surroundings. A virtual object was placed in front of them. This object was used to start the gesture authoring process. Participants were instructed to wrap their hands around the object as if they would grab them in real life. Then, users had to wait three seconds in order to capture the desired gesture. After the three seconds, the gesture was captured and the object was attached to the users hand. It is to note that these three seconds were empirically determined and could be reduced. If the hand was not kept still, the gesture authoring process is aborted and restarted. To determine if a hand is kept still, the palm and all individual finger tips should not move farther than 1 cm in 10 consecutive hand tracking frames. Changing the hand shape releases the object. Users were allowed to grab and place the object in a transparent sphere which made it disappear. This way, users had a reason to grasp an object and were prepared for the task of grasping and placing in the next part of the experiment. The sphere had a diameter of 50 cm while the largest object (hammer) was about 25 cm long. Three buttons within reach of the user allowed to change the virtual object. Two buttons allowed the user to switch to the next or previous object while the third would reset the current object to its initial position and removes all hand gestures attached to it. A total of eight different objects were used in the experiment: Nail, Cube, Small Cube, Hammer, Ball, Plate, Cylinder, and Paper (as shown in Figure 7.1). These objects were chosen in order to include big and small objects as well as simple and complex geometric shapes. The gesture authoring task is depicted in Figure 7.4.

Procedure

Once the user put on the VR HMD, they were immediately in the Gesture Authoring Space. The users could try out different gestures and objects without a strict time limit. However, the experimenter kept this to a maximum of 15 minutes. After the participants captured at least one gesture for each object, a questionnaire was shown to the participants. This questionnaire targeted the subjective user experience.

Results

The following questions were asked after a participant successfully performed the gesture authoring task (5-Point Likert scale ranging from 1 to 5):

- **Q1:** The presented system is useful.
- **Q2:** The presented system is easy to understand.
- **Q3:** I was able to create the gestures I had in mind.
- **Q4:** The presented system is easy to use.
- **Q5:** The fingers of the virtual hand moved exactly like my real hand.

In general, the Gesture Authoring Space was well received by users who found it useful, easy to understand and use, and were able to create the gestures they had in mind. The results of the questionnaire are depicted in Figure 7.8.

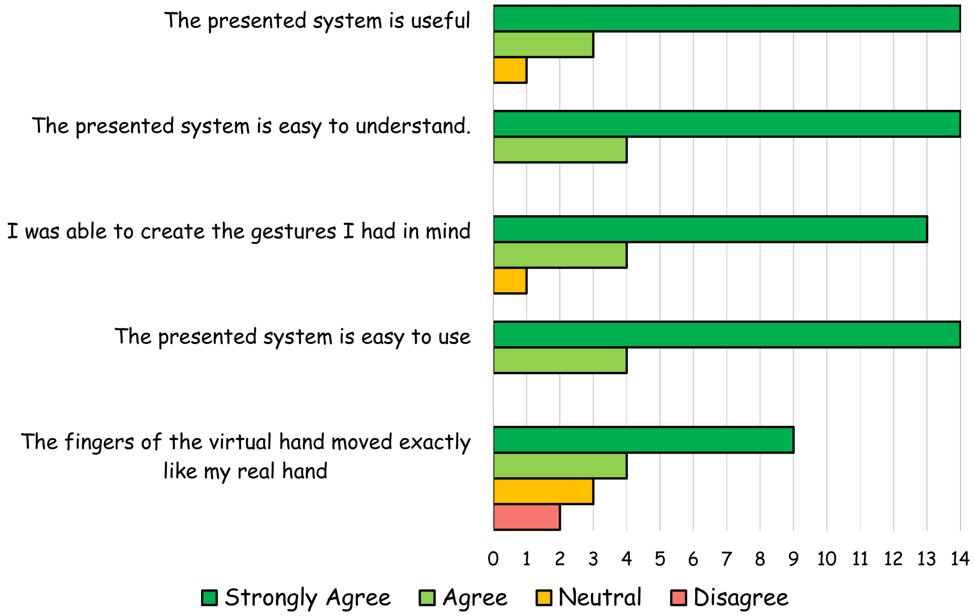

Figure 7.8.: *Questionnaire results after participants performed the task.*

7.4.4. Second Experiment (Part 2): Evaluate Custom Tailored Gestures

Objectives

The main objective of the second part of this experiment was to investigate how custom tailored gestures perform compared to a controller and the pinch gesture. The pinch and controller grasping is implemented as stated in section 7.4.2. The experiment is similar to part one, but the generalised grab gesture was replaced with hand gestures created from the gesture authoring process. The three techniques Controller, Pinch, and Custom Gesture are compared in terms of accuracy, task completion time, usability, and naturalness. Part two of the experiment was conducted directly after the first.

Task

The experiment was conducted as within-subject design. As in the first part of the experiment, participants sat in a virtual space with minimal surroundings and a virtual object was placed in front of them. This object should be grabbed with one of the three interaction techniques. Participants had to grab the object in front of them and place it in a transparent sphere, the target area, to measure the accuracy of a technique. The subjects got a quick introduction how to grasp an object with each technique and had time to practice each technique for a short time. The participants were told to move the virtual object as close as possible to the center of the transparent sphere, which was indicated with an opaque point in the middle of it. Once an object is placed inside the target area, it will disappear after one second. The sphere was the same as in experiment part one and had a diameter of 50 cm. The target area was always within reach of the user but changed its position subsequently after placing an object. This was used to check whether the gestures are still reliable when the user extends his/her arm or moves it from left to right and vice versa. Eight different objects should be grabbed and placed three times with each technique. This resulted in 72 placed objects for each participant (1.296 objects placed in total). The objects are: Nail, Cube, Small Cube, Hammer, Ball, Plate, Cylinder, and Paper (As shown in Figure 7.1).

Procedure

The order of techniques was counterbalanced using the Balanced Latin Square algorithm. Participants had to complete the grab and place task for a technique and then the SUS questionnaire was answered within VR. The questionnaire was answered either with the controller or with hand gestures, depending on which technique was previously used. A final questionnaire is shown after the last SUS questionnaire. The aim of this questionnaire is to allow participants to rate each technique on a scale from 1-10 and rate how natural the interaction was on a 5-Point Likert scale (ranging from 0 to 5).

Quantitative Results

Once a participant placed an object, the distance between the center of the object and the target area is stored. Furthermore, the time between grabbing an object and placing it inside the target area is recorded as well. The dependent variables *Accuracy* and *Task Completion Time* are shown in Figure 7.9. Levene's test assured the homogeneity of the input data ($p > 0.05$) and therefore one-way ANOVA was used for statistical analysis. Tukey's Honest Significant Difference (TukeyHSD) was used as post hoc analysis of the data.

Regarding *Task Completion Time*, the ANOVA result ($F(2, 51) = 8.726; p < 0.001$) showed significant differences between the techniques. Controller was significantly faster than Pinch ($p < 0.05$) and Custom Gesture ($p < 0.01$). However, the pinch gesture showed no statistically significant difference towards the custom gestures. In terms of *Accuracy*, the ANOVA result showed no significance between techniques ($p > 0.05$). Significance between techniques is depicted in Figure 7.9. It is to note that the accidental dropping of an object was also recorded. However, not a single object was accidentally dropped by users.

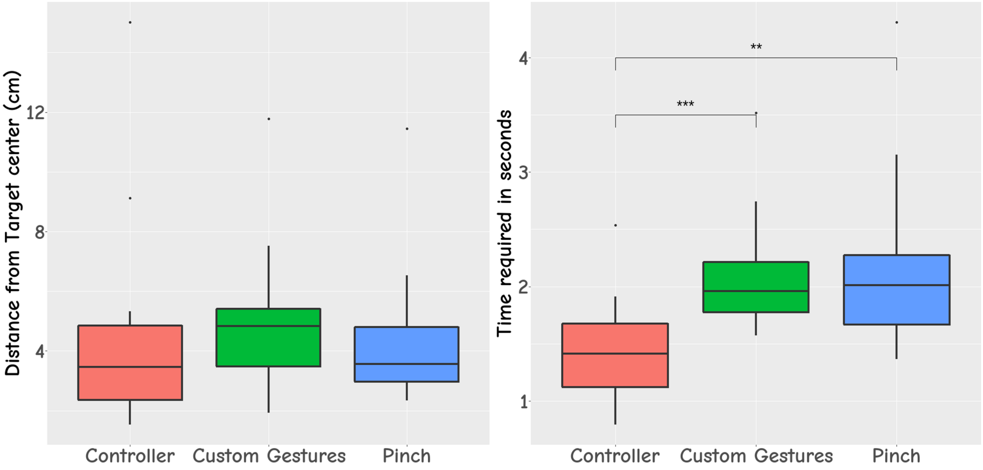

Figure 7.9.: *Quantitative results of the second experiment. Custom gestures were created by the participants following the gesture authoring process. The accuracy of techniques is depicted on the left and task completion time on the right. Significance Levels: *** = 0.001; ** = 0.01; * = 0.05.*

Qualitative Results

Each technique was rated with the SUS by participants (See Figure 7.10). The

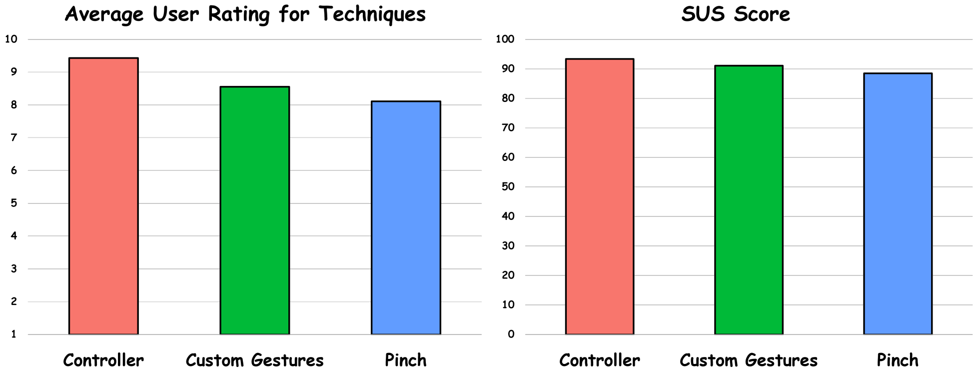

Figure 7.10.: *Average user rating on a scale of 1 - 10 (left) and SUS scores (right) for the techniques.*

SUS scores for each technique are the following: Controller 93.3; Custom Gesture 91.1; Pinch 88.5. After performing each technique, a final questionnaire is shown. This questionnaire allowed participants to rate each technique on a scale from 1 - 10. The ratings of techniques are the following: Controller 9.4; Custom Gesture 8.5; Pinch 8.1. In addition, for each technique, participants were asked a question about how natural a technique felt on a 5-point Likert scale (ranging from 1 to 5). The participants' answers for perceived naturalness of each technique is depicted in Figure 7.11. Generally, most users found that moving objects with custom tailored hand gestures felt very natural while using a controller felt unnatural to many. Using the pinch gesture, most users had a neutral attitude.

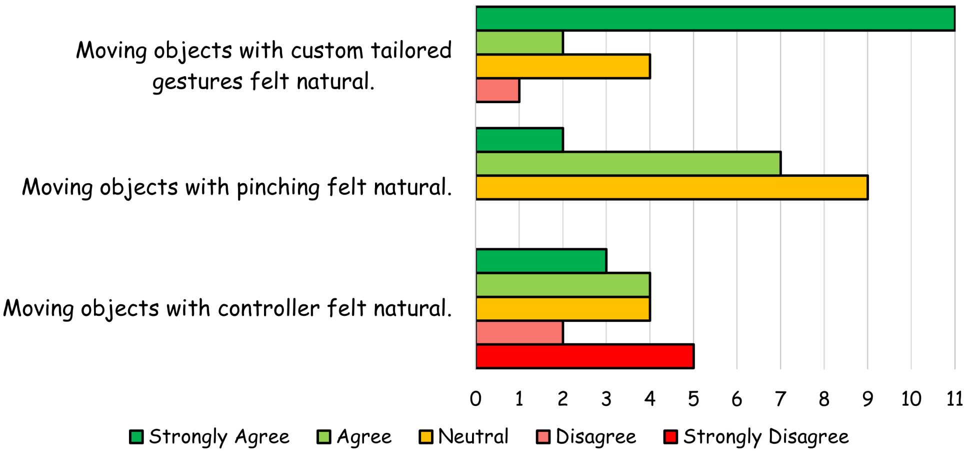

Figure 7.11.: *Evaluation results of the second experiment. Participants were asked which technique felt natural when grabbing objects.*

7.5. Discussion

The results from the first experiment gave interesting insights into how template based gestures perform compared to controller and the pinch gesture. Not surprisingly, controller was outperforming the bare handed techniques in terms of task completion time and usability. While the quantitative results show that Grab and Pinch are not significantly different in terms of accuracy and task completion time, the user rating was in favor of Pinch. This rating is likely due to the fact that the pinch gesture is easier to perform and recognise. A template-based gesture requires a relatively precise hand configuration, whereas the pinch gesture depends only on the position of the thumb and index finger. The aim of the experiment was achieved by showing that Pinch and Grab are quantitatively almost indistinguishable, but subjective opinion clearly goes towards the pinch gesture. Therefore, the next experiment investigated how object tailored custom gestures can be designed and implemented to compare it with Pinch and Controller instead of another generic gesture such as Grab in the first experiment. In particular, it should be investigated whether custom gestures perform better than Grab on subjective assessments when compared to Pinch and Controller.

In the second experiment, the Gesture Authoring Space was well accepted by the users. Participants created complex gestures which could be recognised without issues. The lowest rating (neutral score) came from a participant who had problems in replicating a gesture for a small object. This was due to the fact that the gesture capturing process was triggered when the virtual hand was not properly mapped to the real hand due to hand tracking failures. The participant then tried to grab the object as desired but the gesture was captured differently.

Participants rated the system as useful, easy to understand, and easy to use. Additionally, the participants were able to create the gestures they had in mind. It should be noted that some participants had problems creating complex hand configurations when the view of subsequent fingers was blocked. The gesture authoring process is therefore highly dependent on the quality of the hand tracking system used. This was anticipated before the experiment, which is why question 5 (*The fingers of the virtual hand moved exactly like my real hand*) was included in the questionnaire.

The second part of this experiment used gestures from the Gesture Authoring Space. The qualitative analysis showed promising results in the custom questionnaires as well as the SUS. Sauro [226] concludes that a total SUS score above 68 is considered above average. Albert and Tullis [2] state that a score above 70 can be interpreted as acceptable usability. It was observed that all techniques are above that threshold. Furthermore, the custom tailored hand gestures scored higher in the SUS than the pinch gesture. It should be noted that the SUS was initially designed for more complex systems rather than evaluating a simple system for grabbing objects. However, direct comparison of SUS scores between different techniques gives insights how they compare to each other in terms of perceived usability. The subjective scoring of a system on a scale from 1 - 10 was also in favor of the custom tailored gestures as compared to Pinch. The single template-based grab gesture in the first experiment scored relatively low with a score of 3.75. However, the custom tailored gestures scored 8.5 which indicates a significant improvement. Unsurprisingly, Controller outperformed the other techniques in terms of accuracy, task completion time, and usability which is consistent with existing research findings [32], [151], [230].

No accidental drop of a virtual object was recorded with each technique. This is rather surprising since Masurovsky *et al.* [151] reported significant differences between objects dropped by different techniques where a hand gesture interface had a mean of 5 dropped objects (out of 30) across participants. This is probably due to todays improved hand tracking and the presented techniques are quite robust. An accidental drop in the proposed system would be recorded once an object is grabbed and placed outside the target area. Once the virtual object was in the target area it counted as a correct placement. Some participants did unintentionally place an object because the hand tracking failed sometimes when participants were reaching out a full arm length. However, it was counted as correct placement since this happened in the target area. These "accidental drops" are therefore reflected in the accuracy as the objects were placed quite far away from the center of the target area.

Regarding naturalness, the custom tailored gestures received best results. Previous studies reported that grabbing virtual objects with bare hands did not feel more natural compared to a controller [120], [151]. However, moving objects with custom tailored gestures as presented in this work has received better rating by users in terms of naturalness.

7.5.1. Answering the Research Questions

RQ1: Is a template-based gesture matching approach for picking up virtual objects reasonable?

To answer this question, the first experiment was conducted. It can be said that a template-based hand gesture was on par to the pinch gesture but inferior to the controller. In terms of accuracy and task completion time there was no significant difference to the pinch gesture, showing that it can be a viable bare handed alternative. In terms of usability the template-based gesture was ranked the lowest but still got above average usability results with a SUS score of 75. However, it became only a score of 3.75 out of 10 from users. All of these results indicate that template-based gestures are useful but it is highly depending on which gesture is actually designed. Therefore, the Gesture Authoring Space aims to give users the opportunity to create their own gestures with a template matching approach.

RQ2: Can users define and use their own gestures without help and technical knowledge?

Experiment two was conducted to answer this question. Observing the users as well as analysing the questionnaire shows promising results. Even people with no technical background could use the system seamlessly and without noticeable issues. The questionnaire underlines this assumption. Therefore, it can be said that users can define and use their own gesture without help and technical knowlege with the presented system.

RQ3: How do custom gestures compare to the state of the art for picking up virtual objects in terms of accuracy, task completion time, and perceived naturalness?

The current gold standard for moving virtual objects is using a controller. As far as the state of the art in moving virtual objects with bare hands is concerned, the pinch gesture has been placed in this position to answer this research question. Controller was not significantly more accurate than the bare handed techniques. However, controller was significantly faster than both bare handed techniques. The custom gestures were not significantly faster or more accurate than the pinch gesture. Users' average rating (from 1 to 10) gave custom gestures a higher rating than Pinch. Additionally, the SUS score of custom gestures was marginally higher than Pinch. However, custom gestures are perceived as more natural than both Pinch and Controller. Therefore, it can be said that the customised gestures from the Gesture Authoring Space are a more natural input compared to the others.

7.6. Limitations

Controller still received best scores in the SUS, accuracy, and task completion time across experiments. Therefore it is still recommended to use controller for simple tasks which require fast movement and high precision. It has yet to be investigated how bare hand interaction compares to a controller in more complex rather than a simple grab and place task.

It should also be noted that experiment one and two had grab and place tasks but differed slightly. The first difference is that the first experiment used a different virtual environment. The second difference is that in the first experiment there were only four objects while in the second experiment there were eight. It was found that for custom tailored gestures there should be more objects in order to properly evaluate custom tailored gestures. This led to different results regarding accuracy, task completion time, and usability for Pinch and Controller in experiment one and two. There is also the limitation regarding the number of participants. Due to the COVID-19 pandemic, only a limited amount of participants could be recruited. The Gesture Authoring Space is also limited to one-handed gestures for grasping objects. Grabbing an object with custom tailored gestures for both hands simultaneously is not yet supported.

7.7. Conclusion

This work presents the Gesture Authoring Space to create custom tailored hand gestures for grabbing virtual objects. The proposed solution does not require users to have expert knowledge to design and create gestures they have in mind. A two-step mechanism is used to capture desired hand gestures: 1) Users wrap the hand around this object as if they would grab it in the real world. After the hand is kept still for three seconds the hand gesture is captured. The system was evaluated with two experiments. The first experiment aimed to investigate how the proposed template matching approach compares to state of the art techniques. It was found that a template-based gesture is a viable choice regarding accuracy and task completion time but still lacks behind in usability. Based on these results, experiment two used the template matching approach to record and recognise more natural gestures. The first part of the second experiment evaluated if users can define and use their own gestures without help and technical knowledge. It was found that users with different background knowledge can effectively use the proposed technique and could create the gestures they had in mind. In the second part of this experiment, the custom tailored hand gestures are compared to a controller and the pinch gesture. Gestures created by the Gesture Authoring Space performed better than the initial grab gesture from the first experiment. Specifically, it was found out that controller is still outperforming the bare handed techniques in terms of accuracy, task completion time, and perceived usability. However, the custom tailored gestures outperforms the other techniques regarding naturalness.

Chapter 8

Conclusions

8.1. Goals of the Thesis

The goal of the thesis was to develop novel interaction techniques for immersive environments. It also aimed to investigate how interaction takes place in current systems and how it can be improved. For this purpose, a survey about immersive technologies in the context of remote collaboration was conducted. A first prototype combining novel hand gestures and a virtual environment based on panorama images was implemented. Additionally, essential interactions such as locomotion and virtual object interaction in VR should be advanced with new techniques. Therefore, several locomotion techniques based on hand gestures are presented, compared, and evaluated. The unique implementation details are discussed and comprehensively evaluated by users. Another goal of the thesis was to develop new ways of creating and testing possible gestures. To this end, the thesis discusses the implementation of a framework for arbitrary one-handed gestures ready to be used for immersive applications based on XR technologies. The framework was evaluated in a user centred manner.

8.2. Summary of Thesis Achievements

The achievements of the thesis cover a range of interaction techniques for immersive applications using XR technologies. In essence, the main contribution of the work increases the acceptability and accessibility of hand gestures for use in immersive virtual environments. Finely dissected, the contributions are listed below:

- A comprehensive survey about the use of immersive technologies for remote collaboration.

- Introduction of photorealistic and interactive VR scenarios based on panorama images.

- Conducting a user study for visual comparison of 3D modelled and panorama based virtual environments.
- Introducing techniques to use hand gestures for locomotion in VR and comparison of the techniques in terms of usability and user preference.
- A study of whether lay people can use hand gestures to control locomotion in VR.
- The introduction of a framework and authoring tool to rapidly design, test, and implement one-handed gestures for immersive applications based on XR technologies.
- Comparing novel hand gestures with state of the art techniques for interacting with virtual objects.

The contributions include practical and theoretical solutions for the continued development of today's interaction techniques for immersive environments. These contributions were evaluated in a user-centred way, using both common and customised evaluation techniques.

8.3. Future Work

The most important contributions of this work include hand gestures for interaction. However, the field of natural user interfaces has more to offer. Especially speech and eye gaze are promising approaches to be combined with hand gestures. Utilising the users gaze, a hand gesture can invoke different actions depending on what is looked at, i.e. a possible future direction could be context aware hand gestures. Speech can also help users interact intuitively with a virtual environment and its objects. For example, to enable or disable locomotion features or menus. It can be expected that more and more real motion of the user is conveyed to the virtual environments in the future of XR technology. Intuitive interactions will allow the mass adoption of XR technology to take place, but there is still a lot of research to be done: Hardware needs to become lighter and more comfortable, tracking needs to be improved, and finding intuitive ways to interact with the virtual environment.

List of Figures

List of Tables

Bibliography

[1] M. Al Zayer, P. MacNeilage, and E. Folmer, "Virtual Locomotion: A Survey," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 6, pp. 2315–2334, Jun. 2020, Conference Name: IEEE Transactions on Visualization and Computer Graphics, ISSN: 1941-0506. DOI: `10.1109/TVCG.2018.2887379`.

[2] W. Albert and T. Tullis, *Measuring the user experience: collecting, analyzing, and presenting usability metrics*. Newnes, 2013.

[3] L. Alem, F. Tecchia, and W. Huang, "Handsonvideo: Towards a gesture based mobile ar system for remote collaboration," in *Recent Trends of Mobile Collaborative Augmented Reality Systems*, L. Alem and W. Huang, Eds., New York, NY: Springer New York, 2011, pp. 135–148, ISBN: 978-1-4419-9845-3.

[4] H. P. Andres, "A comparison of face-to-face and virtual software development teams," *Team Performance Management: An International Journal*, vol. 8, pp. 39–48, 2002, ISSN: 13527592. DOI: `10.1108/13527590210425077`.

[5] D. Ashbrook and T. Starner, "Magic: A motion gesture design tool," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10, Atlanta, Georgia, USA: Association for Computing Machinery, 2010, pp. 2159–2168, ISBN: 9781605589299. DOI: `10.1145/1753326.1753653`.

[6] E. R. Bachmann, E. Hodgson, C. Hoffbauer, and J. Messinger, "Multiuser redirected walking and resetting using artificial potential fields," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 5, pp. 2022–2031, 2019. DOI: `10.1109/TVCG.2019.2898764`.

[7] H. Bai, P. Sasikumar, J. Yang, and M. Billinghurst, "A user study on mixed reality remote collaboration with eye gaze and hand gesture sharing," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–13.

[8] R. A. J. de Belen, H. Nguyen, D. Filonik, D. Del Favero, and T. Bednarz, "A systematic review of the current state of collaborative mixed reality technologies: 2013–2018," *AIMS Electronics and Electrical Engineering*, vol. 3, no. 2, p. 181, 2019.

[9] D. Ben Ghida, “Augmented reality and virtual reality: A 360 immersion into western history of architecture,” *International Journal*, vol. 8, no. 9, pp. 6051–6055, 2020.

[10] L. Berger and K. Wolf, “Wim: Fast locomotion in virtual reality with spatial orientation gain & without motion sickness,” in *Proceedings of the 17th International Conference on Mobile and Ubiquitous Multimedia*, ser. MUM 2018, Cairo, Egypt: Association for Computing Machinery, 2018, pp. 19–24, ISBN: 9781450365949. DOI: `10.1145/3282894.3282932`.

[11] M. Billinghurst, H. Kato, and S. Myojin, “Advanced interaction techniques for augmented reality applications,” in *International Conference on Virtual and Mixed Reality*, Springer, 2009, pp. 13–22.

[12] I. Bishop and M. R. Abid, “Survey of Locomotion Systems in Virtual Reality,” en, in *Proceedings of the 2nd International Conference on Information System and Data Mining - ICISDM ’18*, Lakeland, FL, USA: ACM Press, 2018, pp. 151–154, ISBN: 978-1-4503-6354-9. DOI: `10.1145/3206098.3206108`.

[13] C. Boletsis, “The new era of virtual reality locomotion: A systematic literature review of techniques and a proposed typology,” *Multimodal Technologies and Interaction*, vol. 1, no. 4, p. 24, 2017.

[14] C. Boletsis, “Virtual Reality for Prototyping Service Journeys,” *Multimodal Technologies and Interaction*, vol. 2, no. 2, p. 14, 2018. DOI: `10.3390/mti2020014`.

[15] L. Bölling, N. Stein, F. Steinicke, and M. Lappe, “Shrinking circles: Adaptation to increased curvature gain in redirected walking,” *IEEE transactions on visualization and computer graphics*, vol. 25, no. 5, pp. 2032–2039, 2019.

[16] P. Boonbrahm and C. Kaewrat, “Assembly of the virtual model with real hands using augmented reality technology,” in *Virtual, Augmented and Mixed Reality. Designing and Developing Virtual and Augmented Environments*, R. Shumaker and S. Lackey, Eds., Cham: Springer International Publishing, 2014, pp. 329–338, ISBN: 978-3-319-07458-0.

[17] D. Bowman, E. Kruijff, J. J. LaViola Jr, and I. P. Poupyrev, *3D User interfaces: theory and practice, CourseSmart eTextbook*. Addison-Wesley, 2004.

[18] D. A. Bowman, D. B. Johnson, and L. F. Hodges, “Testbed evaluation of virtual environment interaction techniques,” *Presence: Teleoperators & Virtual Environments*, vol. 10, no. 1, pp. 75–95, 2001.

[19] D. A. Bowman and R. P. McMahan, “Virtual reality: How much immersion is enough?” *Computer*, vol. 40, no. 7, pp. 36–43, 2007.

[20] E. Bozgeyikli, A. Raij, S. Katkoori, and R. Dubey, "Locomotion in virtual reality for individuals with autism spectrum disorder," in *Proceedings of the 2016 Symposium on Spatial User Interaction*, ser. SUI '16, Tokyo, Japan: Association for Computing Machinery, 2016, pp. 33–42, ISBN: 9781450340687. DOI: `10.1145/2983310.2985763`.

[21] E. Bozgeyikli, A. Raij, S. Katkoori, and R. Dubey, "Locomotion in virtual reality for individuals with autism spectrum disorder," in *Proceedings of the 2016 Symposium on Spatial User Interaction*, 2016, pp. 33–42.

[22] E. Bozgeyikli, A. Raij, S. Katkoori, and R. Dubey, "Point & teleport locomotion technique for virtual reality," in *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*, 2016, pp. 205–216.

[23] E. Bozgeyikli, A. Raij, S. Katkoori, and R. Dubey, "Point & teleport locomotion technique for virtual reality," in *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play*, ser. CHI PLAY '16, Austin, Texas, USA: Association for Computing Machinery, 2016, pp. 205–216, ISBN: 9781450344562. DOI: `10.1145/2967934.2968105`.

[24] J. Brooke, "Sus: A "quick and dirty'usability," *Usability evaluation in industry*, p. 189, 1996.

[25] J. Brooke, "Sus: A retrospective," *Journal of usability studies*, vol. 8, no. 2, pp. 29–40, 2013.

[26] L. Bruno, J. Pereira, and J. Jorge, "A new approach to walking in place," in *Human-Computer Interaction – INTERACT 2013*, P. Kotzé, G. Marsden, G. Lindgaard, J. Wesson, and M. Winckler, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 370–387, ISBN: 978-3-642-40477-1. DOI: `\url{10.1007/978-3-642-40477-1_23}`.

[27] V. Buchmann, S. Violich, M. Billinghurst, and A. Cockburn, "Fingartips: Gesture based direct manipulation in augmented reality," in *Proceedings of the 2nd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia*, ser. GRAPHITE '04, Singapore: Association for Computing Machinery, 2004, pp. 212–221, ISBN: 1581138830. DOI: `10.1145/988834.988871`.

[28] E. A. Bustamante and R. D. Spain, "Measurement invariance of the nasa tlx," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, SAGE Publications Sage CA: Los Angeles, CA, vol. 52, 2008, pp. 1522–1526.

[29] F. Buttussi and L. Chittaro, "Locomotion in Place in Virtual Reality: A Comparative Evaluation of Joystick, Teleport, and Leaning," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 1, pp. 125–136, Jan. 2021, Conference Name: IEEE Transactions on Visualization and Computer Graphics, ISSN: 1941-0506. DOI: `10.1109/TVCG.2019.2928304`.

[30] G. Caggianese, N. Capece, U. Erra, L. Gallo, and M. Rinaldi, "Freehand-steering locomotion techniques for immersive virtual environments: A comparative evaluation," *International Journal of Human–Computer Interaction*, vol. 36, no. 18, pp. 1734–1755, 2020. DOI: `10.1080/10447318.2020.1785151`.

[31] G. Caggianese, L. Gallo, and P. Neroni, "Design and preliminary evaluation of free-hand travel techniques for wearable immersive virtual reality systems with egocentric sensing," in *Augmented and Virtual Reality*, L. T. De Paolis and A. Mongelli, Eds., Cham: Springer International Publishing, 2015, pp. 399–408, ISBN: 978-3-319-22888-4. DOI: `\url{10.1007/978-3-319-22888-4_29}`.

[32] G. Caggianese, L. Gallo, and P. Neroni, "The vive controllers vs. leap motion for interactions in virtual environments: A comparative evaluation," in *Intelligent Interactive Multimedia Systems and Services*, G. De Pietro, L. Gallo, R. J. Howlett, L. C. Jain, and L. Vlacic, Eds., Cham: Springer International Publishing, 2019, pp. 24–33, ISBN: 978-3-319-92231-7.

[33] D. Calandra, A. Cannavò, and F. Lamberti, "Improving ar-powered remote assistance: A new approach aimed to foster operator's autonomy and optimize the use of skilled resources," *The International Journal of Advanced Manufacturing Technology*, pp. 1–18, 2021.

[34] A. Cao, L. Wang, Y. Liu, and V. Popescu, "Feature guided path redirection for vr navigation," in *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, IEEE, 2020, pp. 137–145.

[35] J. C. S. Cardoso, "Comparison of gesture, gamepad, and gaze-based locomotion for vr worlds," in *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology*, ser. VRST '16, Munich, Germany: Association for Computing Machinery, 2016, pp. 319–320, ISBN: 9781450344913. DOI: `10.1145/2993369.2996327`.

[36] J. C. S. Cardoso and A. Perrotta, "A survey of real locomotion techniques for immersive virtual reality applications on head-mounted displays," en, *Computers & Graphics*, vol. 85, pp. 55–73, Dec. 2019, ISSN: 0097-8493. DOI: `10.1016/j.cag.2019.09.005`.

[37] J. C. Cardoso and A. Perrotta, "A survey of real locomotion techniques for immersive virtual reality applications on head-mounted displays," *Computers & Graphics*, vol. 85, pp. 55–73, 2019.

[38] V. Chat, "Create and play in virtual worlds," *VRChat Inc*, 2018.

[39] A. Chaudhary, J. Raheja, K. Das, and S. Raheja, "A survey on hand gesture recognition in context of soft computing," in *International Conference on Computer Science and Information Technology*, Springer, 2011, pp. 46–55.

[40] H. Chen, A. S. Lee, M. Swift, and J. C. Tang, “3d collaboration method over hololens™ and skype™ end points,” in *Proceedings of the 3rd International Workshop on Immersive Media Experiences*, 2015, pp. 27–30.

[41] L. Chen, F. Wang, H. Deng, and K. Ji, “A survey on hand gesture recognition,” in *2013 International conference on computer sciences and applications*, IEEE, 2013, pp. 313–316.

[42] X. Chen *et al.*, “Immertai: Immersive motion learning in vr environments,” *Journal of Visual Communication and Image Representation*, vol. 58, pp. 416–427, 2019.

[43] L.-P. Cheng, E. Ofek, C. Holz, and A. D. Wilson, “Vroamer: Generating on-the-fly vr experiences while walking inside large, unknown real-world building environments,” in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, IEEE, 2019, pp. 359–366.

[44] M. J. Cheok, Z. Omar, and M. H. Jaward, “A review of hand gesture and sign language recognition techniques,” *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 1, pp. 131–153, 2019.

[45] S. Chowdhury and M. A. Schnabel, “Laypeople’s collaborative immersive virtual reality design discourse in neighborhood design,” *Frontiers in Robotics and AI*, vol. 6, p. 97, 2019.

[46] C. G. Christou and P. Aristidou, “Steering versus teleport locomotion for head mounted displays,” in *International conference on augmented reality, virtual reality and computer graphics*, Springer, 2017, pp. 431–446.

[47] J. Clifton and S. Palmisano, “Comfortable locomotion in vr: Teleportation is not a complete solution,” in *25th ACM Symposium on Virtual Reality Software and Technology*, ser. VRST ’19, Parramatta, NSW, Australia: Association for Computing Machinery, 2019, ISBN: 9781450370011. DOI: `10.1145/3359996.3364722`.

[48] W. P. Corp, *Sansar*, 2018.

[49] C. Davide, G. P. Filippo, M. Massimo, V. Vittorio, and F. Lamberti, “A multi-role, multi-user, multi-technology virtual reality-based road tunnel fire simulator for training purposes,” in *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021)*, ScitePress, 2021, pp. 96–105.

[50] F. De Pace, F. Manuri, A. Sanna, and D. Zappia, “A comparison between two different approaches for a collaborative mixed-virtual environment in industrial maintenance,” *Frontiers in Robotics and AI*, vol. 6, p. 18, 2019.

[51] G. D. de Dinechin and A. Paljic, “Automatic generation of interactive 3d characters and scenes for virtual reality from a single-viewpoint 360-degree video,” 2019. DOI: `10.1109/VR.2019.8797969`.

[52] H. Dinh, N. Walker, L. Hodges, C. Song, and A. Kobayashi, "Evaluating the importance of multi-sensory input on memory and the sense of presence in virtual environments," in *Proceedings IEEE Virtual Reality (Cat. No. 99CB36316)*, 1999, pp. 222–228. DOI: `10.1109/VR.1999.756955`.

[53] L. Dipietro, A. M. Sabatini, and P. Dario, "A survey of glove-based systems and their applications," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 4, pp. 461–482, 2008. DOI: `10.1109/TSMCC.2008.923862`.

[54] J. Dominic and A. Robb, "Exploring effects of screen-fixed and world-fixed annotation on navigation in virtual reality," in *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, IEEE, 2020, pp. 607–615.

[55] T. Dong, X. Chen, Y. Song, W. Ying, and J. Fan, "Dynamic artificial potential fields for multi-user redirected walking," in *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, IEEE, 2020, pp. 146–154.

[56] R. Du, D. Li, and A. Varshney, "Geollery: A Mixed Reality Social Media Platform," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ser. CHI, ACM, May 2019. DOI: `10.1145/3290605.3300915`.

[57] R. Du, D. Li, and A. Varshney, "Project Geollery.com: Reconstructing a Live Mirrored World With Geotagged Social Media," in *Proceedings of the 24th International Conference on Web3D Technology*, ser. Web3D, ACM, Jul. 2019, pp. 1–9. DOI: `10.1145/3329714.3338126`.

[58] C. Elvezio, M. Sukan, O. Oda, S. Feiner, and B. Tversky, "Remote collaboration in ar and vr using virtual replicas," in *ACM SIGGRAPH 2017 VR Village*, ser. SIGGRAPH '17, Los Angeles, California: Association for Computing Machinery, 2017, ISBN: 9781450350136. DOI: `10.1145/3089269.3089281`.

[59] D. Englmeier, F. Fan, and A. Butz, "Rock or roll–locomotion techniques with a handheld spherical device in virtual reality," in *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, IEEE, 2020, pp. 618–626.

[60] B. Ens *et al.*, "Revisiting collaboration through mixed reality: The evolution of groupware," *International Journal of Human-Computer Studies*, vol. 131, pp. 81–98, 2019, 50 years of the International Journal of Human-Computer Studies. Reflections on the past, present and future of human-centred technologies, ISSN: 1071-5819. DOI: `https://doi.org/10.1016/j.ijhcs.2019.05.011`.

[61] Facebook, *Oculus quest 2*, 2020.

[62] A. Ferracani, D. Pezzatini, J. Bianchini, G. Biscini, and A. Del Bimbo, "Locomotion by Natural Gestures for Immersive Virtual Environments," en, in *Proceedings of the 1st International Workshop on Multimedia Alternate Realities*, Amsterdam The Netherlands: ACM, Oct. 2016, pp. 21–24, ISBN: 978-1-4503-4521-7. DOI: `10.1145/2983298.2983307`.

[63] A. Ferracani, D. Pezzatini, J. Bianchini, G. Biscini, and A. Del Bimbo, "Locomotion by natural gestures for immersive virtual environments," in *Proceedings of the 1st international workshop on multimedia alternate realities*, 2016, pp. 21–24.

[64] C. Fleury, N. Férey, J.-M. Vézien, and P. Bourdot, "Remote collaboration across heterogeneous large interactive spaces," in *2015 IEEE Second VR International Workshop on Collaborative Virtual Environments (3DCVE)*, 2015, pp. 9–10. DOI: `10.1109/3DCVE.2015.7153591`.

[65] T. Franke, C. Attig, and D. Wessel, "A personal resource for technology interaction: Development and validation of the affinity for technology interaction (ati) scale," *International Journal of Human–Computer Interaction*, vol. 35, no. 6, pp. 456–467, 2019.

[66] J. Frommel, S. Sonntag, and M. Weber, "Effects of controller-based locomotion on player experience in a virtual reality exploration game," in *Proceedings of the 12th international conference on the foundations of digital games*, 2017, pp. 1–6.

[67] P. Galambos, P. Baranyi, and I. J. Rudas, "Merged physical and virtual reality in collaborative virtual workspaces: The virca approach," in *IECON 2014-40th Annual Conference of the IEEE Industrial Electronics Society*, IEEE, 2014, pp. 2585–2590.

[68] P. Galambos *et al.*, "Design, programming and orchestration of heterogeneous manufacturing systems through vr-powered remote collaboration," *Robotics and Computer-Integrated Manufacturing*, vol. 33, pp. 68–77, 2015, Special Issue on Knowledge Driven Robotics and Manufacturing, ISSN: 0736-5845. DOI: `https://doi.org/10.1016/j.rcim.2014.08.012`.

[69] J. Galván-Ruiz, C. M. Travieso-González, A. Tejera-Fettmilch, A. Pinan-Roescher, L. Esteban-Hernández, and L. Domínguez-Quintana, "Perspective and evolution of gesture recognition for sign language: A review," *Sensors*, vol. 20, no. 12, p. 3571, 2020.

[70] G. Gamelin, A. Chellali, S. Cheikh, A. Ricca, C. Dumas, and S. Otmane, "Point-cloud avatars to improve spatial communication in immersive collaborative virtual environments," *Personal and Ubiquitous Computing*, pp. 1–18, 2020.

[71] L. Gao, H. Bai, G. Lee, and M. Billinghurst, "An oriented point-cloud view for mr remote collaboration," in *SIGGRAPH ASIA 2016 Mobile Graphics and Interactive Applications*, 2016, pp. 1–4.

[72] L. Gao, H. Bai, R. Lindeman, and M. Billinghurst, "Static local environment capturing and sharing for mr remote collaboration," in *SIGGRAPH Asia 2017 Mobile Graphics & Interactive Applications*, 2017, pp. 1–6.

[73] L. Gao, H. Bai, T. Piumsomboon, G. Lee, R. W. Lindeman, and M. Billinghurst, "Real-time visual representations for mixed reality remote collaboration," 2017.

[74] S. Gauglitz, C. Lee, M. Turk, and T. Höllerer, "Integrating the physical environment into mobile remote collaboration," in *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services*, 2012, pp. 241–250.

[75] S. Gauglitz, B. Nuernberger, M. Turk, and T. Höllerer, "In touch with the remote world: Remote collaboration with augmented reality drawings and virtual navigation," in *Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology*, 2014, pp. 197–205.

[76] S. Gauglitz, B. Nuernberger, M. Turk, and T. Höllerer, "World-stabilized annotations and virtual scene navigation for remote collaboration," in *Proceedings of the 27th annual ACM symposium on User interface software and technology*, 2014, pp. 449–459.

[77] C. C. Gava and D. Stricker, "A generalized structure from motion framework for central projection cameras," in *International Joint Conference on Computer Vision, Imaging and Computer Graphics*, Springer, 2015, pp. 256–273.

[78] T. Grechkin, J. Thomas, M. Azmandian, M. Bolas, and E. Suma, "Revisiting detection thresholds for redirected walking: Combining translation and curvature gains," in *Proceedings of the ACM Symposium on Applied Perception*, ser. SAP '16, Anaheim, California: Association for Computing Machinery, 2016, pp. 113–120, ISBN: 9781450343831. DOI: `10.1145/2931002.2931018`.

[79] K. Grønbæk, K. K. Gundersen, P. Mogensen, and P. Ørbæk, "Interactive room support for complex and distributed design projects.," in *INTERACT*, 2001, pp. 407–414.

[80] N. Gu, M. J. Kim, and M. L. Maher, "Technological advancements in synchronous collaboration: The effect of 3d virtual worlds and tangible user interfaces on architectural design," *Automation in Construction*, vol. 20, no. 3, pp. 270–278, 2011.

[81] K. Gupta, G. A. Lee, and M. Billinghurst, "Do you see what i see? the effect of gaze tracking on task space remote collaboration," *IEEE transactions on visualization and computer graphics*, vol. 22, no. 11, pp. 2413–2422, 2016.

[82] P. Gurevich, J. Lanir, and B. Cohen, "Design and implementation of teleadvisor: A projection-based augmented reality system for remote collaboration," *Computer Supported Cooperative Work (CSCW)*, vol. 24, no. 6, pp. 527–562, 2015.

[83] P. Gurevich, J. Lanir, B. Cohen, and R. Stone, “Teleadvisor: A versatile augmented reality tool for remote assistance,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2012, pp. 619–622.

[84] M. J. Habgood, D. Moore, D. Wilson, and S. Alapont, “Rapid, continuous movement between nodes as an accessible virtual reality locomotion technique,” in *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, IEEE, 2018, pp. 371–378.

[85] M. Haller, M. Billinghurst, J. Leithinger, D. Leitner, and T. Seifried, “Coeno: Enhancing face-to-face collaboration,” in *Proceedings of the 2005 international conference on Augmented tele-existence*, 2005, pp. 40–47.

[86] J. Han, A. V. Moere, and A. L. Simeone, “Foldable spaces: An overt redirection approach for natural walking in virtual reality,” in *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, 2022, pp. 167–175. DOI: `10.1109/VR51125.2022.00035`.

[87] S. Hanson, R. A. Paris, H. A. Adams, and B. Bodenheimer, “Improving walking in place methods with individualization and deep networks,” in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, IEEE, 2019, pp. 367–376.

[88] S. G. Hart, “Nasa-task load index (nasa-tlx); 20 years later,” *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 50, no. 9, pp. 904–908, 2006. DOI: `10.1177/154193120605000909`.

[89] S. G. Hart and L. E. Staveland, “Development of nasa-tlx (task load index): Results of empirical and theoretical research,” in *Human Mental Workload*, ser. Advances in Psychology, P. A. Hancock and N. Meshkati, Eds., vol. 52, North-Holland, 1988, pp. 139–183. DOI: `https://doi.org/10.1016/S0166-4115(08)62386-9`.

[90] B. Hatscher and C. Hansen, “Hand, foot or voice: Alternative input modalities for touchless interaction in the medical domain,” in *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, 2018, pp. 145–153.

[91] O. Hayashi, K. Fujita, K. Takashima, R. W. Lindernan, and Y. Kitarnura, “Redirected jumping: Imperceptibly manipulating jump motions in virtual reality,” in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, IEEE, 2019, pp. 386–394.

[92] Z. He, R. Du, and K. Perlin, “CollaboVR: A Reconfigurable Framework for Multi-User to Communicate in Virtual Reality,” in *2020 IEEE International Symposium on Mixed and Augmented Reality*, ser. ISMAR, IEEE, Nov. 2020. DOI: `10.1109/ISMAR50242.2020.00082`.

[93] M. L. Heilig, *Sensorama simulator*, US Patent 3,050,870, Aug. 1962.

[94] K. Higuch, R. Yonetani, and Y. Sato, "Can eye help you? effects of visualizing eye fixations on remote collaboration scenarios for physical tasks," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, 2016, pp. 5180–5190.

[95] K. Higuchi, Y. Chen, P. A. Chou, Z. Zhang, and Z. Liu, "Immerseboard: Immersive telepresence experience using a digital whiteboard," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2015, pp. 2383–2392.

[96] T. Hilfert and M. König, "Low-cost virtual reality environment for engineering and construction," *Visualization in Engineering*, vol. 4, no. 1, p. 2, 2016.

[97] J. Holler, H. Shovelton, and G. Beattie, "Do iconic hand gestures really contribute to the communication of semantic information in a face-to-face context?" *Journal of Nonverbal Behavior*, vol. 33, no. 2, pp. 73–88, 2009. DOI: `https://doi.org/10.1007/s10919-008-0063-9`.

[98] S. W. Hong, A. El Antably, and Y. E. Kalay, "Architectural design creativity in multi-user virtual environment: A comparative analysis between remote collaboration media," *Environment and Planning B: Urban Analytics and City Science*, vol. 46, no. 5, pp. 826–844, 2019.

[99] G. HQ, *Glue universal collaboration platform*, 2020.

[100] T.-W. Hsu *et al.*, "Design and initial evaluation of a vr based immersive and interactive architectural design discussion system,"

[101] HTC, *Vive hand tracking sdk*, 2020.

[102] R. Huang, C. Harris-adamson, D. Odell, and D. Rempel, "Design of finger gestures for locomotion in virtual reality," en, *Virtual Reality & Intelligent Hardware*, vol. 1, no. 1, pp. 1–9, Feb. 2019, ISSN: 2096-5796. DOI: `10.3724/SP.J.2096-5796.2018.0007`.

[103] H. Ibayashi *et al.*, "Dollhouse vr: A multi-view, multi-user collaborative design workspace with vr technology," in *SIGGRAPH Asia 2015 Emerging Technologies*, 2015, pp. 1–2.

[104] N. A. Ibraheem and R. Z. Khan, "Survey on various gesture recognition technologies and techniques," *International journal of computer applications*, vol. 50, no. 7, 2012.

[105] W. IJsselsteijn, Y. De Kort, and K. Poels, "The game experience questionnaire," *Eindhoven: Technische Universiteit Eindhoven*, 2013.

[106] B. inc., *Big screen*, 2020.

[107] G. Inc., *Garou*, 2020.

[108] I. Inc., *Irisvr*, 2019.

[109] S. Izadi, A. Agarwal, A. Criminisi, J. Winn, A. Blake, and A. Fitzgibbon, “C-slate: A multi-touch and object recognition system for remote collaboration using horizontal surfaces,” in *Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP’07)*, IEEE, 2007, pp. 3–10.

[110] D. Jo, K.-H. Kim, and G. J. Kim, “Spacetime: Adaptive control of the teleported avatar for improved ar tele-conference experience,” *Computer Animation and Virtual Worlds*, vol. 26, no. 3-4, pp. 259–269, 2015.

[111] D. Jo, K.-h. Kim, and G. J. Kim, “Effects of Avatar and Background Types on Users’ Co-presence and Trust for Mixed Reality-Based Teleconference Systems,” *Casa 2017*, pp. 27–36, 2017.

[112] H. Kang *et al.*, “Jumping further: Forward jumps in a gravity-reduced immersive virtual environment,” in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, IEEE, 2019, pp. 699–707.

[113] H. J. Kang, J.-h. Shin, and K. Ponto, “A Comparative Analysis of 3D User Interaction: How to Move Virtual Objects in Mixed Reality,” in *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, ISSN: 2642-5254, Mar. 2020, pp. 275–284. DOI: `10.1109/VR46266.2020.00047`.

[114] T. Kantonen, C. Woodward, and N. Katz, “Mixed reality in virtual world teleconferencing,” *Proceedings - IEEE Virtual Reality*, no. October, pp. 179–182, 2010. DOI: `10.1109/VR.2010.5444792`.

[115] G. Kato, Y. Kuroda, K. Kiyokawa, and H. Takemura, “Force rendering and its evaluation of a friction-based walking sensation display for a seated user,” *IEEE Transactions on visualization and computer graphics*, vol. 24, no. 4, pp. 1506–1514, 2018.

[116] J. Keil, D. Edler, D. O’Meara, A. Korte, and F. Dickmann, “Effects of virtual reality locomotion techniques on distance estimations,” *ISPRS International Journal of Geo-Information*, vol. 10, no. 3, p. 150, 2021.

[117] J. W. Kelly, A. G. Ostrander, A. F. Lim, L. A. Cherep, and S. B. Gilbert, “Teleporting through virtual environments: Effects of path scale and environment scale on spatial updating,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 5, pp. 1841–1850, 2020.

[118] S. Kelly, M. Healey, A. Ozyurek, and J. Holler, “The communicative influence of gesture and action during speech comprehension: Gestures have the upper hand,” *The Journal of the Acoustical Society of America*, vol. 131, no. 4, pp. 3311–3311, 2012. DOI: `https://doi.org/10.1121/1.4708385`.

[119] R. Z. Khan and N. A. Ibraheem, “Survey on gesture recognition for hand image postures,” *Computer and information science*, vol. 5, no. 3, p. 110, 2012.

[120] C. Khundam, V. Vorachart, P. Preeyawongsakul, W. Hosap, and F. Noël, "A comparative study of interaction time and usability of using controllers and hand tracking in virtual reality training," *Informatics*, vol. 8, no. 3, p. 1, 2021, ISSN: 2227-9709. DOI: `10.3390/informatics8030060`.

[121] S. Kim, M. Billinghurst, and G. Lee, "The effect of collaboration styles and view independence on video-mediated remote collaboration," *Computer Supported Cooperative Work (CSCW)*, vol. 27, no. 3-6, pp. 569–607, 2018.

[122] S. Kim, G. Lee, W. Huang, H. Kim, W. Woo, and M. Billinghurst, "Evaluating the combination of visual communication cues for hmd-based mixed reality remote collaboration," in *Proceedings of the 2019 CHI conference on human factors in computing systems*, 2019, pp. 1–13.

[123] J.-S. Kim, D. Gračanin, K. Matković, and F. Quek, "Finger walking in place (fwip): A traveling technique in virtual environments," in *Smart Graphics*, A. Butz, B. Fisher, A. Krüger, P. Olivier, and M. Christie, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 58–69, ISBN: 978-3-540-85412-8. DOI: `\url{10.1007/978-3-540-85412-8_6}`.

[124] J.-S. Kim, D. Gračanin, K. Matković, and F. Quek, "The effects of finger-walking in place (fwip) for spatial knowledge acquisition in virtual environments," in *Smart Graphics*, R. Taylor, P. Boulanger, A. Krüger, and P. Olivier, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 56–67, ISBN: 978-3-642-13544-6. DOI: `\url{10.1007/978-3-642-13544-6_6}`.

[125] A. Kitson, B. E. Riecke, A. M. Hashemian, and C. Neustaedter, "Navichair: Evaluating an embodied interface using a pointing task to navigate virtual reality," in *Proceedings of the 3rd ACM Symposium on Spatial User Interaction*, 2015, pp. 123–126.

[126] P.-O. Kristensson and S. Zhai, "Shark¡sup¿2¡/sup¿: A large vocabulary shorthand writing system for pen-based computers," in *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '04, Santa Fe, NM, USA: Association for Computing Machinery, 2004, pp. 43–52, ISBN: 1581139578. DOI: `10.1145/1029632.1029640`.

[127] L. Kruse, E. Langbehn, and F. Steinicke, "I can see on my feet while walking: Sensitivity to translation gains with visible feet," in *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, IEEE, 2018, pp. 305–312.

[128] S. Kumar *et al.*, "Second life and the new generation of virtual worlds," *Computer*, vol. 41, no. 9, pp. 46–53, 2008.

[129] T. Kurata, N. Sakata, M. Kourogi, H. Kuzuoka, and M. Billinghurst, "Remote collaboration using a shoulder-worn active camera/laser," in *Eighth international symposium on wearable computers*, IEEE, vol. 1, 2004, pp. 62–69.

[130] E. Langbehn, T. Eichler, S. Ghose, G. Bruder, and F. Steinicke, "Evaluation of an Omnidirectional Walking-in-Place User Interface with Virtual Locomotion Speed Scaled by Forward Leaning Angle," en, *GI AR/VR Workshop*, pp. 149–160, 2015.

[131] E. Langbehn, P. Lubos, and F. Steinicke, "Evaluation of locomotion techniques for room-scale vr: Joystick, teleportation, and redirected walking," in *Proceedings of the Virtual Reality International Conference - Laval Virtual*, ser. VRIC '18, Laval, France: Association for Computing Machinery, 2018, ISBN: 9781450353816. DOI: `10.1145/3234253.3234291`.

[132] D.-Y. Lee, Y.-H. Cho, and I.-K. Lee, "Real-time optimal planning for redirected walking using deep q-learning," in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, IEEE, 2019, pp. 63–71.

[133] D.-Y. Lee, Y.-H. Cho, D.-H. Min, and I.-K. Lee, "Optimal planning for redirected walking based on reinforcement learning in multi-user environment with irregularly shaped physical space," in *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, IEEE, 2020, pp. 155–163.

[134] G. A. Lee, T. Teo, S. Kim, and M. Billinghurst, "A user study on mr remote collaboration using live 360 video," in *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, IEEE, 2018, pp. 153–164.

[135] G. A. Lee, T. Teo, S. Kim, and M. Billinghurst, "Sharedsphere: Mr collaboration through shared live panorama," in *SIGGRAPH Asia 2017 Emerging Technologies*, 2017, pp. 1–2.

[136] Y. Lee, K. Masai, K. Kunze, M. Sugimoto, and M. Billinghurst, "A remote collaboration system with empathy glasses," in *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*, IEEE, 2016, pp. 342–343.

[137] V. D. Lehner and T. A. DeFanti, "Distributed virtual reality: Supporting remote collaboration in vehicle design," *IEEE Computer Graphics and Applications*, vol. 17, no. 2, pp. 13–17, 1997.

[138] C. Li, X. Zhang, and L. Jin, "Lpsnet: A novel log path signature feature based hand gesture recognition framework," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2017, pp. 631–639.

[139] G. Li, D. Rempel, Y. Liu, W. Song, and C. H. Adamson, "Design of 3d microgestures for commands in virtual reality or augmented reality," *Applied Sciences*, vol. 11, no. 14, p. 6375, 2021.

[140] L. Liu and Y. Huai, “Dynamic hand gesture recognition using lmc for flower and plant interaction,” *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 33, no. 01, p. 1 950 003, 2019.

[141] F. T. LLC, *Facebook horizon*, 2020.

[142] F. T. LLC, *Oculus SDK*, `https://developer.oculus.com/develop/`, [Online; accessed 24-May-2021], 2021.

[143] R. Lovreglio *et al.*, “Prototyping virtual reality serious games for building earthquake preparedness: The Auckland City Hospital case study,” *Advanced Engineering Informatics*, vol. 38, pp. 670–682, 2018, ISSN: 14740346. DOI: `10.1016/j.aei.2018.08.018`. arXiv: `1802.09119`.

[144] H. Lü and Y. Li, “Gesture coder: A tool for programming multi-touch gestures by demonstration,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY, USA: Association for Computing Machinery, 2012, pp. 2875–2884, ISBN: 9781450310154.

[145] S. Lukosch, R. Poelman, O. Akman, and P. Jonker, “A novel gesture-based interface for crime scene investigation in mediated reality,” in *Workshop at CSCW 2012: Exploring collaboration in challenges environments: From the car to the factory and beyond, 11-15 februari 2012, Seattle, Washington, USA*, ACM, 2012.

[146] A. Luxenburger, A. Prange, M. M. Moniri, and D. Sonntag, “Medicalvr: Towards medical remote collaboration using virtual reality,” in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, 2016, pp. 321–324.

[147] N. Magrofuoco, P. Roselli, J. Vanderdonckt, J. L. Pérez-Medina, and R.-D. Vatavu, “Gestman: A cloud-based tool for stroke-gesture datasets,” in *Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, ser. EICS ’19, Valencia, Spain: Association for Computing Machinery, 2019, ISBN: 9781450367455. DOI: `10.1145/3319499.3328227`.

[148] J. Malik, A. Elhayek, and D. Stricker, “Structure-aware 3d hand pose regression from a single depth image,” in *International Conference on Virtual Reality and Augmented Reality*, Springer, 2018, pp. 3–17.

[149] L. Marie Prinz, T. Mathew, S. Klüber, and B. Weyers, “An Overview and Analysis of Publications on Locomotion Taxonomies,” in *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, Mar. 2021, pp. 385–388. DOI: `10.1109/VRW52623.2021.00080`.

[150] K. Masai, K. Kunze, M. Sugimoto, and M. Billinghurst, “Empathy glasses,” in *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, 2016, pp. 1257–1263.

[151] A. Masurovsky, P. Chojecki, D. Runde, M. Lafci, D. Przewozny, and M. Gaebler, "Controller-free hand tracking for grab-and-place tasks in immersive virtual reality: Design elements and their empirical study," *Multimodal Technologies and Interaction*, vol. 4, no. 4, p. 1, 2020, ISSN: 2414-4088. DOI: `10.3390/mti4040091`.

[152] E. Matsas, G. C. Vosniakos, and D. Batras, "Prototyping proactive and adaptive techniques for human-robot collaboration in manufacturing using virtual reality," *Robotics and Computer-Integrated Manufacturing*, vol. 50, no. October, pp. 168–180, 2018, ISSN: 07365845. DOI: `10.1016/j.rcim.2017.09.005`.

[153] C. Matthes *et al.*, "The collaborative virtual reality neurorobotics lab," in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, IEEE, 2019, pp. 1671–1674.

[154] D. Medeiros, A. Sousa, A. Raposo, and J. Jorge, "Magic carpet: Interaction fidelity for flying in vr," *IEEE Transactions on Visualization and Computer Graphics*, 2019.

[155] MeetingRoom.io, *Meetingroom*, 2020.

[156] MeetInVR, *Meetinvr*, 2020.

[157] V. M. Memmesheimer and A. Ebert, "Scalable extended reality: A future research agenda," *Big Data and Cognitive Computing*, vol. 6, no. 1, 2022, ISSN: 2504-2289. DOI: `10.3390/bdcc6010012`.

[158] T. Menzner, T. Gesslein, A. Otte, and J. Grubert, "Above surface interaction for multiscale navigation in mobile virtual reality," in *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, IEEE, 2020, pp. 372–381.

[159] J. Messinger, E. Hodoson, and E. R. Bachmann, "Effects of tracking area shape and size on artificial potential field redirected walking," in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, IEEE, 2019, pp. 72–80.

[160] D. Mestre, P. Fuchs, A. Berthoz, and J. Vercher, "Immersion et présence," *Le traité de la réalité virtuelle. Paris: Ecole des Mines de Paris*, pp. 309–38, 2006.

[161] N. V. Metaverse, *Neosvr*, 2020.

[162] V. Metsis, G. Lawrence, M. Trahan, K. S. Smith, D. Tamir, and K. Selber, "360 video: A prototyping process for developing virtual reality interventions," *Journal of Technology in Human Services*, vol. 37, no. 1, pp. 32–50, 2019.

[163] Microsoft, *Hololens 2*, 2020.

[164] Microsoft, *Mixed Reality Toolkit (MRTK)*, `https://docs.microsoft.com/windows/mixed-reality/mrtk-unity/`, [Online; accessed 07-April-2022], 2021.

[165] P. Milgram, H. Takemura, A. Utsumi, and F. Kishino, "Augmented reality: a class of displays on the reality-virtuality continuum," in *Telemanipulator and Telepresence Technologies*, H. Das, Ed., International Society for Optics and Photonics, vol. 2351, SPIE, 1995, pp. 282–292. DOI: `10.1117/12.197321`.

[166] D.-H. Min, D.-Y. Lee, Y.-H. Cho, and I.-K. Lee, "Shaking hands in virtual space: Recovery in redirected walking for direct interaction between two users," in *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, IEEE, 2020, pp. 164–173.

[167] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 3, pp. 311–324, 2007.

[168] G. B. Mo, J. J. Dudley, and P. O. Kristensson, "Gesture knitter: A hand gesture design tool for head-mounted mixed reality applications," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, ser. CHI '21, Yokohama, Japan: Association for Computing Machinery, 2021, ISBN: 9781450380966. DOI: `10.1145/3411764.3445766`.

[169] T. Monahan, G. McArdle, and M. Bertolotto, "Virtual reality for collaborative e-learning," *Computers & Education*, vol. 50, no. 4, pp. 1339–1353, 2008.

[170] P. Monteiro, G. Gonçalves, H. Coelho, M. Melo, and M. Bessa, "Hands-free interaction in immersive virtual reality: A systematic review," *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 5, pp. 2702–2713, 2021. DOI: `10.1109/TVCG.2021.3067687`.

[171] Mozilla, *Hubs by mozilla*, 2020.

[172] S. Mu and A. Sourin, "Virtual assembling using hand tracking with leap motion controller," in *2021 International Conference on Cyberworlds (CW)*, Caen, France: IEEE, 2021, pp. 121–124. DOI: `10.1109/CW52790.2021.00026`.

[173] S. Murugappan, H. Liu, K. Ramani, *et al.*, "Shape-it-up: Hand gesture based creative expression of 3d shapes using intelligent generalized cylinders," *Computer-Aided Design*, vol. 45, no. 2, pp. 277–287, 2013.

[174] K. Natarajan, T.-H. D. Nguyen, and M. Mete, "Hand gesture controlled drones: An open source library," in *2018 1st International Conference on Data Intelligence and Security (ICDIS)*, IEEE, 2018, pp. 168–175.

[175] M. Nebeling, D. Ott, and M. C. Norrie, "Kinect analysis: A system for recording, analysing and sharing multimodal interaction elicitation studies," in *Proceedings of the 7th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, ser. EICS '15, Duisburg, Germany: Association for Computing Machinery, 2015, pp. 142–151, ISBN: 9781450336468. DOI: `10.1145/2774225.2774846`.

[176] A. Nguyen and A. Kunz, "Discrete scene rotation during blinks and its effect on redirected walking algorithms," in *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*, ser. VRST '18, Tokyo, Japan: Association for Computing Machinery, 2018. DOI: `10.1145/3281505.3281515`.

[177] R. Nguyen, "Integrating in-hand physical objects in mixed reality interactions," in *27th International Conference on Intelligent User Interfaces*, ser. IUI '22 Companion, Helsinki, Finland: Association for Computing Machinery, 2022, pp. 129–133, ISBN: 9781450391450. DOI: `10.1145/3490100.3516476`.

[178] T. Nguyen-Vo, B. E. Riecke, W. Stuerzlinger, D.-M. Pham, and E. Kruijff, "Naviboard and navichair: Limited translation combined with full rotation for efficient virtual locomotion," *IEEE transactions on visualization and computer graphics*, 2019.

[179] N. C. Nilsson, S. Serafin, F. Steinicke, and R. Nordahl, "Natural Walking in Virtual Reality: A Review," *Computers in Entertainment*, vol. 16, no. 2, 8:1–8:22, Apr. 2018. DOI: `10.1145/3180658`.

[180] N. C. Nilsson *et al.*, "15 Years of Research on Redirected Walking in Immersive Virtual Environments," *IEEE Computer Graphics and Applications*, vol. 38, no. 2, pp. 44–56, Mar. 2018, Conference Name: IEEE Computer Graphics and Applications, ISSN: 1558-1756. DOI: `10.1109/MCG.2018.111125628`.

[181] A. S. Nittala, N. Li, S. Cartwright, K. Takashima, E. Sharlin, and M. C. Sousa, "Planwell: Spatial user interface for collaborative petroleum well-planning," in *SIGGRAPH Asia 2015 Mobile Graphics and Interactive Applications*, 2015, pp. 1–8.

[182] M. Norman, G. A. Lee, R. T. Smith, and M. Billingurst, "The impact of remote user's role in a mixed reality mixed presence system," in *The 17th International Conference on Virtual-Reality Continuum and its Applications in Industry*, 2019, pp. 1–9.

[183] NVIDIA, *Nvidia holodeck*, 2020.

[184] D. C. Ogdon, "Hololens and vive pro: Virtual reality headsets," *Journal of the Medical Library Association: JMLA*, vol. 107, no. 1, p. 118, 2019.

[185] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Efficient model-based 3d tracking of hand articulations using kinect.," in *BmVC*, vol. 1, 2011, p. 3.

[186] P. A. Olin, A. M. Issa, T. Feuchtner, and K. Grønbæk, "Designing for heterogeneous cross-device collaboration and social interaction in virtual reality," in *32nd Australian Conference on Human-Computer Interaction*, ser. OzCHI '20, Sydney, NSW, Australia: Association for Computing Machinery, 2020, pp. 112–127, ISBN: 9781450389754. DOI: `10.1145/3441000.3441070`.

[187] VR-ON, *Stagevr*, 2020.

[188] S. Orts-Escolano *et al.*, “Holoportation: Virtual 3d teleportation in real-time,” in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, 2016, pp. 741–754.

[189] J. Ou, X. Chen, S. R. Fussell, and J. Yang, “Dove: Drawing over video environment,” in *Proceedings of the eleventh ACM international conference on Multimedia*, 2003, pp. 100–101.

[190] Y. S. Pai and K. Kunze, “Armswing: Using arm swings for accessible and immersive navigation in ar/vr spaces,” in *Proceedings of the 16th International Conference on Mobile and Ubiquitous Multimedia*, 2017, pp. 189–198.

[191] Y. Pan and K. Mitchell, “Posemmr: A collaborative mixed reality authoring tool for character animation,” in *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, 2020, pp. 758–759. DOI: `10.1109/VRW50115.2020.00230`.

[192] A. Parnami, A. Gupta, G. Reyes, R. Sadana, Y. Li, and G. Abowd, “Mogeste: Mobile tool for in-situ motion gesture design,” in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, ser. UbiComp ’16, Heidelberg, Germany: Association for Computing Machinery, 2016, pp. 345–348, ISBN: 9781450344623. DOI: `10.1145/2968219.2971395`.

[193] A. Patney *et al.*, “Towards foveated rendering for gaze-tracked virtual reality,” vol. 35, no. 6, Nov. 2016, ISSN: 0730-0301. DOI: `10.1145/2980179.2980246`.

[194] S. Pei, A. Chen, J. Lee, and Y. Zhang, “Hand interfaces: Using hands to imitate objects in ar/vr for expressive interactions,” in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, ser. CHI ’22, New Orleans, LA, USA: Association for Computing Machinery, 2022, ISBN: 9781450391573. DOI: `10.1145/3491102.3501898`.

[195] D. N. E. Phon, M. B. Ali, and N. D. A. Halim, “Collaborative augmented reality in education: A review,” in *2014 International Conference on Teaching and Learning in Computing and Engineering*, 2014, pp. 78–83. DOI: `10.1109/LaTiCE.2014.23`.

[196] T. Piumsomboon, A. Day, B. Ens, Y. Lee, G. Lee, and M. Billinghurst, “Exploring enhancements for remote mixed reality collaboration,” in *SIGGRAPH Asia 2017 Mobile Graphics & Interactive Applications*, ser. SA ’17, Bangkok, Thailand: Association for Computing Machinery, 2017, ISBN: 9781450354103. DOI: `10.1145/3132787.3139200`.

[197] T. Piumsomboon, Y. Lee, G. Lee, and M. Billinghurst, “Covar: A collaborative virtual and augmented reality system for remote collaboration,” in *SIGGRAPH Asia 2017 Emerging Technologies*, ser. SA ’17, Bangkok, Thailand: Association for Computing Machinery, 2017, ISBN: 9781450354042. DOI: `10.1145/3132818.3132822`.

[198] T. Piumsomboon, Y. Lee, G. A. Lee, A. Dey, and M. Billinghurst, "Empathic mixed reality: Sharing what you feel and interacting with what you see," in *2017 International Symposium on Ubiquitous Virtual Reality (ISUVR)*, 2017, pp. 38–41. DOI: `10.1109/ISUVR.2017.20`.

[199] T. Piumsomboon *et al.*, "Mini-me: An adaptive avatar for mixed reality remote collaboration," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, ser. CHI '18, Montreal QC, Canada: Association for Computing Machinery, 2018, pp. 1–13, ISBN: 9781450356206. DOI: `10.1145/3173574.3173620`.

[200] V. E. H. PLC, *Engagevr: Communicate, teach, learn*, 2020.

[201] E. Poppe, R. Brown, D. Johnson, and J. Recker, "A prototype augmented reality collaborative process modelling tool," in *Proceedings of the Demo Track of the Ninth Conference on Business Process Management 2011*, H. Ludwig and H. Reijers, Eds., http://ceur-ws.org: Sun SITE Central Europe, 2011, pp. 1–5.

[202] E. Poppe, R. Brown, D. Johnson, and J. Recker, "Preliminary evaluation of an augmented reality collaborative process modelling system," in *2012 International Conference on Cyberworlds*, 2012, pp. 77–84. DOI: `10.1109/CW.2012.18`.

[203] L. Pouliquen-Lardy, I. Milleville-Pennel, F. Guillaume, and F. Mars, "Remote collaboration in virtual reality: Asymmetrical effects of task distribution on spatial processing and mental workload," *Virtual Reality*, vol. 20, no. 4, pp. 213–220, 2016. DOI: `10.1007/s10055-016-0294-8`.

[204] P. Premaratne, "Historical development of hand gesture recognition," in *Human Computer Interaction Using Hand Gestures*. Singapore: Springer Singapore, 2014, pp. 5–29, ISBN: 978-981-4585-69-9. DOI: `10.1007/978-981-4585-69-9_2`.

[205] I. Rae, B. Mutlu, and L. Takayama, "Bodies in motion: Mobility, presence, and task awareness in telepresence," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '14, Toronto, Ontario, Canada: Association for Computing Machinery, 2014, pp. 2153–2162, ISBN: 9781450324731. DOI: `10.1145/2556288.2557047`.

[206] J. Rambach, P. Lesur, A. Pagani, and D. Stricker, "Slamcraft: Dense planar rgb monocular slam," in *2019 16th International Conference on Machine Vision Applications (MVA)*, 2019, pp. 1–6. DOI: `10.23919/MVA.2019.8757982`.

[207] J. Rambach, A. Pagani, and D. Stricker, "[poster] augmented things: Enhancing ar applications leveraging the internet of things and universal 3d object tracking," in *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*, 2017, pp. 103–108. DOI: `10.1109/ISMAR-Adjunct.2017.42`.

[208] S. S. Rani, K. Dhrisya, and M. Ahalyadas, "Hand gesture control of virtual object in augmented reality," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, IEEE, 2017, pp. 1500–1505.

[209] J. Rantala, J. Kangas, O. Koskinen, T. Nukarinen, and R. Raisamo, "Comparison of controller-based locomotion techniques for visual observation in virtual reality," *Multimodal Technologies and Interaction*, vol. 5, no. 7, p. 31, 2021.

[210] H. Regenbrecht, M. Haller, J. Hauber, and M. Billinghurst, "Carpeno: Interfacing remote collaborative virtual environments with table-top interaction," *Virtual Reality*, vol. 10, no. 2, pp. 95–107, 2006.

[211] H. Regenbrecht and T. Schubert, "Real and illusory interactions enhance presence in virtual environments," *Presence: Teleoperators & Virtual Environments*, vol. 11, no. 4, pp. 425–434, 2002.

[212] H. Regenbrecht *et al.*, "Using augmented virtuality for remote collaboration," *Presence: Teleoperators & Virtual Environments*, vol. 13, no. 3, pp. 338–354, 2004.

[213] H. T. Regenbrecht, M. Wagner, and G. Baratoff, "Magicmeeting: A collaborative tangible augmented reality system," *Virtual Reality*, vol. 6, no. 3, pp. 151–166, 2002.

[214] L. Research, *Second life*, 2020.

[215] T. Rhee, L. Petikam, B. Allen, and A. Chalmers, "Mr360: Mixed reality rendering for 360 panoramic videos," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 4, pp. 1379–1388, 2017.

[216] A. Riener *et al.*, "Standardization of the in-car gesture interaction space," in *Proceedings of the 5th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, 2013, pp. 14–21.

[217] R. Room, *Rec room*, 2020.

[218] M. Rymaszewski, W. J. Au, M. Wallace, C. Winters, C. Ondrejka, and B. Batstone-Cunningham, *Second life: The official guide*. John Wiley & Sons, 2007.

[219] B. Ryskeldiev, "Spatial social media: Towards collaborative mixed reality telepresence "on the go"," in *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018, pp. 1–4.

[220] B. Ryskeldiev, M. Cohen, and J. Herder, "Streamspace: Pervasive mixed reality telepresence for remote collaboration on mobile devices," *Journal of Information Processing*, vol. 26, pp. 177–185, 2018.

[221] K. Sakong and T.-j. Nam, "Supporting telepresence by visual and physical cues in distributed 3d collaborative design environments," in *CHI'06 Extended Abstracts on Human Factors in Computing Systems*, 2006, pp. 1283–1288.

[222] A. R. Sarkar, G. Sanyal, and S. Majumder, “Hand gesture recognition systems: A survey,” *International Journal of Computer Applications*, vol. 71, no. 15, 2013.

[223] B. Sarupuri, S. Hoermann, F. Steinicke, and R. W. Lindeman, “Triggerwalking: A biomechanically-inspired locomotion user interface for efficient realistic virtual walking,” in *Proceedings of the 5th Symposium on Spatial User Interaction*, 2017, pp. 138–147.

[224] P. Sasikumar, L. Gao, H. Bai, and M. Billinghurst, “Wearable remotefusion: A mixed reality remote collaboration system with local eye gaze and remote hand gesture sharing,” in *2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, IEEE, 2019, pp. 393–394.

[225] K. A. Satriadi, B. Ens, M. Cordeil, B. Jenny, T. Czauderna, and W. Willett, “Augmented reality map navigation with freehand gestures,” in *2019 IEEE conference on virtual reality and 3D user interfaces (VR)*, IEEE, 2019, pp. 593–603.

[226] J. Sauro, *A practical guide to the system usability scale: Background, benchmarks & best practices.* Measuring Usability LLC, 2011.

[227] E. Sayyad, P. Sen, and T. Höllerer, “Panotrace: Interactive 3d modeling of surround-view panoramic images in virtual reality,” in *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*, 2017, pp. 1–10.

[228] A. Schäfer, G. Reis, and D. Stricker, “A survey on synchronous augmented, virtual and mixed reality remote collaboration systems,” *ACM Comput. Surv.*, vol. 1, no. 1, p. 23, Apr. 2022, Just Accepted, ISSN: 0360-0300. DOI: `10.1145/3533376`.

[229] A. Schäfer, G. Reis, and D. Stricker, “Anygesture: Arbitrary one-handed gestures for augmented, virtual, and mixed reality applications,” *Applied Sciences*, vol. 12, no. 4, p. 21, 2022, ISSN: 2076-3417. DOI: `10.3390/app12041888`.

[230] A. Schäfer, G. Reis, and D. Stricker, “Comparing controller with the hand gestures pinch and grab for picking up and placing virtual objects,” in *2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, IEEE, Christchurch, New Zealand: IEEE, 2022, pp. 738–739. DOI: `10.1109/VRW55335.2022.00220`.

[231] A. Schäfer, G. Reis, and D. Stricker, “Comparing controller with the hand gestures pinch and grab for picking up and placing virtual objects,” in *2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, 2022, pp. 738–739. DOI: `10.1109/VRW55335.2022.00220`.

[232] A. Schäfer, G. Reis, and D. Stricker, “Controlling continuous locomotion in vr with bare hands,” in *EuroXR 2022*, Cham: Springer International Publishing, 2022.

[233] A. Schäfer, G. Reis, and D. Stricker, "Controlling teleportation-based locomotion in virtual reality with hand gestures: A comparative evaluation of two-handed and one-handed techniques," *Electronics*, vol. 10, no. 6, 2021, ISSN: 2079-9292. DOI: `10.3390/electronics10060715`.

[234] A. Schäfer, G. Reis, and D. Stricker, "Investigating the sense of presence between handcrafted and panorama based virtual environments," in *Mensch Und Computer 2021*, ser. MuC '21, Ingolstadt, Germany: Association for Computing Machinery, 2021, pp. 402–405, ISBN: 9781450386456. DOI: `10.1145/3473856.3474024`.

[235] A. Schäfer, G. Reis, and D. Stricker, "Learning effect of lay people in gesture-based locomotion in virtual reality," in *Virtual, Augmented and Mixed Reality: Design and Development*, J. Y. C. Chen and G. Fragomeni, Eds., Cham: Springer International Publishing, 2022, pp. 369–378, ISBN: 978-3-031-05939-1. DOI: `10.1007/978-3-031-05939-1_25`.

[236] A. Schäfer, G. Reis, and D. Stricker, "The gesture authoring space: Authoring customised hand gestures for grasping virtual objects in immersive virtual environments," in *Proceedings of Mensch und Computer 2022*, New York, NY, USA: Association for Computing Machinery, 2022. DOI: `10.1145/3543758.3543766`.

[237] A. Schäfer, G. Reis, and D. Stricker, "Towards collaborative photorealistic vr meeting rooms," in *Proceedings of Mensch Und Computer 2019*, ser. MuC'19, Hamburg, Germany: Association for Computing Machinery, 2019, pp. 599–603, ISBN: 9781450371988. DOI: `10.1145/3340764.3344466`.

[238] R. Schlünsen, O. Ariza, and F. Steinicke, "A vr study on freehand vs. widgets for 3d manipulation tasks," in *Proceedings of Mensch und Computer 2019*, 2019, pp. 223–233.

[239] P. Schmitz, J. Hildebrandt, A. C. Valdez, L. Kobbelt, and M. Ziefle, "You spin my head right round: Threshold of limited immersion for rotation gains in redirected walking," *IEEE transactions on visualization and computer graphics*, vol. 24, no. 4, pp. 1623–1632, 2018.

[240] R. Schroeter and M. A. Gerber, "A Low-Cost VR-Based Automated Driving Simulator for Rapid Automotive UI Prototyping," pp. 248–251, 2018. DOI: `10.1145/3239092.3267418`.

[241] T. Schubert, F. Friedmann, and H. Regenbrecht, "The experience of presence: Factor analytic insights," *Presence: Teleoperators & Virtual Environments*, vol. 10, no. 3, pp. 266–281, 2001.

[242] M. J. Schuemie, P. van der Straaten, M. Krijn, and C. A. van der Mast, "Research on Presence in Virtual Reality: A Survey," *CyberPsychology & Behavior*, vol. 4, no. 2, pp. 183–201, Apr. 2001, Publisher: Mary Ann Liebert, Inc., publishers, ISSN: 1094-9313. DOI: `10.1089/109493101300117884`.

[243] V. Schwind, P. Knierim, N. Haas, and N. Henze, “Using presence questionnaires in virtual reality,” in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–12.

[244] Q. Shao *et al.*, “Teaching american sign language in mixed reality,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 4, pp. 1–27, 2020.

[245] Y. Shen, S. Ong, and A. Nee, “A framework for multiple-view product representation using augmented reality,” in *2006 International Conference on Cyberworlds*, IEEE, 2006, pp. 157–164.

[246] Y. Shen, S. K. Ong, and A. Y. Nee, “Augmented reality for collaborative product design and development,” *Design Studies*, vol. 31, no. 2, pp. 118–145, 2010.

[247] Y. Shen, S.-K. Ong, and A. Y. Nee, “Collaborative design in 3d space,” in *Proceedings of The 7th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, 2008, pp. 1–6.

[248] Y. Shen, S.-K. Ong, and A. Y. Nee, “Product information visualization and augmentation in collaborative design,” *Computer-Aided Design*, vol. 40, no. 9, pp. 963–974, 2008.

[249] R. Sidharta, J. Oliver, and A. Sannier, “Augmented reality tangible interface for distributed design review,” in *International Conference on Computer Graphics, Imaging and Visualisation (CGIV’06)*, IEEE, 2006, pp. 464–470.

[250] A. L. Simeone, N. C. Nilsson, A. Zenner, M. Speicher, and F. Daiber, “The space bender: Supporting natural walking via overt manipulation of the virtual environment,” in *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, IEEE, 2020, pp. 598–606.

[251] D. Simulations, *Rumii*, 2020.

[252] SineSpace, *Breakroom, the social hub for remote teams*, 2020.

[253] F. Škola *et al.*, “Virtual reality with 360-video storytelling in cultural heritage: Study of presence, engagement, and immersion,” *Sensors*, vol. 20, no. 20, p. 5851, 2020.

[254] M. Slater, “The concept of presence and its measurement,” *PEACH Summer School, Santorini, Greece*, 2007.

[255] M. Slater, V. Linakis, M. Usoh, and R. Kooper, “Immersion, presence and performance in virtual environments: An experiment with tri-dimensional chess,” in *Proceedings of the ACM symposium on virtual reality software and technology*, 1996, pp. 163–172.

[256] M. Slater and M. Usoh, “Representations systems, perceptual position, and presence in immersive virtual environments,” *Presence: Teleoperators & Virtual Environments*, vol. 2, no. 3, pp. 221–233, 1993.

[257] M. Slater and S. Wilbur, “A framework for immersive virtual environments (five): Speculations on the role of presence in virtual environments,” *Presence: Teleoperators & Virtual Environments*, vol. 6, no. 6, pp. 603–616, 1997.

[258] A. H. Smeragliuolo, N. J. Hill, L. Disla, and D. Putrino, “Validation of the Leap Motion Controller using markered motion capture technology,” *Journal of Biomechanics*, vol. 49, no. 9, pp. 1742–1750, 2016, ISSN: 18732380. DOI: `10.1016/j.jbiomech.2016.04.006`.

[259] R. S. Sodhi, B. R. Jones, D. Forsyth, B. P. Bailey, and G. Maciocci, “Bethere: 3d mobile collaboration with spatial input,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2013, pp. 179–188.

[260] P. Song, W. B. Goh, W. Hutama, C.-W. Fu, and X. Liu, “A handle bar metaphor for virtual object manipulation with mid-air interaction,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI ’12, Austin, Texas, USA: Association for Computing Machinery, 2012, pp. 1297–1306, ISBN: 9781450310154. DOI: `10.1145/2207676.2208585`.

[261] S. Sorli, D. Casas, M. Verschoor, A. Tajadura-Jiménez, and M. A. Otaduy, “Fine virtual manipulation with hands of different sizes,” in *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Bari, Italy: IEEE, 2021, pp. 304–309. DOI: `10.1109/ISMAR52148.2021.00046`.

[262] M. Speicher, J. Cao, A. Yu, H. Zhang, and M. Nebeling, “360anywhere: Mobile ad-hoc collaboration in any environment using 360 video and augmented reality,” *Proceedings of the ACM on Human-Computer Interaction*, vol. 2, no. EICS, pp. 1–20, 2018.

[263] M. Speicher, B. D. Hall, and M. Nebeling, “What is mixed reality?” In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ser. CHI ’19, Glasgow, Scotland Uk: Association for Computing Machinery, 2019, pp. 1–15, ISBN: 9781450359702. DOI: `10.1145/3290605.3300767`.

[264] M. Speicher and M. Nebeling, “Gesturewiz: A human-powered gesture design environment for user interface prototypes,” in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, ser. CHI ’18, Montreal QC, Canada: Association for Computing Machinery, 2018, pp. 1–11, ISBN: 9781450356206. DOI: `10.1145/3173574.3173681`.

[265] R. R. Strauss, R. Ramanujan, A. Becker, and T. C. Peck, “A steering algorithm for redirected walking using reinforcement learning,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 5, pp. 1955–1963, 2020.

[266] H. Sun, Y. Liu, Z. Zhang, X. Liu, and Y. Wang, "Employing different viewpoints for remote guidance in a collaborative augmented environment," in *Proceedings of the sixth international symposium of Chinese CHI*, 2018, pp. 64–70.

[267] H. Sun, Z. Zhang, Y. Liu, and H. B. Duh, "Optobridge: Assisting skill acquisition in the remote experimental collaboration," in *Proceedings of the 28th Australian Conference on Computer-Human Interaction*, 2016, pp. 195–199.

[268] Y. Sun, A. Armengol-Urpi, S. N. R. Kantareddy, J. Siegel, and S. Sarma, "Magichand: Interact with iot devices in augmented reality environment," in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, IEEE, 2019, pp. 1738–1743.

[269] H. B. Surale, F. Matulic, and D. Vogel, "Experimental analysis of barehand mid-air mode-switching techniques in virtual reality," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–14.

[270] S. Suzuki, H. Suzuki, and M. Sato, "Grasping a virtual object with a bare hand," in *ACM SIGGRAPH 2014 Posters*, ser. SIGGRAPH '14, Vancouver, Canada: Association for Computing Machinery, 2014, ISBN: 9781450329583. DOI: `10.1145/2614217.2630574`.

[271] V. Sync, *Vive sync*, 2020.

[272] S. Systems, *Spatial*, 2020.

[273] M. Tait and M. Billinghurst, "The effect of view independence in a collaborative ar system," *Computer Supported Cooperative Work (CSCW)*, vol. 24, no. 6, pp. 563–589, 2015.

[274] Z. Tan, Y. Hu, and K. Xu, "Virtual reality based immersive telepresence system for remote conversation and collaboration," in *International Workshop on Next Generation Computer Animation Techniques*, Springer, 2017, pp. 234–247.

[275] H. Tang, H. Liu, W. Xiao, and N. Sebe, "Fast and robust dynamic hand gesture recognition via key frames extraction and feature fusion," *Neurocomputing*, vol. 331, pp. 424–433, 2019.

[276] TechViz, *Techviz vr*, 2020.

[277] J. N. Templeman, P. S. Denbrook, and L. E. Sibert, "Virtual locomotion: Walking in place through virtual environments," *Presence*, vol. 8, no. 6, pp. 598–617, 1999. DOI: `10.1162/105474699566512`.

[278] T. Teo, G. A. Lee, M. Billinghurst, and M. Adcock, "360drops: Mixed reality remote collaboration using 360 panoramas within the 3d scene," in *SIGGRAPH Asia 2019 Emerging Technologies*, 2019, pp. 1–2.

[279] T. Teo, A. F. Hayati, G. A. Lee, M. Billinghurst, and M. Adcock, "A technique for mixed reality remote collaboration using 360 panoramas in 3d reconstructed scenes," in *25th ACM Symposium on Virtual Reality Software and Technology*, 2019, pp. 1–11.

[280] T. Teo, L. Lawrence, G. A. Lee, M. Billinghurst, and M. Adcock, "Mixed reality remote collaboration combining 360 video and 3d reconstruction," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–14.

[281] T. Teo, G. A. Lee, M. Billinghurst, and M. Adcock, "Merging live and static 360 panoramas inside a 3d scene for mixed reality remote collaboration," in *2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, IEEE, 2019, pp. 22–25.

[282] TheWild, *Thewild: Immersive collaboration for teams*, 2020.

[283] J. Thomas and E. S. Rosenberg, "A general reactive algorithm for redirected walking using artificial potential functions," in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, IEEE, 2019, pp. 56–62.

[284] Ultraleap, *Leap motion controller*, 2020.

[285] Ultraleap, *Leap Motion SDK*, `https://developer.leapmotion.com/`, [Online; accessed 24-May-2021], 2021.

[286] S. Utzig, R. Kaps, S. M. Azeem, and A. Gerndt, "Augmented reality for remote collaboration in aircraft maintenance tasks," in *2019 IEEE Aerospace Conference*, IEEE, 2019, pp. 1–10.

[287] A. Vaitkevičius, M. Taroza, T. Blažauskas, R. Damaševičius, R. Maskeliūnas, and M. Woźniak, "Recognition of american sign language gestures in a virtual reality using leap motion," *Applied Sciences*, vol. 9, no. 3, p. 445, 2019.

[288] R.-D. Vatavu, L. Anthony, and J. O. Wobbrock, "Gestures as point clouds: A $p recognizer for user interface prototypes," in *Proceedings of the 14th ACM International Conference on Multimodal Interaction*, ser. ICMI '12, Santa Monica, California, USA: Association for Computing Machinery, 2012, pp. 273–280, ISBN: 9781450314671. DOI: `10.1145/2388676.2388732`.

[289] R.-D. Vatavu and L.-B. Bilius, "Gesturing: A web-based tool for designing gesture input with rings, ring-like, and ring-ready devices," in *The 34th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '21, Virtual Event, USA: Association for Computing Machinery, 2021, pp. 710–723, ISBN: 9781450386357. DOI: `10.1145/3472749.3474780`.

[290] B. Verma and A. Choudhary, "Framework for dynamic hand gesture recognition using grassmann manifold for intelligent vehicles," *IET Intelligent Transport Systems*, vol. 12, no. 7, pp. 721–729, 2018.

[291] VirBELA, *Virbela*, 2020.

[292] A. Virtual, *Acadicus*, 2020.

[293] S. Vosinakis and P. Koutsabasis, "Evaluation of visual feedback techniques for virtual grasping with bare hands using leap motion and oculus rift," *Virtual Reality*, vol. 22, no. 1, pp. 47–62, 2018.

[294] vTime, *Vtime xr*, 2020.

[295] P. Wang *et al.*, “2.5 dhands: A gesture-based mr remote collaborative platform,” *The International Journal of Advanced Manufacturing Technology*, vol. 102, no. 5-8, pp. 1339–1353, 2019.

[296] P. Wang *et al.*, “A comprehensive survey of ar/mr-based co-design in manufacturing,” *Engineering with Computers*, pp. 1–24, 2019.

[297] P. Wang *et al.*, “Head pointer or eye gaze: Which helps more in mr remote collaboration?” In *2019 IEEE conference on virtual reality and 3D user interfaces (VR)*, IEEE, 2019, pp. 1219–1220.

[298] R. Y. Wang and J. Popović, “Real-time hand-tracking with a color glove,” *ACM transactions on graphics (TOG)*, vol. 28, no. 3, p. 63, 2009.

[299] X. Wang, “Exploring an innovative collaborative design space through mixed reality boundaries,” in *2007 11th International Conference on Computer Supported Cooperative Work in Design*, IEEE, 2007, pp. 264–269.

[300] X. Wang and R. Chen, “An experimental study on collaborative effectiveness of augmented reality potentials in urban design,” *CoDesign*, vol. 5, no. 4, pp. 229–244, 2009.

[301] X. Wang and P. S. Dunston, “Potential of augmented reality as an assistant viewer for computer-aided drawing,” *Journal of computing in civil engineering*, vol. 20, no. 6, pp. 437–441, 2006.

[302] X. Wang and P. S. Dunston, “User perspectives on mixed reality tabletop visualization for face-to-face collaborative design review,” *Automation in construction*, vol. 17, no. 4, pp. 399–412, 2008.

[303] X. Wang, P. E. Love, M. J. Kim, and W. Wang, “Mutual awareness in collaborative design: An augmented reality integrated telepresence system,” *Computers in Industry*, vol. 65, no. 2, pp. 314–324, 2014.

[304] Z. Wang, H. Wei, K. Zhang, and L. Xie, “Real walking in place: Hex-core-prototype omnidirectional treadmill,” in *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, IEEE, 2020, pp. 382–387.

[305] Wave, *Wavexr*, 2020.

[306] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, “Analysis of the accuracy and robustness of the leap motion controller,” *Sensors*, vol. 13, no. 5, pp. 6380–6393, 2013.

[307] T. Weißker, A. Kunert, B. Fröhlich, and A. Kulik, “Spatial updating and simulator sickness during steering and jumping in immersive virtual environments,” in *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, IEEE, 2018, pp. 97–104.

[308] T. Weissker, P. Bimberg, and B. Froehlich, "Getting there together: Group navigation in distributed virtual environments," *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 5, pp. 1860–1870, 2020.

[309] C. Wienrich, N. Döllinger, S. Kock, and K. Gramann, "User-centered extension of a locomotion typology: Movement-related sensory feedback and spatial learning," in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, IEEE, 2019, pp. 690–698.

[310] J. O. Wobbrock, A. D. Wilson, and Y. Li, "Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes," in *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '07, Newport, Rhode Island, USA: Association for Computing Machinery, 2007, pp. 159–168, ISBN: 9781595936790. DOI: `10.1145/1294211.1294238`.

[311] D. Wolf, K. Rogers, C. Kunder, and E. Rukzio, "Jumpvr: Jump-based locomotion augmentation for virtual reality," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020, pp. 1–12.

[312] WorldViz, *Worldviz: Where virtual becomes reality*, 2020.

[313] B. Yoon, H.-i. Kim, G. A. Lee, M. Billinghurst, and W. Woo, "The effect of avatar appearance on social presence in an augmented reality remote collaboration," in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, IEEE, 2019, pp. 547–556.

[314] R. Yu *et al.*, "Experiencing an invisible world war i battlefield through narrative-driven redirected walking in virtual reality," in *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, IEEE, 2018, pp. 313–319.

[315] N. Zenati, S. Benbelkacem, M. Belhocine, and A. Bellarbi, "A new ar interaction for collaborative e-maintenance system," *IFAC Proceedings Volumes*, vol. 46, no. 9, pp. 619–624, 2013.

[316] N. Zenati, M. Hamidia, A. Bellarbi, and S. Benbelkacem, "E-maintenance for photovoltaic power system in algeria," in *2015 IEEE International Conference on Industrial Technology (ICIT)*, IEEE, 2015, pp. 2594–2599.

[317] N. Zenati-Henda, A. Bellarbi, S. Benbelkacem, and M. Belhocine, "Augmented reality system based on hand gestures for remote maintenance," in *2014 International Conference on Multimedia Computing and Systems (ICMCS)*, IEEE, 2014, pp. 5–8.

[318] F. Zhang, S. Chu, R. Pan, N. Ji, and L. Xi, "Double hand-gesture interaction for walk-through in vr environment," in *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, 2017, pp. 539–544. DOI: `10.1109/ICIS.2017.7960051`.

[319] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang, and J. Yang, "A framework for hand gesture recognition based on accelerometer and emg sensors," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 41, no. 6, pp. 1064–1076, 2011.

[320] J. Zhao, R. An, R. Xu, and B. Lin, "Comparing hand gestures and the gamepad interfaces for locomotion in virtual environments," *arXiv preprint arXiv:2104.12182*, 2021.

[321] D. Zielasko, S. Freitag, D. Rausch, Y. C. Law, B. Weyers, and T. W. Kuhlen, "Blowclick: A non-verbal vocal input metaphor for clicking," in *Proceedings of the 3rd ACM Symposium on Spatial User Interaction*, ser. SUI '15, Los Angeles, California, USA: Association for Computing Machinery, 2015, pp. 20–23, ISBN: 9781450337038. DOI: `10.1145/2788940.2788953`.

[322] D. Zielasko, S. Horn, S. Freitag, B. Weyers, and T. W. Kuhlen, "Evaluation of hands-free hmd-based navigation techniques for immersive data analysis," in *2016 IEEE Symposium on 3D User Interfaces (3DUI)*, IEEE, 2016, pp. 113–119.

[323] D. Zielasko, S. Horn, S. Freitag, B. Weyers, and T. W. Kuhlen, "Evaluation of hands-free HMD-based navigation techniques for immersive data analysis," in *2016 IEEE Symposium on 3D User Interfaces (3DUI)*, Mar. 2016, pp. 113–119. DOI: `10.1109/3DUI.2016.7460040`.

[324] D. Zielasko, Y. C. Law, and B. Weyers, "Take a look around – the impact of decoupling gaze and travel-direction in seated and ground-based virtual reality utilizing torso-directed steering," in *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, 2020, pp. 398–406. DOI: `10.1109/VR46266.2020.00060`.

[325] D. Zielasko, N. Neha, B. Weyers, and T. W. Kuhlen, "A reliable nonverbal vocal input metaphor for clicking," in *2017 IEEE Symposium on 3D User Interfaces (3DUI)*, 2017, pp. 40–49. DOI: `10.1109/3DUI.2017.7893316`.

[326] J. Zillner, E. Mendez, and D. Wagner, "Augmented reality remote collaboration with dense reconstruction," in *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, IEEE, 2018, pp. 38–39.

Appendix A

List of Publications

- Alexander Schäfer, Gerd Reis, and Didier Stricker. 2019. **Towards Collaborative Photorealistic VR Meeting Rooms.** In Proceedings of Mensch und Computer 2019 (MuC'19). Association for Computing Machinery, New York, NY, USA, 599–603. DOI: https://doi.org/10.1145/3340764.3344466
- Alexander Schäfer, Tomoko Isomura, Gerd Reis, Katsumi Watanabe, and Didier Stricker. 2020. **MutualEyeContact: A conversation analysis tool with focus on eye contact.** In ACM Symposium on Eye Tracking Research and Applications (ETRA '20 Short Papers). Association for Computing Machinery, New York, NY, USA, Article 1, 1–5. DOI: https://doi.org/10.1145/3379156.3391340
- Alexander Schäfer, Gerd Reis, and Didier Stricker. 2020. **A Survey on Synchronous Augmented, Virtual and Mixed Reality Remote Collaboration Systems** ACM Computing Surveys (CSUR) (2021). DOI: https://doi.org/10.1145/3533376
- Johanna Menningen, Torben Fetzer, Alexander Schäfer, Gerd Reis, and Siegfried Siegesmund. 2020. **Ultrasound Tomography and 3D Scanning Technologies as a Tool to Constrain theWeathering State of Objects Made of Marble** Proceedings of the International Congress on the Deterioration and Conservation of Stone (STONE), 2020.
- Alexander Schäfer, Gerd Reis, and Didier Stricker. 2021. **Investigating the Sense of Presence Between Handcrafted and Panorama Based Virtual Environments.** In Mensch und Computer 2021 (MuC '21). Association for Computing Machinery, New York, NY, USA, 402–405. DOI: https://doi.org/10.1145/3473856.3474024
- Alexander Schäfer, Gerd Reis, and Didier Stricker. 2021. **Controlling Teleportation-Based Locomotion in Virtual Reality with Hand Gestures: A Comparative Evaluation of Two-Handed and One-Handed Techniques.** Electronics 2021, 10, 715.
  DOI: https://doi.org/10.3390/electronics10060715

- Jason Rambach, Gergana Lilligreen, Alexander Schäfer, Ranja Bankanal, Alexander Wiebel, and Didier Stricker. 2021. **A Survey on Applications of Augmented, Mixed and Virtual Reality for Nature and Environment.** In: Chen J.Y.C., Fragomeni G. (eds) Virtual, Augmented and Mixed Reality. HCII 2021. Lecture Notes in Computer Science, vol 12770. Springer, Cham. DOI: https://doi.org/10.1007/978-3-030-77599-5_45

- Alexander Schäfer, Gerd Reis, and Didier Stricker. 2022. **Learning Effect of Lay people in Gesture-Based Locomotion in Virtual Reality.** Learning Effect of Lay People in Gesture-Based Locomotion in Virtual Reality. In: Chen, J.Y.C., Fragomeni, G. (eds) Virtual, Augmented and Mixed Reality: Design and Development. HCII 2022. Lecture Notes in Computer Science, vol 13317. Springer, Cham. DOI: https://doi.org/10.1007/978-3-031-05939-1_25

- Alexander Schäfer, Gerd Reis, and Didier Stricker. 2022. **AnyGesture: Arbitrary One-Handed Gestures for Augmented, Virtual, and Mixed Reality Applications** Appl. Sci. 2022, 12, 1888.
  DOI: https://doi.org/10.3390/app12041888.

- Alexander Schäfer, Gerd Reis, and Didier Stricker. 2022. **Comparing Controller with the Hand Gestures Pinch and Grab for Picking Up and Placing Virtual Objects** 2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW), 2022. DOI: https://doi.org/10.1109/VRW55335.2022.00220.

- Alexander Schäfer, Gerd Reis, and Didier Stricker. 2022. **The Gesture Authoring Space: Authoring Customised Hand Gestures for Grasping Virtual Objects in Immersive Virtual Environments** In Mensch und Computer 2022 (MuC '22). Association for Computing Machinery, New York, NY, USA, 11 pages. DOI:
  https://doi.org/10.1145/3543758.3543766

- Alexander Schäfer, Gerd Reis, and Didier Stricker. EuroXR 2022. **Controlling Continuous Locomotion in VR With Bare Hands** EuroXR 2022. Lecture Notes in Computer Science, vol 13484. Springer, Cham. DOI: https://doi.org/10.1007/978-3-031-16234-3_11

Appendix B

Curriculum Vitae

Personal Data

Name Alexander Schäfer

Work Experience

2018–2022	**Researcher** TU Kaiserslautern Department of Computer Science, Augmented Vision Group
2016–2018	**Software Engineer for Embedded Systems** Comlet Verteilte Systeme GmbH
2012–2016	**Research Assistant** German Research Center for Artificial Intelligence Augmented Vision Department

Education

2013–2015	**Master of Science in Informatics** HS Kaiserslautern Thesis: *Interaktive Visualisierung komplexer 3D-Szenen im Web*
2009–2013	**Bachelor of Science in Informatics** HS Kaiserslautern Thesis: *Bewegunggsynthese für realistische Bewegungen von 3D-Charakteren*
2006–2008	**Fachabitur** Berufsbildende Schule Rodalben