Conveying Nonverbal Communication in Mixed Reality-based Telepresence Systems

DISSERTATION
zur Erlangung des Grades
'Doktor der Naturwissenschaften'
(Dr. rer. nat.)

am Promotionszentrum Angewandte Informatik (PZAI)
der hessischen Hochschulen für Angewandte Wissenschaften
Hochschule RheinMain & Hochschule Darmstadt &
Hochschule Fulda & Frankfurt University of Applied Sciences

vorgelegt von
Philipp Ladwig, M.Sc.

Betreuer:
Prof. Dr. Ralf Dörner, Hochschule RheinMain
Prof. Dr. Christian Geiger, Hochschule Düsseldorf

Gutachter:
Prof. Dr. Paul Grimm, Hochschule Darmstadt
Prof. Dr. Frank Steinicke, Universität Hamburg

Einreichungstermin: 31.07.2024
Prüfungstermin: 27.01.2025

Erscheinungsort und -jahr: Wiesbaden, 2025

philipp.ladwig@avaluma.ai
philipp_ladwig@hotmail.de

Abstract

Communication represents a fundamental aspect of human interaction, and advancements in technology have enabled the transmission of increasingly complex information over vast distances. Technological advancement has seen the evolution of communication from the use of rudimentary signals, such as smoke signals and Morse code, to the advent of sophisticated solutions, including video conferencing. Recently, Mixed Reality (MR) has demonstrated considerable potential for the transmission of rich spatial data, particularly with regard to nonverbal communication cues such as full-body gestures or authentic eye contact. Despite the existence of early versions of immersive 3D telepresence applications, their widespread adoption is hindered by limitations, notably the obstruction of facial expressions by head-mounted displays (HMDs). The HMD obstructs the ability to discern facial expressions. This dissertation addresses the key challenges of current immersive telepresence systems by combining self-developed hardware prototypes and off-the-shelf hardware with novel software solutions from the field of deep learning.

The core contributions of this work include novel approaches to face tracking under an HMD, face rendering, and face animation. For decades, computer graphics researchers have sought to render human faces in a manner that is as authentic as possible, often requiring a significant amount of manual effort in 3D modeling. This dissertation is focused on the development of photorealistic facial rendering and animation techniques that employ Generative Adversarial Networks (GANs) and Implicit Neural Representations (INRs). These techniques yield superior visual quality with less computing power than traditional methods, while also enabling the automatic creation of a face avatar in a fraction of the time required for manual 3D modeling. To animate these avatars in an immersive MR setting, we introduce a hardware prototype of a face-tracking HMD that captures facial expressions via Convolutional Neural Networks (CNNs).

In addition, we present a middleware that standardizes interfaces for various full-body tracking systems. This simplifies the operation and integration of different systems significantly and standardizes the data representation of gestures and nonverbal communication, for example, through the use of a standardized animation skeleton.

Two user studies provide empirical evidence to support the technological advancements presented in this thesis. The first study demonstrates the influence of personalized avatars on social presence, whereas the second quantifies the efficiency gains in remote collaboration facilitated by nonverbal communication through a shared task space supported by pointing gestures. Additionally, the dissertation presents design guidelines for remote collaboration systems derived from a literature review. By introducing novel solutions for effective remote collaboration, this dissertation has the potential to reduce the necessity for physical travel and its associated environmental impacts in the future.

Keywords: Body Tracking, Coordinate-based Neural Networks, Digital Humans, Face-to-Face, Face Tracking, Implicit Neural Representation, Middleware, Mixed Reality, Neural Rendering, Nonverbal Communication, Generative Adversarial Networks (GAN), Presence, Remote Collaboration, RGB-D, Shared Task Space, Uncanny Valley

Abstract

Zusammenfassung

Kommunikation ist ein grundlegender Aspekt der menschlichen Interaktion, und die Fortschritte in der Technologie haben die Übertragung von immer komplexeren Informationen über große Entfernungen ermöglicht. Der technologische Fortschritt hat die Entwicklung der Kommunikation von der Verwendung rudimentärer Signale wie Rauchzeichen und Morsezeichen bis hin zu hochentwickelten Lösungen wie Videokonferenzen ermöglicht. In jüngster Zeit hat Mixed Reality (MR) ein beträchtliches Potenzial für die Übertragung umfangreicher räumlicher Daten gezeigt, insbesondere im Hinblick auf nonverbale Kommunikationshinweise wie Ganzkörpergesten oder authentischen Blickkontakt. Obwohl es bereits frühe Versionen von immersiven 3D-Telepräsenzanwendungen gibt, wird ihre weitere Verbreitung durch Einschränkungen behindert, insbesondere durch die Verdeckung der Mimik durch Head-Mounted Displays (HMDs). Das HMD behindert die Fähigkeit, Gesichtsausdrücke zu erkennen. Diese Dissertation befasst sich mit den zentralen Herausforderungen aktueller immersiver Telepräsenzsysteme, indem sie selbst entwickelte Hardware-Prototypen und handelsübliche Hardware mit neuartigen Softwarelösungen aus dem Bereich des Deep Learning kombiniert.

Zu den wichtigsten Beiträgen dieser Arbeit gehören neuartige Ansätze des Face Tracking unter einem HMD, zum Face Rendering und zur Face Animation. Seit Jahrzehnten versuchen Forscher im Bereich der Computergrafik, menschliche Gesichter immer authentischer darzustellen, was oft einen erheblichen manuellen Aufwand bei der 3D-Modellierung voraussetzt. Diese Dissertation konzentriert sich auf die Entwicklung von fotorealistischen Face Rendering sowie Animationstechniken, die Generative Adversarial Networks (GANs) und Implizite Neuronale Repräsentationen (INRs) verwenden. Diese Techniken liefern eine bessere visuelle Qualität bei geringerer Rechenleistung als klassische Methoden der Computergrafik und ermöglichen gleichzeitig die automatische Erstellung eines Gesichtsavatars in einem Bruchteil der Zeit, die für die manuelle 3D-Modellierung erforderlich wäre. Um diese Avatare in einer immersiven MR-Umgebung zu animieren, stellen wir Hardware-Prototypen eines Face-Tracking-HMDs vor, der Gesichtsausdrücke über Convolutional Neural Networks (CNNs) erfasst.

Zusätzlich stellen wir eine Middleware vor, die Schnittstellen für verschiedene Ganzkörper-Tracking-Systeme standardisiert. Dies vereinfacht die Bedienung und Integration verschiedener Systeme erheblich und standardisiert die Datendarstellung von Gesten und nonverbaler Kommunikation, z.B. durch die Verwendung eines standardisierten Animationsskeletts.

Zwei Nutzerstudien liefern empirische Belege für die in dieser Arbeit vorgestellten technologischen Weiterentwicklungen. Die erste Studie zeigt den Einfluss von personalisierten Avataren auf die soziale Präsenz, während die zweite Studie die Effizienzgewinne bei der entfernten Zusammenarbeit quantifiziert, die durch nonverbale Kommunikation in einem durch Zeigegesten unterstützten *Shared Task Space* ermöglicht werden. Darüber hinaus werden in dieser Dissertation Design Guidelines für Systeme der entfernten Zusammenarbeit vorgestellt, die aus einer Literaturübersicht abgeleitet wurden. Durch die Entwicklung neuartiger Lösungen für eine effektive entfernte Zusammenarbeit hat diese

Arbeit das Potenzial, die Notwendigkeit physischer Reisen und die damit verbundenen Umweltauswirkungen in Zukunft zu verringern.

Schlagwörter: Body Tracking, Coordinate-based Neural Networks, Digital Humans, Face-to-Face, Face Tracking, Implicit Neural Representation, Middleware, Mixed Reality, Neural Rendering, Nonverbal Communication, Generative Adversarial Networks (GAN), Presence, Remote Collaboration, RGB-D, Shared Task Space, Uncanny Valley

Erklärung zur Autorenschaft

Ich, Philipp Ladwig, erkläre hiermit, dass

- die Dissertation selbständig und ohne unerlaubte fremde Hilfe und nur mit den angegebenen Hilfen angefertigt wurde;
- alle wörtlich oder sinngemäß aus veröffentlichten Schriften entnommene Textstellen und alle Angaben, die auf mündlichen Auskünften beruhen, als solche kenntlich gemacht sind;
- Text-Generierungs-Werkzeuge wie Large Language Modells (LLM) ausschließlich für das Umschreiben, Übersetzen, die Verbesserung der Lesbarkeit oder die Aufbereitung von eigens erstellten Textteilen genutzt wurde, jedoch keine wissenschaftlichen Erkenntnisse oder vergleichbare urheberrechtlich-kritischen Ausgaben des LLMs verwendet wurden. Die Prompts wurden von mir selbst erstellt;
- die Grundsätze guter wissenschaftlicher Praxis eingehalten sind;

Abschnitte der vorliegenden Thesis waren Teil der Arbeiten, die in den folgenden Artikeln und Tagungsbänder vorgestellt und veröffentlicht wurden. Die Artikel sind chronologisch aufgelistet:

Liste eigener Publikationen, die Teil dieser Arbeit sind

[LG19a] **Ladwig, Philipp** and Christian Geiger. "A Literature Review on Collaboration in Mixed Reality". In: *Smart Industry & Smart Education*. Ed. by Michael E. Auer and Reinhard Langmann. Cham: Springer International Publishing, 2019, pp. 591–600. ISBN: 978-3-319-95678-7. DOI: 10.1007/978-3-319-95678-7_65.

[Lad+19] **Ladwig, Philipp**, Bastian Dewitz, Hendrik Preu, and Mitja Säger. "Remote Guidance for Machine Maintenance Supported by Physical LEDs and Virtual Reality". In: *Proceedings of Mensch Und Computer 2019*. Ed. by Florian Alt, Andreas Bulling, and Tanja Döring. MuC'19. Hamburg, Germany: Association for Computing Machinery, 2019, pp. 255–262. ISBN: 9781450371988. DOI: 10.1145/3340764.3340780.

[LG19b] **Ladwig, Philipp** and Christian Geiger. "The Effects on Presence of Personalized and Generic Avatar Faces". In: *Virtuelle und Erweiterte Realität - GI VR/AR Workshop*. Ed. by Paul Grimm, Yvonne Jung, Ralf Dörner, and Christian Geiger. Berichte aus der Informatik. Shaker Verlag, 2019. ISBN: 9783844068870.

[LPG20] **Ladwig, Philipp**, Alexander Pech, and Christian Geiger. "Auf dem Weg zu Face-to-Face-Telepräsenzanwendungen in Virtual Reality mit generativen neuronalen Netzen". In: *17. GI VR / AR Workshop*. Ed. by Benjamin Weyers, Christoph Lürig, and Daniel Zielasko. **Best Paper Award**. Gesellschaft für Informatik e.V., 2020. DOI: 10.18420/vrar2020_15.

[Lad+21] **Ladwig, Philipp**, Damian Zohlen, Manuel Zohlen, and Christian Geiger. "Towards 3D Scanning with Multiple RGB-D Sensors in Virtual Reality". In: *18. GI VR / AR Workshop*. Ed. by Martin Weier, Matthias Bues, and Reto Wechner. Gesellschaft für Informatik e.V., 2021. DOI: 10.18420/vrar2021_15.

[Lad+20a] **Ladwig, Philipp**, Alexander Pech, Ralf Dörner, and Christian Geiger. "Unmasking Communication Partners: A Low-Cost AI Solution for Digitally Removing Head-Mounted Displays in VR-Based Telepresence". In: *IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*. 2020, pp. 82–90. DOI: 10.1109/AIVR50618.2020.00025.

[Lad+20b] **Ladwig, Philipp**, Kester Evers, Eric J. Jansen, Ben Fischer, David Nowottnik, and Christian Geiger. "MotionHub: Middleware for Unification of Multiple Body Tracking Systems". In: *Proceedings of the 7th International Conference on Movement and Computing*. MOCO '20. Jersey City/Virtual, NJ, USA: Association for Computing Machinery, 2020. ISBN: 9781450375054. DOI: 10.1145/3401956.3404185.

[Lad24a] **Ladwig, Philipp**. *Deepfakes: Technische Hintergründe und Trends*. https://www.bpb.de/lernen/bewegtbild-und-politische-bildung/556238/deepfakes-technische-hintergruende-und-trends/. In Dossier: KI, Deepfakes, Soziale Medien. Bundeszentrale für politische Bildung. 2024.

[Lad24b] **Ladwig, Philipp**. *Was ist KI und welche Formen von KI gibt es?* https://www.bpb.de/lernen/bewegtbild-und-politische-bildung/555997/was-ist-ki-und-welche-formen-von-ki-gibt-es/. In Dossier: KI, Deepfakes, Soziale Medien. Bundeszentrale für politische Bildung. 2024.

[Lad+25] **Ladwig, Philipp**, Rene Ebertowski, Alexander Pech, Ralf Dörner, and Christian Geiger. "Towards a Pipeline for Real-Time Visualization of Faces for VR-based Telepresence and Live Broadcasting Utilizing Neural Rendering". In: *Journal of Virtual Reality and Broadcasting (JVRB)*. Vol. 18 (2024) 2024.1. Section: GI VR/AR 2020. 2025. DOI: 10.48663/1860-2037/18.2024.1.

Die folgenden Publikationen sind im Laufe meiner akademischen Laufbahn veröffentlicht worden, sind aber nicht unmittelbar Teil dieser Dissertation. Die Artikel sind ebenfalls chronologisch aufgelistet:

Liste eigener Publikationen, die nicht Teil dieser Arbeit sind

[LF16] **Ladwig, Philipp** and Jannik Fiedler. "Demo: Mesh Modellierung in Virtual Reality". In: *GI-VRAR, Workshop Proceedings / Tagungsband: Virtuelle und Erweiterte Realität - 13. Workshop der GI-Fachgruppe VR/AR,* ed. by Thies Pfeiffer, Julia Fröhlich, and Rolf Kruse. **Best Demo Award**. Aachen: Shaker Verlag, 2016. ISBN: 9783844047189.

[Bal+16] Marina Ballester Ripoll, Jens Herder, **Ladwig, Philipp**, and Kai Vermeegen. "Comparison of two Gesture Recognition Sensors for Virtual TV Studios". In: *GI-VRAR, Workshop Proceedings / Tagungsband: Virtuelle und Erweiterte Realität - 13. Workshop der GI-Fachgruppe VR/AR,* ed. by Thies Pfeiffer, Julia Fröhlich, and Rolf Kruse. 2016.

[KLG16] Okan Sadik Köse, **Ladwig, Philipp**, and Christian Geiger. "Fractal2Mesh: From Implicit 3D Fractal Volumes to 3D Polygonal Geometry". In: *GI-VRAR, Workshop Proceedings / Tagungsband: Virtuelle und Erweiterte Realität - 13. Workshop der GI-Fachgruppe VR/AR,* ed. by Thies Pfeiffer, Julia Fröhlich, and Rolf Kruse. 2016.

[LL16] **Ladwig, Philipp** and Birgit Lohmann. *Reflexion über die Bedeutung virtueller Welten für den Menschen*. Philosophy & Technology. PHILOTEC.de. 2016.

[Dae+16] Jeff Daemen, Jens Herder, Cornelius Koch, **Ladwig, Philipp**, Roman Wiche, and Kai Wilgen. "Semi-Automatic Camera and Switcher Control for Live Broadcast". In: *Proceedings of the ACM International Conference on Interactive Experiences for TV and Online Video*. TVX '16. Chicago, Illinois, USA: Association for Computing Machinery, 2016, pp. 129–134. ISBN: 9781450340670. DOI: 10.1145/2932206.2933559.

[Dae+17] Jeff Daemen, Jens Herder, Cornelius Koch, **Ladwig, Philipp**, Roman Wiche, and Kai Wilgen. "Halbautomatische Steuerung von Kamera und Bildmischer bei Live-Übertragungen". In: *Fachzeitschrift für Fernsehen, Film und Elektronische Medien*. 11. 2017, pp. 501–505.

[LHG17] **Ladwig, Philipp**, Jens Herder, and Christian Geiger. "Towards Precise, Fast and Comfortable Immersive Polygon Mesh Modelling: Capitalising the Results of Past Research and Analysing the Needs of Professionals". In: *ICAT-EGVE 2017 - International Conference on Artificial Reality and Telexistence and Eurographics Symposium on Virtual Environments*. Ed. by Robert W. Lindeman, Gerd Bruder, and Daisuke Iwai. The Eurographics Association, 2017. ISBN: 978-3-03868-038-3. DOI: 10.2312/egve.20171360.

[Her+18a] Jens Herder, **Ladwig, Philipp**, Kai Vermeegen, Dennis Hergert, Florian Busch, Kevin Klever, Sebastian Holthausen, and Bektur Ryskeldiev. "Mixed Reality Experience - How to Use a Virtual (TV) Studio for Demonstration of Virtual Reality Applications". In: *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2018) - GRAPP*. INSTICC. SciTePress, 2018, pp. 281–287. ISBN: 978-989-758-287-5. DOI: 10.5220/0006637502810287.

[Dew+18] Bastian Dewitz, **Ladwig, Philipp**, Frank Steinicke, and Christian Geiger. "Classification of Beyond-Reality Interaction Techniques in Spatial Human-Computer Interaction". In: *Proceedings of the 2018 ACM Symposium on Spatial User Interaction*. SUI '18. Berlin, Germany: Association for Computing Machinery, 2018, p. 185. ISBN: 9781450357081. DOI: 10.1145/3267782.3274680.

Das folgende Google Scholar Profil wird fortlaufend auf dem neusten Stand gehalten:
https://scholar.google.de/citations?user=2eAVtZ8AAAAJ&hl=de&oi=ao

Datum: 28.08.2025 Unterschrift: Philipp Ladwig

Acknowledgment

Special thanks to Chris Geiger, who believed in me early on and gave me the chance to develop freely and explore what I see as meaningful and of value. Thank you, Chris – without you this dissertation would not have been possible!

I would like to thank Ralf Dörner for the very helpful discussions and support throughout the entire time! You have always put me back on the right track!

I would like to express my heartfelt gratitude to my friends and family who have always supported me. Thanks all to you for your patience and considerateness. Thank you, dad and Thomas, for arousing my interest in nature, science and technology.

Big thanks to the entire MIREVI team! Special thanks to Heike, Mitja, Joel and Jochen who always had an open ear and giving me the peace of mind to dedicate myself fully to my tasks - no matter how stressful it was!

Thanks to Prof. Dr. Herder, who has introduced and pushed me to scientific publishing since my Bachelor's degree.

I would like to thank my Bachelor's and Master's students for their productive collaboration: Ben Fischer, Bernhard Wohlmacher, Christin Willscheid, Damian Zohlen, Manuel Zohlen, David Nowittnik, Hendrik Preu, Juan Schupp, Victor-Julian Bringezu, Kester Evers, Lars Janson, Rainer Schiller, Raoul Gödel, René Ebertowski, Zina Ghannadan. Special thanks to René for his excellence work over the last years!

Last but not least, I would like to thank Alex, with whom I have been able to work intensively for over 6 years in such a trusting and productive environment as I have never done with a friend before. I hope we can continue this with avaluma.ai in the coming years :)

This dissertation was partly sponsored by the German Federal Ministry of Education and Research (BMBF) under the project numbers 16SV8182, 13FH022IX6 and 16SV8756. Project names: HIVE-Lab (Health Immersive Virtual Environment Lab), Interactive body-near production technology 4.0 (german: "Interaktive körpernahe Produktionstechnik 4.0" (iKPT4.0)) and "AniBot - Giving digital assistants a face and a voice" (german: "AniBot - Digitalen Assistenten Gesicht und Stimme geben").

Contents

Part I.

Introduction and Basics

"The biggest challenge to developing telepresence is achieving that sense of 'being there'. Can telepresence be a true substitute for the real thing?"
— Marvin Minsky [Min80]

1. Introduction

Humans are social beings and communication is an essential part of their existence. Languages have evolved, as have the body languages that accompany them, which are incredibly complex and appear in many different variations across the planet. One fascinating fact that is common to all languages is that humans have the ability to detect small discrepancies between verbal and nonverbal language. When a person finds themselves in such a situation as a listener, they often can not put into words exactly what was odd about the speaker or the conversation, but they realize that something was "off" or wrong. This ability to read nonverbal cues is critical to human communication because it allows people to adapt their behavior to the context, to understand and to anticipate the needs and intentions of others.

Throughout history, people have sought ways to communicate over long distances. From smoke signals and carrier pigeons to Morse code, the telephone, and video telephony, the technology has advanced significantly, and with each iteration, more information can be transmitted in less time for a richer communication experience. Although video telephony technology has matured, we still invest a great deal of time and money in physically meeting people in the real world. Nonverbal communication (NVC) such as eye contact, gestures, facial expressions, physical distance, and presence in a shared environment are such important components of human understanding and collaboration that we don't want to miss them, especially in important conversations and negotiations.

Mixed Reality technology has emerged as a promising tool for long-distance communication. It allows people to interact spatially with digital objects and avatars as if they were physically present. This technology has the potential to revolutionize long-distance communication by providing a more immersive and engaging experience than traditional videoconferencing. In addition, telepresence is seen as a key technology that could significantly reduce CO_2 pollution.

However, the technology is not yet ready to bridge the "uncanny valley" in real time. In particular, the reconstruction of the human face under a standard head-mounted display (HMD) is still an unsolved problem. This has been partially solved and realized in laboratory environments, but tracking a person's face and animating it authentically without the occurrence of the uncanny valley effect remains the "holy grail" of computer graphics and computer vision. It is a difficult problem to solve because people are very sensitive to subtle differences or small rendering artifacts in the way digital faces are displayed, and a lot of computing power is required to represent human liveliness in real-time applications. This means that current technology falls short of providing a truly immersive experience that feels like a physical encounter. However, the field of neural rendering has been emerging since around 2016, with results of surprisingly high quality. Meanwhile, deepfakes technology has advanced to the point where it is no longer possible to tell the difference between real and synthesized video. While deepfakes are still primarily computationally complex and therefore not real-time and interactive, and often only work in 2D, it is easy to look to the future and realize that deep learning, and in particular neural rendering, has the potential to deliver NVC over long distances within virtual environments.

However, neural rendering is not yet mainstream and photorealistic digital avatars are still difficult to reproduce in real time. In 2024, MR telepresence applications primarily use cartoonish avatar rendering. Human avatars and their perception have already been extensively researched, as a systematic study of social presence concludes: "...multiple studies show that the vivid perception of another person often leads to greater enjoyment and social influence..." [OBW18a]. Although the Media Richness Theory [DL84] is nearly 40 years old, recent studies continue to confirm it [ILC19]. It suggests that individuals exchange the most information during face-to-face conversations, as opposed to digital or analog alternatives such as video telephony. A greater quantity and quality of shared information usually results in more effective communication (see Chap. 2 for more details on this topic).

In this dissertation we will combine existing and established techniques for human body tracking with new approaches, not only from the field of deep learning, but also from classical numerical as well as analytical approaches. The goal is to develop methods for authentic transmission of NVC for immersive telepresence and to bring it one step closer to realization in order to create more effective ways to exchange ideas over large distances, to maintain social contacts and to collaborate in a human-centered way across countries.

1.1. Relevance and Challenges

The goal of this dissertation is to find evidence for the usefulness of transmitting NVC, and it also specifically addresses the technical problem that Mixed Reality is not able to authentically display either gestures or facial expressions one-to-one on the remote side during a teleconference. Use cases are easy to imagine, but we have a concrete practical case from the company Volkswagen. This use case was described to us by a development engineer and a designer during a physical meeting between the University of Applied Sciences Düsseldorf and Volkswagen in early 2018:

When virtual reality became widely and inexpensively available in 2016 with the HTC Vive and the Oculus Rift, Volkswagen investigated the possibilities and potential of the technology. Of particular interest was the interdisciplinary exchange of digital models between designers and engineers in different physical locations. A classic example is changing the position or orientation of components within a vehicle. This often affects the placement of other components or the final design of the product. A central point of criticism from engineers and especially designers was the lack of facial expressions from colleagues when someone proposed a solution or idea. The avatars at that time (2018) had no animation of facial expressions and body language is also limited with only three tracking points (HMD and 2 controllers). For all involved, the added value of spatial perception and other MR-typical advantages did not seem to be sufficient to sacrifice NVC. For Volkswagen, this was one of the main reasons why VR hardware, and especially distributed collaboration using this technology, did not find its way into practical production.

During the COVID-19 pandemic, the term "Zoom fatigue" was coined. Several scientific publications conclude that unnatural communication via a "flat" screen, lacking the full range of NVC, is an important point for the cause of this phenomenon [NW22; WBK07]. These are only a few of the many examples of why real and authentic NVC is important and why this dissertation is relevant. However, the delivery of NVC remains an issue that is perhaps one of the most central to the success of MR. Several companies believe that a realistic face-to-face encounter will be the "killer app" for MR, as evidenced by

the company Meta, which has invested billions of dollars in this research area. Recently, Apple's Vision Pro was introduced, which also makes significant efforts to render a person's face as authentically as possible without an HMD.

However, conveying authentic NVC in immersive telepresence scenarios poses some complex challenges, which can be reduced to two main technical challenges: On the one hand, the tracking of body movements and on the other hand, the authentic reconstruction (rendering) on the remote side. Tracking and rendering photorealistic and believable faces in real time is still considered the "holy grail" of computer vision and computer graphics. This is the main technical challenge of this thesis.

In order to develop meaningful technical solutions, another challenge of this dissertation is to identify what information should be transmitted in what form in order to achieve effective remote collaboration. It is important to assess perceptual-psychological effects and to understand the salience of certain (technically induced) impressions in order to determine the impact of technical solutions on remote collaboration.

1.2. Research Questions and Objectives

In this dissertation, the research methodology is based on and systematically anchored in Nunamaker and Chen's "Research Framework for Information Systems Research' [NC90]. It consists of a multi-stage process that begins with the "observation" stage. The above mentioned case with Volkswagen was an initial observation and was further consolidated through literature research, so that we can state the fundamental research question of this dissertation:

> **RQ1**: How to technically support the transmission of nonverbal communication in Mixed Reality-based telepresence systems?

This question is broad and touches on many different academic areas. It is therefore broken down into further questions and specific objectives. The approach and the formulation of the objectives are also strongly oriented on the framework of Nunamaker and Chen.

In the further course of this work, the transfer of NVC in the context of remote collaboration is divided into the shared task space and the person space. In the person space, an essential exchange of NVC takes place on the basis of e.g. facial expressions, while in the task space, for example, spatial referencing by pointing with a finger takes place when working together on an object. While some people may not directly associate spatial referencing with NVC, it is a component of remote collaboration that is performed via a gesture. In this context, the following question arises:

> **RQ2**: How does the availability of a shared virtual task space, and in particular a referencing tool, affect task efficiency and error rates in remote collaboration?

- RO2.1 is to identify what study design could answer the question.

- RO2.2 is to implement the test environment.
- RO2.3 is conducting the study and evaluate the results.

RQ2 is discussed and answered in Chap. 3.

Along with the task space, the person space is the central space in which NVC is transmitted in a variety of ways. The face is undoubtedly an important mediator for the transmission of NVC. However, due to technical challenges, it has only been possible to a limited extent to conduct user studies with realistic digital personal faces in MR that actually look like the person belonging to the avatar. In the context of this work, we ask whether it is worth the effort to create a personalized avatar that looks like the user, and whether the complex and time-consuming creation process is worthwhile. We measure copresence and social presence in a user study comparing personalized and generic avatar faces. We formulate the following research question:

> **RQ3**: Does a personalized avatar increase copresence and social presence compared to a non-personalized?

- RO3.1 is to identify what study design could answer the question.
- RO3.2 is to implement the test environment.
- RO3.3 is conducting the study and evaluate the results.

RQ3 is discussed and answered in Chap. 6.

A fundamental technology for NVC transmission is body tracking. It captures gestures and spatial referencing. However, there are many different body tracking systems that are often incompatible with each other. Different skeleton hierarchies, coordinate systems or transmission protocols complicate the development of hardware-independent applications. Some kind of middleware that standardizes the large number of different systems would be advantageous in order to be also able to transmit NVC uniformly. Therefore, another research question is:

> **RQ4**: How can different body tracking systems and protocols be standardized to ensure that the representation of nonverbal communication in a telepresence application looks as identical as possible, even with the use of different tracking systems?

- RO4.1 is to identify and create a generic or standardized protocol.
- RO4.2 is the creation of a middleware that receives raw tracking data, "normalizes" it using the above mentioned standardized protocol and sends it to a telepresence application.
- RO4.3 is evaluating the protocol and the middleware.

RQ4 is discussed and answered in Chap. 4.

There are many full-body tracking systems available for Mixed Reality applications, but few that can track the face under an HMD. However, this is a basic requirement for a telepresence system to transmit NVC. Therefore, another question is formulated as follows:

RQ5: How to track a face beneath an HMD?

- RO5.1 is to identify processes and algorithms that could be extended and perform the task.
- RO5.2 is to implement the approach.
- RO5.3 is to evaluate the tracking performance.

RQ5 is discussed and answered in Chap. 5.

The human face can display an enormous variety of different expressions and thus communicate a wide range of information to others. Until now, it has been a challenge to capture facial features in great detail under an HMD, but it has also been a challenge to display these expressions authentically so that other conference participants can perceive them.

RQ6: How to transfer the face in a photorealistic appearance with authentic movement in real time despite wearing an HMD?

- RO6.1 is to identify processes and algorithms that could be extended and perform the task.
- RO6.2 is to implement the approach.
- RO6.3 is to evaluate the reconstruction quality.

RQ6 is discussed and answered in Chap. 7 and Chap. 8.

1.3. Contribution

In this dissertation, we present:

1. Two methods and prototypes using Generative Adversarial Networks (GAN) (Chap. 7) and Implicit Neural Representations (INR) (Chap. 8) that are specifically tailored to the use case of generating photorealistic avatar faces with corresponding facial expressions in real time.
2. A face-tracking HMD that tracks eyebrow and mouth movements and combines the tracking data with data from an off-the-shelf gaze-tracking system in real time (Chap. 5).
3. An open-source middleware that standardizes multiple full-body tracking systems and provides developers with a unified tracking protocol and interface in order to convey NVC in a consistent way (Chap. 4).

4. A literature review on MR systems that supports collaborative remote work that concludes in six design guidelines (Chap. 2).
5. Qualitative and quantitative results on task efficiency from a study examining the collaborative effects of having a shared task space and a spatial referencing tool versus not having these features (Chap. 3).
6. Results of a perceptual-psychological study supporting that social presence is enhanced when using a personalized avatar face that resembles the user for MR-based telepresence applications (Chap. 6).

1.4. Ethical Considerations

The type of technology presented in this dissertation is perceived by the public as "deepfake" technology. The rise of deepfakes presents significant ethical concerns in our increasingly digital society, which relies heavily on the authenticity of images and videos [Paw22]. While the technology behind deepfakes can create highly realistic and engaging content for movies, games, and virtual interactions, it also has the potential for abuse. Deepfakes can be weaponized to create misleading or false videos, leading to misinformation, defamation, and privacy violations.

In addition, ongoing research in digital media forensics is critical to developing effective detection methods to counter the malicious use of deepfakes [Coz+19; Rös+19]. However, it is extremely difficult, almost impossible, to detect deepfakes with the help of only software for image analysis. It is much more effective to train internet users to debunk real-time deepfakes by asking specific questions, e.g. private questions that only the real person can answer, and scrutinizing sources and information.

As this technology continues to evolve rapidly, proactive measures are essential to balance its benefits with its ethical implications, ensuring that it serves the public good rather than undermining trust and security in our digital environments. There are various processes and protocols, such as the Coalition for Content Provenance and Authenticity (C2PA), that can help identify digital content through metadata that is genuine or synthesized by an AI. Media manipulation is as old as media itself, but it is likely to become increasingly difficult to recognize false information in videos, as videos have so far served as a reliable source or even evidence in court. Users of social media and tele- or videoconferencing should therefore be sensitized to the possibility of identity theft.

1.5. Thesis Structure

This dissertation begins with the first part "Introduction and Basics" as an overview that examines the complexities of human communication, analyzes the current state of remote collaboration systems, and highlights how these systems support these complexities. Design guidelines are derived that provide a foundation and motivation for the topics covered in this thesis.

The rest of the dissertation is divided into two further main parts, "Real-Time Body Tracking" and "Real-Time Face Rendering", each consisting of three chapters. The first chapter of the part "Real-Time Body Tracking" investigates the importance of shared virtual task spaces in telepresence, especially through deictic gestures. A user study

shows that a shared task space significantly improves communication clarity and efficiency, reducing task completion time and interaction errors.

The second chapter addresses the lack of standardization in body tracking systems and proposes a unified protocol implemented in a middleware. This software integrates multiple tracking systems into a standardized format, improving the consistency and quality of nonverbal communication in telepresence environments.

The third chapter presents the development of a face-tracking head-mounted display for AR/VR/MR setups, which provides a low-cost solution for capturing facial expressions. Neural networks and optical sensors are used to create 70 facial landmarks that can be used to render an avatar's face in telepresence.

The third part "Real-Time Face Rendering" starts with a study with 22 participants and shows that personalized avatars could improve social presence and suggests the benefits of investing in such avatars.

The fifth and sixth chapters introduce neural rendering techniques for conveying nonverbal facial communication cues. Through several approaches based on Generative Neural Networks (GAN) and Implicit Neural Representation (INR), advances in visual quality and computational efficiency are demonstrated. Remarkably, GANs and INRs not only surpass the visual quality of previous classic manual modeling approaches, but also require only a fraction of the modeling time to train such an avatar face. This clearly outperforms traditional methods in several aspects.

1.6. Terminology

The following is a brief summary of commonly used abbreviations and definitions of words throughout the thesis:

3DMM – 3D Morphable Model (in this thesis of human heads) are statistical representation of shape variations [BV99].

AR – Augmented Reality [MK94].

BTS – Body Tracking System captures and analyzes the movements of a person's body or face to transfer it into the digital domain [Lad+20a].

CCD – Cyclic Coordinate Descent is an inverse kinematics algorithm that iteratively adjusts each joint in a kinematic chain to minimize the distance to a target point [CD03].

CMC – Computer-mediated Communication refers to the exchange of messages between individuals or groups through all kinds of digital devices.

CNN – Convolutional Neural Networks are a specific architecture, designed for processing structured grid data, such as images. They use convolutional layers to automatically learn spatial hierarchies of features from input data.

FABRIK – Forward And Backward Reaching Inverse Kinematics is an IK solver that alternates forward and backward passes along a kinematic chain to place end-effectors at target positions with smooth convergence [AL11].

FLM – Facial Landmark Map denotes a set of predefined keypoints on a face (e.g., eyes, nose, mouth corners) used for tracking facial expressions.

FPS – Frames per second is a measure of how many individual frames (images) are displayed in one second on an output device.

GAN – Generative Adversarial Networks (GANs) are often implemented, in the field of face rendering, as Convolutional Neural Networks (CNNs) by generating realistic image data. GANs consist of two neural networks: a generator and a discriminator. The generator creates synthetic data, while the discriminator evaluates its authenticity against real data. This adversarial process helps the generator improve its outputs, making GANs effective for e.g. image generation of faces.

HMC – Head-mounted camera

HMD – Head-mounted display

IMU – Inertial Measurement Unit is a sensor that measures acceleration and angular velocity, often combining accelerometers, gyroscopes, and sometimes magnetometers to track orientation and motion.

INR – Implicit Neural Representations (INRs), also called coordinate-based neural networks, encode continuous signals within the parameters of a neural network rather than on discrete grids or meshes. By learning a mapping from spatial (or spatiotemporal) coordinates to signal values INRs produce continuous, high-resolution reconstructions.

LSTM – Long Short-Term Memory networks are RNN variants with memory cells and gating structures (input, forget, output gates) that enable learning long-range dependencies in sequences [HS97b].

MLP – Multilayer perceptron is an artificial neural network, consisting of fully connected neurons with nonlinear activation functions, organized in at least three layers.

MR – Mixed Reality is used as a collective term for all digital stages of Milgram's continuum [MK94].

NeRF – Neural Radiance Fields is an INR for synthesizing novel views of 3D scenes. It encodes a scene within a neural network by mapping spatial coordinates and viewing directions to color and density values. By optimizing this mapping from a set of input images, NeRF can generate high-quality, photorealistic renderings from arbitrary viewpoints [Mil+21].

NVC – Nonverbal communication

RGB-D – Red-Green-Blue-Depth refers to image data that includes both color information (RGB) and depth information (D).

RNN – Recurrent Neural Networks are neural network architectures designed to process sequential data by maintaining hidden states that capture temporal dependencies.

SOTA – State of the Art

Telepresence – is, in the context of this dissertation, the use of VR technology for remote collaboration to create a sense of being physically present in a location other than one's actual location, allowing users to interact and collaborate with others as if they were in the same physical space.

Uncanny valley – is a phenomenon in which humanoid objects that appear almost, but not exactly, like real humans evoke feelings of eeriness and discomfort [MMK12].

VR – Virtual Reality

2. Human-to-Human Communication, Telepresence and Collaboration Revisited

Human-to-human communication is multifaceted, and it is difficult to convey the full range of human expression through a digital channel. The COVID19 crisis has shown that video telephony can be an alternative to physical meetings, but cannot fully replace it. For decades, there has been intense research into how to make remote collaboration productive. Researchers have certainly been inspired by various sci-fi works. The holograms from Star Wars in 1977 or the HoloDeck from Star Trek in 1974 are certainly among the first appearances in this field and have captured the imagination of many. Today, telepresence has many names such as ePresence, MediaSpace, or Metaverse, and is actually pursuing the development of an "ultimate display" for telepresence as envisioned by Ivan Sutherland as early as 1966 [Sut66].

This chapter first introduces the basics of human communication and how mediating technology affects it. This is followed by summaries of related work on remote digital collaboration and immersive CMC (computer-mediated communication) systems over a period of more than 4 decades. One basis for the transmission of NVC is the digital recording of NVC through tracking technologies. In later chapters of this thesis, a distinction is made between body tracking (without face) and face tracking, as the technical approaches used are often fundamentally different.

2.1. Nonverbal Communication in the Physical World

The study of NVC can be roughly divided into three areas, some of which overlap: Proxemics, Kinesics, and Facial Expressions. The field of NVC research is vast, but this chapter will be limited to the most important aspects in such a way that it will be sufficient to provide some grounding in the terminology and concepts discussed throughout this thesis. For a more detailed summary of the basics of NVC, we refer the reader to Chapter 3 of the book by Tanenbaum et al. [TEN14].

The first time in history that the study of NVC was mentioned in academia was with Charles Darwin's book "The Expression of the Emotions in Man and Animals" published in 1872 [Dar72]. In the middle of the 20th century, Ray Birdwhistell begins to research NVC in an academic context almost 100 years later. His book on NVC "Introduction to Kinesics" [Bir52] is based on the idea that human gestures, facial expressions, and body movements communicate as much as spoken language. Birdwhistell introduced the notion that these nonverbal forms of communication are culturally specific and can be learned and understood in a manner similar to language. His work significantly influenced the understanding of nonverbal cues in communication, leading to applications in fields as diverse as psychology, anthropology, sociology, and even law enforcement and conflict resolution. However, Birdwhistell's findings would later become controversial. Analyses of Birdwhistell's work concluded that a "lack of systematic order", "inconsistent repetition of views

and their often unsubstantiated presentation" were core deficiencies. Despite its shortcomings, Birdwhistell's groundbreaking work laid the foundation for **Kinesics.** Kinesics is the study of body language, including facial expressions, hand and arm gestures, and posture. Other researchers have built on this work, including Rudolf Laban. He worked on methods of studying and understanding human movement and its significance for conveying messages, with an emphasis on dance and the performing arts [Mal87]. Albert Mehrabian and his team [MW67; MF67; Meh72] conducted other important research in the field of NVC. Their studies have significantly advanced our understanding of how people communicate emotions and attitudes through nonverbal cues. Their research highlights how subtle nonverbal signals can be and how crucial they are in shaping our perceptions of others, providing valuable insights into human behavior and social interaction.

Proxemics is the study of how individuals mediate interactions and physical distance with others. Edward Hall [Hal+68] coined the term proxemics and identified four types of spaces that people maintain, at least in the Western world: intimate (up to about half a meter), personal (from about half a meter to 1.2 meters), social (1.2 meters to about 3.5 meters), and public (more than 3.5 meters). These spaces are influenced by many factors such as age, gender, culture, and the level of intimacy in relationships [Hal+68].

Another important contribution to the study of proxemics is the work of Argyle and Dean [AD65] called **Equilibrium Theory**. They discovered that in social interactions, people seek a balance between different NVC channels, such as eye contact and physical proximity, to achieve a comfortable level of intimacy. This theory suggests that individuals adjust their physical proximity to others depending on the closeness of their relationship, moving closer or further away to match the desired level of intimacy. Equilibrium theory has implications for understanding social interactions in a variety of contexts, including the workplace, personal relationships, and cross-cultural communication. It provides a framework for understanding how people manage/maintain comfort and effectiveness in their interactions by adjusting their NVC cues.

Over several decades, Paul Ekman, in part with his colleague Wallace V. Friesen, developed a very detailed framework for analyzing NVC, focusing on the use, origin, and coding of nonverbal behaviors [EF69]. They have examined how these behaviors are regularly used in specific contexts, their intentions, and the types of information they convey, including idiosyncratic and communicative aspects. The origins of such behaviors are described as 1.) innate neurological responses, 2.) culturally independent practices, or 3.) personally acquired through factors such as culture and education. The coding system identifies how nonverbal acts relate to their meanings, ranging from arbitrary (no visual resemblance to meaning) to iconic (visual resemblance to meaning) to intrinsic (direct execution of meaning). In addition, they categorized nonverbal acts into high-level groups such as emblems, illustrators, and regulators, each of which serves different functions in communication. This comprehensive system helps to understand the complexity and variety of nonverbal communication across situations and cultures.

Ekman and Friesen also studied **facial expressions** in detail. They created a standard called the Facial Action Coding System (FACS) [EF78], which is still used today in computer animation and psychology. The system is based on earlier work by the anatomist Carl-Herman Hjortsjö [Car69]. It is still a key framework for identifying and categorizing the wide range of human facial expressions and their associated emotions. In the field of computer animation, FACS is used and expressions are classified into so-called Action Units (AUs), such as those used by the 3D Morphable Model (3DMM) ICT FaceKit [Li+20].

2.2. Computer-mediated Nonverbal Communication and Collaboration

Research has shown how important and subtle NVC can be. Although telepresence technology has advanced tremendously in recent years, we are still not able to digitize, transmit, and reconstruct the full range of information from a face-to-face conversation at a remote location. However, the transmission of cues for NVC is already commonplace in today's computer-mediated communication. Smileys are certainly one of the oldest forms of NVC. While smileys usually consist only of Unicode characters, and more complex sequences of characters require some creativity on the part of the sender and receiver, they have largely been replaced by emoticons, which are small but easily recognizable pictograms of facial expressions or other messages. Although the amount of data transmitted is technically small, the actual message sent by a smiley or emoji can vary greatly.

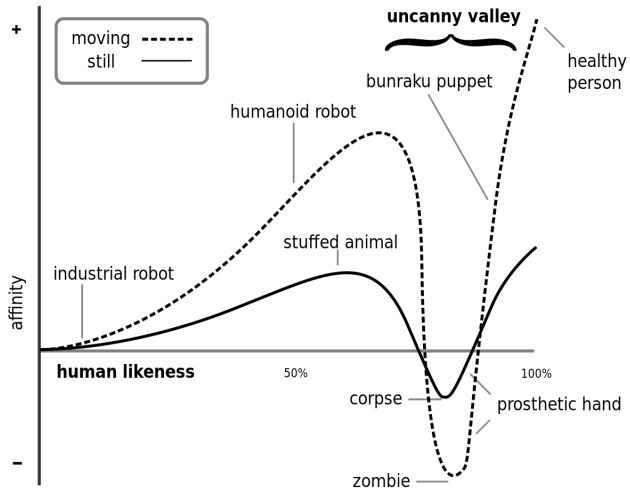


Figure 2.1.: *The uncanny valley effect describes the eerie feeling people experience when they encounter entities that almost, but not quite, look like real people or animals. Image by Smurrayinchester from Wikipedia. License CC BY-SA 3.0 DEED - Image is unchanged.*

Our ancestors have always tried to communicate. Smoke signals, carrier pigeons, and Morse code were severely limited, but this has changed with technology advancements such as better sensors, higher network bandwidth, and increasing storage and computing capacity. Some researchers have explored the possibilities and limitations of computer-mediated communication and developed theories and frameworks. Although some of these theories are relatively old, they are still relevant today. Basically, however, many researchers agree: We humans are evolutionarily "optimized" for face-to-face communication, which has been our primary form of information exchange for presumably millions of years. Being able to read, understand and, if necessary, trust other people was essential for our survival in the past. Humans are therefore extremely sensitive to minimal signs and cues from other humans, which could, for example, indicate a (contagious) disease. This is one of the possible theories behind the so-called **uncanny valley** effect proposed by Mori et al. [MMK12].

The uncanny valley effect is a phenomenon in which humanoid objects (such as robots or animated characters) that closely resemble humans evoke feelings of eeriness and discom-

fort in some observers. The term is used to describe the dip in a graph shown in Fig. 2.1. Although this effect has been studied for decades, it is still not fully understood [Pal+18]. Basically, it can be said that this effect only occurs in humans and animals, as MacDorman and Chattopadhyay were able to show in an extensive study with 548 participants, the effect does not occur in objects [MC16]. In the field of **Computer-mediated communication (CMC)**, this effect is a serious problem. Many mainstream VR platforms, such as Altspace, Meta Horizons, or VRChat, deliberately use catroon-like representations of avatars because today's off-the-shelf technology cannot yet overcome the uncanny valley. However, recent advances in neural rendering show that the uncanny valley can be bridged using deep learning methods, as discussed in detail later in the Chap. 7 and 8.

The terms **telepresence** and **remote collaboration** have some ambiguous definitions in the literature, but there are some basic similarities between all definitions. While "telepresence" is primarily found in Google search results for high-end videoconferencing systems in which conference participants meet virtually in life-size on high-resolution, stationary screens, the word telepresence refers also to the sense of being present in a remote space or environment within an academic context. Often telepresence is also implemented by a robot at the remote counterpart to physically interact with objects. In the context of this thesis, telepresence is understood as a hybrid of both definitions. However, instead of using robots, persons or spaces and objects are "teleported". Participants meet in life-size virtual spaces using wearable (non-stationary) high-definition displays (HMDs). In addition, immersive technology creates the feeling of being in another place. The concept of remote (often called distributed) collaboration has a broader range of definitions than telepresence. CMC has existed for decades and is an established research topic due to globalization. In the context of remote collaboration, several theories have been developed over the last decades that overlap with the topic of telepresence.

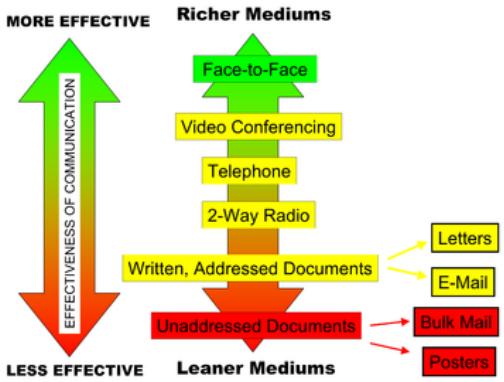


Figure 2.2.: *According to Media Richness Theory [DL84], face-to-face conversations are the medium in which the most information can be exchanged between participants. The richer the amount of information, the more effective the communication. Image by Tntdj from Wikipedia. License CC BY 3.0 DEED - Image is unchanged.*

There are several studies in the area of telepresence, **computer-supported cooperative work (CSCW)**, and CMC. One of the oldest theories regarding the transmission of computer-mediated NVC is the **Media Richness Theory** according to [DL84]. This theory describes a one-dimensional continuum in which text is one of the media with the least information and real face-to-face communication is one of the media with the most information, as can be seen in Fig. 2.2. Research on Media Richness Theory has yielded

mixed results over 40 years of its existence. Some studies challenging it, others supporting its predictions and even a recent study confirms it [ILC19].

Based on the Media Richness Theory [DL84], a number of new theories have emerged. In the context of face-to-face telepresence, the **Media Naturalness Theory** or **Psychobiological Model Theory** according to [Koc04] is worth mentioning. This argues that human non-lexical methods of communication, such as facial expressions, gestures, and body language, have evolved over millions of years and as such must be important to the naturalness of communication between humans. Media Naturalness Theory hypothesizes that because face-to-face communication is the most "natural" method of communication, we want our other (technical) methods of communication to be as close as possible to face-to-face communication.

The above studies may explain why we still like to travel for important meetings. After the COVID19 pandemic, digital meetings have become an accepted way of holding short and informal meetings in developed countries. However, when it comes to important meetings or meetings over a longer period of time, the physical meeting is often preferred. Unlike a face-to-face meeting, video telephony is associated with some problems such as the reduction of NVC due to several facts such as limited camera resolution, field of view, poor lighting conditions, and the inherent reduction from a 3D space to a 2D image. All of these problems negatively affect the detection and transmission of signals related to kinesics, proxemics, and facial expressions.

Nass and Reeves developed the concept of the **media equation** [NR96] based on the idea that people interact with media as if it were real people. This concept was later further developed in a study by Blascovich et al [Bla+02]. This study tested Argyle and Dean's (1965) equilibrium theory of an inverse relationship between mutual gaze, a nonverbal cue signaling intimacy, and interpersonal distance. They successfully demonstrated that participants displayed the same intimacy cues in VR as in reality. It is worth noting that this study was conducted in 2002 and the visual quality of the avatars or agents was very rudimentary. However, they were able to demonstrate this effect even with avatars/agents that were clearly not human to observers due to the low visual quality. This work was ahead of its time, and the researchers already stated the following sentence: "It seems inevitable that as we use these virtual environments more and more, interactions between avatars will become routine" [Bla+02].

The Computer-Supported Cooperative Work (CSCW) Matrix [Joh89] categorizes technology-enabled collaborative activities along two dimensions: time and place. It distinguishes between synchronous (same time) and asynchronous (different time) collaboration, and between co-located and remote interactions. In this way, the matrix represents a spectrum of collaborative scenarios, from direct, face-to-face meetings, such as a video call, to remote, asynchronous work using online platforms, such as a shared Google document in the cloud. This dissertation focuses on the top left quarter, where "same time, same place" face-to-face interactions occur.

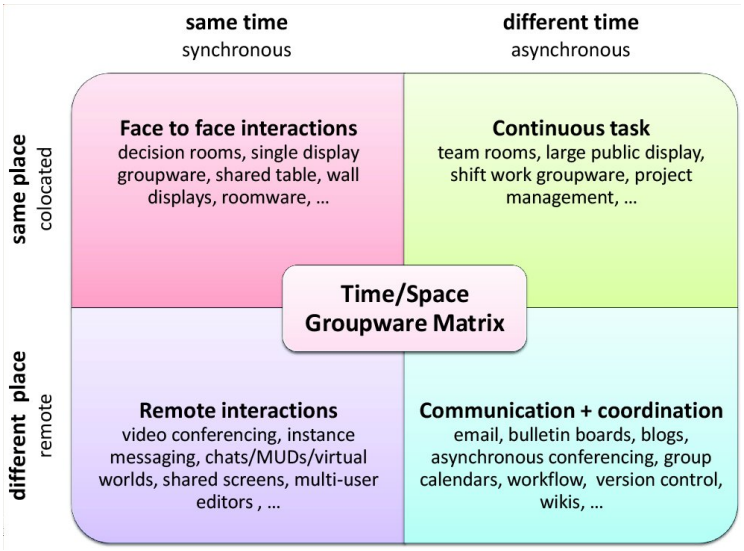


Figure 2.3.: *The Computer-Supported Cooperative Work (CSCW) Matrix, invented by Robert Johansen [Joh89], is a framework used to categorize and analyze the various dimensions of collaborative activities. Image by Momo54 from Wikipedia. Licenseis public domain. Image not changed.*

Not directly related to NVC, but important in explaining the following frameworks and theories, is the description of the continuous transition between reality and virtuality and its intermediate stages. Milgram and Kishinio [MK94] introduced the concept of the *Reality-Virtuality Continuum*, as illustrated in Fig. 2.4. This continuum is divided into four distinct stages: Reality, which is the direct perception of the actual environment without technological intervention; AR, which involves overlaying virtual elements and additional information on the real world; *Microsoft HoloLens* is a typical example of an AR device; *Augmented Virtuality* (AV), in which real objects are integrated into virtual environments, exemplified by the display of a real person in a virtual scene; and *Virtual Reality* (VR), which completely replaces the real environment with computer-generated images, with devices such as *HTC Vive* or *Oculus Rift* representing this technology. This dissertation focuses on *Mixed Reality*, defined by Milgram and Kishinio.

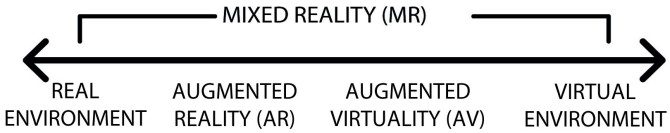


Figure 2.4.: *The Reality-Virtuality Continuum by Milgram and Kishinio [MK94] describes a continuous transition between the real and virtual world. Adapted from Milgram and Kishinio [MK94], modified and redrawn by the author.*

Hiroshii Ishii [IKG93], Bill Buxton [Bux92; Bux09], and Billinghurst and Kato [BK02a] start using the terms **person space** and **(shared) task space** around the same time. They each use slightly different terms, but they mean the same thing. They describe the person space as an area where eye contact, gestures, and conversation take place. The task space, on the other hand, is a subset of the communication spaces where people work together on objects, such as making a sketch on paper or spatially pointing at an architectural model, as shown in Fig. 2.5a. Video telephony, on the other hand, creates a

task space that is usually isolated from the communication space (Fig. 2.5b). Even when participants meet locally in one space and collaborate on digital content, the two spaces are usually separated as well, as shown in Fig. 2.5c. The main challenge is to support awareness of social cues. A major improvement when using AR/VR/MR technology is that the system can be built so that the spaces are not isolated, as shown in Fig. 2.5d.

Person and task space are important concepts and help to understand why remote collaboration sometimes leads to good and sometimes to bad results. In general, it can be said that task space and person space are separated in many remote collaboration applications today. A good example is screen sharing in a videoconference. Typically, the shared screen is maximized and the faces of the participants are significantly reduced. Few video conferencing software solutions offer the option to maximize the video feeds of the people next to the screen sharing on a second monitor in 2023. This is also a problem for the person conducting the screen sharing, as they may not be able to properly recognize the reactions of each person. However, it is especially valuable for the presenter to be able to see how the other participants are reacting to certain parts of the presentation. This is where VR/AR/MR technology has tremendous potential to connect people and task space. However, the face is obscured by the HMD, which ultimately leads to a blockage of the person space.

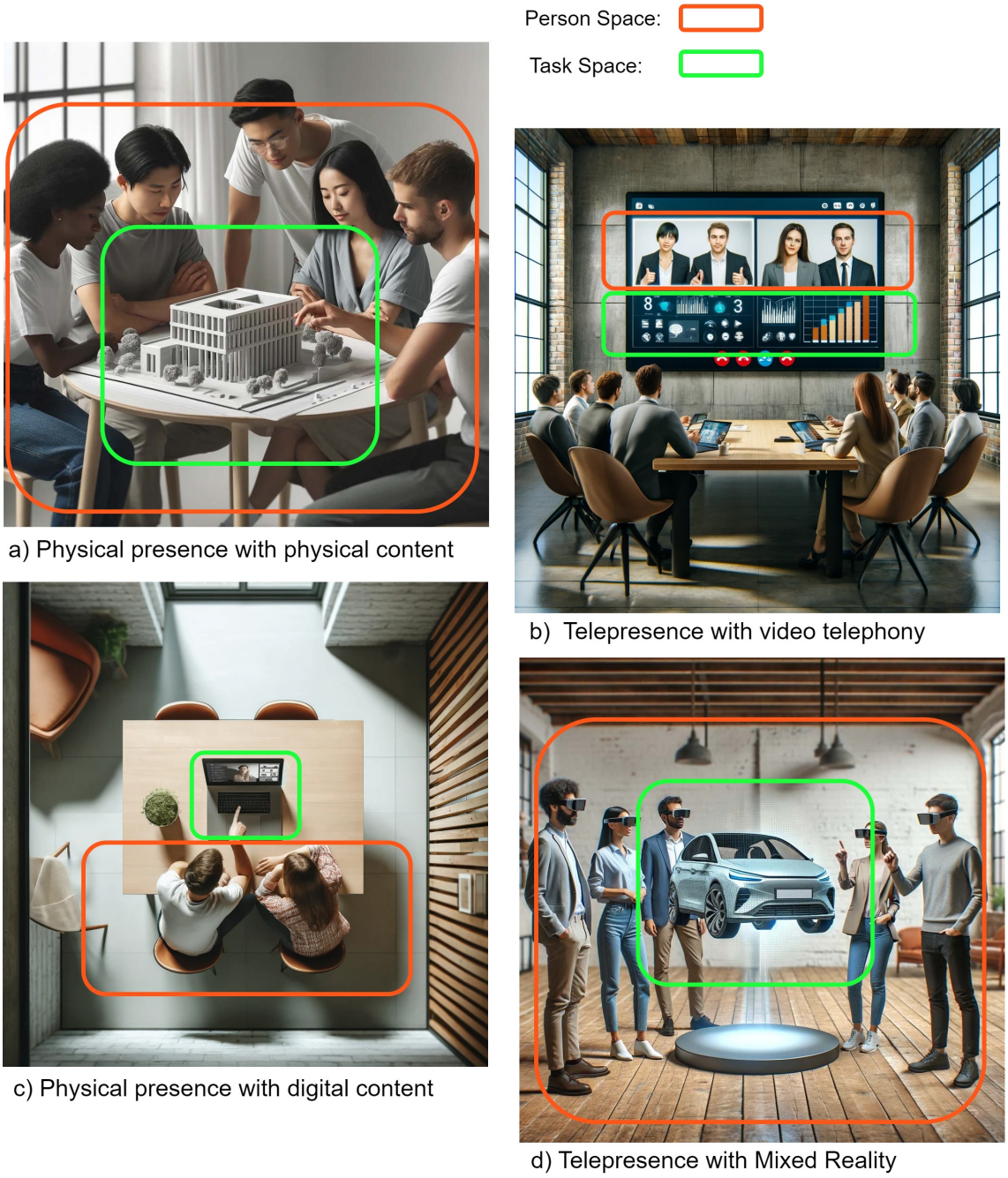


Figure 2.5.: *AR/VR/MR technology can merge person and task space in the virtual world, such as shown in d: The image in a) shows the difference between communication space and task space in the case of physical presence and b) in the case of telepresence in the form of video telephony. Person and task space are separated. c) Separation of these spaces can also occur in local collaboration on digital content. Main challenge: Supporting awareness of social cues. d) Local or remote teamwork in AR/VR/MR can bring the spaces together again. Images generated by Dall-E from OpenAI.*

2.3. Remote Collaboration Systems

Recent advances in electronics research and development are moving us ever closer to the ultimate device, making it increasingly difficult to distinguish between virtual environments and reality. The concept of *Star Trek's Holodeck* represents the pinnacle of such displays. This system would provide realistic, comprehensive experiences that engage all human senses, including touch, sound, and even smell and taste. If it were possible to transmit such sensory data over a network to replicate these experiences remotely, it could revolutionize collaborative efforts by making it seem as if individuals are physically present where assistance is needed.

Currently, however, the technology has not reached the sophistication of the Holodeck as depicted in Star Trek. Olson and Olson [OO00; OO14] have noted that our technological capabilities are still evolving and that "distance matters" in the context of remote collaboration. Nevertheless, many organizations operate in multiple locations, which means that specialists in different technical areas are often spread across regions or even globally. The core strength of any business lies in the expertise of its people, and knowledge sharing between employees and customers is critical to success. While remote collaboration tools such as *Skype, DropBox, or Evernote* are available, they often only support basic forms of communication such as text, images, or video. As the complexity of machines, assembly tasks, and 3D CAD data increases, the need to share and interact with 3D data in real time remains a challenge [OO00; OO14]. This is where Mixed Reality (MR) could significantly alleviate many of the current barriers to remote collaboration.

Over the past thirty years, considerable research has been conducted on the application of Mixed Reality (MR) for collaborative purposes, such as facilitating assembly tasks over the Internet [BB99; Tan+03; Oda+15; GLC15; BCL15], enabling geographically dispersed experts to conduct car design reviews [Nvi; Kli+02] or remotely investigating crime scenes [Poe+12; Dat+14]. These are only a few examples of the usefulness of MR in collaboration. In particular, the fields of remote engineering and virtual instrumentation can greatly benefit from MR for remote guidance. This technology can be critical when specialized, expensive equipment requires maintenance by expert personnel who are not readily available on-site. In addition, remote training facilitated by MR could preemptively address potential emergencies and facilitate wider dissemination of specialized knowledge.

In user studies, a common scenario involves a remote user assisting a local user in performing a task. While different authors may use different terminology to refer to the participants in a remote session, in this context we will use the abbreviations ***RU* for the remote user** and ***LU* for the local user**.

2.3.1. Technological Constraints in Pre-2012 Mixed Reality Research

In each of the studies mentioned in this chapter, a fundamental element is the two-way exchange of speech. Speech is the core of communication in any application. However, descriptions of spatial locations and actions can be imprecise or unclear when conveyed through speech alone. The effectiveness of collaborative tasks is greatly improved when verbal communication is augmented with physical gestures, as Heiser et al. [HTS04] found. Early collaborative systems using Mixed Reality (MR) were primarily video-mediated applications, as described by Ishii et al. [Ish90; IM91; IKG93]. In these systems, a video camera placed above the participant's workspace captured their activities and transmitted

them to other participants via a monitor. A similar system was developed by Kirk and Fraser [KF05], who conducted a study in which participants engaged in a Lego assembly task. They found that using AR not only accelerated collaboration, but also made it easier for participants to remember the assembly steps 24 hours later compared to simply following verbal instructions.

Baird and Barfield [BB99] along with Tang et al.[Tan+03] have demonstrated that AR reduces cognitive load during assembly tasks. Billinghurst and Kato[BK99] reviewed research on collaborative MR in the late 1990s and concluded that while the results are promising, they only begin to explore the potential applications of MR. It remains to be seen where MR can be effectively applied. In addition, Billinghurst and Kato built their own AR tracking system, the ARToolkit [KB99], and emphasized that the traditional *WIMP-Interface* (*Windows-Icons-Menu-Pointer*) is inappropriate for MR environments and needs to be redesigned for such platforms. Schmalstieg et al. [Sch+02] created the Studierstube. This was a kind of middleware that could combine different hardware and software interfaces and emphasized collaborative work. It was found that complicated three-dimensional relationships of mathematical or scientific tasks, for example, could be easily represented.

Klinker et al. [Kli+02] developed the *Fata Morgana* system for collaborative car design reviews, providing the ability to focus on details and compare different designs.

Monahan, McArdle, and Bertolotto [MMB08] highlight the educational benefits of *Gamification*, noting that "computer games have always been successful at capturing people's imaginations, the most popular of which use an immersive 3D environment in which players take on the role of a character." [MMB08]. Similarly, Li, Yue, and Jauregui [LYJ09] presented a VR e-learning system, noting that virtual "e-learning environments can maintain students' interest and keep them engaged and motivated in their learning." [LYJ09]

Gurevich, Lanir, and Cohen [GLC15] created *TeleAdvisor*, a wheeled, remote-controlled robot equipped with a camera and projector on an adjustable arm. The remote user (RU) can view the camera feed and manipulate the robot and its arm using a desktop PC, projecting visual aids onto surfaces. The stability of the robot-mounted camera provides a more comfortable viewing experience for the RU compared to a head-mounted camera, which can be shaky and cause discomfort. In addition, this system allows the RU to control the movement and reduces the cognitive load on the LU, who no longer has to manage the Point of View (PoV) for the RU.

In summary, until 2012, information transfer was limited by insufficient sensors, displays, bandwidth, and processing power. Many systems that relied primarily on video transmission failed to convey the sense of "being there", limiting mutual problem understanding and spatial awareness. Many systems have suffered from the same technical limitations.

2.3.2. New Technology Introduces Sustainable Changes

A chronological review of the literature reveals significant technical advances in systems since around the year 2012. This progress is largely attributed to technological improvements that have increased the availability of sensor data and processing for MR collaboration. Real-time, cost-effective acquisition and triangulation of 3D environmental point clouds became feasible around 2012. This is primarily due to the introduction of the PrimeSense RGB-D sensor and the first Kinect from Microsoft. This led to improved spatial understanding of the environment and more reliable tracking of VR/AR/MR de-

vices [New+11]. In addition, developments in display technology facilitated the creation of affordable head-mounted displays (HMDs). Tecchia, Alem, and Huang [TAH12] pioneered a system that captures the workspace and the arms and hands of the RU and LU with a 3D camera. This system integrates a triangulated, textured virtual scene accessible through a head-tracking HMD, outperforming traditional 2D gesture systems. Sodhi et al. [Sod+13] used the Microsoft Kinect along with a short-range depth sensor to reconstruct a desktop-scale workstation in 3D and transmit a hand avatar to a RU, enabling more complex gesture execution and improving communication and understanding between participants.

In addition, the system developed by Sodhi et al. [Sod+13] can recognize real surfaces, allowing hand avatars to interact realistically with physical objects such as tables. Knowing the location of real surfaces in the virtual environment allows virtual objects to be snapped to those surfaces, reducing the time required to place objects such as furniture or assembly parts.

A textured 3D representation of the environment also allows the RU to freely navigate the environment. Tait and Billinghurst [TB15] developed a system that includes a textured 3D scan of a workstation, controllable by keyboard and mouse on a monoscopic monitor, and supports the selection of spatial annotations. They found that allowing full view independence, as opposed to static views, accelerated collaborative task completion and reduced communication time. Lanir et al. [Lan+13] found similar results, noting the asymmetry in remote assistance tasks. They emphasized that the helper (RU), who usually has more knowledge, benefits from controlling the point of view (PoV) rather than having symmetrical ownership with the worker (LU), who has the necessary physical tools and a better view of the environment.

Oda et al. [Oda+15] use "virtual replicas" in assembly operations, defining them as virtual versions of actual, tracked assembly components. These replicas exist physically for the LU and are represented as 3D models in Virtual Reality for the RU. The position of the virtual model is continuously updated to match the real world environment. Assembly components often have complex shapes, making it difficult for the LU to follow the RU's instructions for proper orientation and positioning. To assist, the RU can overlay virtual replicas in AR for the LU, thereby reducing the cognitive load associated with the task. Oda et al. [Oda+15] found that demonstrating the physical alignment of the virtual replica with another machine component is faster than using spatial annotations on the virtual replicas for visual guidance, which facilitates positioning by the LU. In addition, Oda et al. [Oda+15] implement physical constraints such as object snapping to speed up tasks, similar to the methods used by Sodhi et al. [Sod+13].

Poelman et al. [Poe+12] developed a system capable of generating a real-time 3D map of the environment, specifically designed to address the challenges of remote collaborative crime scene investigation. Datcu et al. [Dat+14] used Poelman et al.'s system and demonstrated that MR enhances the RU's situational awareness, defined by Endsley [End95] as the ability to perceive, understand, and anticipate the future state of a situation.

Pejsa et al. [Pej+16a] developed an AR-based life-size telepresence projection system that uses the Microsoft Kinect v2 to capture remote scenes and reproduce them using a projector at the other end for the RU. This system allows for better perception of nonverbal communication cues, such as facial expressions, which are often obscured in systems where participants wear head-mounted displays (HMDs) that cover parts of the face.

Mueller et al. [MRR17] observed that in remote collaborative tasks, such as locating specific virtual items in a virtual environment, task completion times were improved by using

simple "shared virtual landmarks". These landmarks, such as virtual furniture, help interpret deictic expressions such as "under the ceiling lamp" or "behind the floating cube".

Another permanent change in remote collaboration and NVC research is the introduction of affordable but powerful HMDs such as Oculus Rift, HTC Vive, or even standalone devices such as Microsoft Hololens around 2016. Piumsomboon et al. [Piu+17a; Piu+17b] developed a hybrid AR and VR system. This system uses a Microsoft HoloLens to scan and texture a real room, and then replicates this environment for a remote user who accesses it through an HTC Vive. It tracks and displays the hands, fingers, head gaze, eye gaze, and *Field-of-View* (FoV) of both users. Piumsomboon et al. show that incorporating eye gaze and FoV as cues in collaborative tasks can reduce physical effort (measured by distance traveled) and subjectively simplify tasks. They also introduced variable scaling of virtual space, where shrinking the virtual environment improves orientation and planning through a miniature model, similar to the approach of Stoakley, Conway, and Pausch [SCP95].

To date, there have been many attempts to solve the problems of 2D video telephony with new telepresence systems. However, creating the feeling of being really present at the remote location, which was formulated by Minsky in 1980, remains a major problem. Fuchs et al. [Fuc+94] were ahead of their time and built a collaborative multi-user system that processed data from multiple RGB sensors in the room to create a three-dimensional display of people and their surroundings at a remote location. However, due to the low resolution of the sensors and the lack of computing power for depth estimation, this was only a vision at the time and far from feasible in real time.

In order to provide a comprehensive sense of presence, a trend in research can be recognized: More and more information about the environment and the participants is being captured and transmitted with increasing quality. While sensors are achieving higher resolutions with improved noise behavior, and network bandwidth, memory, and processing power are constantly increasing, the first works have been created that can record and display the environment and the people in it in real time at interactive frame rates in three dimensions. Until recently, our technology was not advanced enough to digitize entire human avatars and environments in real time and reconstruct them at a remote location. But with today's technology, this is becoming more and more possible, as demonstrated by many different research groups from Maimone and Fuchs [MF11; Mai+13], Strotko et al. [Sto+19b; Sto+19a], Kowalski et al. [KND15] or Kainz et al. [Kai+12], Kreskowski et al. [KBF22], Rendle et al. [RKF23] or more recent systems such as the work of Meta's Reality Labs [Bag+21]. Seminal works include Beck et al. [Bec+13] as well as Orts et al. [Ort+16]. All these solutions have in common that the environment is scanned in three dimensions with an interactive repetition rate, thus transmitting a kind of 4D video (3D + temporal dimension). However, even today, in 2024, this technology can only be realized under laboratory conditions and requires a great deal of technical effort and know-how. It is interesting to note that many of the papers presented focus on a technical system, a user study or on a perceptual psychology experiment. A combination is rare. Exceptions prove the rule [Lat+17a; Wal+18a]. In general, however, it can be observed that new technical systems tend to be studied less in the area of effectiveness or quality of (remote) collaboration. Typically, such studies are only conducted once the technology has become mainstream and off-the-shelf hardware is available.

Yu et al. [Yu+21b] conducted a comparison between an avatar created using point cloud volumetric reconstruction and a pre-created and rigged virtual human avatar of the user in a telepresence context. Despite the lower visual quality of the point cloud avatar, including depth noise and some missing features, it outperformed the virtual human avatar in

the areas of copresence, behavioral realism, and humanity. These results were confirmed in a parallel study by Sasikumar et al. [Sas+21], who also reported superior copresence with volumetrically reconstructed avatars. Gamelin et al. [Gam+20] conducted a similar study with point-cloud and pre-created, rigged, mesh-based avatars, but instead of measuring perceptual psychological parameters, the focus was on quantifying measures such as completion time and number of measured errors in the context of a collaborative task. The result also showed that the point-cloud-based avatar is significantly superior to the mesh-based avatar.

While the advent of off-the-shelf RGB-D sensors and HMDs marked the lasting changes mentioned in this section, deep learning is the third change that has or will significantly alter remote collaboration [Tew+20; Tew+22]. The work of Thies, Zollhöfer, and Nießner and their teams demonstrates the potential of deep learning primarily from a technical perspective. While the earlier work on face reanactment did not use deep learning in the classical sense, the visual results and quality of these researchers were already outstanding at the time [Thi+15; Thi+18b; Thi+18a]. Later, the researchers also used Generative Adversarial Networks (GAN) to improve their results even further [TZN19; Elg+20; Thi+20; ZBT22b; Gra+22; ZBT22a; Qia+23]. However, the researchers primarily presented the technical aspects of their work and did not analyze the systems in terms of their impact on remote collaboration.

The field of deep learning can be divided into further iterative technical stages. Variational Auto Encoders (VAEs) and Generative Adversarial Networks (GAN) have been increasingly used since around 2016, demonstrating the potential of neural representation. The key advantage of deep learning is that the systems implement an end-to-end pipeline, which means they achieve very good results without significant manual effort such as 3D modeling. A technical advancement with many advantages over GANs and VAEs are Neural Radiance Fields (NeRFs) [Mil+21; Mül+22] and point (or Gaussian-) based differentiable rendering approaches [Ker+23; Zhe+23]. While it is a challenge to render NeRFs in real time on low-capacity devices such as mobile phones, rendering faces or even whole bodies using geometric primitives such as "Gaussian Splatting" [Ker+23] is much more resource-efficient because point-based rendering can take advantage of hardware-accelerated routines much more efficiently. The visual quality of both approaches is similar and, at least to the untrained eye, it is generally impossible to tell which method was used.

The work of Meta's Reality Labs (formerly Facebook) is worth mentioning. The results of this research division are unique in many ways, but often focus on evaluating technical implementations and shed little light on the implications for remote collaboration [Lom+18; Wei+19; Raj+21; Lom+21; Cao+22]. Furthermore, the reproducibility of the results is often limited due to the use of specialized and expensive laboratory hardware and the fact that the source code is rarely published.

In summary, as technology has evolved to accurately scan and model environments in real time, significant improvements have been made in collaborative tasks, increasing the efficiency of remote interactions. The 3D reconstruction of both body parts and the environment enables: 1.) improved spatial awareness of the remote location (free PoV); 2.) improved communication through the transmission of nonverbal cues such as gaze and gestures; and 3.) the integration of real surfaces with virtual objects for more realistic interactions (virtual collision, snapping). In addition, this reconstruction leads to 4.) more reliable tracking of various devices (phones, tablets, HMDs, virtual replicas) and 5.) the development of new display technologies that enhance immersive experiences, thereby improving spatial understanding and problem awareness for all participants. One possible

reason for the broad progress in many areas of technology could be the intensive research efforts in the smartphone sector for performance enhancement and miniaturization.

2.3.3. Research Agenda, Technology Trends and Outlook

Despite significant progress and research in recent years, the ultimate collaborative display – akin to Star Trek's holodeck – remains a distant goal. Research on how to effectively work together in multi-team, or group-to-group, collaboration has been minimal and therefore represents uncharted academic territory. Previous studies have focused primarily on two-person collaboration, leaving the dynamics of data exchange and interaction between multiple groups largely unexamined. Lukosch et al. [Luk+15b] initiated research in this area, but acknowledged the need for further research. Beck et al. [Bec+13] developed a group-to-group telepresence system shortly after the appearance of the first off-the-shelf RGB-D sensors and continued to research the system [KBF22; RKF23; Sch+24]. Piirainen, Kolfschoten, and Lukosch [PKL12] identified achieving consensus on problem definitions and specifications as a challenge in collaborative remote teamwork. Both situational and team awareness cues are critical and should be addressed in further research.

Another critical issue is maintaining user focus on specific events and elements within an environment, also called "mutual awareness". Ongoing research on awareness cues is essential. Müller, Rädle, and Reiterer [MRR17] recognized the need for techniques that highlight events, collaborators, or objects outside the immediate field of view. Pejsa et al. [Pej+16b] and Masai et al. [Mas+16a] emphasized the importance of nonverbal communication cues such as facial expressions, posture, and proxemics, which are essential for empathy but remain difficult to convey with current technology and avatar technology. Some studies investigating interactive 3D scanning and transmission of interactive 4D video also conclude that the resolution of the sensors is not sufficient to transmit facial expressions with adequate resolution and quality. [Bec+13; Ort+16; MF11]

Comfort in HMDs, while critical to long-term use, is rarely studied. Consider a scenario where a worker is performing a long, complex assembly task remotely, using an HMD that becomes increasingly uncomfortable over time, pointing and pinching in the air, leading to fatigue (e.g., the "gorilla arm"), which can ultimately lead to errors. Piirainen et al. [PKL12] emphasize the importance of not overlooking user needs and human factors, noting that system usability is critical. Recent research in MR has focused primarily on technical aspects and productivity comparisons between MR and non-MR applications, often overlooking comfort and usability. Masai et al. [MFJ16] also emphasized as a design guideline that comfort is essential, because if the system is used in a daily work routine, it could be used for 8 hours a day. However, studies such as Lubos et al. [Lub+16], have begun to address and evaluate comfort in MR applications.

In addition, the challenge of replicating virtual haptic sensations in MR remains unresolved. Researchers such as Oda et al. [Oda+15] are exploring alternatives such as virtual constraints including collisions and snaps. Lukosch et al. [Luk+15a] and Billinghurst [BK99] also highlighted the need for further investigation into what tasks MR can effectively facilitate.

Advances in tracking technologies, network speeds, sensor capabilities, and processing power will continue to push the boundaries of achieving and potentially surpassing a holodeck-like experience. Emerging technologies such as deep learning for object detection, segmentation, and recognition will open new avenues of research [TGG20]. Future MR

devices will not only detect environmental surfaces, but also objects such as machine parts, tools, and people.

2.3.4. Design Guidelines

Many of the studies mentioned above complain about similar problems. The problems have changed since around 2012 due to a wave of affordable and efficient new hardware, however it is striking that many researchers describe similar shortcomings for their specific use cases. When these problems, after 2012, are abstracted and the recommendations are generalized, they can be summarized as design recommendations for the development of AR/VR/MR applications:

1. **Maximize information about the remote environment:** Providing video is essential for situational and spatial awareness, but a 3D mesh of the environment is preferable [Piu+17a; Piu+17b; Sod+13; Poe+12; Dat+14; Oda+15]. An updated 3D mesh should be available in real time, and the high resolution of the mesh seems less important than the resolution of the texture on it [TAH12; MF11; Mai+13; Sto+19b; Sto+19a; Bec+13; Ort+16].
2. **Maximize information about the avatars:** The literature is consistent in showing that when more expressive avatar communication capabilities are implemented for NVC, there is invariably an improvement in various desirable parameters of collaboration. Not only do metrics related to perceptual psychology, such as social presence or copresence, increase, but the effectiveness of collaboration, as measured by the reduction of errors or the time it takes to complete a task, also improves [LG19b; Lad+19; Sod+13; Wu+21; SN18; Lat+17b; Rot+18]. In particular, this includes the detailed and authentic reproduction of facial expressions [Bai+06; Tar+23]. It should be noted that facial anomalies are more disruptive than significant body motion errors, emphasizing the need to reconstruct facial expressions with solid tracking methods and high-quality face rendering [Hod+10].
3. **Provide an independent viewpoint for exploring the remote scene:** This enhances both spatial and situational awareness and helps to understand problems [TAH12; Piu+17a; Piu+17b; Lan+13; TB15; Bec+13]. As for the head-mounted camera (HMC) worn by the local worker, this also avoids nausea for the remote expert due to shaking or constantly moving images from the HMC.
4. **Provide as many awareness cues as possible:** Transmission of speech is essential. It is also beneficial to include posture information from the collaborators, such as head position, head gaze, eye gaze, and field of view (FoV) [Piu+17a; Piu+17b]. While a virtual ray may be sufficient for indicating direction by hand, a static hand model [Sod+13] or a fully tracked hand model provides better communication of natural gestures [Piu+17a; Piu+17b]. It's useful to indicate events outside the user's FoV [BR03; Piu+17a] and to provide common local landmarks [MRR17]. To avoid clutter, these cues should be toggleable [MRR17].
5. **Rather use point cloud avatars than pre-modeled rigged mesh-based avatars:** Given a choice between point-cloud avatars (even of low visual quality) and pre-generated or modeled rigged mesh-based avatars, you should prefer point-cloud avatars [Gam+20; Yu+21b; Sas+21].
6. **Prioritize usability and convenience:** The following points are not only rele-

vant for remote collaboration applications, but also for any MR solution, so they should not be omitted: If the application is intended for prolonged use, make sure the interface is comfortable for the user and consider human factors in the application design [Lub+16; LHG17; PKL12; GLC15]. Avoid the "gorilla arm", caused by pinching gestures with half-stretched arms in front of the user's face for more than a few minutes. It is also important to ensure a consistently high refresh rate for the user throughout the runtime of the application, as otherwise nausea and low comfort levels can occur. This is often a challenge because many standalone MR devices have limited processing power.

2.4. Conclusion

Human-to-human communication is complex and multifaceted. Nonverbal communication includes whole-body gestures, posture and, most importantly, facial expressions. Over millions of years, our brains have evolved to read and interpret the smallest movements and discrepancies in the face of the person we are talking to. This has made tracking and digitally reconstructing a human face extremely difficult.

In general, most of the remote-collaborative systems and prototypes presented try to reproduce the real environment as best as possible. This is true for the 3D environment (rooms, machines, tools) as well as for the people as avatars involved with gestures and facial expressions. Basically, the main goal in science is to perform a kind of "teleportation" of the remote site as well and as authentically as possible, mediated by technology. As a result, researchers today face technical hurdles that are limited by hardware performance (e.g., computing capacity, network bandwidth, or sensor resolution) and by challenges in authentic reconstruction and animation of avatars, such as complex manual modeling and animation of believable faces.

In the last 15 years, there have been two permanent changes in the technology that are or will be essential for remote collaboration in the future: 1.) The advent of off-the-shelf hardware, such as RGB-D sensors or inexpensive but powerful HMDs, and 2.) advances in deep learning will make telepresence much more realistic. It is clear that advances in deep learning can lead to much better tracking, faster transmission and more authentic rendering of NVC than with traditional methods.

Part II.

Real-time Body Tracking

3. Impact of Shared Virtual Task Spaces on Efficiency and Error Reduction in Remote Collaboration

As we continue to explore immersive telepresence and remote collaboration, particularly in terms of delivering NVC, the importance of a shared virtual task space emerges as a key element. The ability to communicate consistently, intuitively, and with minimal cognitive load in remote collaboration scenarios is important. Deictic gestures, as a form of NVC, play a crucial role by establishing a direct connection between participants. In this chapter, we conduct a user study to illustrate how such a space not only improves the clarity of communication, but also increases the efficiency of remote collaboration systems. We quantify the importance of the availability of a shared task space (by providing a digital referencing/pointing method) in terms of time, errors, and type of verbal communication during a remote collaboration task. This chapter demonstrates and highlights the significance of integrating NVC with gestures for effective remote collaboration and motivates the development of further technical prototypes and tools in the later course of this thesis.

Many machines are complicated and require repair by experts with specialized training. In the era of Industry 4.0, sensors, pattern recognition, and artificial intelligence facilitate the prediction of optimal maintenance times, but they cannot completely prevent breakdowns. A machine breakdown can result in significant financial damage to an organization, requiring additional time and specialized personnel. While minor issues can be resolved through email, phone, or video calls, more complex issues are often difficult to resolve remotely.

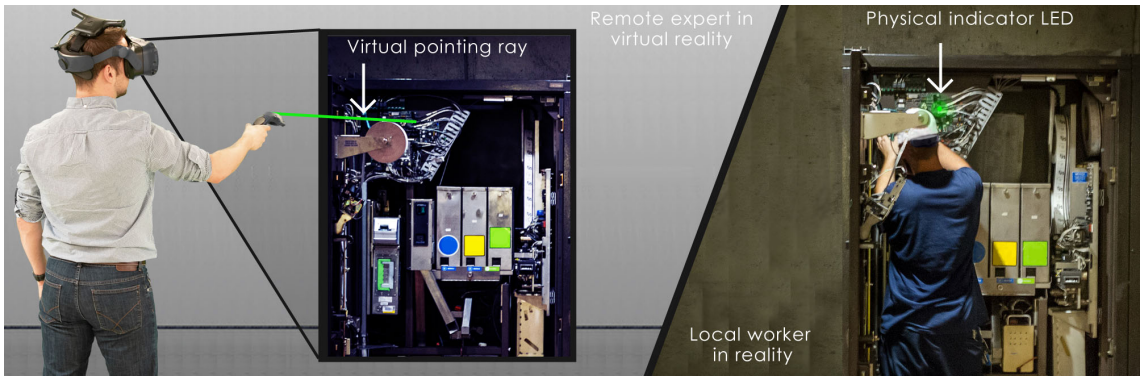


Figure 3.1.: *Collaboration between a remote expert in virtual reality (left) and a local worker in reality (right). The remote expert uses a virtual laser ray to indicate certain parts of the machine, which are then highlighted by LEDs on the local worker's side. Parts of the image by Eric Ourcell [CC BY-NC 2.0] via Flickr [Our19]. Fig. from [Lad+19].*

There is considerable interest in using AR/VR/MR for maintenance support and remote assistance tasks [HF09; Lan+13; GLC15; Tea24]. Many studies support the multiple ben-

efits of AR/VR/MR, but these technologies are often not mature enough. The balance between benefits, costs and convenience is often unsatisfactory. Our approach aims to simplify the overlay of computer graphics onto the real world by simply activating inexpensive LEDs (light emitting diodes) on a physical object, controlled by a low-cost microcontroller. The remote expert interacts with a digital 3D model of the machine, selecting specific components to be highlighted by flashing LEDs visible to the local worker, as shown in Fig. 3.1. Our system, which uses bidirectional verbal communication, enhances the connection and significantly speeds up the process.

We intentionally simplified our system to focus on certain effects, recognizing that this choice has limitations for real-world use. Our experiment did not include video transmission in order to clearly isolate the effects of LED signaling, and did not include body or hand tracking for the local worker. This means that the remote expert cannot correct the local worker's actions if he or she mishandles the machine. Only real-time changes to the machine are monitored and communicated. This setup is not advisable in real-world scenarios due to potential risks to both machine and worker safety. Improvements are needed to ensure safety.

To the author's knowledge, this is the first user study of referencing from a VR application to reality using physical signaling with LEDs for machine maintenance. We discuss the advantages and limitations of this method and the experiment conducted, in which we collected quantitative and qualitative data on usability and performance. This chapter contains the following contributions:

- Demonstrating the positive impact of the availability of a shared virtual task space in remote collaboration environments.
- The presentation, evaluation and discussion of a new and inexpensive approach with LEDs of interaction between remote collaborators.

3.1. Related Work

To understand the following study and its implications, it is important to delineate the smooth transition between AR and actual reality in our setting, noting that the use of LED displays actually occupies a space between these two realms. Our methodology departs from the superimposition of computer-generated images on the real scene. Instead, our system uses stationary LEDs on a machine to generate visual signals in the real world that ultimately have exactly the same goal: to convey spatial information (or to replace a collaborator's referencing action as a form of NVC). Following the definitions of Milgram et al. [MK94] and Azuma [Azu97], our system does not make use of an HMD or other tracking technology on the local worker side, positioning the LED display method closer to actual reality than AR, in terms of Milgram's continuum shown in Fig. 2.4 in the last chapter. Therefore, our system is a VR-to-reality framework. The academic field of AR and VR collaboration has been thoroughly explored, but the niche of collaboration facilitated by LED technology, especially in remote environments, has received limited attention, but we argue that the findings are likely to apply to various combinations of AR/VR/MR-to-AR/VR/MR systems. For this reason, the full range of collaborative systems in Milgram's continuum is mentioned in the following section. Some of the following work has already been mentioned above in Chap. 2, but the following will go into more detail on key work specifically in the context of referencing tasks, LEDs, and proposed cross-continuum systems that utilize AR, VR, MR, and reality simultaneously.

3.1.1. Shared Task Space

Collaboration usually involves two spaces. Hiroshii Ishii [IKG93], Bill Buxton [Bux92; Bux09] as well as Billinghurst and Kato [BK02a] introduced the concepts of "person space" and "task space" in the context of direct face-to-face communication and the emergence of telepresence systems. Person space encompasses the exchange of verbal and nonverbal signals, such as speech and gestures, between participants. Conversely, task space is defined as the environment in which collaborative physical tasks occur, such as manipulating an architectural model or operating machinery. This space is also where objects are pointed to. As Buxton described, remote collaboration often divides the task space into two disjoint spaces - resulting in a separate task space for each of the collaborators. This division is exemplified by videoconferencing, where collaborating and referencing/pointing to a physical object is difficult because there is no shared 3D reference space. Pointing with a finger can lead to misunderstandings due to perspective errors, depending on the position of the webcams.

One of the most relevant works for the following study was conducted by Heiser et al. [HTS04]. They examined the effects of having and not having a co-located task space through a study in which dyads were tasked with finding an optimal evacuation route on a campus map. The study compared two conditions. The first scenario includes a natural, co-located face-to-face communication with a physical shared task space. The second condition includes an environment where participants are separated by a curtain, limiting them to verbal communication only, with no shared task space. The results highlighted the significant impact of referencing/pointing on collaboration efficiency, results (in the form of sketches), and phrases used, with the co-located dyads achieving better results in less time. However, the direct impact of the lack of face-to-face communication remains unclear. The study reported in this chapter is similar to Heiser et al.'s, but the task in our study is different. Instead of drawing sketches of a rescue path on a map, our task involves the correct operation of elements of a machine under the guidance of a remote expert. Similar to Heiser et al. we also measure time, but we can better quantify the results obtained by counting errors or misunderstandings between the dyads perceived by the test conductor or logged by our system. We also count the number of questions during the conversation and the types of verbal comments to analyze the change in the participants' statements.

3.1.2. Remote Collaboration and Machine Maintenance in VR/AR/MR

The integration of VR and AR has been shown to mitigate the challenges associated with referencing objects in remote collaboration by facilitating a shared task space [BK02b]. Numerous studies have highlighted the benefits of AR and VR in improving collaboration efficiency, reducing errors, and fostering shared understanding or "common ground" [OO14; HF09]. In addition, AR and VR solutions are increasingly finding their way into remote industrial collaboration, such as Teamviewer's Assit AR application [Tea24].

Our methodology for the study embraces the concept of a unified task space that enables the visual connection between virtual and physical objects through LED signaling. This approach not only facilitates the transfer of machine information from the physical to the virtual realm, such as valve status, temperature readings, and component wear, but also provides immediate visual feedback on the local worker's progress.

Lanir et al. [Lan+13] and Tait et al. [TB15] demonstrated that giving the remote expert

autonomous control over the scenario view improves remote collaboration. This is particularly relevant in maintenance scenarios, where the local worker's camera feed, which is often unstable and requires explicit, time-consuming verbal instructions to adjust the camera, can hinder task execution. Our system addresses this issue by presenting the remote expert with a 3D-rendered digital twin of the machine in an HMD (HTC Vive Pro), allowing for unrestricted observation.

The TeleAdvisor system of Gurevich et al. [GLC15] has conceptual similarities to our project. It features a remotely controlled robot equipped with a camera and projector, allowing the remote expert to adjust the robot's position and project visual instructions. Unlike this projection-based method, our approach does not rely on a robot, but shares the principle of remote visual guidance. A key difference is the ability of our system to collect and use the machine's internal data.

Sangregorio et al. [San+15] introduced a system to support remote maintenance by gathering and transmitting machine data to the remote expert, using smartphones or laptops for display, without incorporating AR or VR technologies.

Bottecchia et al. [BCJ10] developed a remote maintenance system using custom AR glasses for web-based maintenance of a helicopter turboshaft engine. Similarly, Oda et al. [Oda+15] used "virtual replicas" to bridge VR and reality, tracking physical machine components and rendering them in VR to match their real-world positions. This technique, similar to our approach, facilitates direct visual communication between the local worker and the remote expert by accurately representing the position of rotary encoders and switches.

3.2. User Study: Deictic Gestures in Shared Virtual Tasked Space

3.2.1. Hypothesis

Our study explores remote guidance for machine maintenance through a system that combines VR and physical LEDs to enhance communication between a local worker and a remote expert. The expert is able to turn on and off specific LEDs within the VR environment, which is technically synchronized with a physical counterpart (digital twin, shown in Fig. 3.2) of the machine with the same arrangement of LEDs, enabling precise nonverbal guidance from a remote side. This method uses colocated LEDs on the machine as indicators and represents the referencing method (the deictic gestures) and replaces the real and physical referencing of the expert to areas of the machine, e.g. with his index finger. It is hypothesized that the integration of nonverbal communication methods such as deictic gestures within a shared virtual task space during remote collaboration in an immersive telepresence scenario will significantly increase task efficiency in terms of time and reduce errors by providing unambiguous visual cues.

Figure 3.2.: *Left: View of the digital twin for the remote expert in VR. Right: Physical "machine" seen by the local worker. The expert sees the target states and positions of the elements and can turn on and off LEDs in VR that are synchronized with the real counterpart for the local worker. Fig. from [Lad+19].*

3.2.2. System Overview

The system consists of two main components: a real-world workstation containing the physical machine operated by the local worker, and a virtual environment for the remote expert containing a digital counterpart of the machine, as shown in Fig. 3.2. The preference for VR over a traditional 2D desktop interface stems from the fact that VR significantly improves the ease and speed of wayfinding, navigation, and spatial comprehension (especially in cluttered, complex 3D environments), as evidenced by the findings of Ware et al. [WAB93] and Pausch et al. [PSP93]. Although the task space we are currently using is rather small (430 mm by 330 mm), the benefits and impact of VR are expected to be more pronounced in larger and more complex systems. VR offers the possibility to increase situational awareness by incorporating spatial cues into larger virtual environments [BR03; Gru+17].

In designing the system for our experiment, we were inspired by the concept of creating an escape game, drawing on the established effectiveness of gamification and competitive application design with a scoreboard as a motivational tool, which is supported by numerous studies, including those cited in the literature review by Hamari et al. [HKS14]. Given the critical role of time in our experiment, we hypothesize that incorporating the element of time pressure into an escape game scenario will lead to more consistent results by encouraging all participants to complete the tasks as quickly as possible. Choosing a different scenario could result in participants spending time exploring the environment for its own sake, rather than focusing on completing the specific tasks set in the experiment. Among other factors, engaging people in a user study becomes easier when it involves playing a game, as it tends to increase their willingness to participate and generally fosters a positive attitude towards the experiment.

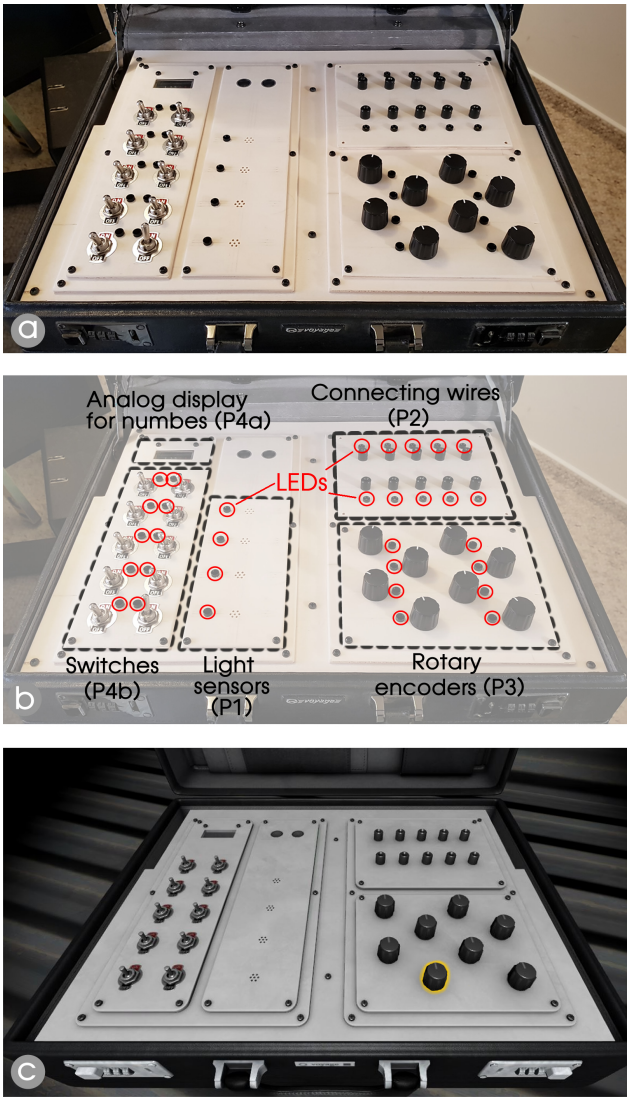


Figure 3.3.: *a) 32 control elements in a suitcase represent the machine b) 32 LEDs (one next to each element) are used for indication (red circles) c) The panel is shown to the remote expert in VR who selects a rotary encoder (yellow glow), which then activates the corresponding LED in the physical suitcase. Fig. from [Lad+19].*

3.2.3. Local Worker and Remote Expert Side

To simulate a tangible machine, a control panel with 32 interactive elements such as switches and encoders was integrated into a suitcase, as shown in Fig. 3.3a. This 430 mm by 330 mm panel not only serves as a practical representation of a machine, but also enriches the narrative of an escape game inspired by "Keep Talking and Nobody Explodes" [Ste24]. Next to each interactive element is an LED, shown in Fig. 3.3b, to provide guidance, along with a numeric display screen with a backligh for improved visibility. The control system is powered by six Arduino Nano microcontrollers, each connected via an I2C bus, allowing real-time Bluetooth communication of each component's status (e.g., the rotational state of a rotary encoder) to a Unity 3D instance of the workstation. Fig. 3.4 shows the back side of a module.

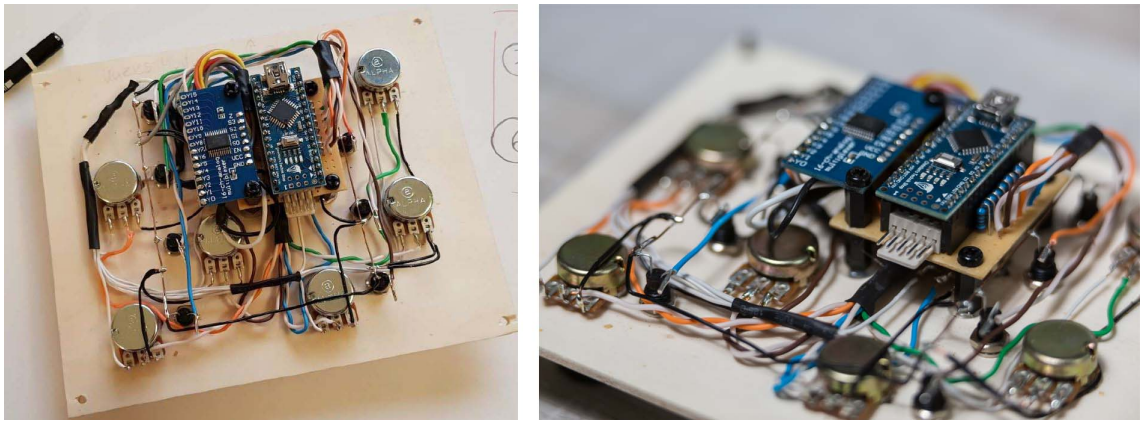


Figure 3.4.: *The back side of Panel P3 (Fig. 3.3). The states of the elements and the LEDs of each panel are processed and controlled by Arduino Nano microcontrollers. Fig. from [Lad+19] and images by Hendrik Preu.*

On the counterpart side, the Unity 3D application renders a virtual replica of the suitcase for the remote expert, as shown in Fig. 3.3c, allowing the expert to monitor the state of the current suitcase's components in real time, such as the state of switches and encoders. The virtual interaction is enabled by a pointing ray controlled by the HTC Vive controller, which allows the activation or deactivation of selected physical LEDs. While the remote expert cannot directly modify the states of the physical components, he/she has the ability to read these states. However, control of the LEDs is entirely in the expert's hands. To improve usability and interaction accuracy, the virtual representation is scaled up by a factor of three, so that the dimensions of the control panel in VR are 1320 mm by 990 mm, allowing for more precise deictic referencing by the user.

3.2.4. Methodology

Prior to the primary study, a pilot study was conducted with three dyads to refine the process and ensure that participants received only the essential information. In the following, we refer to this pilot study at certain points, as it had a significant influence on the final study structure and justifies some of the decisions we made.

3.2.4.1. Participants

The primary study involved 18 dyads, a total of 36 individuals (15 females and 21 males, ranging in age from 22 to 67 years, with a mean age of 35.9 years). This group included students and staff from the local computer science department (17 participants) and employees from two companies specializing in computer-generated special effects, digital content creation, and VR/AR/MR/mobile app development (19 participants). We screened for color blindness and severe visual impairment, as these could affect performance in a referencing task, but we did not screen for stereopsis, as its impact was minimal in our primarily two-dimensional task environment. Participants' proficiency in VR and AR was measured using a 6-point Likert scale to gauge at least a general tendency, summarized in Fig. 3.7. Each dyad spent an average of 35 minutes on the experiment, including time for post-experiment questionnaires and debriefing.

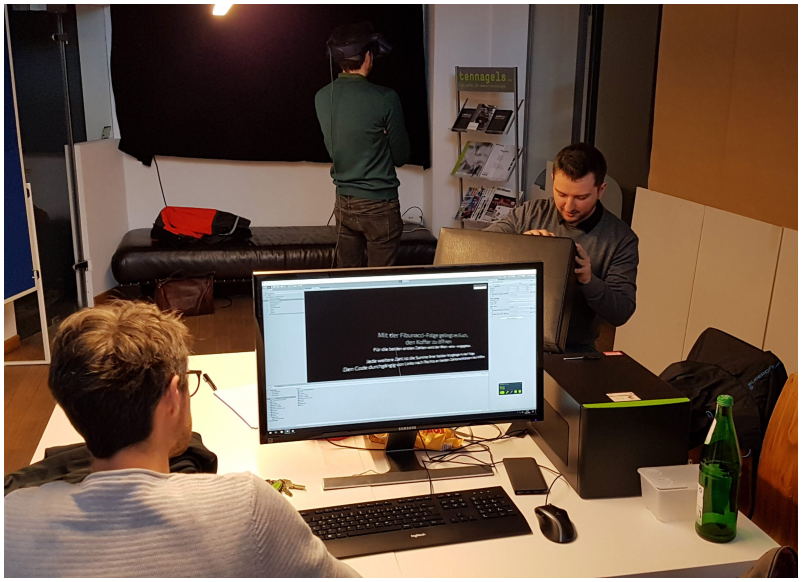


Figure 3.5.: *Experiment setup: The experimenter observes the test in front of the monitor. The local worker works with the suitcase while seated. The remote expert wears an HTC Vive Pro and sees the digital twin of the suitcase. Fig. from [Lad+19].*

3.2.4.2. Material

In Fig. 3.5, the remote expert participant used an HTC Vive Pro head-mounted display and Vive controllers to perform referencing tasks, while the local worker participant sat facing away from the remote expert, next to the suitcase, as shown in Fig. 3.5. Although typical immersive telepresence scenarios would use a headset or microphone and speaker on the HMD, in this study both roles were in the same room, back to back, to simplify coordination and mitigate issues with digital communication channels. This was a result of the pilot study, which initially began in separate rooms. The decision was made to balance the potential for digital communication errors against the benefits of having a single experimenter oversee the experiment in person. This co-location allowed for verbal communication without visual contact, emulating a remote setup as proposed and conducted by Heiser et al. [HTS04]. The system, powered by an appropriate PC configuration (Intel Core i7 4770K CPU, 16GB RAM, Nvidia GTX 1070), ensured that the Unity 3D application (version 2018.3.2f1) ran with optimal performance with a sampling rate of 11.1 ms (90 frames per second - maintained by the main rendering thread of the application) for timing the tasks.

3.2.4.3. Method

The experimenter led each dyad through a pre-defined onboarding process to ensure consistency in the dissemination of information so that each participant started with the same knowledge and conditions. First, participants were informed about the synergy between VR and the physical world, its potential benefits, and that the study aimed to monitor the duration, errors, and types of questions and answers that occurred throughout the experiment. Errors were identified as misunderstandings that led to incorrect use of the equipment. Within the Unity software, a random number generator was used to determine who would assume the roles of remote expert and local worker, as well as the

experimental condition they would experience - either "with LED indication" or "without LED indication". The experiment was structured as a between-groups design, ensuring that each dyad, as well as individual participants, could only participate once to avoid biased results. Participants were asked not to share any details or information with others who had not yet participated in the experiment but who might be potential participants, so as not to influence the other participants.

The experimenter made adjustments to the HTC Vive Pro to account for interpupillary distances and to ensure clarity of vision. The experimenter ensured that the referencing and selection methods were understood. Initially, the remote expert was presented with instructions on a virtual information text panel positioned 1.5 m above the floor. This approach ensured that the expert had a clear view and could identify the necessary details and responsibilities as the experiment progressed. It was critical for the expert to accurately point out specific elements to avoid compromising the data integrity of the study. Upon completion of a task, the remote expert's panel was updated with new information for the next task. The tasks were:

Task A The first task served as an introduction to the experimental setup without any referencing task to get familiar with the setup. It consisted of unlocking a suitcase with a 6-digit combination lock by solving a simple numerical puzzle.

Task B The next task (Panel P1 in Fig. 3.3b) marked the first LED indication task and involved accurately positioning a small flashlight on one of four light sensors. The instructions on the floating panel in front of the expert in VR tell him which of the four positions is the correct one. Now he has to select the correct LED with the pick ray attached to the controller and press the trigger button on the controller. The expert receives visual feedback in his VR application that the LED lights up for the local worker. Correct placement by the local worker triggered positive feedback for the remote expert, and the next task was automatically initiated. Any incorrect actions, such as shining light on the wrong sensor, were visually communicated to the expert in the application and automatically logged by our software and noted by the experimenter as an error.

Task C In the second referencing task, the remote expert instructed the local worker to correctly connect three colored cables (Panel P2). While the correct connections were visible to the remote expert in their application, the cables were initially placed in a bag at the top of the case, easily visible and accessible to the local worker. A total of six referencing instructions were required, and incorrect connections were logged as errors.

Task D The third referencing task required the remote expert to indicate which of five rotary encoders should be rotated to specified angles (Panel P3). To do this, the remote expert had to highlight the appropriate LEDs next to each encoder and verbally communicate the desired rotation as shown in the VR application. The actual rotation of the encoders was then transferred to the VR application, allowing the remote expert to verbally suggest adjustments such as "turn more", "180 degrees", or "twelve o'clock". An error was logged if an incorrect encoder was adjusted, as three should remain stationary.

Task E The fourth referencing task required the communication of four numbers displayed in sequence on a physical display inside the case (Panel P4a in Fig. 3.3b). This sequence was the key to completing the flip switch panel task (Panel P4b). Only the remote expert could see the arrangement of the switches, which were randomized

for each dyad. The local worker had to operate the switches based on the sequence provided by the expert, with visual feedback in the VR application indicating which switch was operated. An error was logged if an incorrect switch was selected.

Task F The final task did not involve referencing and required locating an object in the suitcase. Completion of Task F represents the end of the experiment.

A total of 17 LEDs were used for referencing tasks. The "without LED indication" test condition did not use LEDs and relied solely on verbal instructions and explanations. All errors and timings for each task were documented by our software in a text file. Throughout the experiment, the frequency of deictic and explanatory expressions used was noted by the experimenter. At the end of the experiment, participants were given a post-experiment questionnaire with demographic questions as well as questions about their level of VR experience and other information about the experience of the study (questions shown in Fig. 3.7). Before the next dyad entered the room, the experimenter reset the setup. This was done by returning the case to a previously defined state. This includes, for example, the rotational state of the encoders, the position of the switches, and the cables and flashlight, so that each dyad starts with the same conditions.

3.3. Findings

Tab. 3.1 presents the results for the "with LED indication" and "without LED indication" scenarios, including metrics such as time, number of errors, queries, and both deictic and explanatory phrases. A single dataset for the "with LED indication" scenario was excluded because one participant admitted prior knowledge about the solutions to some tasks because he had previously spoken to a previous participant in the study.

An independent samples t-test with a significance threshold of $p = 0.05$ was used to evaluate the completion times for all referencing tasks (B, C, D, E). In addition, the data were subjected to a Levene's test for homogeneity of variance, which revealed no significant differences. The Shapiro-Wilk test also confirmed that the normality assumption was not violated. It was found that there was a significant difference in completion times between the "with LED indication" condition ($M = 344\,s$, $SD = 90.3$) and the "without LED indication" condition ($M = 493\,s$, $SD = 149$). The results $t(15) = 2.44$ with $p = 0.028$ indicate that the LED indication significantly affects the completion time.

Additionally, a Mann-Whitney U test was performed to analyze the variance in error counts, queries, and both deictic and explanatory expressions. This analysis revealed a significant discrepancy in the number of errors between the groups, but no significant differences were found in the number of queries or the two types of expressions. Further details are shown in Tab. 3.2 and Fig. 3.6. The feedback from the post-experiment questionnaires is shown in Fig. 3.7.

In summary, the results of the user study underscore the significant benefits of a shared virtual task space in remote guidance scenarios. In particular, the inclusion of visual aids via LEDs in a virtual space significantly improved task completion times and reduced error rates.

	time overall	time for referencing tasks	errors	questions	deictic expr.	explan. expr.
with LED indication						
	451 s	348 s	2	7	6	21
	534 s	246 s	1	7	4	30
	522 s	362 s	0	6	7	21
	722 s	349 s	0	1	6	13
	586 s	386 s	2	16	14	48
	828 s	473 s	0	13	2	24
	720 s	405 s	0	9	1	47
	257 s	186 s	0	10	4	31
M'	560 s	356 s	0	8	5	27
∅	578 s	344 s	0.6	8.6	5.5	29.4
without LED indication						
	919 s	657 s	4	16	5	37
	604 s	315 s	6	7	2	36
	980 s	691 s	15	9	6	47
	601 s	371 s	6	5	3	30
	575 s	336 s	4	23	6	27
	1226 s	638 s	4	30	10	48
	871 s	580 s	3	11	1	45
	757 s	455 s	6	4	1	18
	969 s	392 s	1	7	1	50
M'	871 s	455 s	4	9	3	37
∅	834 s	493 s	5.4	12.4	3.9	39.0

Table 3.1.: *Summary of the data collected during the experiment. ∅ represents the mean. M' represents the median. Each row represents the results of one dyad. Table from [Lad+19].*

	errors	questions	deictic expr.	explan. expr.
U-value	2.50	35.0	33.5	18.0
p-value	.002	.96	.85	.21
Mean "With LED"	4.81	8.89	8.72	7.31
Mean "Without LED"	12.7	9.12	9.31	10.5
Significant?	yes	no	no	no

Table 3.2.: *Results of Mann-Whitney U test with critical value of 15 (significance level of .05)). Table from [Lad+19].*

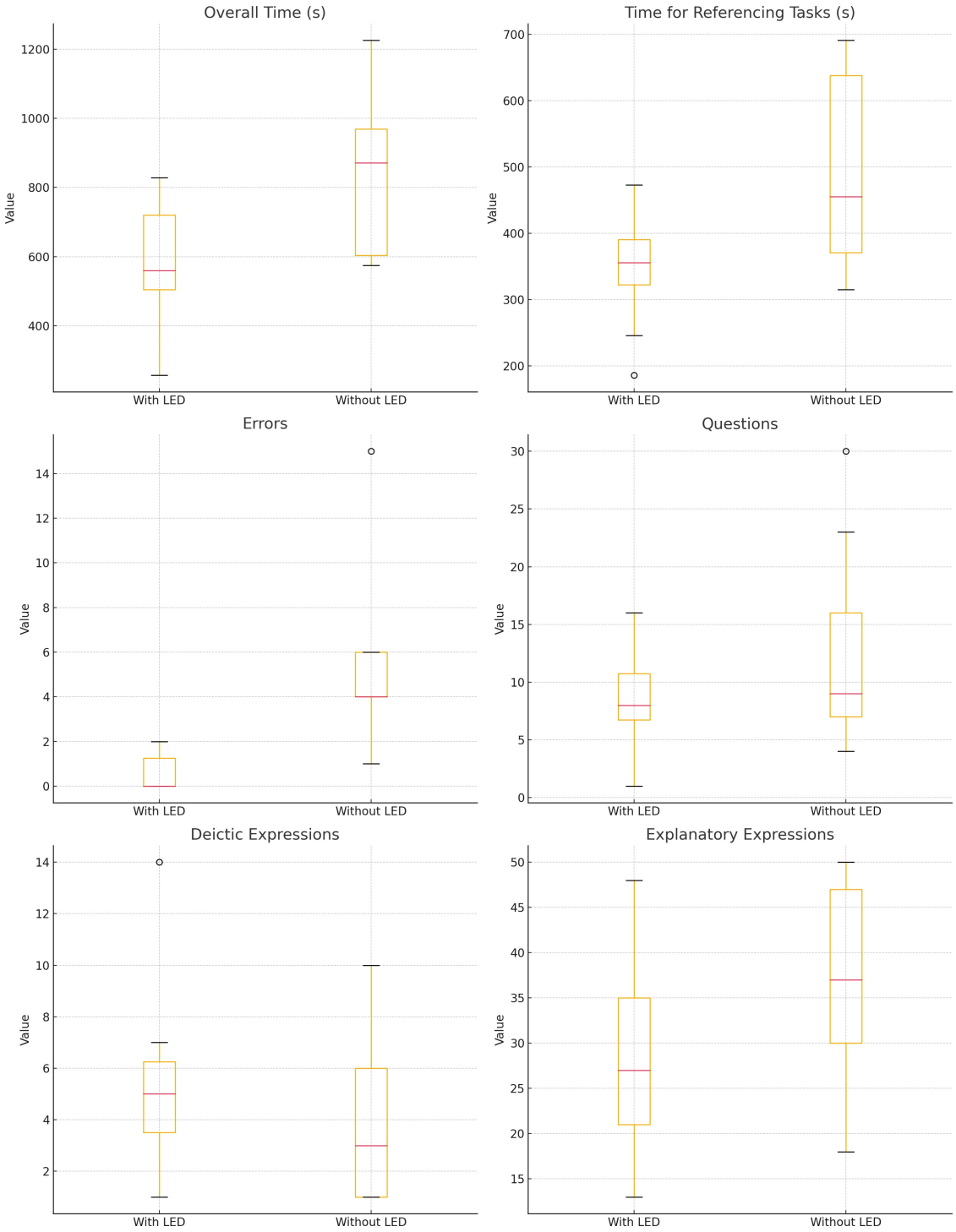


Figure 3.6.: *Box plots of the data from Tab. 3.1.*

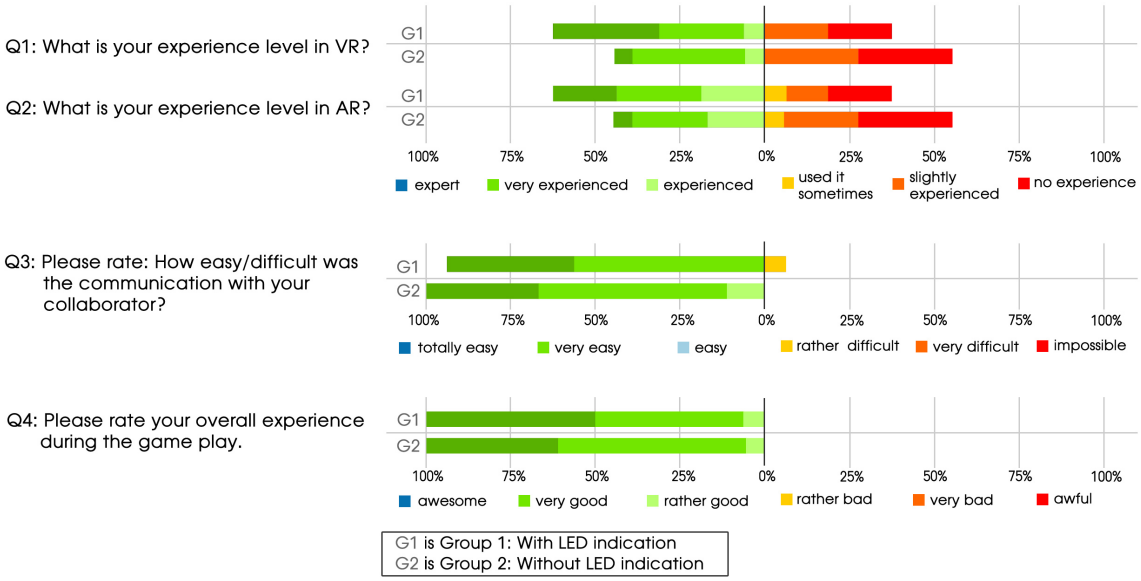


Figure 3.7.: *The results of the post questionnaire. Fig. from [Lad+19]*

3.4. Discussion and Future Work

Our initial hypothesis was that the integration of nonverbal communication methods such as deictic gestures within a shared virtual task space during remote collaboration in an immersive telepresence scenario would significantly increase task efficiency in terms of time and reduce errors by providing unambiguous visual cues. Our results confirm this hypothesis.

In principle, this is not a direct face-to-face collaboration in which the personal space is transferred, but it still underlines the initial statement from the literature review in Chap. 2 that the transfer of more information, in this case by including referencing methods, leads to more effective and also more efficient collaboration than when no referencing is implemented. Because we did not integrate a face-to-face person space, we were able to focus exclusively on the effects of the shared task space in our study.

Feedback on the ease of communicating with their partner (Q3 in Fig. 3.7) was unexpectedly similar in both groups. We expected more pronounced differences and speculated that the non-LED group would rate this question more negatively. In addition, the responses to "Please rate your overall experience during the game" (Q4 in Fig. 3.7) were nearly identical between the groups, with no significant differences. Although solving tasks without LEDs takes longer and is more error-prone, it appears that participants do not subjectively rate the simplified version with LEDs significantly more positively in terms of communication quality and ease of use. This could mean that nonverbal communication in the context of telepresence improves task efficiency but does not necessarily affect subjective user experience or "enjoyment", even though a referencing method leads to an improvement in task performance, at least in terms of quantitative metrics such as time and errors.

Group 1, labeled "with LED indication" (in Fig. 3.7), had reported a slightly higher familiarity with VR and AR technology, which could have influenced the results. However, we analyzed the distribution and found no significant difference between the two groups.

Although the "without LED indication" condition generated a higher number of questions, deictic and explanatory expressions, no significant difference was found. We did notice a difference in communication styles between the dyads, but it was difficult to categorize them definitively. The "with LED indication" dyads exhibited a more assertive, decisive and straightforward communication style, in contrast to the "without LED indication" group, which adopted a more exploratory approach.

Heiser et al. [HTS04] have already shown in a similar experiment that the task space, fully performed in reality without MR technology, is important for task efficiency. We have brought this concept to virtual space and found similar results in this domain, but with more quantified results.

Following the publication of this study, the author of this dissertation spoke with Prof. Dr. Mark Billinghurst about further possible work in this area. Mr. Billinghurst suspected that the length of the questions, and possibly the length of the statements, may have changed between the two study groups. The availability of a referencing/pointing method could possibly be used to identify shorter questions and statements. In addition, the cognitive load might differ between the groups, which could be determined using appropriate questionnaires in the future.

Furthermore, it would be interesting to investigate the influence of a person space on such an experiment. For example, nodding or shaking the head, as well as facial expressions such as frowning or looking scared when the machine is operated incorrectly, could lead to more efficient task performance.

In addition, we are interested in exploring whether providing more comprehensive visual feedback to the remote expert could speed up the maintenance process. This improvement could include monitoring the worker's hand movements and projecting this visualization into the expert's VR environment, similar to what Sodhi et al. [Sod+13] did, potentially shortening the feedback cycle and allowing the expert to intervene if the worker is at risk of operating the machine incorrectly.

3.5. Conclusion

In this chapter, we evaluated the concept of establishing a shared task space in a telepresence scenario with and without the possibility to perform deictic gestures for a simulated machine repair task, facilitated by signaling with physical LEDs. The experiment revealed a significant difference in both task completion time and error rate between the scenarios with the two test conditions "with LED indication" and "without LED indication". However, no significant statistical difference was observed in the frequency of questions or the use of deictic and explanatory phrases.

In conclusion, deictic gestures with spatial reference are important for effective remote collaboration in the context of a shared virtual task space. Research question 2 (RQ2) was "How does the availability of a shared virtual task space, and in particular a referencing tool, affect task efficiency and error rates in remote collaboration?". This can be answered as follows: The availability of a shared task space with the ability to spatially reference objects can save about 30% time and cause about 90% fewer errors, when comparing the mean values of the experimental results. The study design was deliberately chosen to minimize distractions and focus on solving tasks as quickly as possible due to the gamification and competitive nature of the study.

4. Standardizing Body Tracking

There are a large number of different open source and commercial body tracking systems (BTS) that can record and transmit NVC. There is a wide range of technologies with their specific advantages and disadvantages. In addition, the tracking quality and the required computing power vary widely. One of the biggest problems in providing meaningful NVC in immersive telepresence is the lack of interoperability of individual BTSs across different hardware and software. An application developed with a particular BTS will only work with that BTS without further adaptation. This severely limits the distribution and interoperability of different applications and hardware and software ecosystems, and the visualization of body movements can vary significantly with different BTSs and be interpreted differently by viewers. From a technical point of view, this concerns the digital tracking skeleton, e.g. the assignment of individual joints, the hierarchy of joints in a skeleton, and its control data. Control data can have different assignments (x-, y-, z-axis) or orientations (right- or left-handed coordinate systems), or have different scales such as meters, centimeters, or millimeters.

This chapter addresses research question 4 ("How can different body tracking systems and protocols be standardized to ensure that the presentation of nonverbal communication in a telepresence application looks as identical as possible, even when different tracking systems are used?). A software architecture as well as a standardization of a body tracking protocol is proposed. For this purpose, a prototype application has been developed as a middleware called *MotionHub*, which implements five different state-of-the-art BTSs and sends standardized messages over a network to a client, such as the Unity 3D game engine. The goal of the MotionHub is to unify face, finger, and body tracking systems into a single interface, and could be thought of as the SteamVR for body tracking. The MotionHub is a research effort to investigate what is needed to transfer NVC in a simple form of body movements. Several real-world use cases are presented, and the system is evaluated for applicability and performance at the end of this chapter. The system is already used in several commercial and research projects, as demonstrated by Cannav et al. [Can+23], Greve et al. [Gre+22], and Geiger et al. [GGS23].

Please see the oral presentation (`https://youtu.be/GRZqkAN6I9k`) and the source code and documentation: `https://github.com/Mirevi/MotionHub`.

4.1. Related Work

The body tracking domain has seen extensive research and development over many years, resulting in a wide variety of software, hardware approaches, standards, and file formats. This section summarizes basic technologies, hardware and software, file formats, streaming protocols, and general standards for body tracking and motion capture (MoCap). This section focuses on body tracking without facial expression tracking. The related work on face tracking is discussed in Sec. 5.1.

4.1.1. Fundamental Body Tracking Technologies

Motion capture technologies can be broadly categorized into 4 areas: RGB-sensor-based, RGB-D-sensor-based, infrared-sensor-based (often called marker-based), and IMU-sensor-based (IMU: Inertial Measurement Unit). Other technologies such as electromagnetic tracking [Raa+79] or EMG-based (electromyography) [MP04] are alternative approaches, but have not been discussed further because none of these technologies have been accepted for accurate body tracking. Under controlled conditions, electromagnetic tracking is highly accurate, fast, and inexpensive, but the main problem is inference with other ferromagnetic objects such as steel and reinforced concrete. This makes this tracking technology only applicable to specific applications, controlled environments and often requires a calibration phase. However, when all conditions are met, this technology can be very accurate and is even used in medical context. Although optical tracking methods are also very precise, they have the problem that they stop working when the sensor no longer has a clear view of the object being tracked. Electromagnetic methods have the advantage that occlusion is not a problem and biological material does not affect the tracking quality. This means that objects inside a patient's body can be localized in real time during surgery [Pol24]. Electromyography is available as surface EMG (on the skin) and intramuscular EMG. Only the former is suitable for everyday use, but it does not provide a high enough level of accuracy to reliably transmit body movements in the context of immersive telepresence.

4.1.1.1. RGB-based Body Tracking

Human motion capture based on RGB sensor data has been extensively studied for several decades [MHK06; Pop07; Cao+18]. The advantage of this technology is the short preparation time, since no marker or calibration is required for each person. RGB-based tracking systems can be further divided into monocular and multi-view systems. More sensors in the room provide more data, which usually leads to better tracking results. However, there is generally a trade-off between tracking accuracy and speed. The focus in this section is on real-time analysis, as this is the only way to interactively transmit NVC with low latency.

The commercially available tracking system from *The Captury* is based on a certain number of intrinsically and extrinsically calibrated cameras in a room, all facing inwards towards a common center. The system is based on "Sums of spatial Gaussians" [Sto+11], runs in real time even with a multi-camera setup, and was considered one of the most accurate and fastest RGB-based systems until the advent of deep learning based systems.

With the advent of Convolutional Neural Networks (CNNs), the ill-posed problem of extracting and regressing 2D or even 3D limb or joint positions, even from a single camera view, has been successfully solved. The main research results that have contributed to this, besides CNNs, are *ResNets* [He+16] and, in general, *Encoder-Decoder Networks* [SVL14]. In addition to the detection of landmarks such as limbs in images, the effective segmentation of images also plays an important role, since neural networks can be used to recognize people in images in a comparatively resource-efficient manner and to determine the region of interest in a comparatively resource-efficient manner.

Based on these achievements, Newel et al. proposed the *Stacked Hourglass Networks* [NYD16] and achieved significantly better results in various benchmarks compared to the state of the art. They stacked several encoding and decoding networks, resulting in a system that receives an RGB image and reports 16 2D positions of the joints, such as

knees, elbows, and so on. Further improvements were made in the work of Cao et al. using *OpenPose* [Cao+18]. Stacked Hourglass and OpenPose are computationally intensive, and at the time were barely real-time capable. Google has been shown to use further improved versions of the Stacked Hourglass networks for its face, hand, and body tracking, but does not go into all the details in its publications [Baz+20; Kar+19]. They use a Procrustes analysis [Ken89] to speed up the inference time. The unique feature is that the heat map generation of the Stacked Hourglass process could be bypassed and tracking could even be realized on mobile devices in real time. Although not all technical details are disclosed, some trained models are available as part of the *MediaPipe* project [Lug+19].

A recent trend is the use of *Recurrent Neural Networks* (RNN) to realize a stronger relationship between successive images in a sequence. This way, better temporal coherence can be achieved and the results are more fluid. In the early days of deep learning for body tracking, prediction was typically performed on individual images. This approach has changed as the complexity of the networks has increased. Thus, *long short-term memory (LSTM)* [HS97a] and *Gated Recurrent Unit (GRU)* layers [Cho+14] are increasingly used. More advanced systems, such as VIBE [KAB20], use CNNs in combination with GRU and attention layers to fit a parametric model. The tracking is near real-time with accurate results. The use of the *attention mechanism* seems to be one of the key improvements for the academic field of tracking, as it provides a more precise data description and structure for neural networks of 2D or 3D positions [Vas+17; Mil+21].

One of the major drawbacks of monocular RGB-based tracking in direct comparison to other technologies (RGB-D, IR marker-based, and IMU-based) is that even with sensors that have intrinsic and extrinsic calibration data, a metric estimate of the depth or even size of the person being tracked is possible, but ambiguous and usually prone to error. However, this problem is increasingly being solved, as can be seen in SHAPY [Cho+22]. This system is able to display metric data from images, such as height or waist circumference, in addition to body shape and pose.

Meta introduced *MEgATrack* [Han+20], which is able to accurately track hand and finger movements via 4 cameras in an HMD. The metric estimation is done by an initial calibration where the user's hand must be seen by at least 2 cameras simultaneously for a short time. This stereo calibration determines the size of the hand for the subsequent tracking process and helps to limit the ambiguity between hand size and distance to the sensor. Unfortunately, Meta does not publish further details about the architecture of the neural network in this paper. However, they indicate that they also implement temporal dependencies and even extrapolation of landmarks as additional input to the network to improve tracking quality.

Directly comparing depth-only systems with RGB-based systems, the RGB systems usually have the advantage of a higher 2D resolution and a lower noise level. Furthermore, it is also possible to make predictions on different resolutions of the target images, which allows adaptive adjustment of the threshold between speed and accuracy, since RGB sensors often have high resolution with many megapixels. As a result, this technology is well suited to quickly (in real time) obtain a "first and rough guess" of the body pose to further optimize this pose with more complex and accurate algorithms.

4.1.1.2. (RGB-)D-based Body Tracking

Many of the approaches used in RGB-based solutions can be applied in a similar way to RGB-D-based systems, as the additional depth information can simply be used as an additional input channel in a neural network. Furthermore, compared to RGB data, the depth modality is much more insensitive to illumination variations, color and texture changes, and provides rich 3D structural information of the scene. However, there are fewer research articles on depth-based body tracking compared to RGB-based methods in 2023 when writing this text. There may be several reasons for this, but one reason is certainly the much wider availability of color cameras compared to depth sensors. Depth cameras, such as ToF or LIDAR systems, are comparatively expensive, have lower resolution, and require significantly more power in mobile devices compared to RGB sensors. In addition, their range is often limited (often to a maximum of about 6-10m) and the noise in the depth data increases with the distance between the object and the sensor. Nevertheless, providing an additional depth channel generally makes tracking for complex applications such as hand tracking much more robust. For a comprehensive overview of RGB-D tracking methods til to 2018, we refer the reader to [Wan+18a]. While it can be seen from around 2014 that deep learning is gaining ground and delivering good results, "traditional" tracking methods such as optimization-based methods, e.g. with the Iterative Closest Point (ICP) algorithm [RL01; Sha+15], are still superior in the trade-off between computation time and tracking quality. However, it can be seen that in very complex scenarios, such as occlusion, deep learning methods generally deliver better results.

Convolutional Neural Networks (CNNs) have not only revolutionized RGB-based tracking, but are equally applicable to RGB-D streams. The depth information provides an important additional channel of data that enhances the network's ability to understand and segment images in three-dimensional space, leading to more accurate modeling of objects and environments. This application is particularly effective in overcoming the ambiguities associated with depth perception in monocular RGB systems.

CNNs are one of the most common approaches to body tracking. However, there are other architectures that have their specific advantages and problems and are suitable for RGB-D data. Alternative approaches that also give good results with RGB-D data are *Graph Convolutional Networks (GCN)* by Caetano et al. [Cae+19] and various forms of RNNs such as LSTMs [HS97b]. The problem with RNNs is their high memory requirements and limited ability to handle longer sequences of past and future information. This problem has been successfully solved by the Transformer architecture, which has further improved the quality of results by allowing temporal relationships to be better understood [Goe+23].

4.1.1.3. Feature-based Outside-In and Inside-Out Body Tracking

Outside-in systems rely on the placement of reflective or active markers (also called features) on specific anatomical landmarks of the human body. These markers are then tracked by multiple calibrated infrared cameras that capture the reflected or emitted IR light from the markers. The 3D positions of the markers are triangulated using data from multiple camera views, allowing accurate reconstruction of the body's motion. Well-known systems are available from *Natural Point* with the name *OptiTrack* [Nat20] or from the company *Vicon* [Vic20].

Until around 2012, inside-out was difficult to implement in consumer hardware because it requires sensors inside the moving device that needs to be tracked. On the one hand,

sensors were too large, too heavy, and often too inaccurate to be installed in an HMD, and on the other hand, there was a lack of algorithms and processing power to provide real-time tracking data. This has changed, and many of today's HMDs use cameras for position detection and are classified as inside-out. They look for contrasting and distinct features in the environment and use them as anchor points. Typically, these systems are combined with IMU-based systems to provide higher sampling rates and more robust tracking.

Valve's Lighthouse tracking system [YS19] is a hybrid between outside-in and inside-out tracking, and differs significantly from other systems. Sensors are also located in the device being tracked, but these sensors are not cameras, but photodiodes with a bandpass filter. Unlike cameras, these photodiodes provide only binary information. Around the hardware to be tracked there are so-called "lighthouse stations" with rotating laser fans with a fixed angular speed and a bunch of LEDs. The LEDs send a synchronization signal in the form of a flash to measure the time when the laser fan hits the respective sensors (binary signal) on the device to be tracked. Using information about how fast the sensor is rotating and how much time has elapsed since the synchronization flash hit the sensor, the system can calculate where the device is located in space relative to the lighthouse station [YS19].

Feature-based systems offer high accuracy and temporal resolution, making them suitable for applications that require precise motion capture, such as the transmission of NVC gestures with even the smallest movements. However, they have limitations, including the need for line-of-sight between markers and cameras, potential marker occlusion, and the time-consuming process of marker placement and calibration. In addition, these systems are typically more expensive compared to RGB and RGB-D systems.

4.1.1.4. Inertial Measurement Unit (IMU)-based Body Tracking

Inertial Measurement Unit (IMU)-based body tracking systems, such as those from *xSens* [XSe], *Perception Neuron* [Noi20], or *Rokoko* [Ele23], represent a technology that can provide high-quality tracking data. These systems work by using a network of sensors distributed throughout the body, each containing an IMU sensor.

Each IMU typically consists of a triad of accelerometers, gyroscopes, and sometimes magnetometers. Accelerometers measure linear acceleration, gyroscopes measure angular velocity, and magnetometers measure magnetic fields to provide heading information. The IMUs continuously record data about their specific orientation and accelerations. Sensor fusion algorithms are used to combine the data from the accelerometers, gyroscopes, and magnetometers to estimate the orientation of each sensor module relative to a global reference frame. This process compensates for the individual limitations of each sensor type, resulting in a more robust and accurate motion measurement. Typically, various filtering and smoothing techniques are applied to the data.

While IMUs are commonly used to track smartphones in *Simultaneous Localization and Mapping (SLAM)* applications, or to improve the temporal resolution of optical tracking systems such as Valve's Lighthouse Tracking, it is worth noting that IMUs alone are now often sufficient to accurately represent spatial position and orientation. Today, the level of accuracy is sufficient to deliver NVC with reasonable quality in real time.

4.1.2. File Formats and Standards

Perhaps the best known and oldest de facto standard for VR data exchange is the *Virtual-Reality Peripheral Network (VRPN)* by Taylor et al. [Tay+01]. It provides simple and unified interfaces for various devices from different vendors. Many of these devices have common functions, such as 6-DOF tracking or key input, but access to these functions differs between manufacturers. VRPN unifies functions across devices with generic classes such as *vrpn_Tracker* or *vrpn_Button*. As a result, it can be considered both a standard and middleware. MotionHub takes a similar approach, but focuses on body tracking. The first official international standard for humanoid animation is *H-Anim*, which was developed within the *Extensible 3D (X3D)* standard framework and is the successor to *Virtual Reality Modeling Language* (VRML) [Sta97; Sta03]. H-Anim was released in 2006 [Sta06], updated in 2019 [Sta06], and represents one of the only attempts to date to establish an official open standard for humanoid avatar motion and data exchange.

COLLADA and *FBX* are exchange file formats for 3D applications and are widely used today. While humanoid animation is not the primary focus of COLLADA, its open and flexible structure allows developers to store body tracking data. In contrast, the proprietary FBX format emphasizes motion data but lacks clear documentation, resulting in incompatible versions.

Although COLLADA and FBX can be used to write and read 3D geometry, the *Biovision Hierarchy* (BVH) file format is designed exclusively for managing skeletal motion data and is therefore simpler in structure. It is supported by numerous body tracking applications and is often used for real-time humanoid motion data transfer due to its simplicity and reduced overhead compared to other file formats.

4.1.3. Software and Hardware

In 2010, Microsoft began shipping the Kinect, which greatly expanded the body tracking community by providing an affordable and powerful sensor. During this time, *PrimeSense* released *OpenNI* [Occ20] and *NITE* [Occ20, p.15]. OpenNI provides low-level access to the Microsoft Kinect and other PrimeSense sensors, while NITE acts as middleware, allowing users to perform higher-level operations such as body tracking and gesture recognition. PrimeSense stopped distributing OpenNI and NITE after it was acquired by Apple Inc. Although OpenNI remains available on other websites [Occ20], active development has been discontinued.

Building on the success of the Kinect and other PrimeSense sensors, several proprietary BTSs have been developed, including *iPi Mocap Studio* [Sof24], *Brekel Body* [Bre], and *Nuitrack* [Inc] based on RGB-D streams. Examples of BTSs using IMU-based tracking are *Xsens MVN* [XSe], *Perception Neuron* [Noi20], and *Rokoko Smartsuit II.* [Ele23]. Optical solutions with passive or active (infrared) markers include *OptiTrack Motive: Body* [Nat20], *Qualisys Track Manager* [Qua24], *ART-Human* [Gmb20], *Vicon Tracker* [Vic20], and *Motion Analysis Cortex* [Inc20c].

The concept behind *OpenVR* [Lud20], implemented in *SteamVR* [Val24] and *OpenXR* [Inc20a] is similar to the idea behind MotionHub. It consolidates the interfaces of many MR input and output devices into a single library. Its open source nature probably contributes to its success. While OpenVR and OpenXR will not focus on full-body tracking until 2021, efforts are being made to extend their functionality.

The company *IKenima* [IKi20] has developed an application called *Orion*, which is based on inverse kinematics and uses HTC Vive trackers attached to the hips and feet. IKenima was acquired by Apple in 2019, and Orion's distribution was subsequently discontinued.

MiddleVR [Mid20] is a proprietary middleware that supports various commercial input and output devices and provides a generic interface to the Unity game engine [Tec19]. Although body tracking is supported, it is not the primary focus of the software. The commercial software most closely related to MotionHub is *Reallusion iClone* with its plug-in *Motion LIVE* [Inc20d], which emphasizes real-time body tracking and supports multiple tracking systems for face, hand, and body capture.

4.1.4. Research Systems

In contrast to the previous section, this section presents non-commercial work from the academic field, the source code of which was or still is mostly publicly available. The research work most directly related to our work includes *OpenTracker* [RS05] and *Ubiquitous Tracking (UbiTrack)* [New+04; Wag+04]. Both systems are generic dataflow networking libraries for various tracking systems. Unlike MotionHub, they do not focus exclusively on body tracking. They provide a generic interface to object tracking systems for mixed reality applications, similar to the concept behind VRPN. Although OpenTracker and UbiTrack have a different focus than MotionHub and their research was done more than 16 years ago, the unification concept is similar and can be reused for our work. Suma et al. [Sum+11] developed the *FAAST (Flexible Action and Articulated Skeleton Toolkit)* based on OpenNI and NITE. It provides a VRPN server for streaming the user's skeleton joints over a network. However, development has been discontinued. Damasceno et al.. [DCL13] presented a middleware for multiple low-cost motion capture devices applied to virtual rehabilitation. Eckert et al. [Eck+16] proposed a similar system for playing exergames. *OpenPTrack* [MBM16] is one of the most recent systems. It is not described as middleware itself, but rather as a BTS. However, because OpenPTrack supports processing multiple RGB-D camera streams from different vendors and uses different tracking algorithms (such as its own or *OpenPose* [Cao+17]), it acts more like middleware. Similar to MotionHub, OpenPTrack is open source and focuses primarily on body tracking. The difference between the two systems is that OpenPTrack focuses solely on working with RGB-D streams, while MotionHub aims to incorporate different tracking technologies, such as optical or IMU-based BTSs, to take advantage of the unique benefits of each technology. Consequently, our approach requires a more generic method to fuse, calibrate, and merge the heterogeneous data from different BTSs. To the best of the authors' knowledge, no currently available middleware supports recent high- and low-cost BTSs, different technologies, and is open source. Although the idea of a body tracking middleware is not new, MotionHub addresses these aspects, making it a unique system and a valuable contribution to the community.

4.2. Motion Hub System

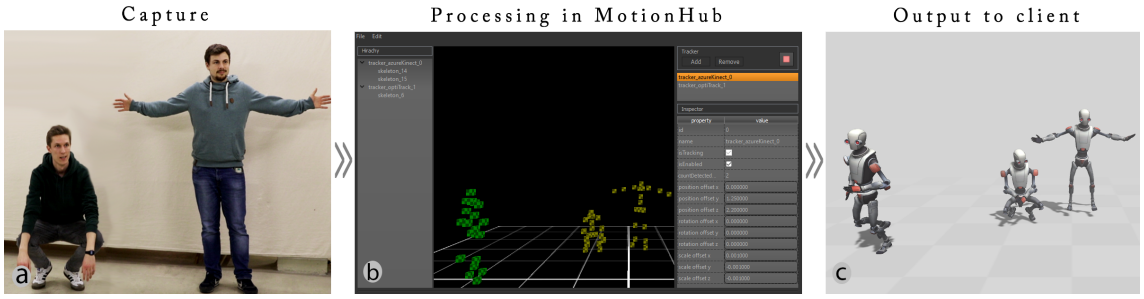


Figure 4.1.: *Overview: The MotionHub is able to perform live capture, process and merge the tracking data. The yellow skeletons in b) are live captured by Azure Kinect and the green skeleton is an OptiTrack recording streamed via UDP from the Motive software. Finally, the combined tracking data is sent to a client. In this case it is the Unity3D game engine, shown in c). Fig. from [Lad+20a].*

The previous section on related work has shown that there is a wide variety and number of different tracking systems. To answer the research question 4 (RQ4: " How can different body tracking systems and protocols be standardized to ensure that the representation of nonverbal communication in a telepresence application looks as identical as possible, even with the use of different tracking systems?"), the first design rationale would be that the "target" BTSs need to send their tracking data to the MotionHub. The second rationale would be that the MotionHub detects which BTS the data comes from, converts it accordingly to a defined standard, and then forwards it to the "final" client, in our case the Unity game engine.

In order to realize the MotionHub's role as middleware, several requirements have to be met. The first idea was to write a separate server for each BTS as a standalone program that sends the tracking data to the MotionHub for further processing. However, we realized that 1.) the usability and user acceptance decreases if the MotionHub is fragmented into different programs and different configuration windows (e.g. an additional window for each BTS) and 2.) we realized that low latency is a non-negligible factor when building a middleware for interactive applications. Each transmission between a server and a client causes a latency of only a few milliseconds. However, in a fragmented system consisting of different building blocks, the accumulation of delays can gradually grow. This had to be avoided.

As a kind of "lowest common denominator" of communication, it turned out that every manufacturer of a BTS - without exception - provided an implementation as a C or C++ SDK. Since these two languages are compatible and among the most powerful, the MotionHub was built as a C/C++ framework to integrate with the native SDKs. In this way, the MotionHub receives raw skeleton data from different BTSs via the native SDKs, processes it to create a unified skeleton in real time, and transmits it to the client via a UDP-based network message using the OSC protocol *(Open Sound Control)* [Wri05].

4.2.1. Unified Skeleton

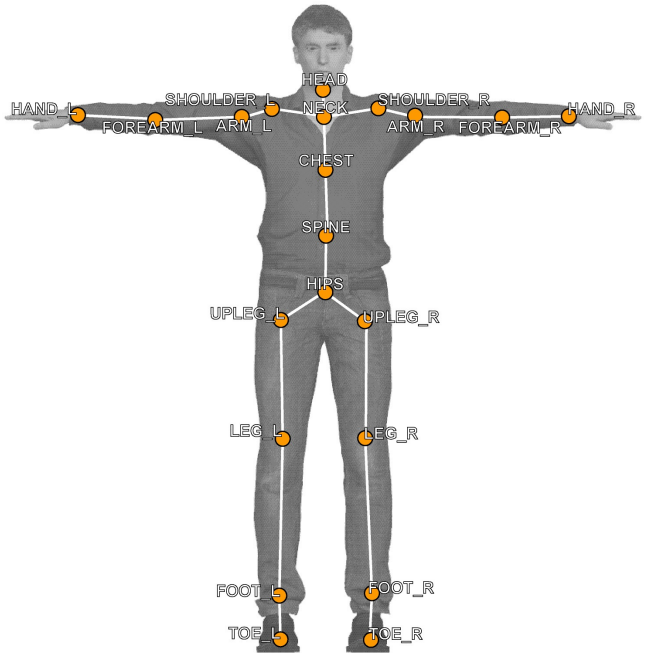


Figure 4.2.: *The unified skeleton structure that MotionHub streams to its customers. The skeleton of each BTS is standardized to this structure. Fig. from [Lad+20a].*

To standardize the output data from different BTSs, we convert the different skeleton data structures into a universal skeleton model with 21 joints, as shown in Fig. 4.2. The joint structure is adapted from the Unity humanoid avatar skeleton, [Tec20], which is similar to the *H-Anim LAO-1* standard [Bru06, p.20] used by BTSs such as the Azure Kinect Body Tracking SDK. This standardization of joint names and indices facilitates a consistent data representation of the converted skeletal data for avatar animation in third-party applications. Each joint is characterized by a global position (*vector3*), a global rotation (expressed in quaternions), and a confidence value (ranked as [none, low, medium, and high]) within a right-handed coordinate system. Some BTSs, such as the Azure Kinect SDK, provide joint confidence values based on distance and visibility. For joints from BTSs that do not provide confidence values, such as OptiTrack, we use "high" as the default. All original skeletal data is manipulated based on this standardization, the details of which will be discussed later in this chapter.

4.2.2. Subsystem Architecture

An overview of the data flow and architecture of all subsystems is shown in Fig. 4.3. Each BTS is shown at the top of this figure. Each BTS operates at its own unique acquisition frequency and refresh rate. For fast processing of incoming data and low latencies, the data handling code must run independently of the main program loop. To address this issue, we embedded independent threads for each BTS within the MotionHub. A tracking thread collects raw skeleton data from the corresponding SDK, processes it, and immediately sends it to the client (game engine), as shown in the box with the orange border in the following Fig. 4.3:

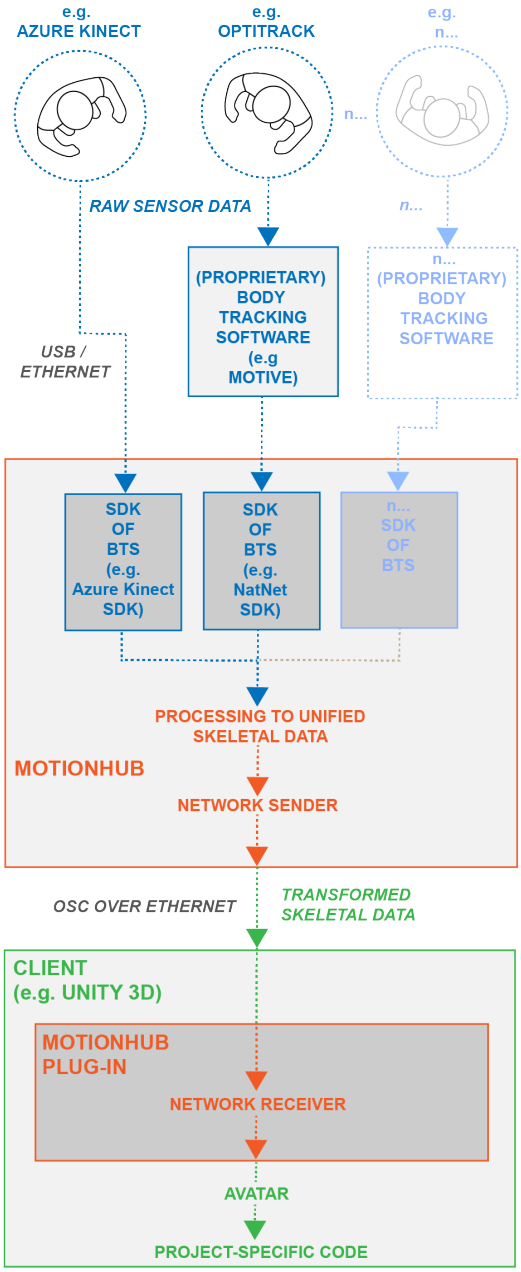


Figure 4.3.: *Data flow and architecture of the Motion Hub. Image template courtesy of Daemen et al. [Dae+16]). Fig. from [Lad+20a].*

When dealing with threads, it is crucial to protect memory areas from concurrent access through appropriate atomic structures. Several internal processing pipelines, such as the tracker threads, the UI thread, the render thread, or the network sender thread, access these protected regions. We found that copying the protected memory to pre-allocated areas first and then processing on that copy, rather than locking critical areas during processing, resulted in improved performance and reduced latency in processing and transmitting data.

To pass skeleton data to the game engine client, we used a UDP-based iteration of the OSC protocol [Wri05], chosen for its simplicity, rapid integration, and speed. The structure of our protocol mirrors the Biovision Hierarchy (BVH) file format, but is augmented with

additional MotionHub-specific control messages necessary for communication between the MotionHub and the client side. In addition, we have developed a more compressed skeleton data representation than the BVH or VRPN-based data streams to further reduce network latency. In a local area network (LAN), we chose UDP over TCP to prioritize fast connectivity and lower latency over packet loss recovery, since packet loss is rare in a LAN environment. While in our case the players were physically located in the same place, it would also be possible to establish a remote connection over the Internet using a TCP connection. Each OSC packet contains translation (three float values) and rotation (four float values) data for each joint, along with the skeleton ID (as an integer).

In addition, the UI module includes a rendering window to display both incoming and converted skeleton representations. The joints of the skeleton are displayed in different colors based on their confidence values, as shown in Fig. 4.4.

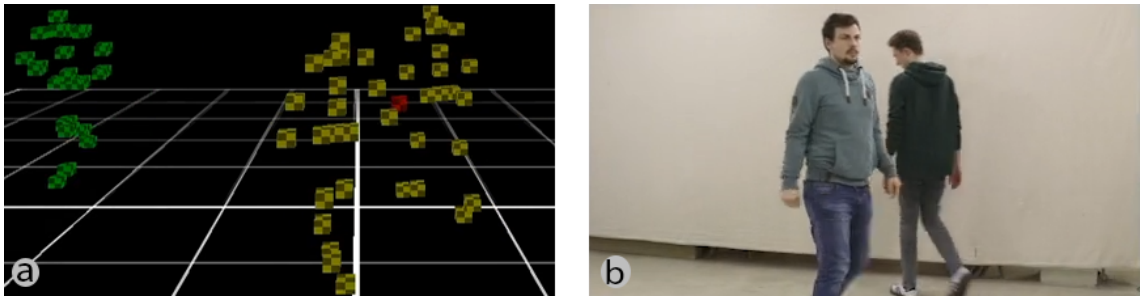


Figure 4.4.: *The OpenGL render window in a) uses color to indicate the current tracking confidence of the joints. Some BTSs, such as Azure Kinect, provide such values. Yellow joints have a "medium" confidence value, while red joints have "low" confidence because they are occluded, as seen in b). For BTSs that do not provide a confidence value, a default value can be selected manually. In this figure, the OptiTrack value is set to "high" by default and displayed in green. Fig. from [Lad+20a].*

4.2.3. Supported BTS and Dependencies

The MotionHub was developed in C++ for two reasons: First, C++ is one of the fastest programming languages available [Gou20], which is a critical feature for real-time body tracking middleware. Second, creating an interface with different SDKs is only feasible in C++, as virtually every BTS offers an SDK or API rooted in C or C++.

Since Microsoft's Body Tracking SDK [Mic20b; Mic20a] uses neural networks for processing, it requires the use of the NVIDIA CUDA Deep Neural Network library (cuDNN) [NVI19]. To manage matrix, vector, and quaternion computations, MotionHub uses the Eigen header-only library [Eig20] and the user interface is built using the Qt 5 framework [Qt20].

To maximize code openness and ease of use, MotionHub automatically downloads and configures several software dependencies during the build process using CMake [Inc20b]. This greatly reduces developer workload and paves the way for more streamlined future development. Compiled binaries are also available. We chose Windows as our target operating system because all SDKs are exclusive to this platform and a significant number of BTSs do not provide interfaces for Macintosh or Linux-based systems.

The following BTSs are supported by the MotionHub:

- Microsoft Azure Kinect
- Natural Point OptiTrack
- The Captury
- Movella XSens MVN Animate
- OpenVR
- OpenVR with multiple Vive Trackers with an inhouse-developed inverse kinematik (IK) solver

As mentioned in the related work Sec. 4.1, these BTSs can be broken down into their underlying technologies: RGB-D, marker-based, IMU-based and additionally "Lighthouse with IK". Each of these technologies has its specific advantages and disadvantages. For this reason, one representative of each has been implemented in MotionHub and is compared directly below in the Tab. 4.1. Each technology offers unique advantages in terms of refresh rate, latency, accuracy, tracking area, setup time, and cost. The following discussion compares these technologies and explores potential multi-modal approaches that combine their strengths.

We have added "Lighthouse with IK" as an additional technology to the Tab. 4.1 below, as our own experiments have shown the potential of combining a tracked HMD, two controllers, and multiple Vive trackers with inverse kinematics solvers. The values from Tab. 4.1 are based on the implementation and analysis of a self-developed system integrated into MotionHub. Straps with Vive trackers are attached to the feet and hips. Together with the tracking data from the HMD and the controllers, we feed and solve a kinematic chain with a combination of *Forward And Backward Reaching Inverse Kinematics (FABRIK)* [AL11] and *Cyclic Coordinate Descent (CCD)* [CD03].

Technology	**RGB**	**RGB-D**	**Marker-based**	**IMU-based**	**Lighthouse with IK**
Refresh rate	30 - 120Hz	30Hz	10 – 10 000Hz	240Hz	120Hz
Latency	∼50ms	∼70-100ms	∼1 - 10ms	∼20ms	∼5-10ms
Precision	∼20mm	∼10-40mm	<0.3mm	∼10mm	<2mm
Tracking area	100m^2 and more	max. 5m Distance	100m^2 and more	Radius Wifi 50m	100m^2
Setup time per Person	<1min	<1min	∼15min	∼30min	∼10min
Price	∼30.000€	from ∼350€	∼25.000€ and more	∼6.000 - 25.000€	1.500 - 3.500€
Examples	Captury	Microsoft Kinect, Intel Real Sense	OptiTrack, QualiSys, Vicon	xSens, Perception Neuron	HTC Vive Tracker

Table 4.1.: *A rough classification and direct comparison of different body tracking technologies that are implemented into the MotionHub.*

Marker-based systems offer a wide range of refresh rates from 10Hz to 10,000Hz, making them suitable for high-speed motion capture such as biomechanical analysis in sports, but are not required for real-time transmission of NVC at their highest level. A BTS should typically have more than 30 Hz to smoothly transmit nonverbal signals. Both RGB and Lighthouse with IK systems support up to 120Hz, providing high responsiveness suitable for real-time applications such as virtual reality. IMU-based systems also offer a high frame rate of 240Hz, which is beneficial for dynamic motion capture. In contrast, RGB-D systems typically operate at lower refresh rates (30 Hz), which can limit their usefulness in fast-paced environments.

Marker-based systems have the lowest latency (approximately 1-10ms), followed by Lighthouse with IK systems (5-10ms). IMU-based systems have a moderate latency (about 20ms), while RGB and RGB-D systems have the highest latencies, which can affect user immersion, embodiment, and interaction quality in virtual environments. The bottom line is that latency for interactive remote collaboration should not exceed about half a second in our experience. Furthermore, we observed that the verbal and nonverbal communication channel can be shifted so far that communication becomes uncomfortable.

Accuracy is critical for applications that require precise and detailed motion tracking. Marker-based systems offer unparalleled precision (<0.3mm), ideal for applications requiring micro-gestures. However, this precision is not typically required for NVC. Lighthouse with IK follows with excellent accuracy (<2mm). In contrast, RGB-D and IMU-based systems offer moderate accuracy, and standard RGB systems have the lowest accuracy (about 20mm) with a relatively high noise level. IMU-only systems typically need to be recalibrated after a period of time (a few minutes to an hour) as they record a constantly increasing offset.

The spatial range in which motion can be accurately captured varies widely. While RGB, marker-based and Lighthouse with IK systems offer large tracking volumes (up to 10 m x 10 m for Lighthouse with IK), RGB-D systems are limited to smaller distances due to their limited field of view (maximum 5 m). IMU-based systems have the unique advantage of not being limited to a field of view and offer a large tracking radius (up to 50 m and more). They are only limited by the WiFi signal.

Setup time per person and cost are considerations that affect the deployment of tracking technologies, practical applicability, ease of use, and the chance to gain mainstream acceptance due to low price. Marker-based systems are the most time-consuming and expensive to set up. RGB and RGB-D systems offer fast setup times and lower costs, making them accessible to more casual users for a "quick call" without any preparation time. Lighthouse with IK and IMU based systems represent a middle ground, requiring moderate setup time and offering a range of prices that can be justified by their benefits such as good precision and therefore good reconstruction of NVC gestures.

4.2.4. User Interface

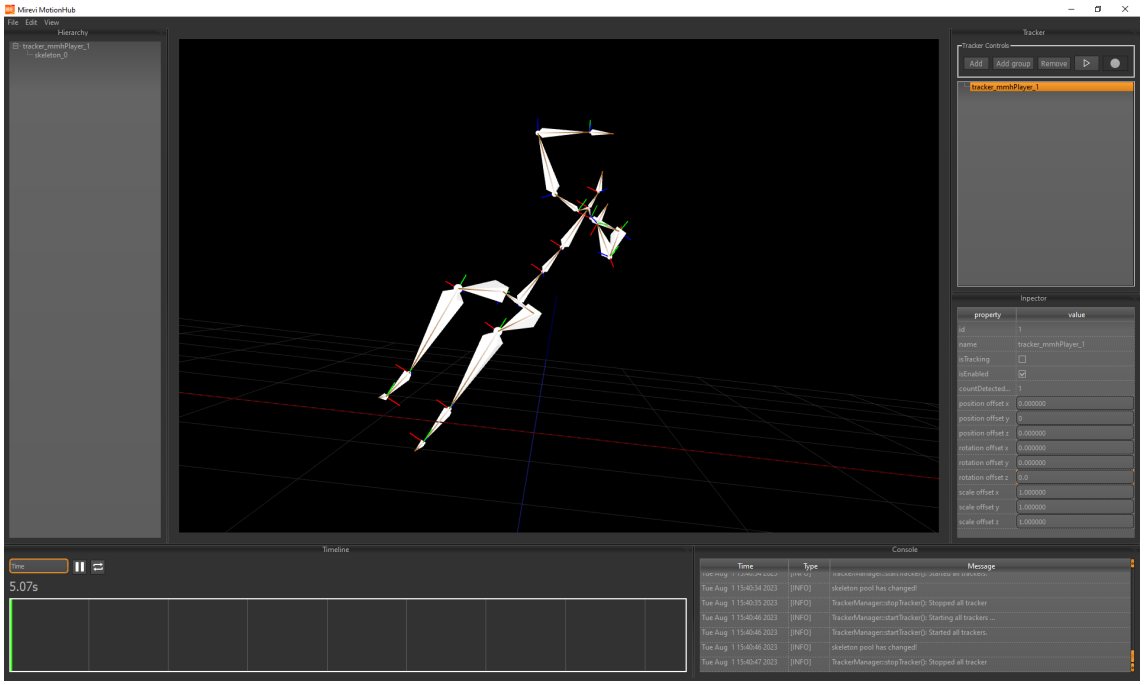


Figure 4.5.: *The MotionHub user interface. The left side shows all currently running BTS and found skeletons. The bottom left is the timeline. In the lower right corner is the log window. On the right side is the tracker overview and the corresponding detail window called Inspector – similar the UI of Unity.*

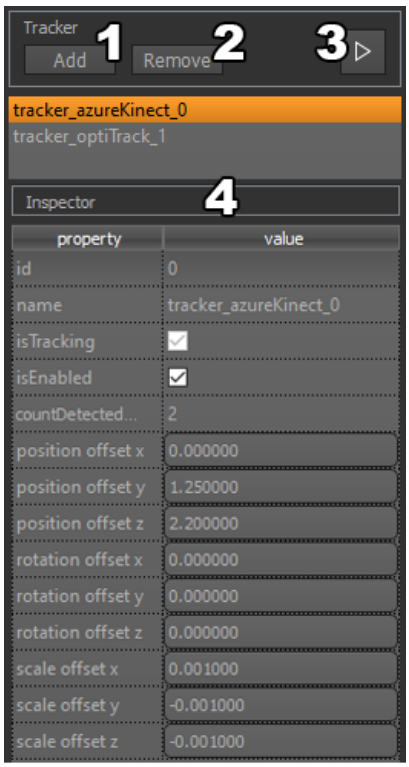


Figure 4.6.: *MotionHub user interface: Tracker Control Panel (1–3) and Tracker Property Inspector window (4). Fig. from [Lad+20a].*

The main MotionHub user interface is shown in Fig. 4.5. A close-up of the right side of the interface is shown in Fig. 4.6. Numbers 1 and 2 in white mark the buttons for adding and removing BTSs. Number 3 identifies a toggle button that either starts or stops the global main tracking loop. All BTSs have the ability to be enabled or disabled collectively or individually.

As shown below number 4 in Fig. 4.6, selecting a tracker from the list (highlighted by the orange box) reveals its properties in the *Tracker Property Inspector*. Different tracking systems report in different coordinate systems. For example, OptiTrack's center is in the middle of the tracking area on the ground, while Azure Kinect's origin is inside the depth sensor itself. When multiple BTSs are integrated into a single tracking area, their origins must be spatially aligned. Therefore, the Tracker Property Inspector allows the user to manually adjust the coordinate system origin of different BTSs. However, manual alignment can be tedious. The following section introduces and discusses several methods for automatically aligning and merging different BTSs in the same physical space.

4.2.5. Conversation Matrices

Processing joint position data into the MotionHub's unified coordinate space involves several steps: applying translation, rotation, and scale offsets to merge tracking spaces, and mirroring the correct axes if necessary. Joint rotations, however, are the most complicated and vary from BTS to BTS. A list of the specific rotations of the implemented BTSs can be found online in the MotionHub documentation: `https://github.com/Mirevi/MotionHub/tree/master/doc`. In addition to the coordinate system, we also had to consider the skeletal structure. In some BTSs, the skeleton is hierarchically structured so that the joint rotations are in local coordinate spaces. These local values are transformed into global rotations by the MotionHub before being transmitted to the receiver side. For example, each joint rotation of the Azure Kinect system is offset by different values.

$$R_i = I_i O_i^{-1} T$$

The output rotation quaternion R for all joints i of a tracker is the product of the tracker-specific global coordinate system transformation T, the inverse global offset orientation O, and the raw input rotation I.

4.2.6. Game Engine Client

In order to receive skeleton data in a game engine, we developed a receiver package for the Unity engine. It contains code that creates a character for each received skeleton and animates it with given rotation values as shown in Fig. 4.7. The character animation is solved in the plug-in code by multiplying all inverse joint rotation values of the character in a hierarchy order and the joint rotation in T-pose.

$$R_{i(client)} = I_i T_i \prod_{k=j(i)+1}^{n} r_{f(i,k)}^{-1}$$

For all joints i, the transmitted rotation quaternion I is multiplied by the joint rotation of the character in T-pose T and the product of all inverse joint rotations r in the skeleton hierarchy above the current joint. While *f(i,k)* returns the joint that is k nodes above i in the hierarchy, *j(i)* shows on which hierarchy level the joint is located. The process iterates

upward through the joint hierarchy, starting with the parent of the joints and ending with the root joint, where n is the number of iterations. Then the product quaternion R is applied to the local rotation of the character.

When developing a plug-in and integrating a BTS into MotionHub, it is critical to preview the processed data to identify and debug rotation offsets on different axes. Typically, there is little documentation from the BTS developer as to how exactly the received data is oriented in Cartesian space. To facilitate this process in Unity, our plugin is able to visualize debug options, as can be seen in Fig. 4.7b. These options include toggling the display of the skeleton ID, avatar position, joint axes, joint names and avatar meshes. The skeleton ID is the same as the MotionHub's internal skeleton ID and is passed to the client via OSC data packets.

The plugin is also designed to be avatar mesh independent. This means that it is possible to switch between different skeletons without any code changes, as long as Unity recognizes them as humanoid.

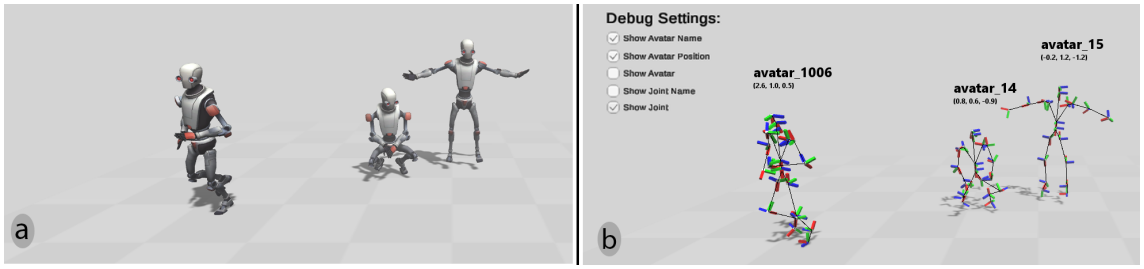


Figure 4.7.: *The Unity game engine plug-in: a) shows the avatar view in the Unity renderer. b) shows the debug view, which shows the position and rotation of all joint axes. Fig. from [Lad+20a].*

4.3. Spatial Alignment of Different Body Tracking Systems

Manually adjusting and aligning two BTSs in the same physical space using the *Tracker Property Inspector* window (Fig. 4.6) is usually very tedious and difficult, since several degrees of freedom such as translation and rotation have to be taken into account. However, there are some methods that can superimpose the coordinate systems of different BTSs in an automated procedure. In this section, we will briefly discuss how different BTSs based on different base technologies, such as an optical BTS and an IMU-based system, can be spatially calibrated.

Basically, the basic technologies used for BTSs can be seen in Tab. 4.1. These are RGB, RGB-D, infrared marker-based, IMU-based, and as an exotic that is gaining more and more importance in the BTS field, the Lighthouse system invented by Valve [YS19; okr16]. The Lighthouse system uses several basic technologies and is, strictly speaking, not a single technology, but a combination of several. It is also sometimes classified in the literature as outside-in tracking and sometimes as inside-out. However, an explicit listing in this section of the Lighthouse system is useful because working with other BTSs has several practical advantages, such as using Lighthouse-based HMDs with other body tracking technologies. The following Tab. 4.2 shows the specific calibration procedures that can be used when combining different BTS technologies.

	RGB	**RGB-D**	**IR marker**	**Lighthouse**	**IMU**
RGB	1 or 2	1 or 2	1	1	1
RGB-D	1 or 2	1 or 2 or 3	1 *	7	1
IR marker	1	1 *	4	1 ***	1
Lighthouse	1	7	1 ***	4	1
IMU	1	1	1	5	6

Table 4.2.: *When spatially aligning different BTS technologies in the same physical space multiple algorithms can be applied depending on the combination of technologies. Numbers and asterisks are explained below.*

The numbers in the table have the following meaning:

1. Singular value decomposition (SVD) based Iterative Closest Point (ICP) on skeleton joint locations (aligning both skeletons in multiple frames)
2. Perspective-n-Point solution (e.g. with OpenCV's solvePNP function)
3. Non-SVD-based-ICP on both point clouds
4. No alignment necessary
5. Procedure available for xSens [BVb]
6. Alignment procedure from manufacturer [BVa]
7. Alignment procedure proposed below in this section

*: In our experiments, we observed interference between OptiTrack and Azure Kinect, which use the same infrared spectrum for their measurements. The Kinect showed significantly higher errors than OptiTrack. The errors can be reduced by manually lowering the frame rate of OptiTrack significantly (e.g. to 30 fps). It is also possible to connect OptiTrack and Kinect to a synchronization device [Nat23] to achieve higher frame rates.

**: Combining the Lighthouse technology with infrared-based RGB-D sensors such as Intel RealSense or Azure Kinect can affect the tracking performance. We have investigated which combinations work and which do not, as shown in Tab. 4.3 and 4.4:

	Lighthouse v1 receiver	**Lighthouse v2 receiver**
RealSense D415	No interference	Strong interference
Azure Kinect	<1m: bad // >1m: OK	<1m: bad // >1m: acceptable

Table 4.3.: *Compatibility with RGB-D sensor when using Lighthouse **v1** base stations.*

	Lighthouse v1 receiver	Lighthouse v2 receiver
RealSense D415	Not compatible with LH v2	<1m: bad // >1m: acceptable
Azure Kinect	Not compatible with LH v2	No interference

Table 4.4.: *Compatibility with RGB-D sensor when using Lighthouse **v2** base stations.*

***: Combining Lighthouse and OptiTrack requires a synchronization device [Nat23].

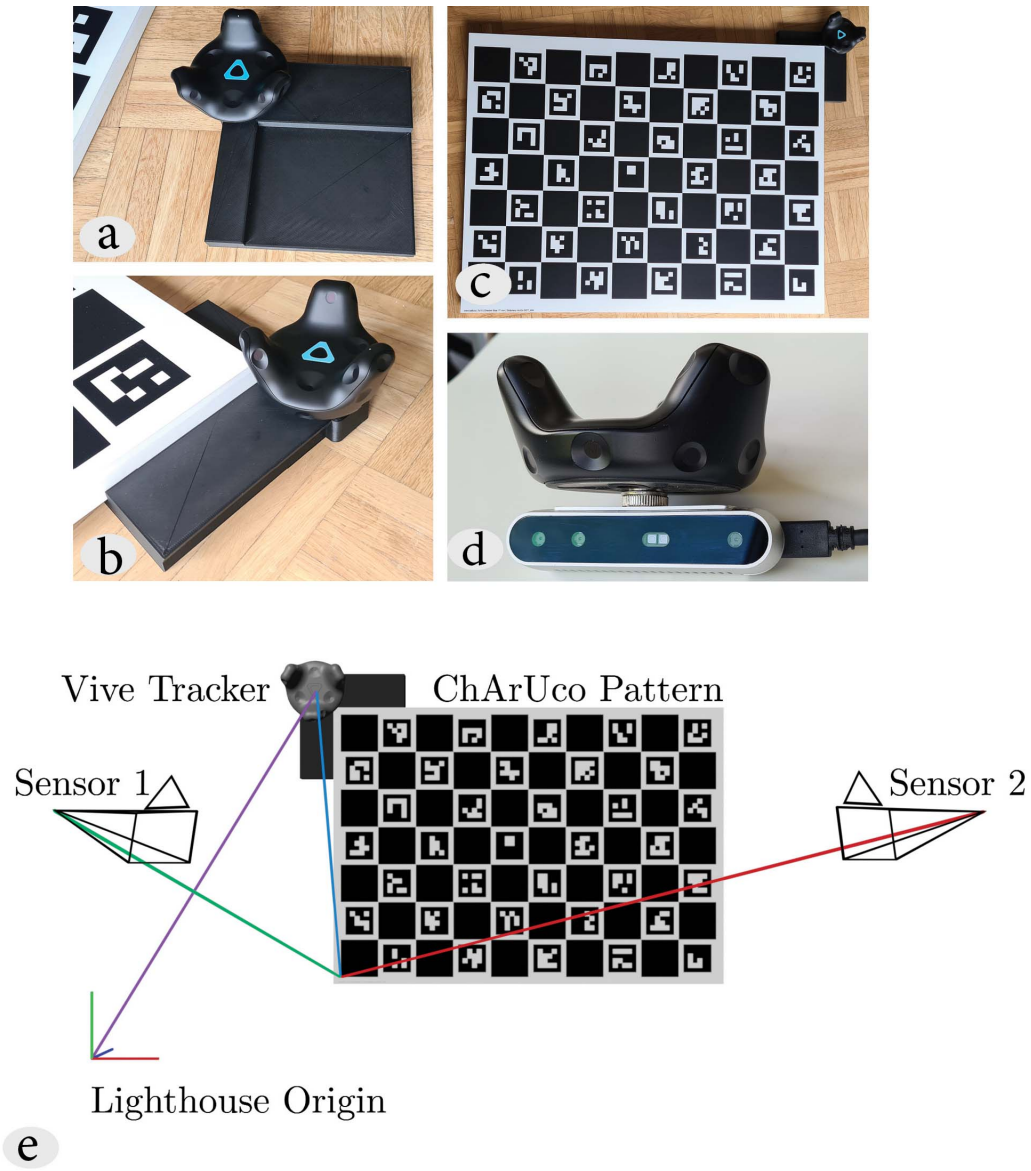


Figure 4.8.: *a) 3D-printed Vive tracker mount; b) Tracker mount attached to the ChArUco pattern; c) Complete calibration pattern (60 cm x 80 cm) with Vive tracker v2 in the upper right corner; d) Real-time camera pose estimation without a pattern can also be performed by a Vive tracker attached directly to an RGB-D sensor; e) Relationships between the global coordinate system, RGB-D sensors, a Vive tracker, and the ChArUco pattern during the calibration phase. Fig. from [Lad+21].*

Off-the-shelf HMDs such as Meta Quest or HTC's devices do not provide reliable full body tracking. Although arms and individual fingers are tracked, the correct position of the hips is missing and legs are usually completely ignored. For better embodiment, it is possible to use external sensors to transfer the missing limb into the virtual space. This section prototypically shows how to calibrate the RGB-D sensors Azure Kinect or Intel RealSense and the Lighthouse tracking system.

RGB-D sensors usually locate their coordinate origin at the position of the depth sensor with the Z-axis pointing outwards as the depth. Lighthouse tracking typically places the origin somewhere in the middle of the floor of the tracking area. The goal is to spatially align the origins as well as find the correct orientation between the rooms.

The bespoken systems lack an inherent interface to spatially unify them. Some kind of visual connection and information transfer must be established between them. Beck and Fröhlich, [BF17] introduced a method for registration between RGB-D sensors and an optical marker-based infrared tracking system. We have incorporated their concept into our system framework and improved its efficiency. While Beck and Fröhlich used a standard chessboard calibration pattern, our approach uses a ChArUco pattern as can be seen in Fig. 4.8. This pattern not only adds robustness, but also speeds up the calibration process. In addition to Beck and Fröhlich, we present the calibration of a novel combination of devices: Azure Kinect and Intel Real Sense using the Lighthouse system.

The standard chessboard pattern used by Beck and Froehlich [BF17] shows a significant decrease in detection performance when the board is not fully captured in the camera frame. However, the ChArUco board remains reliable even with partial occlusion, which is critical for speeding up processing during camera pose estimation. This is made possible by uniquely identifiable visual markers (also called patterns) from the ArUco library. With these ArUco markers, each corner point of the checkerboard pattern can be uniquely identified. This makes it possible to uniquely identify only parts of the pattern and use them for calibration. This is especially important for the calibration of the IR-based depth sensor.

In the course of this thesis, a large number of different samples from different printing companies were tested. It was found that not all printed Charuko patterns reflect infrared light in such a way that the printed pattern is visible to the IR sensors for performing the calibration. However, with a diffuse surface coating, which is offered by several companies, the infrared light is fully reflected and can be used for calibration. However, the printed pattern can sometimes produce bright reflected spots (much stronger than visible light spots), so that parts of the pattern are not detected. This is where the Charuco pattern shows its advantage. Despite only partial areas of the pattern being overexposed due to reflection artifacts, camera calibration is still possible. We have observed this problem on a large scale, especially with the Azure Kinect. However, the Azure SDK can be used to enable a "passive AR" illumination mode, which disables the active IR light source on the sensor and uses only ambient light. In order to illuminate the scene sufficiently, it may be necessary to set up additional light sources, preferably IR light sources, to achieve better calibration results.

We present a two-step method for identifying fixed cameras in the Lighthouse tracking setup. The first step is to calibrate and determine the intrinsic properties of each camera separately in order to accurately determine the position and orientation of the pattern relative to the camera's own coordinates in a second step. We compute the orientation of the camera and translate these measurements to the coordinates of the Lighthouse tracking

system. To facilitate this transformation, a Vive tracker is attached to the ChArUco pattern using a 3D-printed mount, as shown in Fig. 4.8 a–c).

For effective spatial calibration, it is essential to first optically calibrate each sensor to identify its unique intrinsic camera parameters and distortion coefficients. This involves taking multiple RGB images of the ChArUco pattern with each sensor from different angles and distances. From these images we can derive the intrinsic camera data. From our experience we can say that 50 images per sensor are sufficient for calibration.

Our experience has shown that even sensors from the same model series can have different intrinsic parameters. Although some sensors also provide intrinsic and extrinsic data from the factory for RGB and D sensors (which are often mounted in the same housing with an offset of a few centimeters, such as Intel RealSense or the Azure Kinect), this data is often inaccurate. The data provided is sufficient for a quick and initial test setup of the system, but if accurate scans are required, each sensor must be recalibrated individually. We suspect that the internal factory calibration becomes less accurate over time due to temperature changes or forces applied to the sensor, such as during shipping.

Once all the cameras are set up, their exact position and angle in the room must be determined (extrinsic calibration). This is done by presenting the ChArUco pattern to each sensor in the acquisition area again. Additionally, the position and orientation of the pattern is documented with the attached Vive tracker. This allows the transformation matrix of each camera to be calculated as follows:

$$\mathbf{M}_{cam} = \mathbf{M}_{tracker} \cdot \mathbf{M}_{offset} \cdot \mathbf{M}_{pattern}^{-1} \tag{4.1}$$

with the following components, also shown in Fig. 4.8 e):

- $\mathbf{M}_{pattern}$ (red and green line in Fig. 4.8 e): The transformation matrix of the pattern in camera coordinates extracted from the captured image using a 3D-2D point correspondence function and the camera's intrinsic properties.
- $\mathbf{M}_{offset}$ (blue line): The constant offset from tracker to pattern provided by the user.
- $\mathbf{M}_{tracker}$ (purple line): The tracker matrix in global coordinates.

4.4. Evaluation

To test the real-world applicability and additional value of MotionHub in creating and running an application, we created an interactive game called "Human Tetris". This game involves the movements of two players. The procedure is as follows: Player #1 starts with a random body pose. The shadow of this pose is cast on a wall and is fixed when player #1 says he or she is ready. Next, player #2 gets ready and tries to mimic player #1's body pose and shadow as a virtual wall appears and moves toward them. When the wall reaches player #2, the game calculates and displays a score based on how accurately the pose was copied. The game is shown in Fig. 4.9, and a video of the game is uploaded here: `https://youtu.be/O_5hiweZQhE`

Figure 4.9.: *Playing "Human Tetris" with the MotionHub. a) shows the physical set-up with OptiTrack and Azure Kinect. b) shows our evaluation game created in Unity 3D. Figures from [Lad+20a].*

4.4.1. Procedures

We performed three experiments: 1.) a game of Human Tetris, 2.) timing the switch between different BTSs, and 3.) measuring the latency introduced by MotionHub.

For the first experiment, we played two rounds of Human Tetris using two Azure Kinects and an OptiTrack motion tracking setup with 24 cameras (12x Prime 13 and 12x Prime 17W) and a suit containing 50 passive retroreflective markers. We ran one instance each of the game, OptiTrack Motive, and MotionHub, which processed data from

three BTSs simultaneously. Both Azure Kinects and OptiTrack Motive were run on a single PC (Intel i7 6700K, Nvidia GTX 1080), and communication between the BTSs and MotionHub was managed through localhost. In the first round, only one Kinect and the OptiTrack were activated, while the second Kinect was covered with a small blanket to demonstrate the deactivation of the second Kinect for the demo video. At the start of the second round, the OptiTrack player left the tracking area and a second player uncovered the second Kinect and joined the game. At this point, the OptiTrack system was deactivated and both players were tracked by one of the two Kinects. After the test, we collected qualitative feedback from the players through informal interviews.

The second experiment involved measuring the time it took to switch between the OptiTrack and Kinect systems within MotionHub, using our Human Tetris game setup.

In the final experiment, we recorded visual reaction times using a high-speed camera to determine the delay induced by each system. We recorded the time lapse from a physical movement to its detection by each BTS, both with and without MotionHub integration. For the MotionHub tests, visual responses were analyzed in both the MotionHub rendering window and the Unity game engine renderer. MotionHub and the Unity engine ran on a single PC (Intel i7 6700K, Nvidia RTX2080) connected via a 1 Gigabit localhost UDP connection. A 144 Hz refresh rate monitor and a 1000 fps camera (Sony DSC-RX10M4) were used. Both OptiTrack and Kinect updated at 30 Hz (33.3 ms each). Notably, increasing the update rate of OptiTrack reduced the tracking quality of the Kinect due to infrared interference. The software versions used were Motive:Body v2.0.2 and Azure Kinect Body Tracking SDK v1.0.1.

4.4.2. Results

The first experiment described above confirmed the effectiveness of the MotionHub concept. The participants, who were already familiar with both tracking systems, did not observe any irregularities in the system behavior, except for a slightly increased delay with the Azure Kinect.

In the second test, it took 8 seconds to switch from OptiTrack to Azure Kinect within the GUI. In the third test, the measurement results of the induced delay, performed with a high speed camera, are presented in the following tab,4.5.

Table 4.5.: *Delays between physical motion and recognized motion by two specific BTSs (MH stands for MotionHub). Tab. from [Lad+20a].*

System @30 Hz	With MH	Without MH	Induced delay
OptiTrack	127 ms	114 ms	13 ms
Azure Kinect	222 ms	151 ms	71 ms

Previous tests showed that delays were significantly longer when the packet transmission frequency was tied to the MotionHub's main application loop. Reduced delays were achieved by using separate tracker threads that transmit new data as soon as it becomes available, without waiting for the main application loop, which also handles the user interface and user input processing and therefore delays the entire processing pipeline. More details on this approach are provided in the following Sec. 4.2.2.

We speculate that the difference in induced delays between OptiTrack (13 ms) and Azure

Kinect (71 ms) observed in MotionHub may be due to the different refresh rates of their respective SDKs. While OptiTrack Motive and NatNetSDK's thread within MotionHub interact at over 240 Hz (even though the cameras operate at 30 Hz), the Azure Kinect body tracking SDKs communicate at 30 Hz.

We have omitted the delay between MotionHub's internal renderer and Unity's renderer in the Tab. 4.5 because no visible difference could be detected in the high-speed camera images; both renderers had synchronized outputs, thus showing the same delay. However, to assess the network delay between MotionHub and the game engine, we analyzed the timestamps of network packets. Our measurements of the time to send and receive a UDP packet on the same PC (localhost) consistently showed delays of less than 1 ms, even during MotionHub's data transmission.

4.5. Future Work

The concept and functionality of the MotionHub can be extended significantly. One obvious way to extend the concept of MotionHub would be to add additional tracking modules such as face or hand tracking. Many manufacturers and research publications focus on either body, face or hand tracking. With some minor modifications, MotionHub could be extended to track these areas of the body.

Another idea would be to set up a node-based network where different MotionHub instances, each with different BTSs, work together. In this setup, each MotionHub node could receive tracking data, convert it, and send it to a central master node. This master node would then integrate the data into a unified coordinate system and perform sensor fusion. This setup could improve tracking accuracy for individual tracking or enable simultaneous tracking of many individuals. In this way, data acquisition for machine learning could also be realized, as (unified) data from different sensors could be acquired simultaneously within one application. For example, RGB-D data could be acquired simultaneously with high quality tracking such as OptiTrack or xSens. With this high quality data, you would have solid ground truth data to train a tracking algorithm based on the RGB-D. This approach has already been used by Shaikh and Douglas [SC21]. In addition, this functionality could be used to benchmark different systems simultaneously.

As an unexpected benefit, we found that the MotionHub reduces the initial time to prototype an application or game in Unity, since some Unity plug-ins of certain BTSs require special handling of skeletal data and assignments for joints. The MotionHub plugin automatically adapts to any humanoid skeleton in Unity, significantly speeding up the process.

4.6. Conclusions

We have introduced and evaluated MotionHub, our open source platform designed to integrate tracking data from different body tracking systems (BTS) into a unified skeleton structure that can be streamed to client applications such as game engines. In this way, MotionHub is able to transfer NVC more consistently between collaborators. We believe that an open and comprehensive body tracking standard is necessary for efficient remote collaboration and the sustainable development of the so-called "metaverse". Our system demonstrates how this can be realized in an applied system. The software and the evaluation is the answer to research question 4 (RQ4) that was: "How can different

body tracking systems and protocols be standardised to ensure that the representation of nonverbal communication in a telepresence application looks as identical as possible, even with the use of different tracking systems?".

MotionHub adds a delay of 13 ms when using a marker-based optical tracking system (OptiTrack) and 71 ms when using a markerless system (Azure Kinect). The platform allows users to seamlessly switch tracking systems without additional configuration on the receiver side. MotionHub demonstrates the potential for features that extend and unify the capabilities of existing BTSs. In the future, we plan to extend these capabilities by introducing additional automatic calibration processes to align different coordinate systems between BTSs, and by extending MotionHub to support additional systems such as OpenPose [Cao+17], as well as hand and face tracking. We believe, we have provided a valuable tool to the community and academic field.

5. Face-Tracking Head-Mounted Display

A critical component of future interaction in immersive environments will be the capture of the user's facial expressions, providing a rich layer of nonverbal communication. The CTO of Meta (formerly Facebook) still acknowledges in Nov. 2023 that the current high cost of eye and mouth tracking technologies makes them infeasible for mainstream consumer devices [Mix24]. However, it is expected that these capabilities will eventually be built into every device. The current difficulty lies in retrofitting existing models, such as the Meta Quest, with these advanced modules. Competitor HTC already offers aftermarket devices for mouth and eye tracking [HTC24b; HTC24a; HTC24c]. It underscores the need for further research in this area to provide affordable mainstream technology and demonstrates the gradual evolution of MR, VR, and AR technologies where future devices will likely have these capabilities built in from the start. With this in mind, we investigated potential eye and face-tracking technologies and designed our own face-tracking system for an HMD, as shown in Fig. 5.1.

This chapter extends the previous chapters, which focused on body tracking, with the topic of face tracking. For this purpose, established full-face tracking methods are examined to see how they react to partial face occlusion or how they deal with unusual camera angles or focal lengths. It turns out that established full face-tracking solutions cannot be used in an HMD due to the cropped face area, steep angles and fish-eye focal length as shown in Fig. 5.2 and 5.3. Therefore, in this dissertation, a neural network is designed that performs the detection of 36 landmarks of the lower part of the face using a miniature wide-angle camera. The gaze tracking is realized using off-the-shelf eye-tracking hardware.

An additional image processing algorithm is developed to track the movement of the user's eyebrows. Furthermore, a simple solution with pressure sensors in the foam of the HMD (the contact area between the face and the HMD) was developed and evaluated, but was found to be error-prone due to head movements such as nodding.

An alternative approach using optical sensors and image processing was developed to track

Figure 5.1.: *To expand and improve the face tracking capabilities of an off-the-shelf eye-tracking HMD, we added three miniature IR sensors/cameras to it. Two of these sensors are used to track the movement of the eyebrows (right), while the remaining sensor focuses on the lower part of the face (left).*

the eyebrows. For this purpose, additional IR sensors were placed next to the Fresnel lenses in the HMD (see Fig. 5.1 right) to track the user's eyebrow movements. Since our approach uses the off-the-shelf eye-tracking solution of an HMD, special attention must be paid to the LEDs already installed to illuminate the area and the HMD. Typically, they operate with their own pulse-width modulation, which leads to image artifacts with our sensors.

The goal of our efforts is to reliably track facial movements under an HMD, which will later be used for facial reconstruction. The contribution of this chapter is the introduction and evaluation of a system that recognizes 68 facial landmarks of the user in real time under an HMD. Later in this thesis, the user's face will be reconstructed and animated almost photorealistically according to his facial movements.

We started our research at the beginning of 2019, while there were few solutions in this research area. In recent years, there has been a significant increase in research activities in this field, which is certainly also caused by the establishment of mainstream VR hardware. In addition, we would like to point out that even beyond the analysis and transmission of NVC, this technology can improve human-machine interaction by providing an alternative input modality to mouse, keyboard, finger gestures, or controller input.

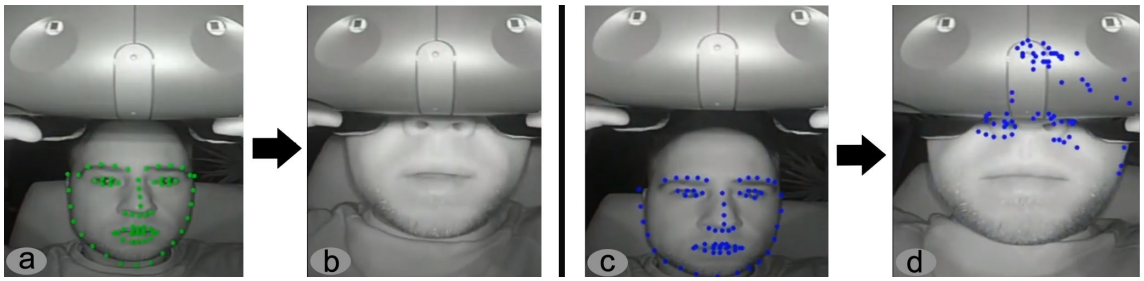


Figure 5.2.: *SOTA methods fail when processing image streams with the upper part of the face covered. The tracking results with green landmarks in a) and b) show the tracking method "ensemble of regression trees" [KS14]. In b) no landmarks are shown because the method fails in the first of two steps, finding a face. The second step would be to determine the landmarks. The blue landmarks in c) and d) show the results of the Facial Alignment Network (FAN) [BT17]. Both methods work as long as a "full face" is shown. However, when only the lower part of the face is shown, both methods fail.*

Figure 5.3.: *Some methods, such as the "ensemble of regression trees" approach by Kazemi and Sullivan [KS14], fail even when a full face is shown, particularly when the camera is too close to the face. The figure shows the same input image on the left and right, but the application of the algorithm yields different tracking results, as shown by the green overlays. This leads to unusable tracking results.*

5.1. Related Work

This section first gives an overview of available and established full-face tracking algorithms. This is followed by a section on specific methods that can also deal with partial occlusion, e.g. occlusion while wearing an HMD. Full-face methods are discussed because partial-face tracking is usually derived from full-face solutions.

Full-face tracking is a well researched topic. In contrast, tracking the user's facial expressions while the face is partially occluded, e.g. with an HMD, is a much less researched topic and is particularly challenging when real-time frame rates are required along with robust tracking quality. Beyond the tracking and transmission of natural communication cues, face tracking is considered a method with as yet unrealized potential for human-computer interaction [JHH05; LHT17; Yam+17]. It can not only receive explicit user input, but could also implicitly provide valuable information for intelligent assistants [Ess00]. For example, it provides data about the user's mood, subconsciously transmitted information, or even ironic speech, for example, as indicated by a small grin while speaking.

A well-known system for classifying facial expressions is the Facial Action Coding System (FACS), developed by Paul Ekman and Wallace V. Friesen in 1978 [EF78]. While it is a comprehensive tool for objectively categorizing the physical expression of emotion through the analysis of facial movements, and is also used in the context of facial animation for special effects in movies, it has relatively little application in academic face-tracking research, especially in the context of applications with neural networks. Facial expressions are much more often expressed in quantitative terms, such as the positions of facial landmarks or the blendshape values of a statistical face model (e.g., a 3DMM), than in FACS parameters. Our presented system also provides position information via landmarks. This section is based on the detailed state of the art report by Zollhöfer et al. [Zol+18] and has been extended with recent research results, especially from the field of deep learning.

5.1.1. Statistical Face Models

Most of the methods mentioned in this section are based on the analysis-by-synthesis approach, which uses some kind of loss function or energy minimization function based on the difference between the original image and a synthesized image. This method is based on the seminal work of Blanz and Vetter [BV99]. It involves rendering a parametric face model and iteratively fitting it to the target image using a loss function (usually L1 or L2 loss) until the (photometric or geometric) error falls below a certain threshold and the optimization loop is terminated. The advantage of using a parametric model is that the high dimensionality of the face alignment problem can be reduced to a limited number of parameters of a system of equations, which transforms it into an optimization problem. Proposed solutions can be categorized into optimization-based methods (e.g. [Thi+15; Thi+18a]) and (deep-)learning-based methods (e.g. [Den+19; Fen+21; ZBT22b; DBB22]). These methods regress parameters of a statistical face model within a linear shape space. Some works first determine identity parameters from a single or a few frames of the person's face with neutral expression, and then estimate expression parameters. Determining the specific identity shape of a person usually leads to better tracking solutions in later steps.

When estimating the parameters of a statistical shape model, determining facial landmarks is free of charge if a landmark embedding is known. These embeddings correspond to

vertices or faces of the model that represent standardized landmark positions such as Multi-PIE [Gro+10]. While many accurate dense alignment solutions are slow, sparse feature detection is much faster and is often used to initialize dense tracking.

Statistical face models are an important part of today's face-tracking algorithms and are also called 3D morphable models (3DMM). These models are generated by non-rigidly deforming a face template to multiple high-quality scans of different subjects with different expressions. These deformations are stored as vertex offsets that are added to an initial generic neutral face representation. These offsets have many names, but are often referred to as blendshapes, morph targets, or shape keys. Each of these offsets is linear and can be weighted independently of other offsets. Optimization algorithms use these vertex offsets to minimize an energy function (in machine learning it is called *loss function* rather than *energy minimization function*), such as the Euclidean distance between a face scan acquired by a depth sensor and the statistical face model.

By integrating these statistical face models, face-tracking methods benefit from a strong inductive bias. In particular, the 3DMM acts as a powerful regularizer that aligns the derived functions. Thus, these models not only provide a fundamental framework for understanding and replicating human facial dynamics, but also ensure that the extrapolations made by face-tracking algorithms are based on realistic and physically plausible facial movements and expressions.

Statistical face models commonly used in academic research include the Basel Face Model [Pay+09] and FLAME [Li+17]. The ICT FaceKit [Li+20] is newer than the previous two models and less referenced in academic work, but offers the most detailed representation with a complete mouth interior with tongue and teeth, as well as detailed eye areas with, for example, geometry to represent tear fluid at the contact points between sclera and skin. All three models are available for research purposes and the FLAME and ICT FaceKit are free for commercial use as well. For more details on 3DMMs, the author refer the reader to Egger et al. [Egg+20].

5.1.2. Sparse Feature Alignment

Sparse feature detection and alignment are important algorithms to provide initial cues for more accurate alignment in later processing steps. A first step in training a face-tracking algorithm is to detect the region of interest in an image, in our case the face. Simple algorithms such as Histogram of Oriented Gradients (HoG) [DT05] or Haar Cascade Classifier [VJ01] can perform this task in a few milliseconds today. HoG is a feature detection algorithm that counts the occurrences of gradient orientation in patches of an image. Haar Cascade Classifiers use image filters (kernels) to detect certain structures in images, and are more robust to variations in face orientation than HoG.

Once the image region containing the face has been detected, further alignment algorithms can provide more information about the face's orientation and expression. Kazemi and Sullivan introduced the "ensemble of regression trees" to faces [KS14] and were able to detect 68 face landmarks, based on the Multi-PIE scheme of Gross et al. [Gro+10], in a few milliseconds.

However, major advances in accuracy and speed for sparse feature matching have been achieved in the last decade by neural networks. As mentioned in the full-body tracking Chap. 4 above, the architecture of the Stacked Hourglass Network [NYD16] also plays a central role in facial tracking. The Facial Alignment Network (FAN) by Bulat and

Tzimiropoulos [BT17] has been used in many research papers for both face detection and more detailed face landmark detection in images. The combination of speed, accuracy, and open source has long established the FAN as an important building block in many face-tracking systems. It uses an advanced architecture of four Stacked Hourglass Networks and has also been trained with the Multi-PIE landmark scheme. A major drawback of the FAN for interactive applications is its speed. On a high-end gaming machine from 2020, the FAN runs at about ten to 15 frames per second. This approach was later extended by Google's Mediapipe face landmark regressor. This solution provides 400 landmarks, higher accuracy (especially around the eyes due to image cropping transformations and cascading nets), and gaze tracking while reducing the overall inference time of the system. MediaPipe's landmark regressor uses a fraction of the computational resources of the FAN with comparable tracking quality.

While the ultimate accuracy of current neural networks is limited by inaccuracies and errors in the available datasets, the authors of "Fake it until you make it" [Woo+21] sought to further increase accuracy by creating a synthetic "perfect" dataset. To do this, the authors developed a pipeline to generate photorealistic images of a variety of different synthetic humans. The humans were generated based on parametric 3D models with different skin colors, hairstyles, facial shapes, ethnicities, ages, and wearing different clothing and jewelry. Based on the availability of the exact positions of individual landmarks of the synthetic images, which are directly related to the polygon mesh of the face, a perfect dataset can be generated. In addition, a large number of images can be easily generated by scaling the entire dataset. With such a dataset, it is possible to significantly improve the accuracy of landmark prediction and to have much less noise in the prediction of similar images. This is especially noticeable when processing video.

5.1.3. Dense Photometric Alignment on RGB Data

Usually, one of the methods from the previous section "Sparse Feature Alignment" is used to roughly fit a (parametric) face model to the input image, in order to have a good starting point for further dense photometric (or also geometric, see next section) optimization algorithms. This process is commonly used by many high-precision face alignment approaches [Gar+13; SKS14; Thi+15; Cao+15; Thi+18a; Thi+18b; Wu+16a; ZBT22b; DBB22]. Several works show the importance of a good initialization, which brings the parametric model as close as possible to the convergence region, leading to shorter processing time and better overall results.

Other methods, such as analysis-by-synthesis or inverse rendering, can then be used in a slower (often offline) process to compute highly accurate results. Good examples in face tracking are MICA and its Metrical Tracker by Zielonka et al. [ZBT22b] and the Video Head Tracker by Grassal et al. [Gra+22]. The pioneering work of Thies et al. [Thi+15; Thi+18a] seems to be one of the best real-time face-tracking solutions on RGB data so far. It is a Gauss-Newton optimizer, highly optimized for data-parallel processing. Unfortunately, the code has never been released to the public.

5.1.4. Dense Geometric Alignment on Depth Data

The Iterative Closest Point (ICP) [RL01] algorithm is critical for fitting statistical shape models in dense geometric alignment tasks within approaches that use depth data streams. ICP is used to resolve depth ambiguities, ensuring that the geometry of the face model is

accurately aligned with the depth data. It often uses point-to-point distance metrics (using simple Euclidean distance) between model faces and the input depth. It can be enhanced with first-order surface approximation (point-to-plane variant) for improved robustness and handling of translational motion. The ICP can be used for rigid and non-rigid alignment. Typically, rigid alignment is used first for transformation and rotation optimization. The non-rigid alignment is then used to optimize the identity shape parameters and also the expression parameters of the statistical face model.

The solutions of Thies et al. [Thi+15] and Weise et al. [Wei+09; Wei+11] are based on the ICP algorithm to determine the dense geometric alignment of a face to a statistical face model. BinaryVR's solution [BinVR19; Upl23] is probably based on this as well. The sensor used by BinaryVR only provides depth data. Therefore, it can be assumed that BinaryVR also uses a variant of the ICP and therefore only uses Dense Geometric Alignment. However, many other solutions use a combination of color and depth data. Apple's current face tracking (in 2022) via ARKit probably also uses the front depth sensor. In general, however, there is a trend that face tracking using RGB data is used more often in academic work than depth data. The practical advantage is that RGB sensors are more common in smartphones or as webcams on laptops than depth sensors. Compared to RGB sensors, depth sensors are more expensive and require more power for depth measurement and often for data processing, which is done in 3D space instead of 2D.

The following part of this section will focus on data processing using neural networks. Depth information is very helpful for our specific use case, but at the time of development (2019-2020) there was a supply bottleneck for suitable depth sensors.

5.1.5. Face Tracking for Mixed Reality Devices

While full-face tracking with off-the-shelf sensors is a well-researched area, it still poses a challenge in MR applications. Available open-source solutions for full-face tracking, such as Dlib [KS14] or the Facial Alignment Network (FAN) [BT17], fail when the upper face is covered. Some hardware manufacturers have announced or released lip-tracking modules for their HMDs, such as HTC for the Vive, HP for its Reverb G2 Omnicept HMD, or in Meta Quest Pro [HTC24b; HTC24a; Mix24], but these are closed-source hardware and software and cannot be easily extended.

Theoretically, it is possible to segment hidden areas of the face and make only the visible areas available to the tracking algorithm. In practice, however, there are several problems: the solutions described in the above sections are often trained on a fully visible face. This means that the neural networks must be trained, often from scratch, on a new data set. This requires adapting the data set, which is not always easy. On the other hand, prior segmentation of the input data, e.g. using deep learning approaches such as BiSeNet [Yu+21a], means additional computational effort. Since low-latency interactivity is a prerequisite in the telepresence scenario, this can lead to computational bottlenecks. Therefore, prior segmentation should be avoided and alternative solutions should be preferred.

Li et al. [Li+15a] did pioneering work in this area. Similar to our methodology introduced in Chap. 5.4, they 3D printed a mount for a sensor. Because the minimum focal length of their sensor was larger than the one we use in Chap. 5.4, their mount was more protruding, appears to be much heavier, and is rather uncomfortable to wear.

Several approaches have been presented in which ordinary spectacle frames are equipped with photosensitive sensors to measure the distance between the face, especially the cheek, and the spectacle frame [Mas+16b; Asa+17; Yam+17]. This approach is only suitable for AR HMDs, such as Microsoft HoloLens. The majority of VR HMDs are placed directly on the user's skin. Common to this group of devices is that facial expression reconstruction provides mediocre to poor results for authentic nonverbal communication. The error rate of misrecognized expressions is high, and a continuous representation or a linear transition between expressions is often not possible. In addition, tracking noise is high and moving the HMD leads to tracking errors.

Olszewski et al. [Ols+16] used a simple RGB camera attached to the bottom of an HMD. They created a plausible mouth animation using a CNN and implemented a direct regression from the images to the blendshape parameters of their 3D face model. This approach requires a significant amount of manual work by a 3D artist and does not provide landmarks without an additional software extension, which means that it cannot be trained continuously.

Thies et al. [Thi+18b] achieve reasonable real-time lip tracking results with good visual reconstruction quality. This work is based on optimizing the expression parameters of a 3DMM presented in [Thi+18a] and [Thi+15]. The tracking pipeline code is not open source. Offline full-face trackers for 3DMMs were presented later, such as the *Metrical Tracker* by Zielonka [ZBT22b] or the *Video Head Tracker* by Grassal et al. [Gra+22]. They could theoretically be extended to optimize only the lower part of the face. For full-face tracking, they provide good results, but are not capable of real-time frame rates. Analyzing a frame with these offline trackers usually takes several seconds.

Lombardi et al. [Lom+18] and Wei et al. [Wei+19] proposed the most advanced systems so far. It is based on a high-quality 3D morphable face model previously acquired with 40 DSLR cameras pointed at the user's face from different directions. Wei et al. [Wei+19] use image-to-image translation based on Generative Adversarial Networks (GAN) to transfer infrared images from cameras of a face-tracking HMD to the style of the rendered avatar. Differentiable rendering methods allow regression of real images to rendered images and enable face tracking. The tracking system is person-specific and does not generalize between faces. Although the tracking quality is superior to other approaches, the system is probably one of the most expensive solutions invented so far. Due to the complexity of the pipeline and the computing power required, these systems can only be used in a laboratory environment.

BinaryVR [BinVR19] was a company that offered lower face tracking for various virtual reality HMDs until 2020. The company was acquired by Epic Games in 2020 and the products are no longer available [Upl23]. The software was closed source, but they offered an SDK with compiled binaries.

The work of Brito and Mitchel [BM19] is closely related to our approach presented in Chap. 5.4. They reprocessed a given facial landmark dataset with an image distortion function to obtain images similar to those provided by introspectively mounted cameras in an HMD. These cameras typically have wide-angle lenses and are positioned close to the user's face. The new dataset was used to train a shape predictor based on the "ensemble of regression trees" (ERT) of Kazemi and Sullivan [KS14]. The difference with our work is that we use a lightweight CNN without using ERT. In addition, we create a person-specific dataset and do not reuse other sources for supervised learning.

5.1.6. Tracking Methods without Optical Sensors

For the sake of completeness, this section also discusses other tracking alternatives that do not use photosensitive sensors. Bernal et al. [Ber+18; Ber+22] use sEMG (surface electromyography), EEG (electroencephalography), EDA (electrodermal activity), and ECG (electrocardiography) sensors. These sensors provide physiological data that can be used for more than face tracking. However, despite their complexity and price, these sensors do not appear to be superior to photosensitive sensors in terms of face-tracking accuracy. There is also the factor of wearer comfort, which is generally inferior because the sensors often require direct skin contact and are poor at wicking sweat. However, off-the-shelf products such as those from Emteq Labs [Emt24] have become established, especially in academic research. Emteq primarily uses sEMG (surface electromyography) to track muscle movement.

Strain gauges, such as those used by Li et al. [Li+15a], have a similar problem with comfort and sweat removal. These sensors were attached to the foam of the HMD and thus represent a relatively large surface on which the user's sweat cannot be dissipated, as the surface of the sensors is largely made of non-breathable plastic film. The advantage of strain gauges is that they are relatively inexpensive. However, measurement errors can also be introduced by head movements, which must be removed from the tracking data in order to achieve more reliable face-tracking results.

In summary, photosensitive sensors provide a lot of data per unit of time for their low cost, while alternatives such as various forms of electrography are more expensive with only mediocre data quality. However, it is worth noting that there has been little research on pressure sensors in the contact area and foam between the skin of the face and the HMD. Therefore, in Sec. 5.5.1, pressure sensors will be investigated for their suitability for face tracking.

5.2. Design Requirements and Rationale

Based on the literature review in this area and the previous development of several prototypes, the following points can be summarized:

1. **Real-time processing and low latency:** The primary goal of our face-tracking system is to operate in real time with minimal latency. This ensures that the system can immediately interpret and respond to user expressions, enhancing the immersive experience and ensuring that users feel truly connected to the virtual environment or another person during a teleconference.
2. **Compactness:** Given the spatial constraints of an HMD, the face-tracking system should be designed to be small so that it does not interfere with the aesthetics or add unnecessary weight, allowing for extended comfortable use.
3. **Cost-effective:** In order to make the face-tracking system accessible to a wider audience, it should preferably be designed with low-cost hardware components where available.
4. **Open source:** It allows researchers, developers, and users to contribute, refine, and distribute face tracking in MR.
5. **Fisheye camera with short focus distance:** To minimize the distance between

the camera and the user's face while maximizing the captured area of the face, a fisheye camera with a short minimum focus distance of less than 10 cm should be used.

6. **Handling steep angles:** Because of the distorted field of view characteristic of a fisheye camera, the tracking algorithm must be able to interpret the steep angles.
7. **Minimum frame rate of 30 Hz:** For fluid face-tracking, our system guarantees a refresh rate of at least 30 Hz. This rate is essential to maintain real-time tracking and ensure that even subtle facial movements are captured without any noticeable lag.
8. **Minimum image noise of the HMC:** Given the potential difficulties of neural networks with noisy images, our system should use cameras with low image noise and good light sensitivity to ensure optimal tracking performance and accuracy.
9. **Infrared capable:** Due to the limited illumination inside the HMD, it is necessary to illuminate the hidden parts of the face with infrared light, which cannot be seen by the human eye. Therefore, the tracking cameras should be equipped with an IR band-pass filter that allows only the wavelength of the IR LEDs used to pass through.
10. **Reporting facial landmarks:** We have found that landmarks are a good input modality, compared to blendshapes, for rendering a photorealistic face using neural rendering (see Chap. 7) and especially using image-to-image/video-to-video translation networks such as the pix2pix GAN by Isola et al. [Iso+17].

5.3. Sensors and Illumination

Cameras are the sensors that provide the richest and most detailed information at relatively low cost and good availability. The majority of related work in the field of face tracking therefore relies on RGB or RGB-D cameras. Unfortunately, very few publications describe which exact models were used with which lenses or filters. Therefore, in this thesis, we will research in advance which type of camera sensor (IR, RGB or D or RGB-D) should be used with which specifications.

5.3.1. Depth Sensors

Face tracking based on a data stream from an RGB-D sensor is well researched and produces reliable results. Most current solutions use a form of the *Iterative Closed Point* algorithm [RL01], as explained in Sec. 5.1.4. However, the choice of sensor for our specific application of face tracking with an HMD is very limited. RGB-D sensors like the Intel Real Sense or Microsoft Kinect are out of the question due to their size, weight and too small closest focusing distance (sometimes much more than 10 cm). At the time of the research in 2019, there was only a small selection of RGB-D or only-D sensors small enough to be considered for our application. The majority of these sensors had a resolution that was much too low (often less than 32×32 pixels) and a temporal sampling rate of less than 10 Hz. The D-sensor that met all our requirements was the PMD PicoFlexx [PMD23], which was not available at the time. We received a PicoFlexx camera from BinaryVR (acquired by Epic Games [Upl23] in 2020), but the firmware was flashed with a BinaryVR-specific version, so we could not access the raw data with the original

camera manufacturer's SDK from PMD. BinaryVR did not provide raw data in their SDK either. Therefore, using RGB-D or D cameras for our application scenario was not possible at this time.

5.3.2. RGB and IR Sensors

As mentioned in the previous chapter, a depth sensor was not available for our specific application at the time of development. Therefore, RGB sensors were used, which could be converted to IR sensors with an appropriate filter. The main advantage of an IR sensor over an RGB sensor is that IR cannot be registered by the human eye - it is therefore invisible. Since there is usually only a small amount of illumination in the space between the HMD and the user's eye area, or even below the HMD, these areas need to be illuminated. If visible light were to illuminate the space between the HMD and the user, it would significantly reduce user comfort.

Another reason for using IR sensors, which also block visible light, is to better control external lighting situations. For example, the displays built into HMDs are optimized to emit mostly visible light (RGB light) and relatively little IR radiation for efficiency. This is helpful when sensors are aimed at the user's eyes, which are illuminated by the HMD displays. Significant changes in the brightness of the displays can change the illumination of the eye area, which would negatively affect tracking performance.

It is often not possible to read relevant parameters, such as the closest focusing distance, from a datasheet because they are usually not specified. Other parameters, such as image quality, are difficult to describe objectively in a datasheet. Therefore, we pre-selected seven sensors based on their data sheets and ordered them for further testing in the lab. The data for the preselection were mainly the resolution, which had to be at least 640×480 pixels, the size, which had to be at most 18×18×20mm (20mm is the height from the PCB to the top of the lens), and the field of view, which had to be at least 160°. In a further step, we subjectively evaluated these sensors for our specific application using 5 criteria on a scale of 1 to 6. The evaluation is based on the German school system. Here, 1 is the best and 6 is the worst. With such a metric, we were able to quantify different features and decide on a sensor. The sensors were installed one after the other in the HMD and evaluated. The table below shows the results in Fig. 5.1:

Camera							
Manufacturer	Caddx	ePath	Kobert-Goods	Kobert-Goods	ePath	Kobert-Goods	jcheng
Name and/or serial number	Turbo EOS2	LYSB00IEXTO	U1-MWD	205IRLWD	EPC_CCT_528	1959	MINIC800W002
Field of view	1 (160°)	5 (<90°)	1 (160°)	2 (150°)	5 (<90°)	2 (150°)	1 (160°)
Closest focusing distance	1 (20mm)	1 (20mm)	1 (20mm)	1 (20mm)	1 (20mm)	4 (40mm)	1 (20mm)
Light sensitivity	1	3	5	2	5	3	4
Size (L,W,H in mm)	3 (14x14x16)	1 (12x12x5)	2 (12x12x10)	2 (12x12x10)	4 (15x15x10)	2 (12x12x10)	4 (15x15x18)
Subjective image quality	1	3	2	3	3	2	1
Frame rate	30Hz	30Hz	30Hz	30Hz	30Hz	30Hz	30Hz

Table 5.1.: *Seven miniature cameras in direct comparison. Since some specifications are often not mentioned in the data sheet, or certain features are difficult to quantify objectively, such as image quality, these seven cameras were installed in the HMD and subjectively rated for the given task based on 5 criteria on a scale of 1 to 6 (based on the German school grading system).*

The best sensor for this task is the Caddx Turbo EOS2 with a 1/3 inch CMOS sensor, a 2.1 mm wide angle lens (160° field of view) and a size of 14x14x16 mm. The weight is 3.72 g, which is negligible as additional weight for the HMD. With a resolution of 720×480 pixels, the sensor achieves a refresh rate of 30 Hz. The sensor transmits the image stream as a CVBS signal (Color, Video, Blanking, and Sync) and can be received by a PC with a Composite-CVBS-to-USB converter. The camera is recognized as a USB Video Class device version 1.5 and can be read by OpenCV via the standard interface. Each camera, including a Composite-CVBS-to-USB converter, is affordable and costs approximately 40 USD in 2019. The complete wiring of a camera with power supply is shown in Fig. 5.4:

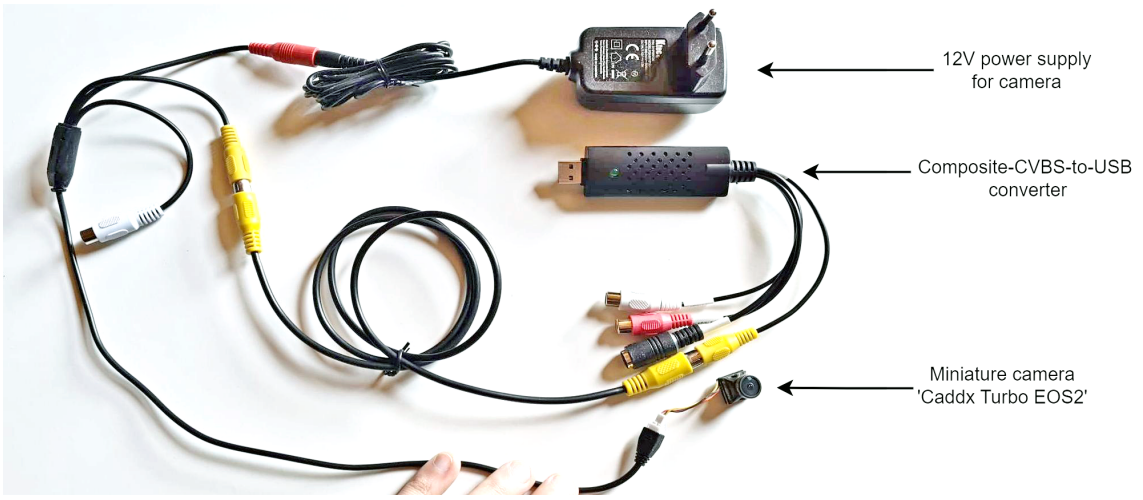


Figure 5.4.: *Wiring of a camera. 12V power supply with a CVBS to USB converter. The image stream is received as a USB video class device on a PC and can be read with OpenCV.*

Possible advantages of choosing more expensive sensors could be the manual control of exposure and ISO speed (light sensitivity) via software and higher frame rates such as 60 Hz or 120 Hz. The sensor we chose supports 30 Hz.

As IR-pass filter a thin polyester filter from Lee Filters (product name "LEE 87") is used. This filter blocks light below 730 nm. To attach the filter, the built-in IR blocking filter can be removed from the lens without damaging the sensor, and the IR filter can be glued in place, as shown in Fig. 5.5 below:

Figure 5.5.: *The RGB filter was removed and a IR filter, that passes light above 730nm, was inserted.*

5.3.3. Illumination Safety Considerations

IR radiation can have several effects on the human body. It can damage the retina and cornea of the eye and cause burns to the skin. The retina is particularly sensitive to IR radiation in the near and mid-infrared spectrum and is a sensitive target for thermal damage because it has no pain receptors and the damage is not immediately apparent, which can lead to irreversible damage if exposure is severe or prolonged. However, the cornea can also be damaged and the formulas and limits for the cornea are also briefly summarized.

During the development of our prototypes, we used Harvatek HT-170IRPJ LEDs. These are 0805-sized SMDs with 140° radiation at 7 mW/sr radiant intensity with a near infrared spectrum at 850 nm, which is almost invisible to the human eye.

There are two guidelines for evaluating the safety of our setup. First, EU Directive 2006/25 [Eur06] focuses on protecting workers in the EU from health risks arising from exposure to artificial optical radiation at work, and requires employers to assess and manage these risks. Second, IEC 62471 is a standard for manufacturers that provides guidelines for the photobiological safety of lamps and lamp systems to ensure they are safe for human eyes and skin. In essence, the EU directive is about protecting workers, while IEC 62471 is about ensuring product safety. Both directives set the same limit of $100\,W/m^2$ for exposure times greater than 1000 s for the risk to the human eye from IR radiation.

In the following two subsections, we summarize the relevant information and equations for our specific scenario of IR exposure as outlined in the two directives. IR exposure to the eye typically lasts as long as the teleconference, which can range from a few minutes to several hours. These guidelines use different equations depending on the length of exposure to the eye. Typically, values greater than 10 seconds and up to 1000 seconds (16.66 seconds) are considered for maximum exposure. Therefore, calculations are always based on the maximum exposure duration. Another critical aspect is the very short distance between the eye and the LED, which is only a few centimeters. The limits, formulas and additional information are derived from the two guidelines mentioned above and two application notes from OSRAM GmbH [JS24; Hal14]. The cases for the cornea and the retina are considered separately as both may be affected by exposure.

5.3.3.1. Corneal exposure limits

The maximum allowable IR exposure to the cornea as a function of time can be calculated for t > 1000 seconds as follows:

$$E_{IR} = \sum_{780}^{3000} E_\lambda * \Delta_\lambda \leq 100 \frac{W}{m^2} \tag{5.1}$$

E_λ is the spectral irradiance per unit area and per unit frequency interval in $\frac{W}{m^2 \cdot nm}$, $\Delta\lambda$ is the wavelength of the emitted light in nm, and t is the exposure time in seconds.

The irradiance can be calculated with the radiation intensity I_{IR} and the squared distance between the radiation source and the target using the formula 5.2:

$$E_{IR} = \frac{I_{IR}}{d^2} \tag{5.2}$$

The corresponding information can be derived from the data sheet of the IR LED.

5.3.3.2. Hazard exposure limits for the retina

The pupil diameter, the size of the emitting source area of the LED, and the wavelength are required to determine the threshold for the retina. Since IR radiation does not exert any visual stimulus such as photopic adaptation or aversive response, the natural protective mechanism of the iris to adapt and protect the eye to ambient light conditions does not function. Therefore, we must assume the maximum value of an open pupil, which is 7 mm.

The next step is to calculate the angular expansion α of the light source. This is done by dividing the length L and width W of the emitting area by 2 times the distance d between the eye and the light source, as described in the following equation.

$$\alpha = \frac{L + W}{2d} \tag{5.3}$$

Depending on the exposure/irradiation time, there are different minimum limits used for the calculation. The upper limit for α_{max} is always 0.1 rad. The following table shows how to calculate the lower limits.

Time range	$\alpha_{\text{min,eff}}$
$t \leq 0.25\,\text{s}$	$0.0017\,\text{rad}$
$0.25\,\text{s} < t < 10\,\text{s}$	$0.0017 \cdot \sqrt{\frac{t}{0.25}}\,\text{rad}$
$t \geq 10\,\text{s}$	$0.011\,\text{rad}$

Table 5.2.: *Limits of the angular subtense α and measurement field of view (FoV) for different time ranges. Source of the table is [JS24]*

The following "Burn Hazard Weighting Function" considers thermal stress as a function of wavelength:

$$R(\lambda) = 10^{\left[\frac{700 - \lambda}{500}\right]} \tag{5.4}$$

Based on these formulas, radiation intensity and retinal hazard can be determined with varying degrees of accuracy using different data sheet specifications. An accurate formula that applies to the near IR range of 780 nm to 1400 nm is the following:

$$L_{IR} = \sum_{\lambda=780}^{1400} L_\lambda \cdot R(\lambda) \cdot \Delta\lambda \leq \frac{6000}{\alpha} \left[\frac{W}{m^2 \cdot sr}\right] \quad (\text{t} > 10\text{s}) \tag{5.5}$$

where L_R is the spectral radiance in W/m2/nm/sr and α is in radians and t is in seconds. However, there are other formulas that simplify the calculation and give good approximations based on the radiant intensity and the size of the emitting area of the LED (l and w). It is advisable to take the maximum rather than the typical radiant intensity from the datasheet to exercise caution:

$$L_{IR} \approx I_{max} \cdot \frac{R(\lambda)}{\left(\frac{l + w}{2}\right)^2} \tag{5.6}$$

5.3.4. Pressure Sensors

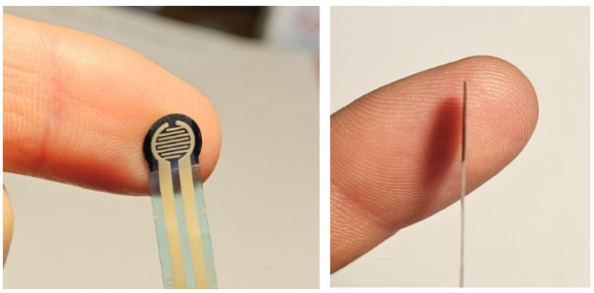


Figure 5.6.: *A pressure sensor RFP-602 that we used in our experiments.*

Pressure sensors in the foam of an HMD seem to be well suited to capture facial expressions in a cost-effective manner. By exerting pressure on the sensors through the movement of facial muscles, it is possible to evaluate differentiable data on specific parts of the face and transfer it to a virtual avatar. We have built several prototypes using thin-film resistive pressure sensors and have found sensors suitable for our application. These sensors are based on the piezoelectric effect. This means that the sensors report a variable voltage proportional to the pressure on the sensor. There is a wide range of pressure sensors available. The sensor model *RFP-602* was preferred to other models because this sensor has a better sensitivity in a range of 10 g - 500 g compared to alternatives that usually have a sensitivity range between 50 g and 2000 g. In our experiments we have found that pressures higher than 250 g rarely occur.

5.4. Lower-face Tracking beneath an HMD

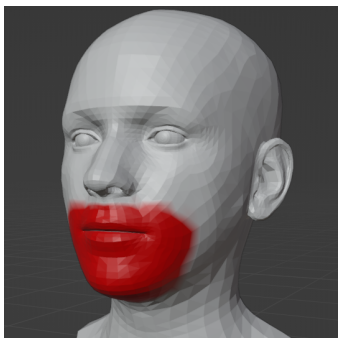


Figure 5.7.: *The red area represents the tracking area covered in this chapter.*

As mentioned earlier in this chapter in the Related Work section, proven face-tracking methods become ineffective when an HMD covers the upper half of the face (see Fig. 5.2 and Fig. 5.3). As mentioned in the introduction of this chapter, there is no special data set of close-up and wide-angle lenses, neural networks, or similar alternatives that allow face tracking of the lower face. The seminal work by Thies et al. [Thi+18a] did not publish any source code. An implementation turned out to be very complex, as the code has to be executed in a highly parallelized and optimized manner on the GPU in order to maintain real-time capabilities. In the meantime (2023), optimization-based face trackers similar to Thies et al. [Thi+18a] have emerged, such as the Video Head Tracker (VHT) by Grassal et al. [Gra+22] and the Metrical Tracker by Zielonka et al. [ZBT22b], but the optimization of a single image takes several seconds. This is too slow for real-time interactive applications. With the FAN [BT17], it turned out that it was not easy to train the network (especially on current hardware) because the training code was based on a machine learning library that was no longer maintained. It was written in Torch, the Lua-based predecessor of PyTorch. Today, only PyTorch is under development, led by Meta (formerly Facebook). Porting or rewriting the code to a newer library turned out to be difficult and buggy during training because of hardware and driver changes. Among other things, the Stacked Hourglass architecture seemed too complex for our use case, with runtimes that were too long.

In the following section, we present a method that can analyze the user's facial expressions

from steep camera angles of the lower face sensor and report them as facial landmarks at more than 900 frames per second. The targeted tracking area is shown in Fig. 5.7. To implement this, we take advantage of freely available "full face" tracking methods to create a labeled dataset of 19 individuals with ground truth facial landmarks. We perform a special case of "transfer learning" by transferring the "knowledge" from a slow but powerful neural network based on a large training dataset to a much smaller network capable of real-time inference. We do not exchange weights and biases, but labeling information. The acquired data is then cropped to the lower part of the face and a traditional convolutional neural network (CNN) with fully connected layers is trained in a supervised learning fashion. Our proposed solution is person-generic, meaning that we do not need any person-specific data to track/infer a new face that is not present in the dataset. We show that our conventional but lightweight CNN is faster in inference than previous full-face-tracking solutions and achieves results comparable to those of SOTA tracking. Our contributions are:

- An end-to-end supervised-learning pipeline for a lower-face-tracking CNN.
- A CNN (with traditional architecture) for the area beneath and HMD with tracking quality comparable to state-of-the-art full face tracking that is suitable for real-time tracking framerates that uses minimal GPU resources to leave resources for other GPU-intensive tasks such as rasterization or network inference.
- To the best of the author's knowledge, the first open source solution for a face-tracking pipeline (data set creation, training and weights) of the lower part of the face targeting VR, AR and MR hardware.
- CAD files for 3D printing the camera mount.

The code for the following section can be downloaded here: `https://github.com/Mirevi/UCP-Framework/tree/main/Lower-Face-CNN`

At the time of development (2019) of this part of the thesis, no system existed that could track only the lower part of the face when the upper part is covered. In the last four years, several commercial systems have become available for purchase, such as HP Reverb G2 Omnicept, Oculus Quest Pro, HTC Vive/Focus Lip Tracker, and Vive XR Elite Full Face Tracker [HTC24b; HTC24a; Mix24; HTC24c]. Please note that these commercial solutions are more expensive compared to our simple monocular approach of about 40 USD in hardware costs.

At the time of development, the only device designed for the special case of face tracking under HMDs was produced by BinaryVR [BinVR19; Upl23]. We received two devices from the company and were able to experiment with them, but due to supply bottlenecks on the part of the company *PMD Sensors*, neither BinaryVR nor PMD were able to send us any more devices. The access to the tracking data of our devices was very limited by the BinaryVR SDK. In addition, the devices used special firmware on the depth sensor that was owned by BinaryVR and was closed source. Due to the lack of access to the raw tracking data as simple 2D landmark coordinates, which we needed for our neural rendering solution presented in Chap. 7, a solution had to be found as it is a basic technology to achieve the goal of this thesis. Therefore, we created our own face-tracking solution. The following section does not present a novel neural network architecture for face tracking, since face tracking is well researched and relies on an established network architecture, but it does explain a novel pipeline for acquiring labeled training data. The network architecture consists of typical encoder-like convolutional layers followed by several fully

connected layers and is based on pioneering work such as LeNet by LeCun et al. [LeC+89] or AlexNet by Krizhevsky, Sutskever, and Hinton [KSH12].

The following Fig. 5.8 shows the process. First, data is collected using an HMC. The data is processed and compiled into a training set. After training, the person's lips can be tracked with a sensor on the HMD, even though the upper half of the face is covered. The last image on the right shows the resulting landmarks. The landmarks are connected by lines.

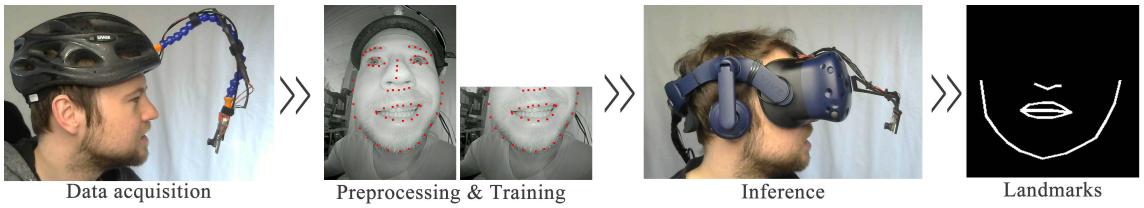


Figure 5.8.: *Pipeline of our approach: In the first step, full-face images of several people are acquired for a training dataset. Next, facial landmarks are detected for the full face and stored as a training dataset with cropped images showing only the lower half of the face. Then, our lightweight CNN is trained on the labeled dataset. After training, the CNN can be used while wearing an HMD. Our CNN provides 36 facial landmarks of the lower face.*

5.4.1. Sensor Mounts and Illumination

To ensure good tracking performance, the sensors must be mounted in the appropriate positions on the HMD. Two mounts have been developed for this purpose. One to acquire the training data set for our CNN (the *training mount*) and another mount that attaches the sensor to the HMD during the inference stage and sends the images to the trained CNN (the *inference mount*).

The training mount consists of a bicycle helmet with a flexible arm, as shown in Fig. 5.9. For illumination purposes, four IR LEDs (Harvatek HT-170IRPJ) were soldered to a circuit board to illuminate the area around the mouth. The camera, which can also be seen in Fig. 5.9, is a modified RGB sensor, as described in Sec. 5.3.2. This setup allows the acquisition of images of the entire face to create a labeled dataset with an existing full-face face tracker solution.

Figure 5.9.: *Data acquisition using the "training mount": A bicycle helmet was modified by adding a mount with 4 infrared LEDs and a modified RGB sensor with an infrared band-pass filter instead of an RGB filter.*

The inference mount consists of three parts, as shown in Fig. 5.10. The parts are held together with miniature screws and nuts (size M1 and M2). The mount is designed so that the distance of the sensor and also the angle at which the sensor faces the user can be manually adjusted by loosening and tightening the corresponding screws. In addition to the sensor, and similar to the training mount, four IR LEDs (Harvatek HT-170IRPJ) were mounted on a circuit board to illuminate the area around the mouth. The mount is significantly smaller and lighter than the mounts of previous work such as that presented by Thies et al. [Thi+18b] and Olszewski et al. [Ols+16], who introduced a similar system, but larger than the mounts currently (writing this in August 2023) available face-tracking systems such as the HTC Vive Lip Tracker. The mounts were modeled in Blender and printed using Fuse Deposition Modeling (FDM) with polylactide (PLA) as the material. The 3D files are available here: `https://github.com/Mirevi/UCP-Framework`

Figure 5.10.: *Top and side view of the 3D prints for attaching the sensors. The coin is for size comparison.*

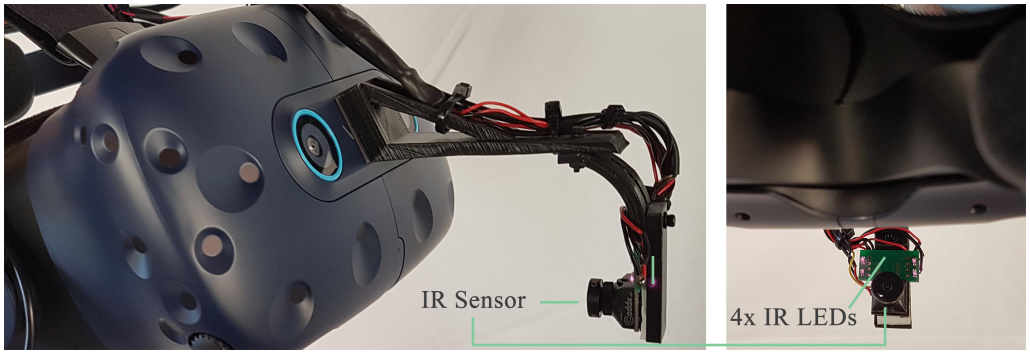


Figure 5.11.: *The assembled and mounted camera mount for the lower face-tracking area. Image from [Lad+20b]. The 3D printing files are available at* `https://github.com/Mirevi/UCP-Framework`

5.4.2. Data Acquisition and Data Set Preperation

The first step in our pipeline is to acquire full-face images of a user and to detect landmarks in those images. To do this, we use the training helmet described in the section above and shown in Fig. 5.9. Our supervised learning approach requires a labeled dataset. The goal is to take advantage of the available (but slow) full-face tracking to trim the acquired images and landmarks to the lower half of the face and thus generate a labeled dataset for training our proposed CNN. For full face tracking, we compared Dlib's [Kin09] method, which is based on "ensemble of regression trees" by Kazemi and Sullivan [KS14], with FAN, which is based on a CNN by Bulat and Tzimiropoulos [BT17]. The challenge in our setting is the small distance between the camera and the face, which results in a difficult viewing angle for the face-tracking solutions. We found that undistorted images from our fisheye camera tended to give better results. Therefore, we calibrated the wide-angle camera using Zhang's method [Zha00] to determine the intrinsic and undistorted images before passing them to the face trackers. The calibration pattern is a 6x9 chessboard with an edge length of 10 mm. The pattern is small so that the sensors can be calibrated at the target working distance of about 10 cm.

Dlib's method fails with our miniature camera when processing images from distances closer than about 40 cm, as shown in Fig. 5.3. FAN fails at a distance of less than 5 cm because parts of the face are cut off at the image boundaries. However, it can still deliver reasonable results from distances greater than 8 cm. For this purpose, our final helmet mounts have a camera-to-face distance of 10 cm and we created the final dataset with the labeling information from the FAN.

For the acquisition process of the training data set, we use the training mount. The acquisition process involves recording a person's face for about 5 min at 30 fps, resulting in a dataset of about 9000 unique images. We recorded 19 individuals (six women, 13 men) while they were speaking and making grimaces, as shown in Fig. 5.12. During the recording, we didn't have a strict protocol, but we talked to the person to record natural mouth movements and asked the person to make at least 13 grimaces that we had predefined. Pictures of these grimaces were shown and the participants were asked to imitate these expressions. This resulted in a total of 171000 unique facial images. Furthermore, we applied data argumentation to increase the dataset to avoid overfitting to 342000 face images. Each of the originally captured images was flipped horizontally (i.e. from left to

Figure 5.12.: *Four images of different people from the training dataset of 19 individuals. The images have different camera positions and lighting parameters.*

right flipped, not top to bottom flipped), doubling the number of images in the dataset. In addition, we randomly cropped, rotated, or scaled each of the 9000 original images, resulting in 27000 images per person and 513000 images for the entire dataset.

After acquiring the full face images, we detect the facial landmarks with the FAN [BT17] and store them with the corresponding images. After detecting the landmarks, we compute the mouth bounding box based on the minimum and maximum coordinates of the landmarks of the lower face over the whole dataset. Then, we crop the images and the previously detected landmarks in the acquired dataset to this bounding box and store these new cropped images as well as the corresponding landmarks as a labeled training dataset for our CNN. These steps are visualized in Fig. 5.13 below:

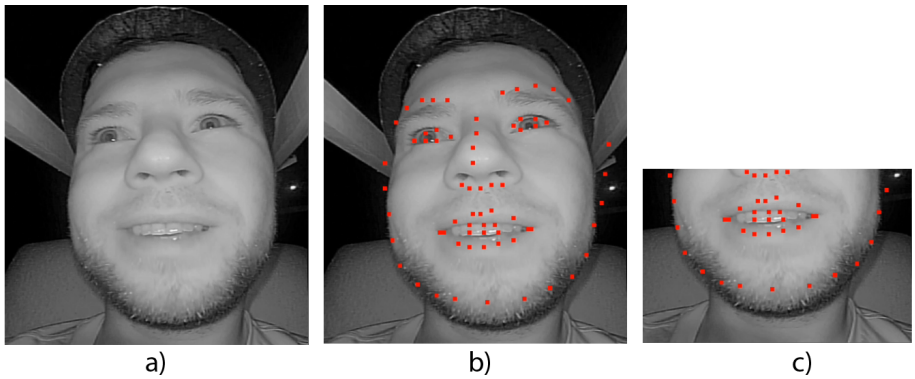


Figure 5.13.: *Building the training set for the lower face-tracking CNN. a) Shows the uncovered face captured from the same angle as the one in the face-tracking HMD. b) The Facial Alignment Network [BT17] determines 68 landmarks based on the MultiePIE scheme. c) The upper face is cropped and the remaining image area and landmarks are used as the labeled training set for the lower face-tracking CNN. Image from [Lad+20b].*

5.4.3. Neural Network Training and Architecture

Our primary goal was to create a CNN capable of delivering solid real-time frame rates on ordinary gaming hardware, while leaving enough GPU resources for rasterization and

inference for other networks, such as generating photorealistic images of the user's face in an immersive telepresence scenario, as described in Chap. 7 and 8. As an architecture for our CNN, we considered the Stacked Hour Glass Architecture by Newell, Yang, and Deng [NYD16], which has been used in several body tracking and face alignment works such as the FAN [BT17]. However, our experiments showed rather unsatisfactory frame rates and results. The architecture is too deep, requires too many network parameters, and needs an additional step at the end of the forward pass to analyze the generated heatmaps, as mentioned by Guo et al. [Guo+20]. Furthermore, when we reduced the network depth, we experienced an unstable training process without satisfactory convergence after several epochs. It also often resulted in heatmaps with high image noise. Lowering the learning rate reduced this effect to some extent, but it still occurs randomly. Therefore, we chose a simpler, more traditional, and well-documented architecture that is faster than FAN and also provides reasonable tracking quality. Furthermore, the classical architecture inspired by LeNet [LeC+89] or AlexNet [KSH12] has already been successfully applied to face landmark detection as shown by Wu et al. [WY17].

The output of the CNN is a set of 36 tuples of landmarks of image coordinates. The network has a total of 1.982 million parameters. We trained the network for over 15 epochs with a batch size of 8 using the Adam optimizer and a learning rate of 0.001. Grid search is also used for hyperparameter tuning [BB12]. Training and validation are split in a ratio of 70 to 30. To speed up training and inference, we implemented Mixed Precision Training by Nvidia APEX, a PyTorch extension [NVI24]. Furthermore, for inference, the trained model was converted into a traced module using TorchScript, which further speeds up the inference pass. Each convolutional layer is batch-normalized.

We used the Normalized Mean Error (NME) as metric and loss function. For face alignment, a reasonable metric is crucial and helps to quantify the tracking quality. The metric for face alignment is usually the point-to-point Euclidean distance normalized by the interocular distance[CC06; Sag+13; She+15]. In our case, we cannot rely on the interocular distance because we do not have eye landmarks. Therefore, we normalize by the bounding box of the lower half of the face. A similar approach based on a bounding box for the whole face was also used by Bulat and Tzimiropoulos [BT17]. In particular, we used the Normalized Mean Error (NME), defined as

$$\text{NME} = \frac{1}{N} \sum_{i=1}^{N} \frac{||x_i - y_i||_2}{d} \tag{5.7}$$

where x denotes the ground truth 2D landmarks for a given face, y the corresponding prediction, and d is the square root of the ground-truth bounding box, computed as $d = \sqrt{width_{bbox} * height_{bbox}}$.

Type	Size	Structure	Output
Conv	5x5	BatchNorm, ReLu, MaxPool (2x2)	16
Conv	5x5	BatchNorm, ReLu, MaxPool (2x2)	32
Conv	3x3	BatchNorm, ReLu, MaxPool (2x2)	64
Conv	3x3	BatchNorm, ReLu, MaxPool (2x2)	128
Conv	3x3	BatchNorm, ReLu, MaxPool (2x2)	256
Conv	2x2	BatchNorm, ReLu	512
Lin	-	ReLu	500
Lin	-	ReLu	500
Lin	-	Sigmoid	72

Table 5.3.: *Architecture of the proposed CNN. The input to the network is a single-channel image of size 156 px × 204 px. The output of the network is 72 values representing the x and y coordinates of 36 2D face landmarks.*

5.4.4. Evaluation

We quantitatively compared our solution with the FAN by Bulat and Tzimiropoulos [BT17] using the NME metric, as shown in Fig. 5.14. Our proposed CNN achieves a tracking error of 1.98% compared to FAN on a 1.5 min image sequence captured by our face-tracking HMD.

Note that we show our unfiltered results without positional smoothing. Brito and Mitchel [BM19] used a Kalman filter for noise reduction of the detected landmark positions. Since they did not provide the source code or the trained model, we were not able to directly compare the two solutions.

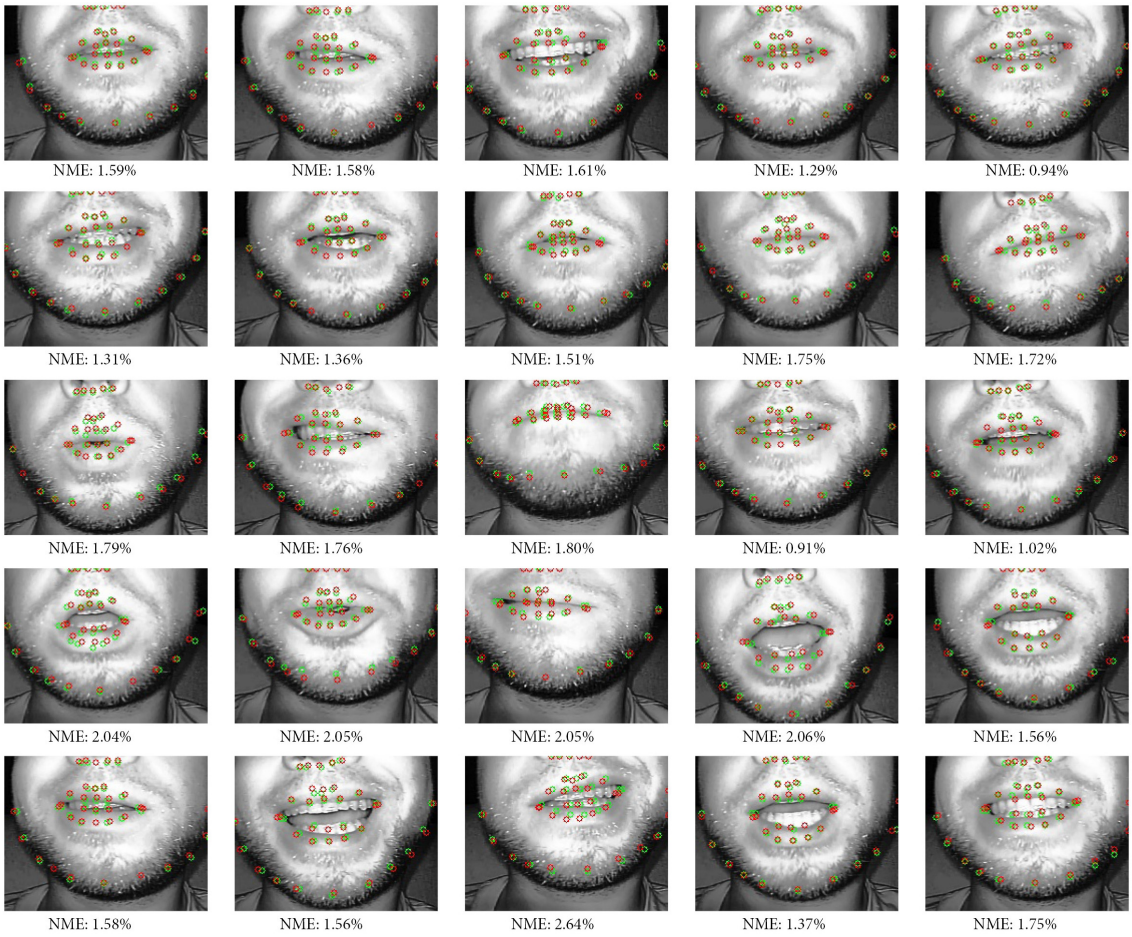


Figure 5.14.: *Results: The images show a comparison of our proposed solution with the FAN [BT17]. The green landmarks are the ground truth data of the full face detected by FAN. The red ones are predicted landmarks of our proposed CNN for the lower half of the face. The images are from the test set. Therefore, the network has not seen these images during training.*

5.4.4.1. Time Measurements

For inference in our experiments, we used PyTorch 1.7.1 on an Intel i7-4790K, 16GB RAM, NVidia RTX 2080Ti with 11GB VRAM, Windows 10 x64 (Build 19042.867), and Nvidia driver 461.09. The network was trained in ∼410 min (almost 7 hours) on a dataset of 513000 images, which we acquired through 95min (19 individuals × 5 min) of recording with the camera in the helmet mount (Fig.5.9). On average, the execution of a forward pass of our network with cropping of the input image to a smaller and predefined area (depending on the position of the camera mount and can be adjusted manually) takes 1.107 ms (903 fps) on the GPU. However, the final frame rate is limited by the native frame rate of the IR sensor from Sec.5.3.2, which is only 30 fps, meaning that we only use 33.21 ms per second for our face-tracking solution.

Compared to the work of Brito and Mitchel [BM19], our solution is more than 29 times faster. Their mouth landmark detection takes 33 ms (30.3 fps) on hardware that is 4 years older. They used an NVIDIA 4G Quadro K2200 with an Intel Xeon CPU E5-2630 v4, but it is not clear whether their application runs on the GPU or the CPU. Our approach is

also faster than FAN. The latter's full-face tracking runs at 71.4 ms per frame (14 fps) on the test hardware, which is 64.5 times faster.

5.4.4.2. Limitations

We observed a degradation of tracking quality with expressive facial play, face movement close to the edge of the camera field of view, and motion blur, as shown in Fig. 5.15. We believe that the occurrence of motion blur can be easily avoided by using brighter IR LEDs and a better camera with shorter exposure time. Another limitation is the fact that the CNN is not able to track the tongue because the ground truth data does not contain the necessary information.

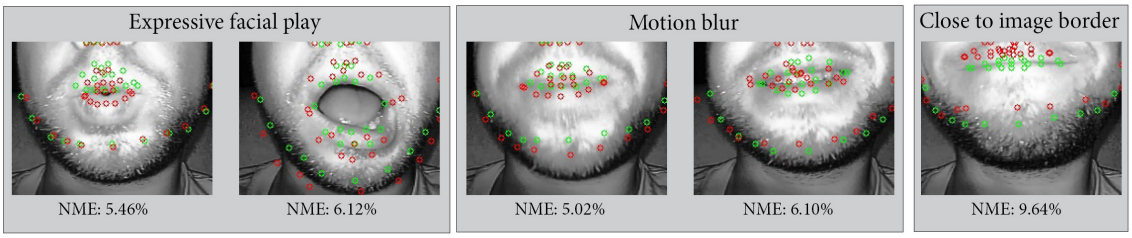


Figure 5.15.: *Limitations: Limitations of our CNN include expressive facial play (first and second images from left). Furthermore, we observed a degradation of tracking quality with motion blur during fast changing expressions (third and fourth image) and when the camera is shifted (far right).*

5.4.5. Discussion und Future Work

As mentioned above, we also experimented with the Stacked Hourglass architecture (encoder-decoder architecture without fully connected layers), but could not find an acceptable trade-off between speed and accuracy. The FAN is too slow and resource intensive for our application and we tried to reduce the stacks in the FAN from 4 to 1 and 2 stacks. As I write this text 4 years later in 2023, I believe that the HourGlass architecture would have yielded a better result if we had done extensive hyperparameter tuning with more appropriate initialization of the network weights. Unfortunately, at the time we were trying to solve the problem in 2019, there was little documentation and few code repositories and best practices for training Stacked Hourglasses, as they had only recently been invented [NYD16]. In contrast, the classic architectures, which are over 30 years old, such as the LeNet [LeC+89], were very well documented. Furthermore, the classical architecture has already been successfully applied to the face landmark regression problem, as demonstrated by Wu et al. [WY17].

We believe that a combination of a Stacked Hourglass architecture and the positional encoding introduced by Vaswani et al. with the Transformer networks [Vas+17] could outperform our current implementation. Current facial landmark prediction research relies on it, and researchers report superior performance over SOTA [Wat+22]. In 2021, position encoding was also one of the key technologies for the invention of Neural Radiance Fields (NERF) [Mil+21]. From our own experiments with coordinate-based networks, such as NeRFs [Mil+21] or creating the FF2EXP-Net in Sec. 8.2.4, we can confirm that positional encoding makes the networks leaner by giving better results during testing time. Furthermore, although current papers on face landmark detection recommend a multi-stage

approach (cascading networks), such as AttentionMesh [Gri+20] (also known as MediaPipe Face Mesh[Lug+19]) or Lv et al. [Lv+17]. We also have the limitation as shown in Fig. 5.15, but we do not recommend this in our specific application because the image area to be analyzed remains relatively the same due to the mount and does not require any further adjustment or cropping. Rather, we recommend a fast method for rough alignment and cropping of the lip area due to minimal sliding of the HMD on the user's face. Simple and fast methods that can perform this task well are Histogram of Oriented Gradients (HOG) [DT05] or Haar-like features by Viola and Jones [VJ01].

We would like to draw the reader's attention to the work of Grishchenko et al. [Gri+20]. This work was done and published after the completion of our solution. Grishchenko et al.'s solution is optimized for fast inference and achieves real-time frame rates for landmark regression on full faces even on mobile GPUs. Unfortunately, they do not present detailed information about the architecture and what kind of layers are used. MediaPipe's landmark regression is one of the most accurate tracking methods available today (January 2024) and is often used as a basis for face alignment problems.

5.5. Eyebrow Tracking

Some of today's HMDs come with eye tracking out of the box or can be upgraded with hardware to make it possible. In addition to the direction of gaze, some systems can also detect the degree to which the eye is open, but few systems can also detect the position of the eyebrows [BMF24]. The following section presents two solutions that can track eyebrows under an HMD. One approach uses pressure sensors and the other uses optical sensors.

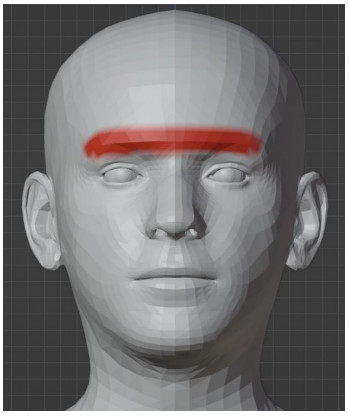


Figure 5.16.: *The red area represents the tracking area covered in this chapter.*

5.5.1. With Pressure Sensors

Related work has shown that little research has been done on using pressure sensors in the contact area between the user's face and the HMD. As mentioned in Sec. 5.3.4, we found a suitable sensor for our application. To read the sensor data, which is reported as voltage changes, the sensors were connected to the general purpose input/output (GPIO) pins of an ESP8266 microcontroller. A script running on the microcontroller sends the measurement data via UDP over Wifi to a computer, which converts the measurement data into corresponding facial expressions.

Three sensors were added above the eyebrows in the foam of the HMD as shown in Fig. 5.18. Due to their narrow design, the thin-film pressure sensors were not able to detect pressure through the foam without further modification. As shown in Fig. 5.19, additional adhesive rubber studs were attached to both sides of the sensor to significantly increase the pressure on the sensors. As a result, reasonable data can be generated and the sensors do not adversely affect user comfort by causing painful pressure points on the forehead.

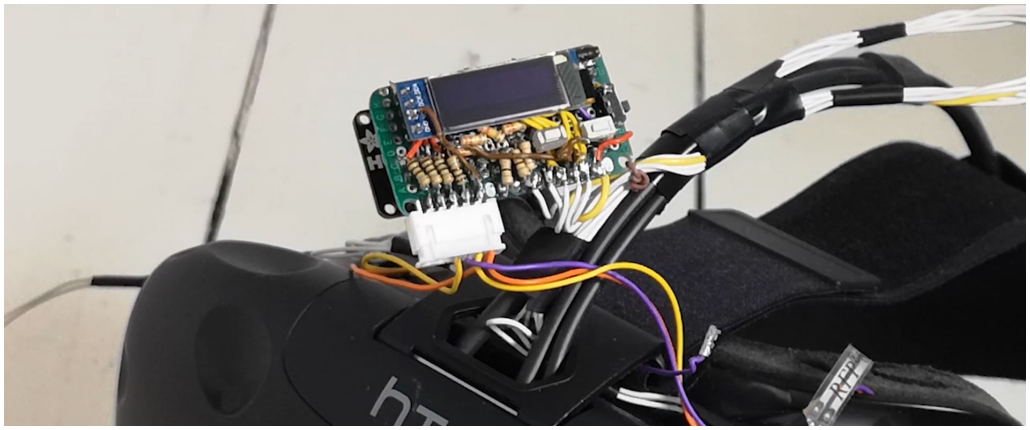


Figure 5.17.: *A ESP8266 microcontroller read out the pressure sensor data in the HMD's foam in order to control an avatar. Image credits by Juan Schupp.*

Figure 5.18.: *Position of the pressure sensors in the preliminary study. Slits were cut into the foam of the HMD frame and three sensors were glued into them.*

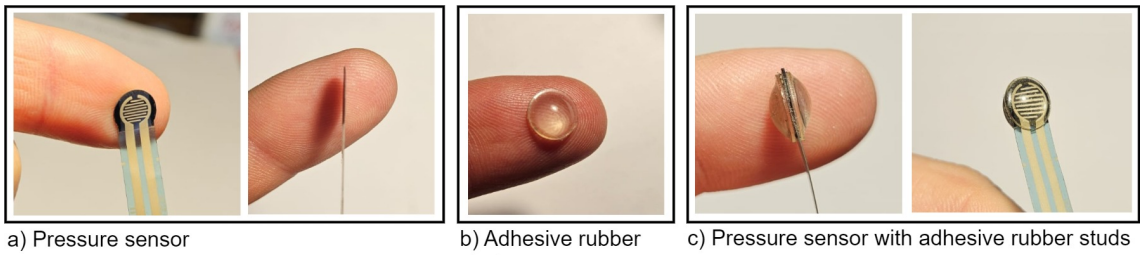


Figure 5.19.: *The RF602 pressure sensor is thin and does not provide reasonable data in the foam of the HMD without rubber studs. a) shows the sensors without modification. b) shows the rubber studs. They have a diameter of 8 mm. c) shows the sensors with two rubber pads on each side.*

Because each person has a different facial geometry, the sensors experience different pressure with different facial expressions, depending on the shape of the face. In addition, each person has a different preference for how tightly they want to wear the HMD. This means that the initial pressure ratios for a neutral facial expression are always different for different people. To get reasonable tracking results, the three sensors must be calibrated in three

simple steps. Each step consists of making an expression and recording the corresponding pressure values. The recorded expressions were 1.) a neutral expression, 2.) a full raised eyebrow (frown) and 3.) an angry expression (full lowered eyebrows). Fig. 5.20 shows the expressions as well as the corresponding tracking data of the sensors. For visualization purposes, we use only the frame of the HMD without the attached displays.

In a preliminary study, we linked the data from the pressure sensors to the corresponding blendshapes of a virtual avatar. To create an avatar, we used the FaceGen SDK [FaceGen24]. If the person does not move and only repeats the recorded expression, the tracking data is generally very reliable. However, there were three reasons against further use.

1. Even slight head movements, such as a slight nod, influenced the tracking data. This means that the eyebrows showed different expressions without any actual movement of the user's brows. It might be possible to filter out such erroneous data using the HMD's tracking data, but the following additional disadvantages meant that we did not pursue the entire pressure sensor approach any further.
2. expressions, such as a laugh, also moved the brows. Laughing raises the cheeks and creates higher pressure on the sensors in the forehead area. In practice, this meant that the eyebrows always moved upwards when people laughed.
3. Another decisive reason against the pressure sensors is the susceptibility of the system to a rapidly occurring lack of calibration of the system. Because the HMD can slip during use and the user may have to reposition the HMD, the pressure ratios for the previously recorded expressions also change. This situation quickly leads to strong facial expressions on the avatar that the user does not intend. The uncanny valley effect quickly occurs here.

Another reason for eliminating the system was the sensor noise. We used the unfiltered sensor data and used the on-chip analog-to-digital converter (ADC) of the ESP8266. Using this device in combination with the RFP-602 thin-film sensors, we were able to determine a noise of +- 15 g. This leads to a slight twitching of the eyebrows, which could be easily reduced by using appropriate filters such as 1€-Filter [CRV12]. In summary, after this preliminary study, we did not pursue pressure sensors any further due to their susceptibility to tracking errors, which quickly lead to uncanny facial animation.

```
S1 : 0 || S2 : 245 || S3 : 0
S1 : 0 || S2 : 244 || S3 : 0
S1 : 0 || S2 : 244 || S3 : 0
S1 : 0 || S2 : 243 || S3 : 0
S1 : 0 || S2 : 239 || S3 : 0
S1 : 0 || S2 : 242 || S3 : 0
S1 : 0 || S2 : 244 || S3 : 0
S1 : 0 || S2 : 242 || S3 : 0
S1 : 0 || S2 : 238 || S3 : 0
S1 : 0 || S2 : 239 || S3 : 0
S1 : 0 || S2 : 240 || S3 : 0
```

a.) Neutral expression.

```
S1 : 22 || S2 : 389 || S3 : 50
S1 : 40 || S2 : 424 || S3 : 50
S1 : 29 || S2 : 437 || S3 : 56
S1 : 29 || S2 : 437 || S3 : 39
S1 : 28 || S2 : 429 || S3 : 39
S1 : 27 || S2 : 435 || S3 : 42
S1 : 27 || S2 : 436 || S3 : 38
S1 : 27 || S2 : 434 || S3 : 37
S1 : 26 || S2 : 429 || S3 : 37
S1 : 27 || S2 : 423 || S3 : 41
```

b.) Rising the eyebrows.

```
S1 : 0 || S2 : 50 || S3 : 0
S1 : 0 || S2 : 49 || S3 : 0
S1 : 0 || S2 : 49 || S3 : 0
S1 : 0 || S2 : 56 || S3 : 0
S1 : 0 || S2 : 58 || S3 : 0
S1 : 0 || S2 : 67 || S3 : 0
S1 : 0 || S2 : 68 || S3 : 0
S1 : 0 || S2 : 68 || S3 : 0
S1 : 0 || S2 : 68 || S3 : 0
```

c.) Lowering the eyebrows.

Figure 5.20.: *Three sensors were embedded in the foam of the HMD frame to detect eyebrow movements. The actual HMD was removed for visualization. S1: sensor above right eyebrow, S2: centered between both eyebrows, S3: left eyebrow. The values of the sensors are read out with a resolution of 1024 steps, while the maximum pressure represents a pressure above 500 g. Subfigure a.) shows a neutral facial expression. S1 and S3 show no pressure, while S2 reports a pressure of about 120g - 130 g. Subfigure b.) shows raised eyebrows. The middle sensor shows almost double the pressure. The two lateral sensors show only a few grams of pressure. Subfigure c.) shows lowered eyebrows. The foam above the eyebrows is significantly relieved. The center sensor shows about 20-30 g of pressure, while the two side sensors show no pressure. Image credits by Juan Schupp.*

Figure 5.21.: *In addition to eyebrow tracking, the system can track other expressions, such as a grin, with two sensors inserted into the foam on the cheeks. When grinning, the person's cheeks exerted a clearly measurable pressure on the sensors in the lower part of the foam. The pressure sensor approach was not pursued further because it was too susceptible to tracking errors during head movements. Image credits by Juan Schupp.*

5.5.2. With Optical Sensors

Although pressure sensors are less expensive than RGB sensors, optical sensors provide significantly more information for analysis. As a result, more reliable tracking data can be generated. Today's SOTA image processing for face tracking is largely based on CNNs. In our specific application case of telepresence, we assume that a computer has to compute the entire face tracking, the photorealistic rendering as well as the representation of the virtual environment. We deliberately decided not to use another CNN for eyebrow tracking in order to save computational resources, since there are already other heavy tasks to be performed. Due to the relatively static image area of the eyebrows in the HMD and the usually strong contrasts and clear boundaries between eyebrows and skin, we rely on classical image processing algorithms, which have significantly lower computational costs. The images we obtain with our system from inside the HMD are of high quality and well suited for image processing, as can be seen in the following Fig. 5.22:

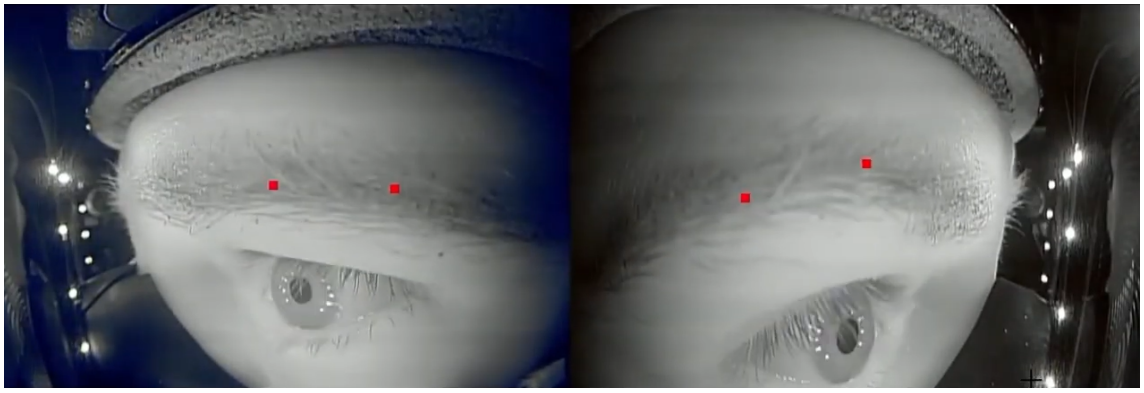


Figure 5.22.: *The goal of this section is to develop an optical tracking solution for eyebrows. This image shows the final results of our eyebrow tracking setup with four landmarks in red.*

5.5.2.1. Illumination and Sensor Attachement

Our goal is to add eyebrow tracking to current eye-tracking systems. In all current eye tracking HMDs, the eye area is already illuminated by several infrared LEDs for eye tracking. Cameras behind the Fresnel lenses use this illumination for eye tracking. In our particular case, we use the HTC Vive Eye Pro, which uses LEDs with a wavelength of 850 nm. These LEDs are arranged in a ring around the Fresnel lenses. The brightness of these LEDs is adjusted by the manufacturer using pulse width modulation (PWM) and synchronized with the eye-tracking cameras behind the Fresnel lenses. Our goal was to place two additional sensors in the cavity between the HMD and the face to perform eyebrow tracking. Unfortunately, we were unable to synchronize the gaze tracking sync signal with our eyebrow tracking sensors, resulting in image artifacts during recording due to the pulse-width modulation of the gaze tracking LEDs. Due to these artifacts, robust eyebrow tracking would not be possible as the individual images were often exposed very differently.

To solve this problem, a first prototype was developed in the form of a PCB ring with several IR LEDs soldered to it, as shown in Fig. 5.23. The IR LEDs (Harvatek HT-170IRPJ) have the same wavelength of 850 nm as the original HTC Vive gaze tracking LEDs. These rings were placed over the original LEDs, blocking the original light source as shown in Fig. 5.24. We did not use pulse width modulation and controlled the brightness of the LEDs by voltage. This solution works well for the gaze tracking of the HTC Vive Eye Pro as well as our eyebrow tracking.

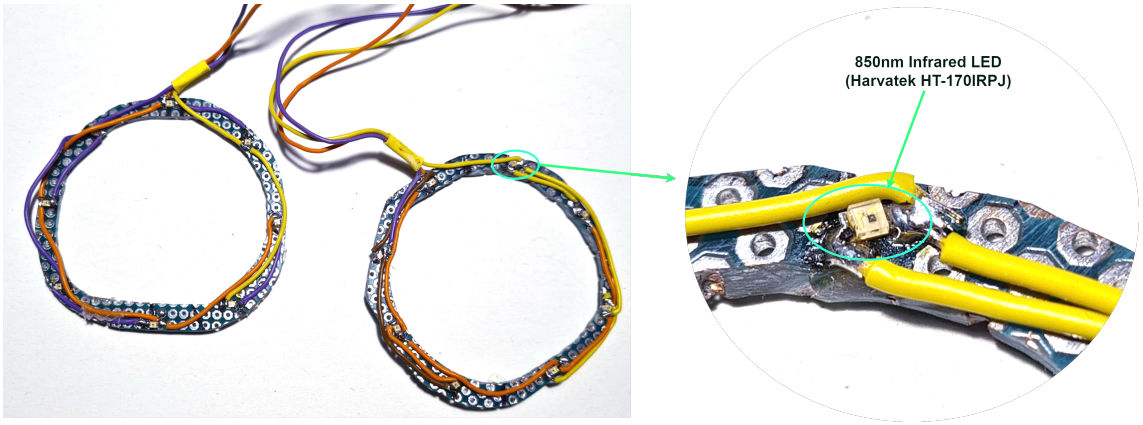


Figure 5.23.: *First prototype of the rings for illumination of the eye area. The ring is equipped with eight SMD IR LEDs.*

Figure 5.24.: *First IR-LED-ring prototype built into the HMD. HMD is a off-the-shelf HTC Vive v1. Image credits by Juan Schupp.*

It is important to note that infrared light, which is invisible to humans, can also cause permanent damage to the eyes or cornea. In Sec. 5.3.3 "Safety Considerations Regarding Illumination" details the limits to be observed and how to calculate them. In the EU Directive 2006/25 [Eur06] limit values can be derived. For our LED with a wavelength of $\lambda = 850\,nm$, we can take the following limit for the continuous total irradiance in the infrared wavelength range from $\lambda = 780\,nm$ to $3000\,nm$: $E_{IR} = 100 W/m^2$. From the LED data sheet we take the radiant intensity, which is $I_{eff} = 6\,mW/sr$. The distance between the LED and the eye in the HMD is about 25mm. E_{IR} can be calculated from $E_{IR} = I_{eff}/d^2$, where d is the distance between the LED and the eye. In our case, $E_{IR} = 9.6\,W/m^2$. There are eight LEDs installed per eye, which simply added together gives $E_{IR} = 76.8\,W/m^2$, which is below the limit of $E_{IR} = 100 W/m^2$. [Hal14]

With the first prototype of the PCB rings, we were able to ensure that our approach worked in principle, but it was prone to loose contacts and cables in the user's field of view in front of the Fresnel lenses. As a result, a new PCB design was created as shown in Fig. 5.25. These PCBs were professionally manufactured and soldered by the company *jlcpcb.com*. In an automated process the PCBs were cut and soldered with SMD resistors (20 Ohm and SMD package size of 0402) and eight Harvatek HT-170IRPJ IR-LEDs. The production of 20 PCBs costs about 30 Euro.

Figure 5.25.: *Final IR LED ring prototype. PCB design by Bernhard "Earny" Wohlmacher.*

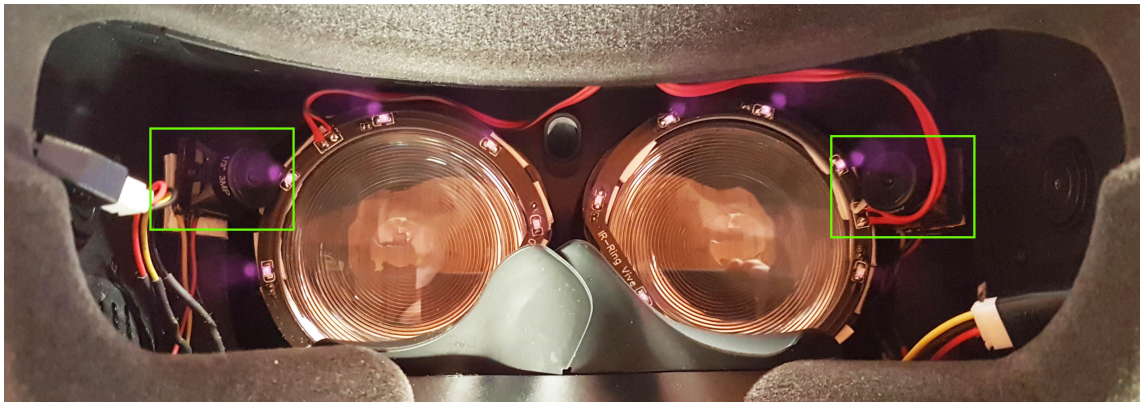


Figure 5.26.: *The second prototype of the IR-LED ring circuit boards is mounted around the Fresnel lenses. For eyebrow tracking, two cameras were mounted inside the HMD, as highlighted by the green frames.*

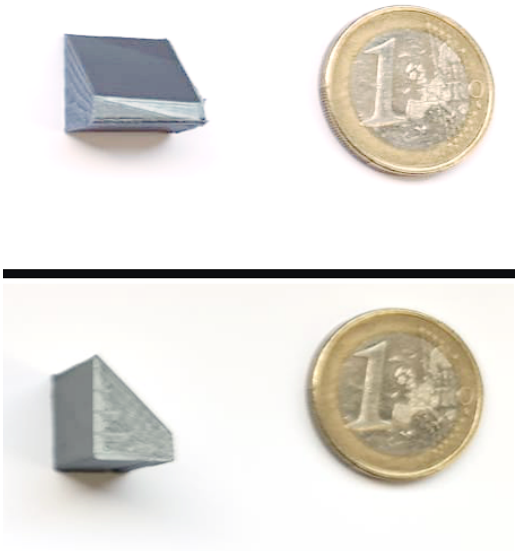


Figure 5.27.: *Top and side view of the wedge for mounting the eyebrow tracking sensor inside the HMD next to the Fresnel lenses as shown in Fig. 5.26. The coin is used for size comparison.*

The first mount is a simple wedge attached to the inside of the HMD with double-sided tape (next to the coin in Fig. 5.10), as shown on the top right and left in Fig. 5.26.

5.5.2.2. Tracking Algorithm

To get details about the positions of the eyebrows, we convert the HMC streams (the sensors in the green frames in Fig. 5.26 above) into a binary image format. Setting a grayscale threshold is necessary to get images where the eyebrows are clearly contrasted with the skin. This means, for example, that darker areas in the grayscale image become black (pixels are set to a value of 0) and lighter areas become completely white (pixels are set to a value of 1 for white). In this way, we can detect the transition between skin and eyebrow in the binary image to track the relative position of the eyebrow, as shown in Fig. 5.28. In this Fig. two vertical bars along the Y-coordinate of the binary eyebrow image can be seen. Within these bars, the average position of all white pixels is calculated and represents the eyebrow position, shown as red dots.

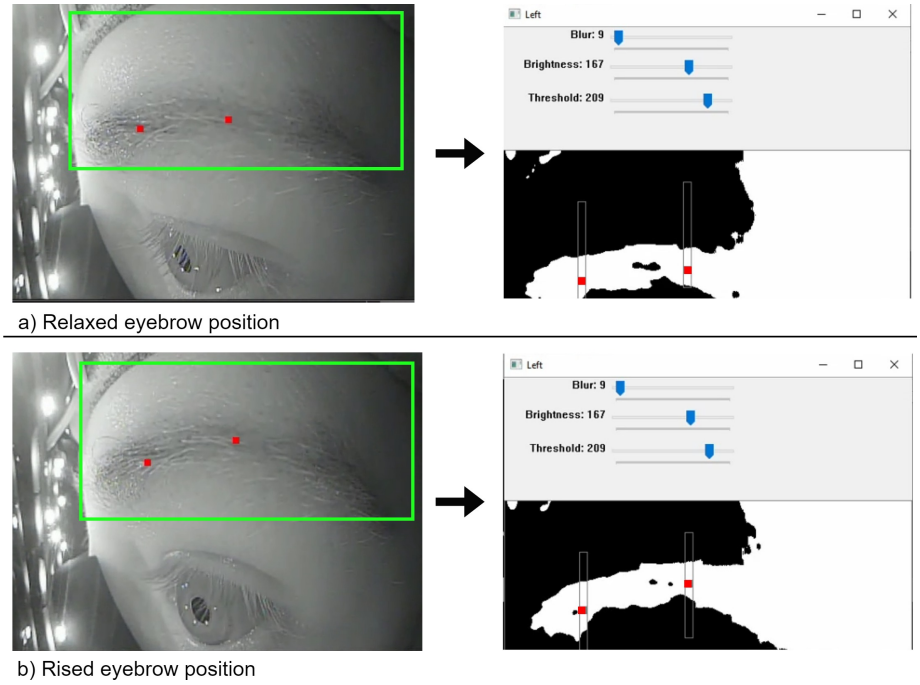


Figure 5.28.: *The section of the image in the green frames is converted into a binary image on the right. This binary image is determined based on a brightness threshold. In the image shown, the eyebrow is represented in white in the binary image on the right. An algorithm recognizes two landmarks (red) that track the average Y-position of the white area and thus the eyebrow. The upper half shows a relaxed eyebrow and the lower half a raised eyebrow.*

In the common Multi-PIE landmark format, each eyebrow actually has five landmarks. In the camera image shown in Fig. 5.28 it can be seen that the exposure of the eye area decreases towards the outside. The presented method is not able to extract these underexposed, low-contrast areas. However, since the eyebrows show little deformation and either go up or down in one piece, we found that two landmarks are sufficient. These two landmarks control all five landmarks of each eyebrow. However, this approach requires individual calibration of the gray threshold for each person, since people have different skin and eyebrow colors. An obvious limitation is that the method fails if the contrast

between eyebrows and skin is too low, or if there are no eyebrows at all. The higher the contrast, the better the tracking. Low contrast can lead to noise, which can have a negative or uncanny effect on the later rendering of the face. To minimize this problem, a 1€-filter [CRV12] with a $min_{cutoff} = 0.004$ and a $beta = 0.007$ was implemented. Among other things, the use of a Gaussian blur is helpful to reduce the flickering of lighter areas in the eyebrows, which can be caused by reflections from individual hairs.

After a firmware update to the HTC Vive Eye Pro in 2021, eye tracking did not work reliably with our ring PCBs. Once the PCBs were removed and the HTC Vive's original IR LEDs were no longer obscured, eye tracking returned to normal. Unfortunately, HTC did not release information on exactly what changes they made to the firmware, and we were unable to restore working eye tracking with our ring PCBs. However, after the firmware update, the image artifacts without our PCB were significantly less severe than before the update. Our solution to restoring our eyebrow tracking was to minimize the minor image artifacts that can be seen in the top row of Fig. 5.29 so that our image processing algorithm described above could work again without error. The solution to the problem is to create an image based on the average of three consecutive images and make it available to the eye tracking algorithm. The formula is simple:

$$\frac{i(t) + i(t+1) + i(t+2)}{3} \tag{5.8}$$

where $i(t)$ is an image of the HMC from time step t. One drawback was that the tracking frequency was cut in thirds, resulting in an effective tracking rate of 10 Hz.

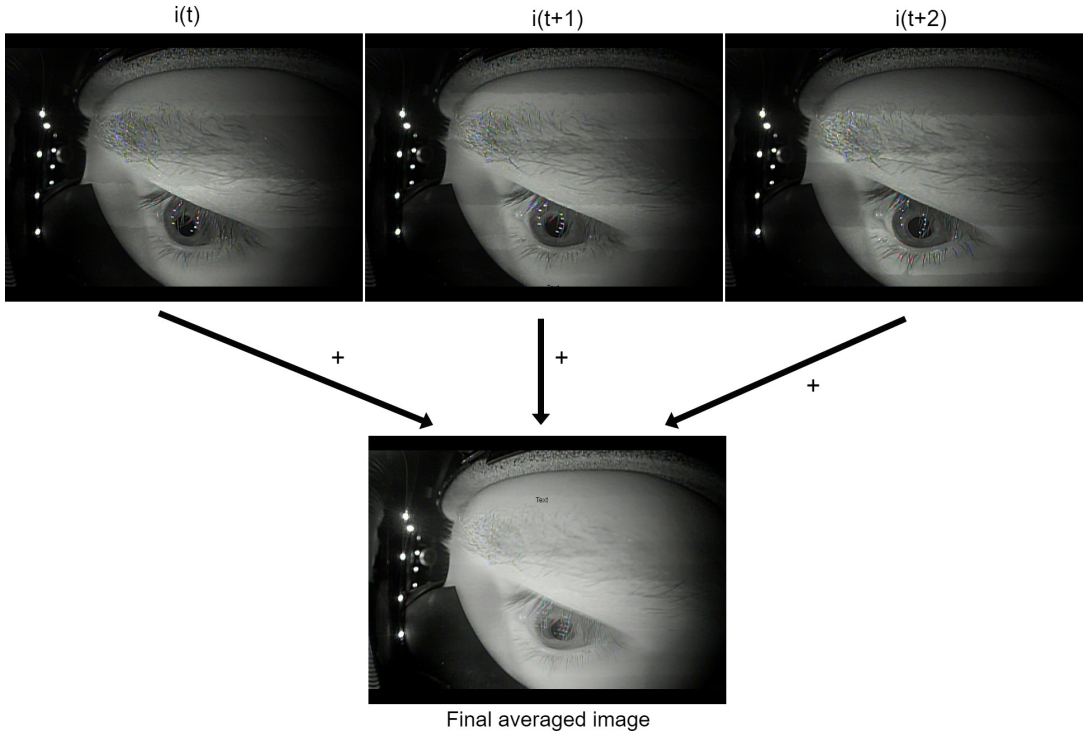


Figure 5.29.: *After updating the Tobii eye tracker firmware on the HTC Vive Pro Eye, image artifacts in the form of flickering became visible. The solution was to overlay the last three images to reduce the flickering significantly. This would not be a solution for eye tracking, as can be seen from the blurred iris, but the eye brown tracking algorithm delivers good results.*

5.6. Merging Tracking Data

The Face Tracking HMD consists of 3 independent tracking modules that must be combined to achieve full face tracking. The modules are the eyebrow tracking from Sec. 5.5, the lower face tracking from Sec. 5.4 and the standard eye tracking solution. Each of these modules provides positions of facial landmarks that are concatenated into 70 facial landmarks, each consisting of two coordinates, similar to the structure of the Multi-PIE dataset with two additional landmarks – the irises. In the following, the concatenated image of all landmarks will be called the *Facial Landmark Map (FLM)*, which will play a key role in face reconstruction in the next chapter. In the next chapter, the FLM is given as input to a generative neural network that can synthesize a photorealistic image of a person based on the facial landmarks. Due to the extreme camera angles from the eyebrow and the lower face sensor, the reported landmarks must be modified in three steps before they are given to the generative neural network. An overview of these steps is given in Fig. 5.30 below:

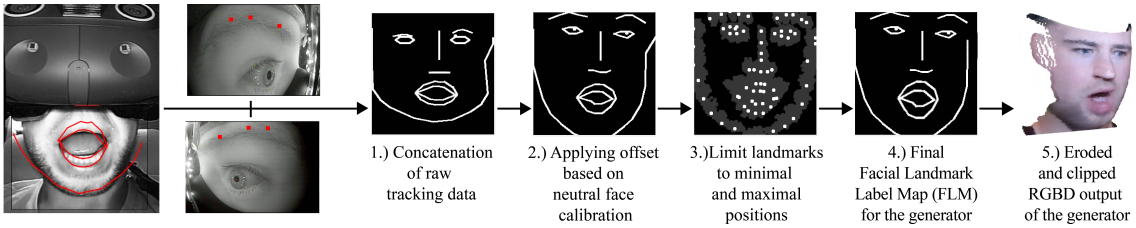


Figure 5.30.: *Processing the data generated by the face-tracking HMD into an FLM. In the following course of the dissertation, the FLM is used to synthesize an avatar face through a generative neural network. Image from [Lad+20b]*

In the first step, the reported landmarks from each tracking module are concatenated into an "uncalibrated" FLM representing the raw face-tracking data.

The second step consists of a calibration step between two neutral facial expressions. The user wearing the HMD is asked to make a neutral face, and offsets are now added to this raw tracking information, which comes from a data set that is used to generate a facial avatar, discussed in the next chapter. A relationship is established between a neutral face in the face-tracking HMD and the neutral face without the HMD. This helps to ensure high quality representation of the face avatar synthesized by a generative neural network. This offset is continuously added to each of the incoming landmarks during real-time face tracking.

In our experiments, we found that the tracking results can deviate significantly from the average in some cases. This happens, for example, when the HMD is put on or taken off, or when the user makes expressive facial play, there are large tracking errors. These errors quickly lead to the occurrence of the uncanny valley effect. Therefore, in the third step, a minimum and maximum position limit is applied to the landmarks to filter out unusual positions. We call this a "cage" because each landmark can only move within a certain range, which was also part of the training set on which the face avatar was trained. In this way, we include all positions in the cage across the entire training dataset and add an additional 3px dilation (at a resolution of 512 x 512 pixels). The gray area around the landmarks in step 3 in Fig. 5.30 represents the cage area. Our tests have shown that the cage increases the range of possible expressions. This is important because the quality of the output of the generative neural network decreases when it receives landmarks that were not in the range of the training data set.

The fourth step is not directly part of the face-tracking HMD process, but is included for completeness and to illustrate the complete use case. It passes the final FLM to the neural network (the generator) to synthesize the facial avatar with the corresponding expression. Note that the system requires very little network bandwidth, since we can choose whether to send the image data (including the FLM or the data of the final generated avatar) or only the position data of the 70 landmarks. This corresponds to only 140 float values 30 times per second, which means a network bandwidth of only 67 kbit/s is required.

5.7. Evaluation

A direct comparison with other systems was difficult after the development in 2019 and 2020, because there were no comparable systems at that time. Therefore, the face-tracking HMD is evaluated with an image-generating neural network, which will be introduced in the next chapter. We have summarized the results in Fig. 5.31. The first column shows the ground truth results captured by the face-tracking HMD. The ground truth images were taken by the person with the shown expression while a second person removed the face-tracking HMD and took an image of the person without the HMD. The second column shows the FLM generated by the face-tracking HMD. The third column shows the images generated by the generative neural network, which receives the FLM and generates the corresponding avatar image. The last column shows the difference image between the first and third columns. The darker an area, the greater the difference. Please note that we also generated the images in Fig. 7.11 on page 138 using the same procedure. These images were also generated with the face-tracking HMD. Lines C, D and E are also included and are analyzed in the context of the generative neural network. Therefore, it also generates depth information and we have removed the background because it is not relevant in a telepresence scenario.

The results show that the face-tracking HMD is able to track and reproduce facial expressions in the majority of cases with sufficient accuracy to recognize the user's expression. However, we found that even small tracking errors, especially around the eyes, can lead to quite different interpretations of facial expressions. For example, only a small difference in the degree of eye opening leads from a neutral expression to a surprised expression, as can be clearly seen in row D of Fig. 5.31.

A limitation of the system, which is difficult to show in single images in this document, is the temporal coherence of the results. This plays an important role for an authentic representation, as jitter or twitching due to noise or tracking outliers or errors can quickly lead to the occurrence of the uncanny valley effect. Filtering the results is a trade-off between minimizing noise and tracking errors and achieving interactive, fast, and authentic face animation. The used 1€-filter [CRV12] performs well in this area compared to e.g. an average value filter, because otherwise movements like laughing are transferred unnaturally slowly to the avatar's face. Under certain circumstances, a strong filter can make it look like a slow motion video of the person is being displayed.

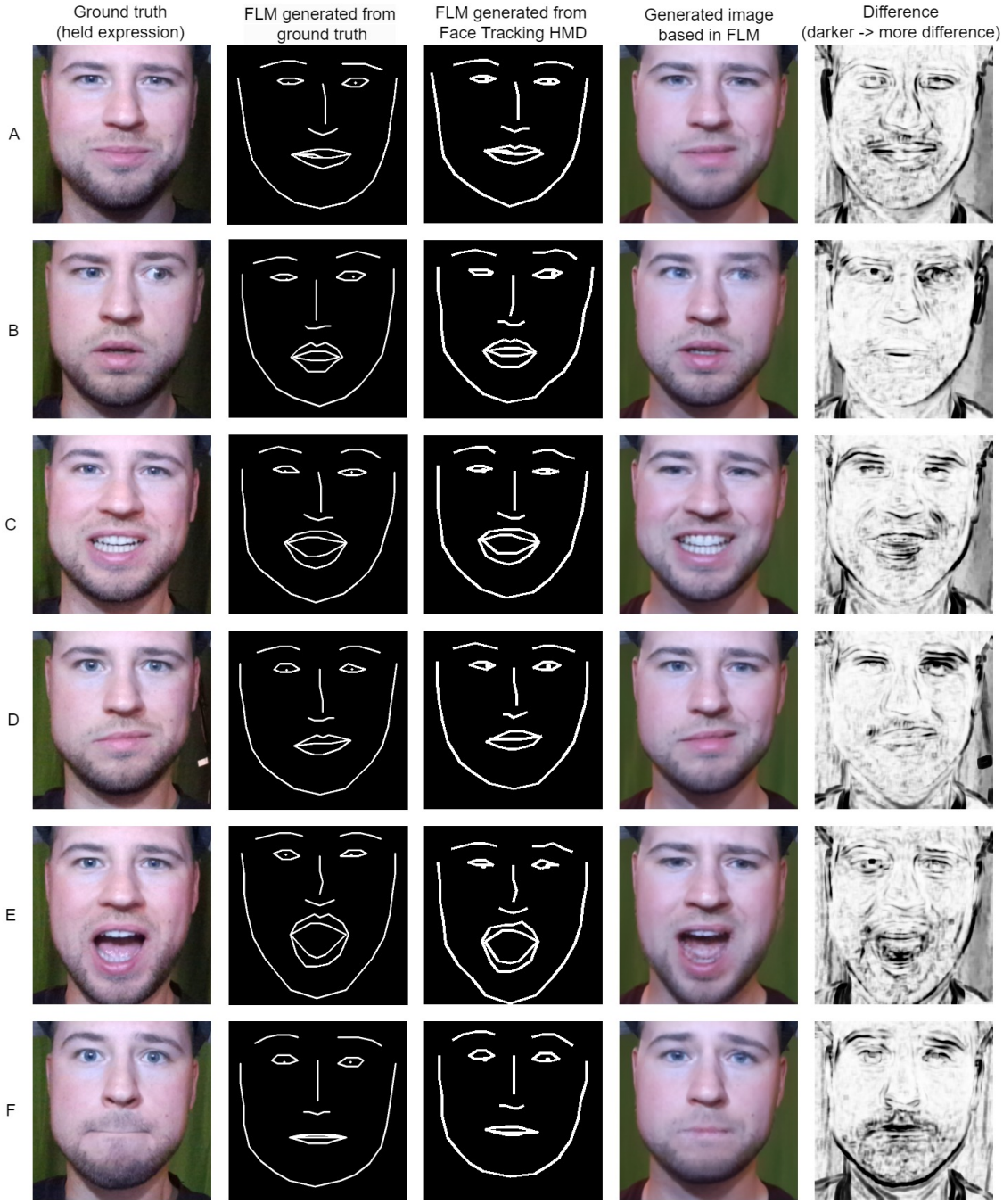


Figure 5.31.: *Results of the face-tracking HMD: The first column shows ground truth images of facial expressions captured without the face-tracking HMD. These were generated by photographing the subject after removing the face-tracking HMD. The second column shows the face landmark model (FLM) generated from the ground truth and the third column shows the FLMs generated by our face-tracking HMD. The fourth column shows avatar images generated by a neural network (GAN) using the FLM. The neural network will be introduced in the next chapter. The last column shows the difference between the ground truth and the generated images, with darker areas indicating greater differences.*

5.8. Discussion and Future Work

A few years have passed (it is May 2024 now) since the development of the face-tracking HMD and some findings from the academic field can be noted. The use of facial landmarks as input to a neural network is decreasing and there is a trend towards the use of 3DMMs. Although facial landmarks are still used to align the 3DMM initial in a good position for further optimization steps, the blendshape parameters, which often consist of only 50 to 60 float values, are finally fed into the network as input. This is significantly less than the 140 values entered in our 70 landmark approach.

In addition, now there are a number of different off-the-shelf tracking systems from well-known HMD manufacturers such as Meta or HTC that offer comparable and often better performance than the system presented here. It should be noted, however, that the tracking data provided is sometimes very encapsulated and available in manufacturer-specific data formats. This means that the tracking information is received in the form of blendshapes designed for a specific predefined animation standard or 3D face model of the company. Connecting other models can be cumbersome. For privacy reasons, the SDKs offered by the manufacturers usually do not allow direct access to the camera streams. Image processing is done internally, so there is no access to the image feeds via an API or open source interface, and there is no technical documentation on how the face-tracking process works.

In our observations, we found that a frame rate of 30 fps can sometimes be insufficient to accurately represent facial expressions, especially lip movements during speech. This problem likely arises because while most elements are rendered at a high frame rate of 90 fps, the lips remain static for a period of 3 frames. To solve this problem, it might be beneficial to implement a blending technique such as a Bezier curve or simple linear interpolation. This would require interpolation of the facial landmarks (FLMs), specifically the 68 landmarks used to define facial features, to achieve a smoother motion representation. Gaze direction interpolation may look unnatural and would need to be further explored.

5.9. Conclusion

In this chapter, we developed and evaluated a face-tracking HMD system designed to capture and interpret facial expressions in real time. We have answered research questions 5 (RQ5) "How to track a face beneath an HMD?" and presented approaches that integrate different types of sensors into the HMD for face tracking. Our results show that traditional full-face tracking methods for "images from the wild" fail under the specific constraints required for HMDs, such as close-distance shots, steep angles and fish-eye lenses. To overcome these limitations, we developed and trained a convolutional neural network focused on the lower face region to analyze the sensor data. This network effectively detects 36 facial landmarks with 900 frames per second using a wide-angle IR sensor. With 900 frames per second, our solution runs 60x times faster than other common networks such as the FAN [BT17] with around 15 frames per second.

In addition, we explored two methods for eyebrow tracking – a feature that many of today's face-tracking methods in the year 2024 still do not support. The first method, using pressure sensors embedded in the foam of the HMD, proved to be unreliable. Therefore, we developed an optical tracking solution using two additional IR sensors, which showed superior performance in tracking eyebrow movements using classical image pro-

cessing algorithms without neural networks. The challenge here is to combine the eyebrow tracking system with off-the-shelf eye-tracking without interference from the illumination, as the off-the-shelf solutions use pulse-width modulation to dim their IR LEDs. Hardware synchronization was not possible, therefore a software-based solution was introduced to remove image artifacts.

The combination of the lower face, eyebrow, and off-the-shelf eye tracking modules results in a facial landmark map (FLM) containing 70 feature points. This map can serve as input to methods that generate photorealistic imagery of the user's face, as shown in later stage of this dissertation in Chap. 7 and 8. Our face-tracking HMD achieved a solid level of accuracy by addressing issues such as noise and tracking errors with effective filtering techniques. Although our neural network theoretically runs at 900 fps, the actual frame rate is limited to 30 fps by the given frame rate of the low-cost IR sensor used. This issue highlights the need for interpolation techniques in the future to ensure smoother transitions and more natural avatar animations. However, our system is not yet able to completely bridge the uncanny valley.

In summary, this chapter contributes a novel face-tracking HMD system that overcome current technological gaps and provides a robust, low-cost solution for real-time facial expression tracking that requires minimal hardware computational capacity. This work lays the foundation for future developments in immersive telepresence applications by enhancing the fidelity of virtual interactions through improved NVC cues driven by the user's face.

Part III.

Real-time Face Rendering

6. The Impact of Personalized and Tracked Face Avatars in Immersive Telepresence Environments

With today's advances in technology, it is becoming easier to create and manipulate personalized and authentic 3D avatar faces for use in social VR applications. However, the process of creating a personalized avatar with facial expressions is resource-intensive, requiring significant time, computational resources and expertise, as well as high-end hardware for interactive rendering. This raises the question of whether the investment in such an elaborate avatar with facial expressions is justified. It is conceivable that a simple, anthropomorphic and generic avatar might suffice, potentially providing an equivalent sense of presence compared to an "expensive" personalized avatar.

In a study with 22 participants divided into two groups, we investigated copresence and social presence. Copresence refers to the state of being in the same physical or virtual space with something or someone at the same time, while social presence describes the degree to which individuals feel a sense of personal and emotional connection with others in mediated communication. We observed evidence that a non-personalized (in this context called "generic") anthropomorphic representation of the interlocutor can lead to a reduced sense of social presence compared to a personalized representation that resembles the interlocutor. However, our results suggest that the sense of copresence remains unchanged by the use of a personalized avatar. In summary, our results suggest that it is useful to generate personalized avatars that resemble their real-world counterparts because it increases participants' sense of social presence.

6.1. Introduction

For many years, researchers in the field of AR/VR/MR have explored the nuances of social interaction, remote collaboration, and their impact on the sense of different types of presence. Different avatar designs and levels of rendering fidelity have been experimented with, but the challenge of creating authentic avatars has remained a technical hurdle with its own set of limitations.

In the early 2000s, researchers Nowak and Boccia [NB03] pioneered the study of the influence of facial expressions on anthropomorphic agents and avatars in virtual environments. They demonstrated that people perceive less copresence and social presence from dialog partners represented by low-anthropomorphic representations than from those consisting of high-anthropomorphic representations. It is worth noting that the technology of the time severely limited the realism of facial representations. However, even 14 years later in 2017, avatar capture systems are still not photorealistic. For example, the creation of full-body avatars with detailed facial capture is described by Achenbach et al. [Ach+17],

who present a fairly sophisticated system. Although the system is complex and delivers comparatively good quality, the avatars are still in the uncanny valley.

Compared to Nowak and Boccia [NB03], more contemporary studies have dealt with the social and perceptual consequences of using realistic self-avatars, as investigated by teams such as Piryankova et al. [Pir+14], Latoschik et al. [Lat+17b], and Waltemate et al. [Wal+18b]. The 3D scanning technologies used by these groups are sophisticated and require laboratory facilities and specialized personnel to operate. However, these technologies have not been used to compare the effects of personalized anthropomorphic avatars, designed to reflect the actual appearance of individuals with facial expressions, to generic avatars with standardized facial expressions in a VR-based remote face-to-face conversation.

Although the study presented in this chapter was conducted in early 2019, and our system is not photorealistic either, it is (in the author's perception) more detailed in the facial domain and less "creepy" than other systems, such as that of Achenbach et al. [Ach+17]. As I write this chapter for my dissertation in March 2024, great progress has been made in the field of photorealistic representation of photorealistic avatars. Although there is still no commercial product that can create unique avatars, more and more work is being presented that can generate photorealistic avatars in real time under laboratory conditions with comparatively little computational effort and time. The chapters 7 and 8 dive deeper into current technologies such as GANs and Implicit Neural Representation (INRs). With these technologies, the effects investigated in this study could presumably be better measured and more clearly isolated.

To the authors' knowledge, this was the first work in 2019 to use a real-time facial capture system under an HMD with rendering of personal facial avatars to investigate its effects on copresence and social presence. Thus, it is an open question whether personalized avatars can promote a greater sense of presence than their generic counterparts. This chapter explores the following hypotheses:

H1: The use of a personalized avatar face, with facial expressions based on predefined blendshapes and animated in real time by a human, enhances the perception of **copresence** compared to a generic avatar face using the same facial expressions (the expressions are the same but the identity is different, see Fig. 6.1).

H2: The use of a personalized avatar face, with facial expressions based on predefined blendshapes and animated in real time by a human, enhances the perception of **social presence** compared to a generic avatar face that uses the same facial expressions (the expressions are the same but the identity is different, see Fig. 6.1).

The contribution of this chapter is to investigate the influence and effects of both personalized and non-personalized (in this chapter referred to as "generic") avatar faces equipped with facial expressions on social presence and copresence. We present both quantitative and qualitative results, and report and discuss participant feedback. To do so, we developed a technical pipeline to generate and control personalized avatar faces while wearing an HMD via eye and lip tracking. The implications of this research extend to social interactions and conferencing applications in AR/VR/MR contexts. The concepts of social presence and copresence are critical to effective communication. The results of this research could enhance our understanding of avatar-based remote interactions and provide preliminary insights into the advantages of using personalized avatars over generic ones. It also motivates further prototype development in the course of this dissertation.

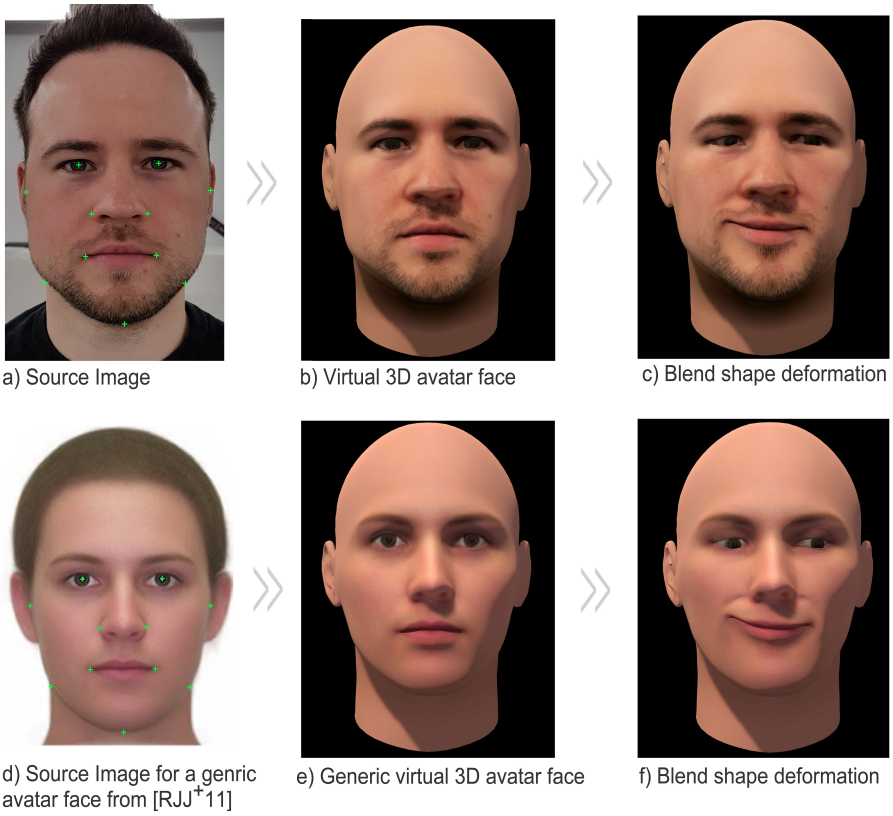


Figure 6.1.: ***a)*** *Input image for creating a personalized avatar. Green crosses are manually annotated landmarks for the 3D avatar face generation algorithm;* ***b)*** *Personalized 3D avatar created from image a);* ***c)*** *Personalized avatar deformed by blendshapes;* ***d)*** *Generic avatar created from the androgynous norm of Rohdes et al. [Rho+11]. Generation of avatar head is identical to a)* ***e)*** *Generic 3D avatar created from image d);* ***f)*** *Generic avatar deformed by identical blendshapes as shown in c). Image from [LG19b].*

6.2. Related Work

6.2.1. Presence, Social Presence and Copresence

The concept of presence is interpreted differently by scholars, leading to a plethora of definitions, interpretations, and subtypes [Gof63; SWC76; BH87; Bio97; BC02; BHB03; NB03; You03; OBW18b]. Some researchers use the terms "immersion" and "presence" interchangeably. However, a distinction can be made between "technological qualities" and "psychological experiences" [SW97]. The characteristics of a technical system, such as resolution, field of view, frame rates, and so on, can affect the degree to which the system is immersive [Wel+96]. In contrast to immersion, presence is the subjective experience of actually being in a mediated virtual environment [SW97]. According to Lee [Lee06] and Oh et al. [OBW18b], presence can be further divided into tele-presence, self-presence, and social presence: Tele-presence can be defined as "the extent to which one feels present in the mediated environment rather than in the immediate physical environment" [Ste06, p. 75]. In contrast, self-presence is the extent to which the "virtual self is experienced as the actual self" [AKB12] and is closely related to immersive virtual body ownership [LLL15].

The third type, social presence, refers to the "sense of being with another" [BHB03] and depends on the ease with which a person perceives having "...access to the intelligence, intentions, and sensory impressions of another" [Bio97, p. 19].

As this chapter focuses on copresence and social presence, it includes a more detailed view of the term and goes beyond Lee's definition of the three subcategories of presence. [Lee06]

The definition of social presence by Youngblut [You03], Biocca [Bio97; BHB03], Oh[OBW18b] and Nowak and Boccia [NB03] divides the term into copresence and social presence itself. Copresence is defined as "... the subjective experience of being together with others in a computer-generated environment, even when participants are physically situated in different sites.". [You03]. Goffman [Gof63] described copresence as a situation in which individuals report that they actively perceive others while also feeling that they are actively perceived by others. Some researchers emphasize that the term "others" does not explicitly mean "humans" because it is possible to feel copresent with computerized agents or inanimate objects. Social presence excludes agents and objects, while it addresses social interaction with a real person as well as "...access to the intelligence, intentions, and sensory impressions of another" [You03]. Biocca stated that "social presence occurs when users feel that a form, behavior, or sensory experience indicates the presence of another intelligence" [Bio97]. Short et al. [SWC76] popularized the concept by defining social presence as "the degree of salience of the other person in the interaction and the consequent salience of the interpersonal relationships.". However, their measures focus more on the user's perception of a medium's ability to convey another's presence than on the actual perceived presence of the other person.

A meta-study by Oh et al. [OBW18a] found that most current evidence suggests that people experience higher levels of social presence when shown a visual representation of a person compared to no representation of a person. The study also summarizes the extent to which the effect of social presence is enhanced when the visual representation of the avatar is animated with the authentic movement and behavioral realism of a real person. Here it is clear that the degree of social presence increases when the movements look real and match what is being said, such as matching hand gestures, head nods or shakes, as well as blushing when a virtual human/agent makes a mistake. [Gar03; RKG09; Kul+11; Kim+16]

In contrast to the consistent effects of behavioral realism, studies of the effects of photographic and anthropomorphic realism on social presence show mixed results. While some studies show an increase in social presence with more realistic visuals, others find no difference or even a decrease [NB03]. This inconsistency may be due to three factors: 1.) photographic realism may be less important than behavioral cues, 2.) questionnaires may not capture subtle differences, and 3.) varying levels of photographic and behavioral realism across studies due to limitations of technology at the time of the study, e.g., the much-cited study by Nowak and Boccia [NB03]. Consistency between behavioral and photographic realism appears to be critical, as higher social presence is reported when both are consistent.

Not directly part of this study, but very relevant in this area, is the uncanny valley effect. It is still unknown how this effect has influenced various previous studies, as it is difficult to quantify. However, it seems clear that expressive human-like movements and appearance, as well as facial expressions, reduce this effect and lead to similar perceptions between virtual characters and real people [RW18; MC16; McD+08].

In light of current developments and discussions about the performance and Turing test

of Large Language Models (LLMs) from companies such as OpenAI, Anthropic, Mistral, and others, it would certainly be necessary to further differentiate the notion of social presence at this point. In this study, we focus on the interaction between real persons in a remote environment.

6.2.2. Face and Body Capture

Human scanning has been the subject of extensive research using a wide range of technological methods. The market also offers numerous commercial solutions for body scanning and capture. Inexpensive and simple systems have been introduced by Nagano et al. [Nag+17], Straub and Kerlin [SK14], Gesslein et al. [GSG17], and Shapiro et al. [Sha+14]. Casas et al. [Cas+15] developed a system for generating different facial blendshapes. More complex face and body scanning systems have been demonstrated and used by Achenbach et al. [Ach+17] and Bogo et al. [Bog+17], as well as in social and perceptual experiments by Latoschik et al. [Lat+17b] and Piryankova et al. [Pir+14]. Some systems are capable of performing scans in real time, capturing multiple frames per second, as demonstrated by Orts et al. [Ort+16]. Systems with high quality results usually require expensive hardware or can only be used in a laboratory environment. A significant challenge for many systems is the accurate capture and animation of facial expressions and the avoidance of the uncanny valley effect, a challenge our system also faces. A comprehensive literature review on avatar face generation and rendering with neural rendering can be found in Sec. 7.2.

6.2.3. Facial Expression Recognition under a Head-Mounted Display

Since 2016, when HMDs became significantly cheaper while getting better technical specifications, a number of papers on face-tracking HMD prototypes have been published. We refer the reader to the extensive literature review in Sec. 5.1. In this subsection, we only mention specific similarities and differences to other works that are related to the technical aspects and not to the user study.

The choice of comparable prototypes is small. Thies et al. [Thi+18b] presented a system called FaceVR, designed to reproduce the user's facial expressions in a video. This system tracks the mouth area using a standard fixed webcam (unlike our approach, which is not attached to the HMD), while eye movement is captured by a single IR eye-tracking camera mounted inside the HMD. The visual quality of the system is impressive and almost photorealistic. In particular, the reproduction of authentic facial expressions looks realistic and does not produce the uncanny valley effect. In comparison, our system uses generic blendshapes, which do not reproduce person-specific characteristics very well. Thies et al.'s system also uses a 3D head like ours, but it is embedded in a video (which may loop) to display the background, scalp hair, and torso. Therefore, Thies' system has a limited ability to move freely around the head. We do not have these limitations, but we also do not display a background, scalp hair, or torso.

Li et al. [Li+15b] presented a system that uses a depth camera mounted on the HMD to track the area around the mouth that is not covered by the HMD. Unlike FaceVR [Thi+18b], Li et al.'s system does not track eye movements, but can detect movements around the eyes using thin strain sensors placed in the foam liner of the HMD. Casas et al. [Cas+16] developed a system capable of capturing personalized face meshes, textures, and corresponding blendshapes via an RGB-D sensor for real-time facial animation. The visual quality is reasonable, but the system seems to be limited to a small set

of facial expressions. Our approach of using generic blendshapes allows us to display a variety of facial expressions.

Lombardi et al. [Lom+18] presented the most sophisticated systems to date. They introduced a deep appearance model for rendering human faces, using Generative Adversarial Networks to create personalized avatars with photorealistic facial animations. The system incorporates cameras both inside and outside the HMD to capture the area around the mouth and eyes, enabling tracking of eye and eyebrow movements and facilitating the creation of highly expressive avatar faces in this research area. The work of Wei et al. [Wei+19] further extended this technical approach, resulting in improved visual quality. Chu et al. [Chu+20] further improved the so-called "codec avatars" and conducted a user study on the subjective perception quality of the avatars.

None of the systems described in this subsection have been evaluated for their impact on the qualitative or quantitative aspects of presence.

6.3. System

This section outlines our technical approach for real-time generation and animation of personalized avatar faces. Our developed pipeline allows us to generate a fully rigged and textured personalized avatar head for an immersive telepresence application within two minutes. A demonstration video of the system running can be found here: `https://youtu.be/_SJYunw6kVU`

6.3.1. Avatar Creation

Custom avatars are created using the FaceGen SDK [FaceGen24]. Three images of the target person's face are taken (front, right, left) and processed by FaceGen. This process is largely automated, with a script processing the input images to construct a 3D avatar head. The only manual step involves manually marking 29 facial landmarks on the front and side images. These landmarks are marked with green crosses in Fig. 6.1a and d. FaceGen generates a single personalized avatar face in less than 90 seconds.

FaceGen applies the input images to a standard base head mesh and modifies specific regions based on these images. This is done using statistical shape models (SSM) and statistical appearance models (SAM). For facial images, SSMs and SAMs quantify the average shape and texture distribution of a face within a given population, along with the primary variations in shape and texture distribution from these averages. Having access to this detailed quantitative data on facial anatomy allows for the precise deformation of the base head mesh to match the input image.

Our workflow excludes the generation, capture, or animation of scalp or beard hair, as well as eyeglasses. Only short beards and eyebrows are included as texture information in the 2D face texture. In addition, we do not perform any form of teeth or tongue capture/scanning due to the technical challenges of reproducing a believable oral cavity. As a result, the personalized avatars have a standard set of teeth and tongue that is consistent across all models. Please note that we do not use physically-based skin rendering or other techniques such as spherical harmonics. Lighting is a simple setup with Lambertian skin shading.

The generic avatar face shown in Fig. 6.1d was created from a single gender-neutral com-

posite frontal face image derived by blending 48 photographs (24 female, 24 male) into a single average face, based on the work of Rohdes et al. [Rho+11].

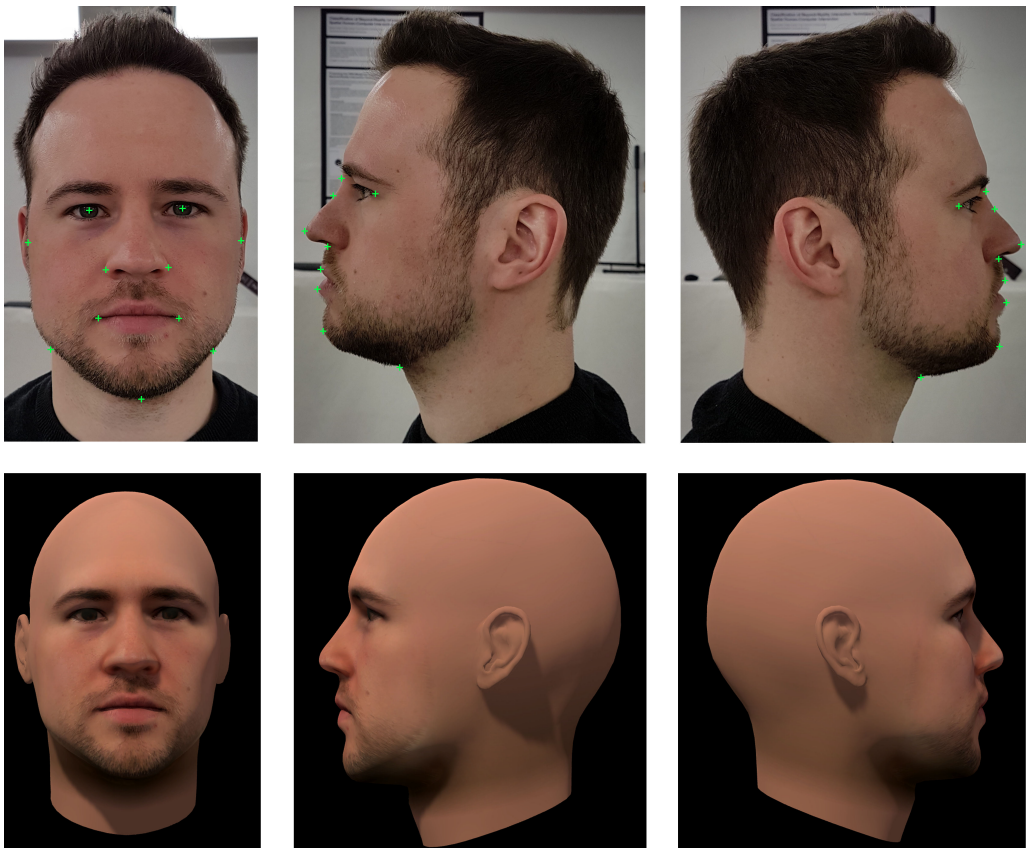


Figure 6.2.: *Example of FaceGen input images and output avatar head. Rendered with the Unity game engine. The green crosses in the upper images are landmarks that need to be set manually for avatar creation. Image from [LG19b].*

6.3.2. Facial Animation

In addition to the mesh and color map, FaceGen outputs 112 different blendshapes for facial animation. For our study, we use only 16 of these blendshapes. The generated blendshapes are universal and not tailored to each individual. This means that facial idiosyncrasies such as wrinkles or the personal style of smiling are not captured and reproduced, as shown for example in Fig. 6.1c and 6.1f.

In the application we used for our study, the blendshapes that control eye movements are driven by a Tobii Pro eye tracking device integrated into an HTC Vive HMD. This eye tracking technology captures the direction of gaze and blink for each eye and is located behind each Fresnel lens. Additionally, mouth movements are tracked by a PMD Pico Flexx depth camera attached to the HMD with a 3D printed mount (Fig. 6.3). The BinaryVR SDK [BinVR19] is used to determine the positions of various parts of the lower face, such as the chin and mouth corners. The system is developed on Unity version 2018.3.11f1. The system is designed so that the head generated by FaceGen can be imported into the Unity application without any further adjustments, and all blendshapes are automatically

registered between the BinaryVR SDK and the Tobii Eye Tracking SDK. At runtime, the system achieves real-time performance, maintaining a rate of 90 fps.

It is worth mentioning that the lip tracking of the BinaryVR device is not accurate enough, and the available blendshapes for the avatar heads are not detailed enough to reproduce the various mouth movements when speaking. In addition, there is a delay of about 70-120 ms for the lip tracking pipeline, which also causes unsynchronized mouth movements when speaking. Please watch the following video for more details in motion of the study and setup: `https://youtu.be/_SJYunw6kVU`.

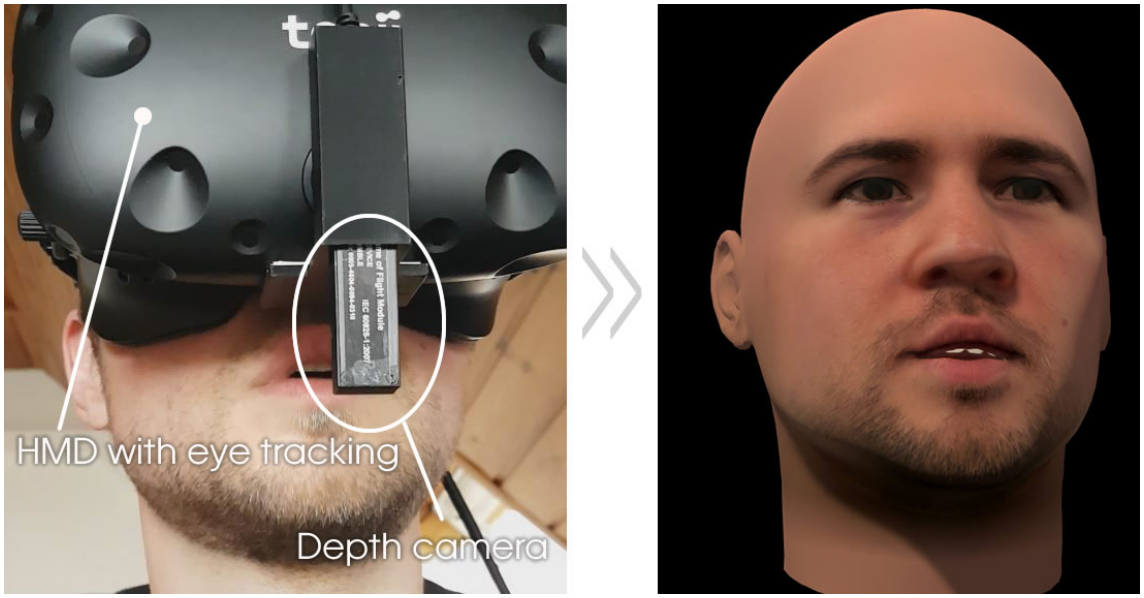


Figure 6.3.: *A depth camera mounted on an HMD tracks the lower facial movements. The BinaryVR software, [BinVR19] maps actual facial expressions to the virtual avatar's expressions. Image from [LG19b].*

6.4. Experiment

6.4.1. Participants

A total of eleven dyads, i.e. 22 German individuals (consisting of two females and 20 males, with an age range of 21-36 years, mean age = 27.04 years, standard deviation = 4.26) participated in the study. These participants were all students affiliated with the local computer science department and had previous experience with MR technology. Each pair within the dyads was acquainted with one another beforehand, allowing them to be familiar with each other's personality traits, facial expressions, and voices. The average time required for each dyad to complete the post-experiment questionnaires, participate in an unstructured interview, and undergo debriefing was approximately 25 minutes. The average time spent in the VR environment was approximately 10 minutes.

Figure 6.4.: *Study setup; a) Participant wears HTC Vive with eye and lip tracking and speaks remotely with a person in another room. The experimenter at the table observes the experiment; b) The participant in another room listens while the remote person speaks. Image from [LG19b].*

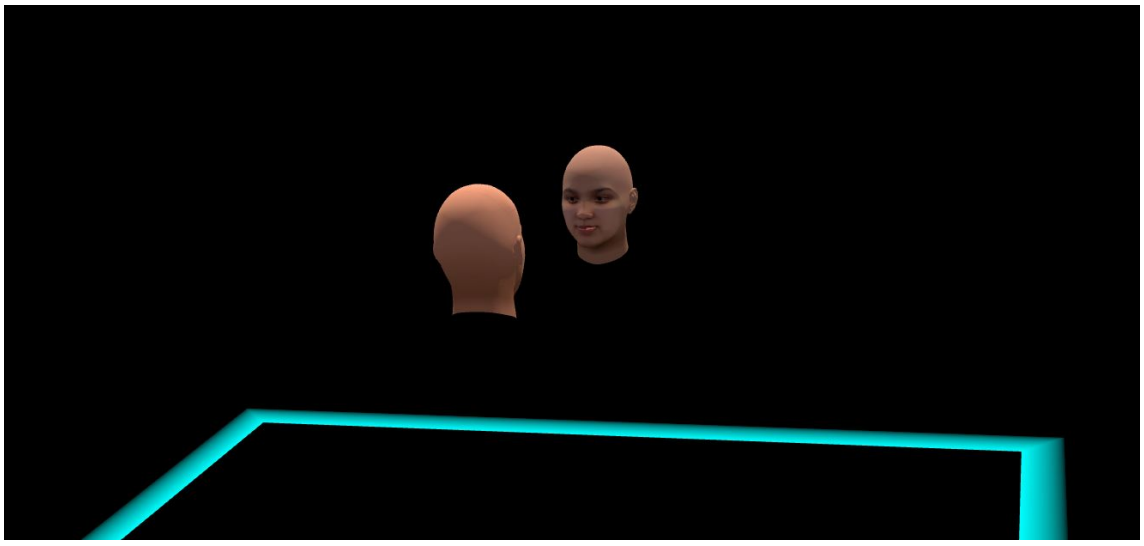


Figure 6.5.: *An in-game camera shows the setup inside the Unity game engine. No hands, controllers, or bodies are shown. Image from [LG19b].*

6.4.2. Method

The experimenter used a standardized and structured procedure to introduce the experiment to the participants, ensuring that everyone received identical information. The main objective of the study was not disclosed to the participants in terms of measuring presence and gaining insight into telepresence with personalized avatars; they were only informed that the experiment aimed to evaluate the impact of eye and lip tracking technology in VR. This approach was adopted to ensure that participants' perceptions of the study were not biased by prior assumptions, thereby minimizing the risk of bias in their responses to the questionnaires. The experimental design was structured as a between-groups design.

Participants were physically located in separate rooms, but met in the same virtual environment. The experimental setup is shown in Fig. 6.4. After agreeing to participate in the experiment, each dyad was asked to sign an informed consent form and was then instructed to don the head-mounted display and headphones. The interpupillary distances (IPD) were adjusted and the eye and lip tracking systems were calibrated. This took approximately 30 to 60 seconds per person. To facilitate the calibration process and to help participants get used to the environment, a virtual mirror was provided at the beginning, allowing participants to see themselves and their partner's avatar. Facial expressions were controlled in real time by the Tobii eye tracker and the depth camera facing the mouth.

Participants and the experimenter could hear each other via a digital audio stream. The task was to engage in casual conversation about topics such as recent weekend activities, upcoming vacation plans, and similar topics. The experiment officially began when the conductor left the audio channel, remaining only as a listener. He removed the virtual mirrors for the participants and they could see each other. The virtual environment is shown in Fig. 6.5. Often, people would start talking immediately when they saw each other for the first time. When the conversation slowed down, the conductor introduced new topics by displaying a virtual board with questions such as "What are you working on?" or "Tell a joke!

After an average duration of 8.25 min in VR (min=5.1 min, max=13.95 min, median=8.07 min), the conductor asked the participants to remove the HMDs and presented a post-experiment questionnaire. The questionnaire collected demographic data and included 14 questions rated on a five-point Likert scale and three questions rated on a ten-point Likert scale. The questions are detailed in Tab. 6.1 and were translated into German and shown to the participants to avoid misunderstandings and bias in the results. All participants were able to read and understand German. Our questionnaire is a reduced version of the study by Nowak and Boccia [NB03] and is also used by other researchers to assess the importance of copresence and social presence, which in turn are inspired by Short et al. [SWC76] and Burgoon and Hale [BH87]. Latoschik et al. [Lat+17b] also used this questionnaire in a reduced version.

Specifically, we reduced the number of questions from a total of 24 to 17 and used a 5-point Likert scale for copresence instead of a 7-point Likert scale. We used 11 questions for copresence and 6 questions for social presence. For social presence, instead of a sliding scale for all questions, we used a mixture of a 5-point Likert scale (Q12-14) and a 10-point Likert scale (Q15 to Q17). We used the questionnaire of Nowak and Boccia [NB03] as a template, but are not sure about the exact interpretation of the documented questions and their metrics due to ambiguities in their paper. However, we have chosen it for reasons of simplicity, ease of use, and the possibility of obtaining more likely at least a tendency due to a lower resolution of the scales (from 7 to 5-point Likert). In addition, the possibility

of choosing a neutral point allowed us to check whether the participants could understand and apply the questions to the experimental setting, which is an important finding in the context of a first study in order to structure further studies in the future.

The questionnaire is available at the following link: `https://docs.google.com/forms/d/e/1FAIpQLSeHlLt6xNs3ER9GzXbNTCQVJPC39R0FBdLCXCIMZ5KFASZW8A/viewform`

Of the 22 participants, ten participated in dialogues with "personalized-face-to-personalized-face" avatars, another ten participated in "generic-face-to-generic-face" dialogues, and two participated in a "generic-face-to-personalized-face" dialog. In total, eleven personalized 3D avatar heads were created using FaceGen for the study.

6.4.3. Results

A Mann-Whitney U test for independent samples was performed for each question, using a significance level of p = .05. The results of the tests are presented in the right column of the Tab. 6.1 and as box plots in Fig. 6.6 and 6.7 for the "personalized avatar face" and "generic avatar face" scenarios. The results suggest that there is no statistical difference in the copresence responses between the two conditions (Q1 to Q11). This could provide a basis for rejecting H1.

However, the analysis reveals a significant difference between the groups in terms of social presence, particularly highlighted by questions Q13 and Q14. The difference in responses to question Q14 is even highly significant. In two out of six questions about social presence, a significant difference between conditions was observed, which could indicate that H2 should be considered confirmed.

The informal interviews conducted with both members of each dyad after the tests support the validation of H2: Four out of eleven people in the group exposed to the personalized avatars praised the quality of the dialogue and mentioned something similar to "the meeting was surprisingly real", whereas none of the group members exposed to the generic avatar emphasized the realism. During and after the interview, there was a sense of excitement in the group that saw the personalized avatar. Note that this is a subjective opinion of the experimenter and the authors and was not recorded in the questionnaire. Two participants in the group that saw the personalized avatars said that especially the faces in a neutral expression are very similar to their real counterparts, but when they make movements, such as laughing or speaking, the representations sometimes deviate strongly from reality.

ID	P. Type	Range	Question	U / p
Q1	Cop.	Likert 5 point	I did not want a deeper relationship with my interaction partner.	U= 43.5 p = .28
Q2	Cop.	Likert 5 point	I wanted to maintain a sense of distance between us.	U= 56 p = .795
Q3	Cop.	Likert 5 point	I was interested in talking to my interaction partner.	U= 56 p = .795
Q4	Cop.	Likert 5 point	My interaction partner was intensely involved in our interaction.	U= 43.5 p = .28
Q5	Cop.	Likert 5 point	My interaction partner seemed to find our interaction stimulating.	U= 52.5 p = .624
Q6	Cop.	Likert 5 point	My interaction partner communicated coldness rather than warmth.	U= 58 p = .897
Q7	Cop.	Likert 5 point	My interaction partner created a sense of distance between us.	U= 49.5 p = .490
Q8	Cop.	Likert 5 point	My interaction partner seemed detached during our interaction.	U= 43.5 p = .280
Q9	Cop.	Likert 5 point	My interaction partner acted bored by our conversation.	U= 60 p = 1.0
Q10	Cop.	Likert 5 point	My interaction partner was interested in talking to me.	U= 55.5 p = .764
Q11	Cop.	Likert 5 point	My interaction partner showed enthusiasm while talking to me.	U= 49 p = .471
Q12	Social P.	Likert 5 point	To what extent did you feel able to assess your partner's reactions to what you said?—Able to assess reactions, not able to assess reactions.	U= 45.5 p = .342
Q13	Social P.	Likert 5 point	To what extent was this like a face-to-face meeting?—A lot like face to face, not like face to face at all.	U= 27 p = .03
Q14	Social P.	Likert 5 point	To what extent was this like you were in the same room with your partner?—A lot like being in the same room, not like being in the same room at all.	U= 11.5 p = .0015
Q15	Social P.	Sliding 1-10	To what extent did your partner seem "real"?—Very real, not real at all.	U= 33 p = .077
Q16	Social P.	Sliding 1-10	How likely is it that you would choose to use this system of interaction for a meeting in which you wanted to persuade others of something?— Very likely, not likely at all.	U= 44.5 p = .308
Q17	Social P.	Sliding 1-10	To what extent did you feel you could get to know someone that you met only through this system?—Very well, not at all.	U= 42 p = .238

Table 6.1.: *The questionnaire used to asses copresence and social presence. Yellow text highlights significant results.*

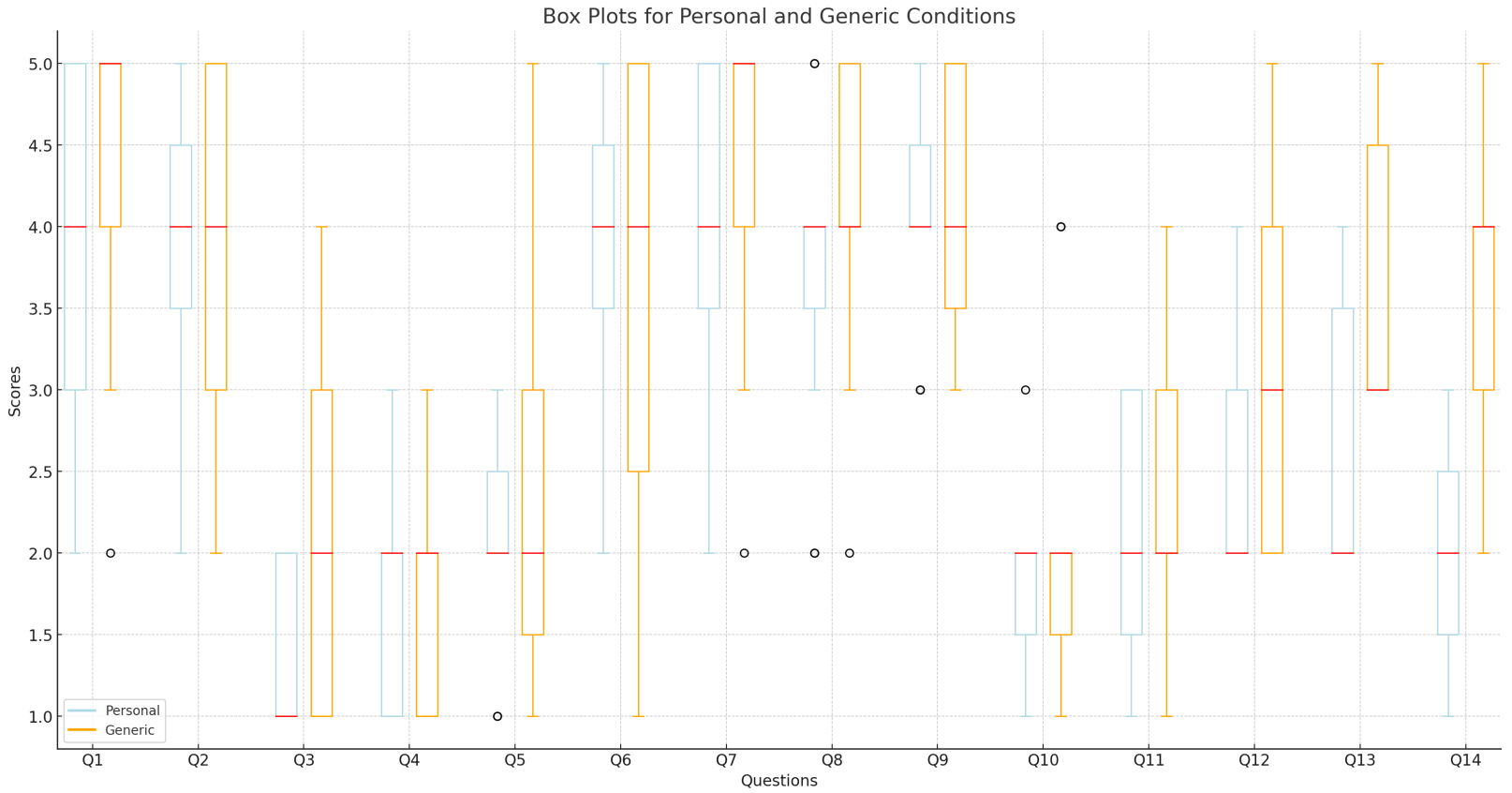


Figure 6.6.: *Results of questions Q1 to Q13 of our questionnaire (Tab. 6.1) visualized as box plots. Each box represents the Likert scale responses of eleven participants. The y-axis is 1 = strongly agree, 2 = agree, 3 = neutral, 4 = not agree, and 5 = strongly not agree.*

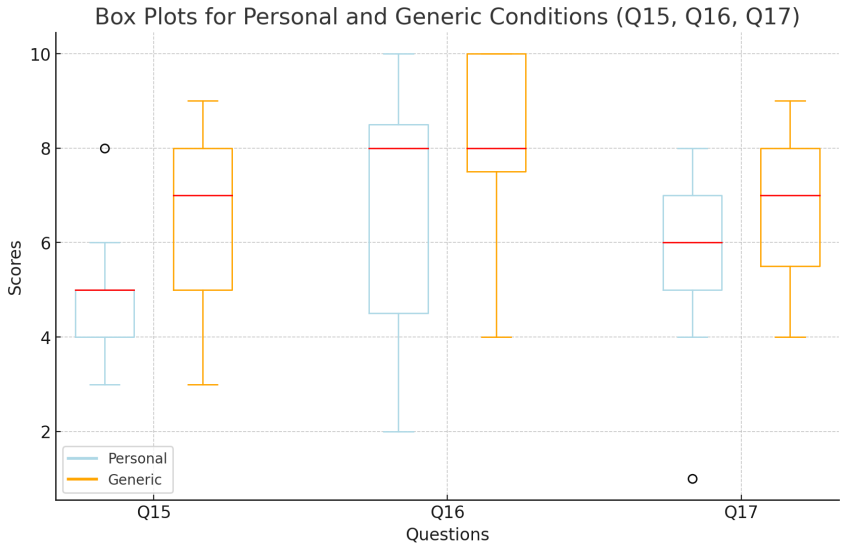


Figure 6.7.: *Results of questions Q15 to Q17 of our questionnaires (Tab. 6.1) visualized as box plots. Each box represents an answer on a sliding scale from 1 to 10.*

6.5. Discussion and Limitations

Despite the lack of photorealistic features of the avatars, such as scalp hair, generic teeth, generic blendshapes, no eyebrow and tongue tracking, and missing bodies, the two avatar face versions were subjectively perceived differently by the groups, as the results show. Another interesting metric might be to measure the perceived level of the uncanny valley effect. For example, Latoschick et al. [Lat+17b] measured the perceived "creepiness" of their avatars in their study.

Another aspect to consider is that the dyads participating in each session were acquainted with each other. We suspect that this may have influenced the experimental results. Feng et al. [Fen+14] conducted research using 3D avatars that were scanned and animated based on a real person, with a set of body gestures recorded and then applied to different human avatars. Their results showed that observers rated the performances of 3D avatars that replicated the body gestures of the original human subject as more similar to the original subject, especially among groups familiar with the subject, compared to avatars that used gestures from a different human subject. Although facial gestures and expressions were not examined in Feng et al.'s study, which is different from our research, we believe that the effect they reported could also apply to facial expressions and could have influenced the questionnaire ratings in our study, i.e. familiarity between participants could be a significant variable. After conducting the experiment, the authors believe that it could also make a difference whether people know each other briefly from work and have only been in direct contact for a few minutes or hours, or whether they have been friends for many years. This could possibly be measured in the form of a multiple-choice question in which the "previous contact time" with the experimental partner is noted.

In our experiment, we had also originally planned to use a logger to measure the virtual distance between the participants every 5 seconds, similar to the experiment by Bailenson et al. [Bai+03]. Unfortunately, due to technical problems, we were unable to record reliable data and therefore did not include these measurements in the study. Our origi-

nal hypothesis was that people who saw a personalized avatar would maintain a shorter distance compared to the other group.

Future research should include various questionnaires such as those introduced by Blascovich et al. [Bla+02], Egerto et al. [EOT64], and Smith and Neff [SN18]. In addition, Slater [Sla04] notes that questionnaires are only one of several tools available. Quantitative measures such as biological signals or the analysis of unconscious and unintentional behavior (as elicited in the Rubber Hand Illusion experiment [IKH06]) could provide additional insight and lend more credibility to the study's conclusions.

6.6. Conclusion

We can affirmitly answer research question 3 (RQ3), at least for social presence. The question was: "Does a personalized avatar increase copresence and social presence compared to a non-personalized?". We have provided evidence that a virtual personalized avatar face that resembles the real person but includes generic facial expressions, as opposed to a generic face with identical derived facial expressions, may not increase the sense of copresence. Nevertheless, our results suggest that it could increase the sense of social presence. Our findings are based on a questionnaire, inspired by Nowak and Boccia [NB03], consisting of 17 questions in total: 11 on copresence and six on social presence. Among these, one question showed a significant difference ($p<0.05$) and another showed a highly significant difference ($p<0.01$) for social presence between the two groups (those who saw a generic face vs. those who saw a personal face).

In the context of remote collaboration, this study highlights the value of using a personalized avatar rather than a generic one to increase participants' social presence. We suspect that the measured effect could be increasingly stronger with more authentic and realistic avatar representations. Our research provides initial evidence that addresses the question posed at the beginning of the chapter: Does the investment in creating personalized avatars pay off? If these avatars enhance the sense of social presence and therefore also the psychological connection within the participants, then personalized avatars may be crucial for authentic social interactions in MR collaborative environments.

7. Neural Rendering for Conveying Nonverbal Facial Communication Cues

7.1. Introduction

Direct, face-to-face communication is multidimensional, involving both spoken words and nonverbal cues. Eye contact, facial expressions, gestures (kinesics), and the physical space between individuals (proxemics) are crucial elements of a conversation [LG19a]. Currently, popular computer-mediated communication tools such as Microsoft Teams, Google Meet, or Apple's FaceTime offer video conferencing. These platforms allow users to observe facial expressions, but they lack features such as real eye contact, full display of broad gestures, spatial interpretation of pointing gestures, and the sense of physical proximity between users.

Today's head-mounted displays provide realistic and engaging 3D experiences, including telepresence applications. However, even in 2024, they fall short in displaying the real face of the users, at least for off-the-shelf hardware and software. When the face is obscured by a VR headset, important nonverbal facial cues are lost, which are essential for interpersonal communication. This limitation is not only relevant in VR meetings (e.g. in VRChat [VRc20], Altspace [Mic22], or Meta's Horizon Worlds [Met22]), but also in scenarios where a person in VR interacts with an audience. Examples include architects presenting building designs in VR, virtual YouTubers (VTubers), Twitch streamers broadcasting from MR environments, or friends playing MR games together. Our proposed method enables real-time reconstruction and animation of "trained" faces for such applications.

Creating and rendering photorealistic humans in real time traditionally requires extensive manual work by skilled 3D artists, including scanning, modeling, and texturing. Current digital humans require the explicit simulation of geometry, materials, and the way light interacts with them, a process that, while effective, is both costly and time-consuming due to the need to meticulously model every three-dimensional detail of the face. Advances in neural rendering, however, propose a transformative approach: by learning from data, this technology aims to transform the complex task of rendering graphics into a problem of model learning and inference. By training neural networks on face datasets, the process of generating avatars could become much simpler, less labor-intensive, and even more realistic.

Few research groups have addressed this challenge and developed methods for generating realistic facial avatars in MR without extensive manual annotation of data and modeling of 3D shapes [Thi+18b; Lom+18; Wei+19; Raj+21]. Academic trends are shifting toward NERFs and Gaussian splatting [Gra+22; ZBT22a; Qia+23], although many of these approaches only work with expensive hardware setups that require many cameras capturing images simultaneously. Other solutions are not publicly available or run only in laboratory environments due to the many manual steps involved in acquisition, training, and

inference [Lom+18; Wei+19]. However, it is clear that it is only a matter of time before mainstream users will be able to create photorealistic avatars and use them in MR.

Studies in various fields have examined human avatars and their perception. A systematic review on social presence notes that vivid perception of another person often increases enjoyment and social influence [OBW18a]. In Chap. 6 we found that a personal avatar face has advantages over a generic avatar in terms of social presence. Despite its age, the media richness theory [DL84], which posits that the richest exchange of information occurs during face-to-face interactions, is still supported by recent research [ILC19]. A major challenge in creating virtual face-to-face presence is the uncanny valley effect [MMK12], where minor unnatural deviations in virtual faces can evoke negative reactions.

Recently, Variational Autoencoders (VAE) and Generative Adversarial Networks (GAN) have successfully crossed the uncanny valley. So-called "deepfakes" powered by VAEs and GANs produce results so lifelike that they are being studied to distinguish them from real images with algorithms, as the human eye often cannot [Rös+19]. In this work, we use similar algorithms from the "deepfake" domain and add an extra dimension (depth data as textured point clouds instead of only RGB images) to create realistic 2.5D facial avatars. Our pipeline requires very little manual work, no annotation and no 3D modeling, and exploits the potential of unsupervised learning. We capture faces from a static frontal view with an RGB-D sensor, do not generate textures for side views, but still maintain stereoscopic perception in virtual face-to-face conversations. GAN image generation is computationally intensive and typically done offline, but our VR applications require high frame rates, typically more than 30 frames per second for an acceptable impression.

This chapter presents three iterative prototype stages that provide an end-to-end learning pipeline for digital faces that is less costly and requires moderate computational resources. A separate paper was published for each iteration [LPG20; Lad+20b; Lad+25]. The goal for all prototypes was to use a standard graphics card to capture and reconstruct facial features with high detail and interactive frame rates, creating authentic avatars that outperform the visual quality of current tools such as VRChat [VRc20] or Meta's Horizon Worlds [Met22]. We aim to reduce the initial barriers for users and provide real-world applicability, focusing on readily available hardware and maintaining real-time frame rates. Our research focuses on enhancing the quality and speed of current GAN technologies. Additionally, we are exploring ways to tailor these technologies specifically for user-friendly telepresence scenarios. We share our work through publicly available repositories of our three iterative prototypes at:
`https://github.com/Alpe6825/RGBD-Face-Avatar-GAN`
`https://github.com/Mirevi/UCP-Framework`
`https://github.com/Mirevi/face-synthesizer-JVRB`

We encourage viewers to watch the accompanying oral presentation videos for visual results and comparisons in motion:
`https://youtu.be/Wa95qDPV8vk`
`https://youtu.be/fBofqRfvoiM`

7.2. Related Work

While face tracking and reconstruction under an HMD for VR is a young field of research, computer graphics researchers have been trying to synthesize computer-generated faces realistically for decades. Raytracing and pathtracing technologies have enabled this ca-

pability offline since the 2000s, but achieving photorealistic rendering of human faces in real time is still considered very difficult in the field of computer graphics. This section discusses systems that primarily use CNNs, such as VAEs and GANs, for image synthesis, as well as earlier systems that used alternative generation methods, such as traditional meshes, optical flow, or similar techniques. In the next chapter of this dissertation, we will discuss systems that use Implicit Neural Representations (INRs), also called coordinate-based neural networks, such as NeRFs [Mil+21] and SIRENs [Sit+20].

Similar to our approach, Casas et al. [Cas+16] captures data with an RGB-D sensor and transforms it into a rigged and textured face mesh. Changes in the texture are realized by a form of optical flow. The system can reconstruct trained poses with high accuracy, but it is time-consuming to create many different facial expressions, as each expression requires several manual steps. Our solution has significantly fewer manual steps and is more automated due to an unsupervised learning approach.

Früh et al. [FSK17] also use an RGB-D sensor to generate facial data as a mesh and capture gaze information. The system is able to reconstruct only the directly occluded face area of an HMD in a green screen environment, where the green screen is replaced by the virtual environment. The mouth is not captured and reconstructed, which means that once the user tilts the head down and covers the mouth region with the HMD, it cannot be reconstructed in the video feed.

The systems of Lombardi et al. [Lom+18], Wei et al. [Wei+19], and Raj et al. [Raj+21] create photorealistic avatars with authentic facial expressions. While previous work completed the generation of personalized avatars in a few minutes, Lombardi et al.'s system requires more than a day to compute. The three-dimensional avatar is generated using a large number of high-resolution images from different angles and facial expressions, with an expensive hardware setup that generates a large amount of data for further processing. The system uses an encoder-decoder convolutional network similar to our work, but requires high-end hardware and is therefore only applicable in a laboratory environment.

A key component of the system of Lombardi et al. [Lom+18] and Wei et al. [Wei+19] is the use of Variational Autoencoders (VAEs). Both VAEs and GANs have been shown to be suitable for authentic face reconstruction. However, since the literature shows that VAEs combined with only L1 loss tend to produce more blurry results, we use GANs [Joh19] to produce more detailed results with higher visual quality. The foundational work on Generative Adversarial Networks (GANs) was introduced by Goodfellow and colleagues [Goo+14], with significant improvements later made by Radford and colleagues [RMC16]. Karras and his team [Kar+17] further advanced the technology to produce images of portraits that are virtually indistinguishable from actual photographs through the implementation of the "Progressive Growing GAN". However, Karras and colleagues noted a limitation of GANs: the lack of direct control over specific features of the generated images, such as hair color, facial expression, or gender, because the input is a latent vector with no clear connection to these attributes. Subsequent developments by Karras and his team refined the GAN architecture, allowing for the separation of high-level attributes (such as pose and identity) from random variations (such as freckles and hair). However, this model still did not provide explicit control over facial expressions.

Conditional GANs (cGANs) have demonstrated the ability to learn and reproduce specific input-output relationships that are understandable to humans. Mirza and Osindero [MO14] extended the generator and discriminator inputs with a label y, facilitating the generation of images within a specific category y. This approach to conditioning

GANs was further developed by Radfort et al. [RMC16] with the DCGAN and by Isola et al. [Iso+17] with the Pix2Pix GAN, where the traditional noise input vector z is replaced by a user-defined input vector. The absence of the noise vector eliminates the latent space Z (since $z \in Z$), and without compensating for the stochastic nature of the noise vector, the GAN risks simply memorizing training examples, leading to poor performance on inputs that differ from the training set, as observed by Isola et al. [Iso+17]. The integration of a U-net architecture [RFB15] with dropouts in the Pix2Pix GAN addresses the stochastic element and the lack of latent space within the generator. The Pix2Pix GAN's discriminator is fed both the input image x and either the generated image $y_{fake} = G(x)$ or the corresponding real image y_{real} from the dataset, following the cGAN concept [MO14] by evaluating the output of the generator with respect to the input. Unlike cGANs, the Pix2Pix GAN's discriminator outputs a matrix instead of a binary decision, with each element of the matrix corresponding to an $n * m$ region of the input image, allowing matrix-based abstract representations to condition the network for controlled output generation. This methodology was further refined by [Wan+18b] in the Pix2PixHD GAN to produce higher resolution and more detailed images. In this section of the thesis, we explore the adaptation of the cGAN framework, specifically the Pix2Pix and Pix2PixHD models, for specific applications in our field of study.

The systems of Thies et al. [Thi+15; Thi+18a], are compared to the aforementioned systems of Lombardi et al. [Lom+18], Wei et al. [Wei+19], and Raj et al. [Raj+21], require only an RGB-D [Thi+15] or only an RGB sensor [Thi+18a] for the recording, but, similar to our system, they can only generate a frontal viewing angle as a result, while the three aforementioned works can generate any viewing angle. From a stationary viewing position, the representation of the synthesized faces is photorealistic and of the same visual quality as the three systems mentioned above. The uncanny valley effect occurs only to a small extent (opinion of the author). Both systems by Thies et al. use a 3DMM (Basel Face Model [BV99]) and optimizes the model parameters with a CUDA-accelerated Gauss-Newton solver using an analysis-through-synthesis approach. This is similar to our approach, but instead of analysis-through-synthesis with a 3DMM, we train GANs to produce RGB-D data without any inductive bias such as a 3DMM.

A further iteration of this system [Thi+18b](called FaceVR) was extended to include a stereoscopic rendering of two images – each for one eye in an HMD. The stereoscopic rendering makes it possible to spatially display the person in VR. However, the position tracking of the HMD was not used. This means that the user cannot look around the person, but always sees the same camera position. A stereo rig consisting of two commercially available webcams is used to capture a person and forms the data basis for the stereoscopic face reanimation. The "Face2Face" method by Thies et al. [Thi+18a] is applied to each of these video streams and rendered for each eye.

All of the previously mentioned systems by Thies et al. did not yet use VAEs or GANs, and used traditional meshes and static textures or, in some cases, an appearance graph for the mouth region for visualization [Thi+18a]. The work of Thies and his teams was further enhanced visually through the use of GANs. In their work Deferred Neural Rendering [TZN19], Thies and team used a generative neural texture on the face model that was able to adapt to variations in facial expressions better than previous approaches. This system was further developed with another iteration, called "Neural Voice Puppery" [Thi+20], and focused on authentic reconstruction of the mouth area with a real-time text-to-speech approach.

GANs have shown in the past that they can perform robust image-to-image translation,

but this can lead to problems when multiple temporally related images are concatenated into a video. This can result in visual temporal incoherence, which can lead to a kind of image noise, especially in detailed regions, or to subjectively perceived visual inconsistencies in general. Elgharib et al. [Elg+20] solved this problem by inserting a sequence of images instead of only a single image into the U-net based Pix2Pix GAN. By inserting previous and subsequent images, the neural network can better learn temporal relationships. The system is also referred to as a video-to-video translation system.

7.3. Design Rationale

This section outlines several requirements and design principles that shape our approach to creating a more immersive and authentic telepresence experience. The design of the system focuses on the use of open-source software and low-cost hardware to automatically capture and control a 3D facial avatar in real time. Our goal is to capture and reconstruct personal expressions without the need for manual modeling or generic expression templates, using only a standard RGB-D sensor such as Microsoft Kinect or Intel RealSense.

Our system is tailored for real-time performance on consumer-grade computers with a single GPU, making the system accessible to mainstream users. To achieve this, we use depth maps instead of meshes or voxels, and optimize the data structure for faster processing by neural networks in an unsupervised learning strategy.

For immersive MR telepresence, it is sufficient to reconstruct only the user's face, since areas not hidden by the HMD can be captured and transmitted "from real reality" by external RGB-D sensors in the room. This eliminates the need for full head reconstruction and allows the face to be integrated into a combined point cloud consisting of data from multiple sensors distributed throughout the room. Our approach therefore focuses on producing a realistic facial animation with minimal computational cost.

7.4. System Overview

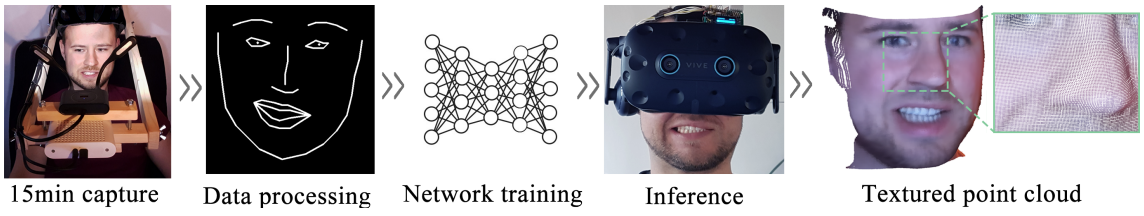


Figure 7.1.: *Our conceptual pipeline: First, we capture several RGB-D images using a helmet camera mount. These images are processed and serve as the input data for our GAN. After training, the GAN produces textured point clouds in real time. Image from [Lad+25]*

In the development of neural network based approaches, there are four pillars that should receive the main attention: 1.) The quality of the training data, 2.) the architecture of the neural network, 3.) the loss function, and 4.) the training parameters. These four pillars will be discussed in the following sections of this chapter. The development of our face-synthesizing GAN progressed through three major iterative stages, each improving on the previous version.

Research prior to 2017, such as Wu et al. [Wu+16b], has shown that three-dimensional

data represented as a voxel-based structure is associated with high training and execution times and is not suitable for interactive framerates. However, since we are aiming for a telepresence system with interactive frame rates, an RGB-D-based solution was pursued, which reduces the three-dimensional problem to a simplified 2.5D approach similar to a geometric projection. The advantage of our specific use case is that this approach works because human faces do not require a full 3D representation, as faces do not have overlapping or occluding surfaces that would be relevant for an authentic representation. The advantages of an RGB-D storage format lie in the compact representation of the data as a point cloud and the ability to adapt previous and successful RGB-based methods. However, one of the main advantages is that the rendering of the points can be done very efficiently and quickly by a graphics card, since point rendering is a hardware-accelerated computational task. Approaches to hardware-accelerated point cloud rendering were introduced by Zheng et al. [Zhe+23].

This short section outlines the design and operation of our proposed system, illustrated in Fig. 7.1: Our pipeline starts with a ten to 15 minutes recording session to collect a personal RGB-D dataset. This data is pre-processed by an automated process. We extract a **Facial Landmark Map (FLM)**, shown in Fig. 7.2, for each RGB image and store it alongside the original RGB and also D image. These FLMs contain 70 facial landmarks and translate the facial expressions from the RGB image into a binary format. Our GAN is trained using the pairs of RGB-D images and their corresponding FLMs. In this way, the FLMs are the intermediate state between the real facial expression of the individual in the training set and also while wearing the head-tracking HMD mentioned above. During preprocessing of the data set, we create FLMs, and later, when the person is wearing the HMD, we concatenate the eye, brow, and lower face tracking information into an FLM.

Each individual's data requires training the GAN from the beginning. We do not use a 3DMM [BV99] and do not learn cross-person correspondences. After training, the system is suitable for real-time applications. In the inference stage, we use only the trained generator module of the GAN. In a virtual reality setup, users would wear a face-tracking head-mounted display capable of generating an FLM in real time, such as proposed in Chap. 5. The FLM is fed into the generator module of our GAN. The GAN then generates both RGB and D images of the "learned" individual based on the FLM. Finally, we merge these generated RGB and D images to create a textured point cloud with interactive refresh rates. Although the GAN is much faster than the usual refresh rates of HMDs with about 250 frames per second, the system is limited by the speed of the tracking systems, which in our case is 30 fps for the mouth area and 10 fps for the eyebrows.

Figure 7.2.: *Visual examples of Facial Landmark Maps (FLMs). Images by René Ebertowski.*

It is difficult to do direct evaluations of the system with others because it is one of the first.

There are a few comparable systems, as described in Related Work, but they either require special laboratory hardware that is not easily accessible, or the software is not published or open source, as in the case of the Gauss Newton Solver for the Basel Face Model by Thies et al. [Thi+15; Thi+18a]. Today in 2024, face trackers with good performance exist in the form of the Video Head Tracker (VHT) by Grassal et al. [Gra+22], DECA by Feng et al. [Fen+21], EMOCA by Radek et al. [DBB22], or the Metrical Tracker by Zielonka [ZBT22b], but this was not the case at the time of developing the pipeline introduced in this chapter. Therefore, it was not possible to reproduce the results of the other researchers.

7.5. First Prototype: The Foundation Network and Data Acquistion Pipeline

The first prototype laid the groundwork by using the Pix2Pix Patch-GAN by Isola et al. [Iso+17] and extending it with a fourth channel to derive a depth map alongside the RGB output, resulting in an animated texture point cloud of a face for telepresence scenarios that can be driven by the aforementioned face-tracking HMD and is capable of conveying nonverbal facial communication cues. The second prototype builds on this foundation and extends it with new ideas from seminal research. The architecture, loss functions, hyperparameters, and training process were refined to increase the learning efficiency and effectiveness of the model.

As mentioned in the introduction of this section, an RGB-D dataset of a specific individual is fundamental to the training process. The data is acquired using a Microsoft Azure Kinect RGB-D camera mounted in a fixed position on a helmet mount, as shown in Fig. 7.3. We chose the Azure Kinect because of its good data quality, but our approach is generic and other RGB-D cameras such as Intel RealSense or PMD sensors could be used for this task.

As learned from previous experiments, by using the helmet mount from Fig. 7.3 to help with a stable position of the face in the images, we saw not only a much better reconstruction quality, but also a faster convergence of the GAN to an acceptable loss (measured by SSIM [Zho+04] and LPIPS [Zha+18]). Furthermore, by using the helmet mount, we were able to reduce the capacity of the neural network in terms of layers and neurons. Reducing the complexity of the neural network is a critical part as we strive for interactive frame rates. Our goal is to ensure that the face, and therefore the landmarks in the FLM, are in the same position across the entire dataset, without changes caused by head rotation, for example. The head rotation is later transferred to the virtual face by the tracked rotation of the HMD. This approach allows us to 1) significantly minimize the size of the data set, 2) reduce the size of the neural network and save training time, and 3) achieve interactive frame rates.

Our initial approach in building the first prototype was to place the RGB-D sensor on a table and let the captured person move his head freely during the recording. We found that the network had to be large to achieve reasonable reconstruction quality. The helmet mount helps to minimize the entropy of the dataset by eliminating varying distances between the sensor and the face (z dimension in sensor space), face positions (x and y dimensions in image space), and head rotations.

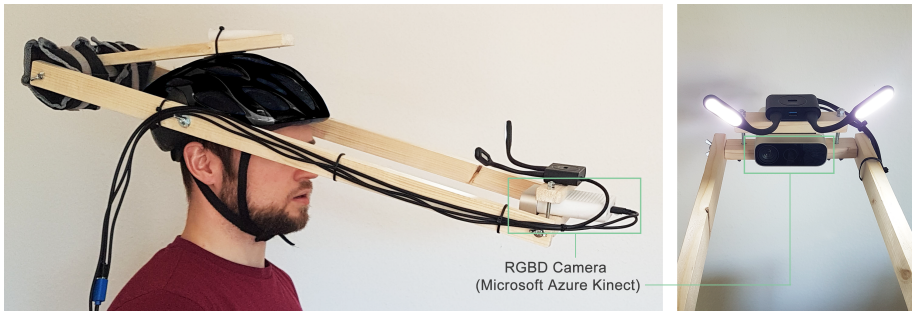


Figure 7.3.: *A low-cost helmet mount helps reduce variance in the data set, resulting in shorter training times, smaller neural networks, and higher reconstruction quality. Image from [Lad+25; Lad+20b; LPG20].*

In order to achieve a reasonable quality of facial identity and expression reconstruction, the dataset for an individual includes approximately 26 facial expressions (captured multiple times), 20 phonetically balanced sentences, and approximately 5 minutes of speech. This comprehensive approach ensures a good balance between recording time, training time, and reconstruction results. In several experiments, we have found that about 600–700 RGB-D and infrared images is a good compromise between training time and final visual quality. The following section describes the steps necessary to generate a dataset for training our GAN.

7.5.1. Training Data Set Acquistion and Processing

The main goal of the dataset processing is to minimize the amount of data to what is necessary for a good tradeoff between visual quality, acquisition, and training time. It is essential to preserve the original quality of the images, as the generator within the GANs relies on learning the transition from FLM to both RGB and depth images using the dataset. Therefore, maintaining high quality images in the dataset is critical for successful training of the model.

1.) Face Landmark Detection
The basis for the following steps is the determination of facial landmarks and their conversion into Facial Landmark Maps (FLM, shown in Fig. 7.2), which serve as crucial data for controlling the GAN output and the person's facial expressions. These landmarks are determined for each RGB image and also correspond to the depth map in the dataset. As described in the chapter 5.1 "Related Work" of the Face Tracking HMD, there are several methods to perform landmark detection. We experimented with the implementation in the DLib library [Kin09] with "Ensemble of Regression Trees" by Kazemi and Sullivan [KS14], but after some testing switched to the Facial Alignment Network (FAN) by Bulat and Tzimiropoulos [BT17]. Although the FAN is much slower and not capable of real-time frame rates, it shows more realistic landmark detection results. After the landmark detection process, we obtained 68 landmarks for each RGB-D frame in our dataset, as shown in Fig. 7.4.

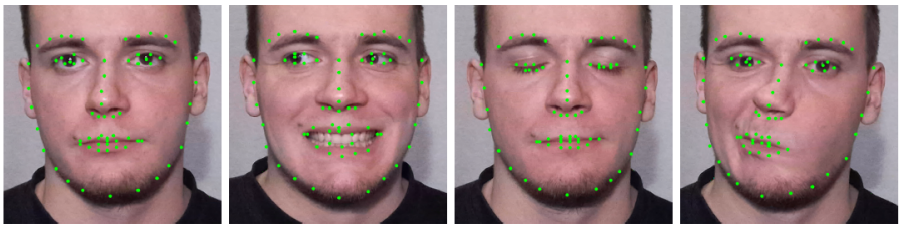


Figure 7.4.: *Examples of facial landmark detection results for the dataset. Images by René Ebertowski.*

2.) Gaze Tracking

The FAN is not able to determine the gaze direction within an image, which is essential for the reconstruction phase of the GAN to accurately reconstruct the person's eyes and gaze direction. In a first experiment, we implemented our own eye gaze tracking algorithm, inspired by Xiong et al. [Xio+14], which works on infrared images provided by the Kinect sensor. The infrared images are produced by the depth sensor of the Azure Kinect. The accuracy of this solution was mediocre. In particular, a lot of tracking noise made the final avatar look unbelievable and not very authentic, as the pupils trembled. Filtering and smoothing the data was difficult due to the high level of noise. The next iteration uses a Tobii 4C eye tracker attached to the helmet mount and calibrated with the RGB-D image from the Kinect, as shown on the left in Fig. 7.5 below the Azure Kinect sensor.

Figure 7.5.: *A Tobii 4C eye tracker was used to reliably track the user's gaze. The eye tracker (black thin bar) is mounted below the Azure Kinect on the left side of the image.*

The additional eye tracker made the helmet mount significantly heavier, but provided better tracking results than the first prototype, especially in terms of tracking noise, and was therefore an acceptable trade-off to improve visual reconstruction quality.

3.) Crop

Based on the minimum and maximum pixel positions per axis of the face landmarks, an axis-aligned bounding box is created around the face. Additionally, we add a margin around this bounding box of 15% of its edge length to improve the final visual image quality of the trained GAN. In our initial experiments, we only used a narrow bounding box, which led to poor visual results when the mouth was wide open or the eyebrows were raised. The larger range of landmarks on the y-axis changed the aspect ratio of the entire face, which caused the landmarks on the x-axis to be compressed. This means that the face became smaller in the FLM and many landmarks left their "usual" position. Usual means that landmarks usually stay in a certain area in a large part of the dataset. GANs learn a stochastic distribution and deliver poor results in such "edge cases" where there

is less training data available. To ensure that the size of the face does not change when the mouth is opened or the eyebrows are raised, we added the margin of 15% of its edge length. Theoretically, we would not need a margin on the left and right sides, but GANs are better suited to process and generate square images (or at least edge lengths of a power of two) than images with different or arbitrary edge lengths.

4.) Normalize, Clip, Mask, Resize, and Sort

To improve the data quality for GAN training, we perform a histogram normalization on all RGB-D images to get a better distribution of pixel values over the whole range. This process is straightforward for the color information, but more complicated for the depth values because it requires an additional preprocessing step:

The RGB-D sensor captures images that store color information as 3-channel PNG files (8-bit RGB) and depth information as 1-channel PNG files (16-bit grayscale). In our initial experiments, we let the neural network predict 16-bit values for the depth pixels. Since the numerical precision of the neural network is limited when passing data through multiple layers, we obtained a "noisy" reconstruction and a "banding effect" in the predicted depth maps, as shown in Fig. 7.6.

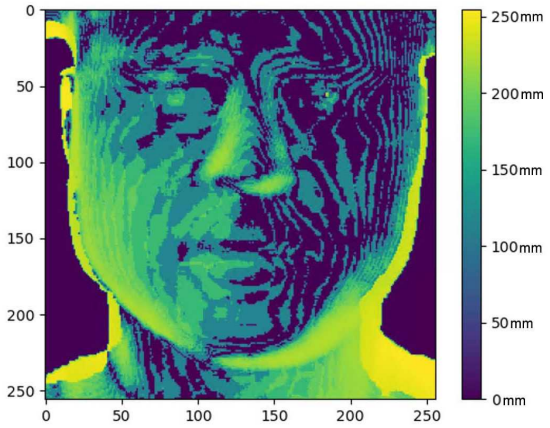


Figure 7.6.: *The image shows "banding" artifacts due to 16 bits per depth pixel when using the full depth resolution of the RGB-D sensor with our GAN architecture. Reducing the depth range from 16 bits (65535 mm) to 8 bits (255 mm) is critical to the visual quality and size of the neural network.*

The 16-bit depth resolution works in millimeters, so the full depth range is 0 to 65535 millimeters. However, the area of the face we want to reconstruct is much thinner. With an 8-bit resolution, we would reduce the area to 255 millimeters, which is sufficient for our application since the depth of the frontal area of a face is smaller than that. By drastically reducing the depth data in our dataset from 65535 to 255 millimeters, we achieve a much higher visual quality of the depth map. On the one hand, the depth noise is significantly reduced, and on the other hand, the face is much better resolved in terms of depth resolution and detail, and does not show any banding effects. Within this downsampling of the depth data, it is important to note that the depth scale is normalized and the actual absolute depth information is rejected. Please note that previous solutions by other researchers store the depth data in a 32-bit EXR image format, which results in much more data [Cas+16].

By clipping and reducing the depth resolution, the values in the background behind the face are removed from the depth map. In the RGB images, however, there is still color

information in the background. Since we have a direct correspondence between the RGB and D images, we can also set the color information to 0 wherever there are no depth values. Let p be a pixel, then the equation of the function is:

$$d_{mask}(p) = \begin{cases} 255 & \text{if } p_d > 0 \\ 0 & \text{else} \end{cases} \tag{7.1}$$

In this way, we cut out the face from the color images, which has another advantage for training the GAN, as it only receives relevant face information. The background does not need to be reconstructed in our application scenario. Furthermore, all images have been resized to 256 pixel edge length.

The final step of the preprocessing consists of a randomized division into a training and a test dataset. In our experiments, 85% for training and 15% for test turned out to be a good ratio. Finally, we obtain a dataset with images and structure that look like the following figure:

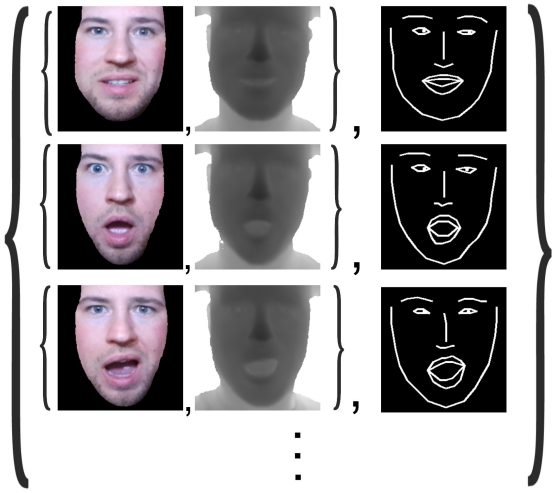


Figure 7.7.: *The final data set: For each pair of acquired RGB-D images, a Facial Landmark Map (FLM) is created. During training, the GAN learns the paired image-to-image mapping from a FLM to an RGB-D image. Image from [LG19b].*

7.5.2. Network Architecture and Training

Our initial prototype was developed using the Pix2Pix-GAN framework created by Isola et al. [Iso+17], which is heavily based on the U-Net architecture created by Ronneberger et al. [RFB15]. We refer the reader to the appendix of the paper by Isola et al. where the architecture of Pix2Pix is explained in detail. In our adaptation of the original model, our generator module produces not only three RGB channels, but also depth data for a fourth channel. In our initial experiments, we modified the discriminator to process eight input feature maps, as shown in Fig. 7.8 on the left. The first four of these maps represent the channels of an RGB-D image, and the last four maps contain the corresponding FLM as a grayscale image. We entered the FLM four times with the hypothesis that the training would be more balanced and the network would produce better results [LPG20]. In further experiments, however, we found that this was not the case and that it only increased the training time, but had no effect on the results. Therefore, in a further iteration of the prototype, we added only five feature maps to the discriminator instead of eight, as shown

in Fig. 7.8 on the right. Formally speaking, our pipeline looks like this: Let $\mathbf{L} \in \mathbb{R}^{2\times 70}$

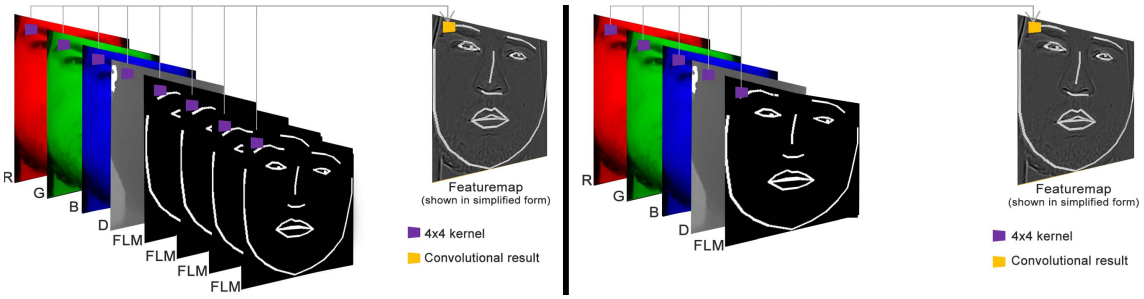


Figure 7.8.: *Example convolution for the discriminator input. Each RGB-D channel and the FLM are individually weighted. For the first iteration of our neural architecture, we entered the FLM four times (shown on the left) with the hypothesis that the training would be more balanced and the network would produce better results. However, this was a false assumption and only increased the training time without improving the visual results. In our final version (shown on the right), the FLM is fed into the discriminator only once. Images from [LPG20] and[Lad+20b]*

be the FLM, which contains 70 facial landmarks in image coordinates x and y, we can illustrate the procedure as

$$\mathbf{T}_t, \mathbf{D}_t \leftarrow \mathcal{G}\phi(\mathbf{L}_t) \tag{7.2}$$

where $\mathcal{G}\phi$ is the generator of our GAN which produces $\mathbf{T} \in \mathbb{R}^{256x256}$ as an RGB texture and $\mathbf{D} \in \mathbb{R}^{256\times 256}$ as depth map at time instant t. A rendered image $\mathbf{R} \in \mathbb{R}^{w\times h}$ of the face can be rendered from a rasterizer $\mathcal{R}$:

$$\mathbf{R}_t \leftarrow \mathcal{R}(\mathbf{T}_t, \mathbf{D}_t, \mathbf{C}_t), \tag{7.3}$$

where $\mathbf{C}$ denotes the camera position and projection function. Please note, the rasterizer is hardware accelerated and does not involve any raycasting techniques or other volume rendering methods. An OpenGL based rendering pipeline is used based on OpenSceneGraph [OSG24].

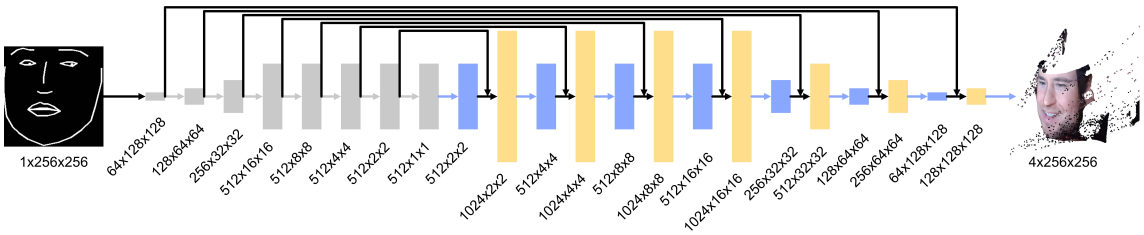


Figure 7.9.: *The generator takes a 1-channel FLM and generates a 4-channel RGB-D image that is displayed as a textured point cloud. The Fig. shows the dimensions of the tensors in the network. The gray blocks are a Conv2D with InstanceNorm+ReLU layers and blue and yellow are ConvTranspose2D layers+InstanceNorm+LeakyReLU layers. Image from [Lad+20b].*

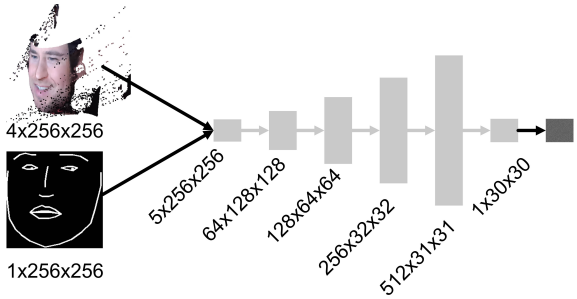


Figure 7.10.: *The discriminator receives an FLM and an RGB-D image and has to decide if it is real or generated by the generator. Image from [Lad+20b].*

The dimensions of the passed tensors of the generator and discriminator of our first prototype are shown in Fig. 7.9 and 7.10. We basically used the original architecture of the U-Net based Pix2Pix with blocks from Convolution-Norm-ReLu. We used the usual parameters like kernel size of 4, strides of 2 and a drop out rate of 50% with up convolutions with normal ReLU and down convolutions with LeakyReLU with a slope of 0.2. The last layer is a Tanh function layer, which has an output range of $[-1, 1]$ and is therefore particularly suitable for image generation. Compared to the original generator from Pix2Pix, we added a fourth feature map to the output of the generator to be able to output a depth channel. In addition, the discriminator receives five feature maps instead of six, each consisting of two images with RGB channels. In our approach, the first four feature maps correspond to the channels of an RGB-D image, the remaining one contains the corresponding FLM. Our first prototype has the following objective to train the generator, which corresponds to the objective function of Pix2Pix:

$$G = \arg\min_G \max_D \mathcal{L}_{GAN}(G, D) + \lambda \mathcal{L}_{L1}(G) \tag{7.4}$$

The losses are explained in more detail later in this chapter in section 7.6.1. For each individual, both the generator and the discriminator are trained from the beginning. The training procedure is the same as for the Pix2Pix network. Before training, all weights are set to random values, following a Gaussian distribution with a mean of 0 and a standard deviation of 0.02. Training uses a batch size of 1 and runs for 100 epochs. The training of both networks starts with a learning rate of 0.0002, which is linearly reduced to zero over the last 70 epochs. The reduction of the learning rate of the discriminator, represented by $Loss_D = (Loss_{Dreal} + Loss_{Dfake}) * 0.5$, slows down its learning speed compared to the generator. This setting is crucial because the discriminator is initially too efficient. Slowing down its learning rate is essential to give the generator enough opportunity to learn how to accurately generate the desired face.

7.5.3. Results and Evaluation

Fig. 7.11 illustrates the facial expressions captured by the face-tracking head-mounted display (HMD) from Chap. 5. The first column shows the face landmark masks (FLM) generated by the HMD, while the second column shows the output of our GAN. The expressions were maintained when the face tracking HMD was removed, the helmet mount (shown in Fig. 7.3) was applied, and an image was captured, shown in the third column. This setup allows a direct comparison between the generated and ground truth images.

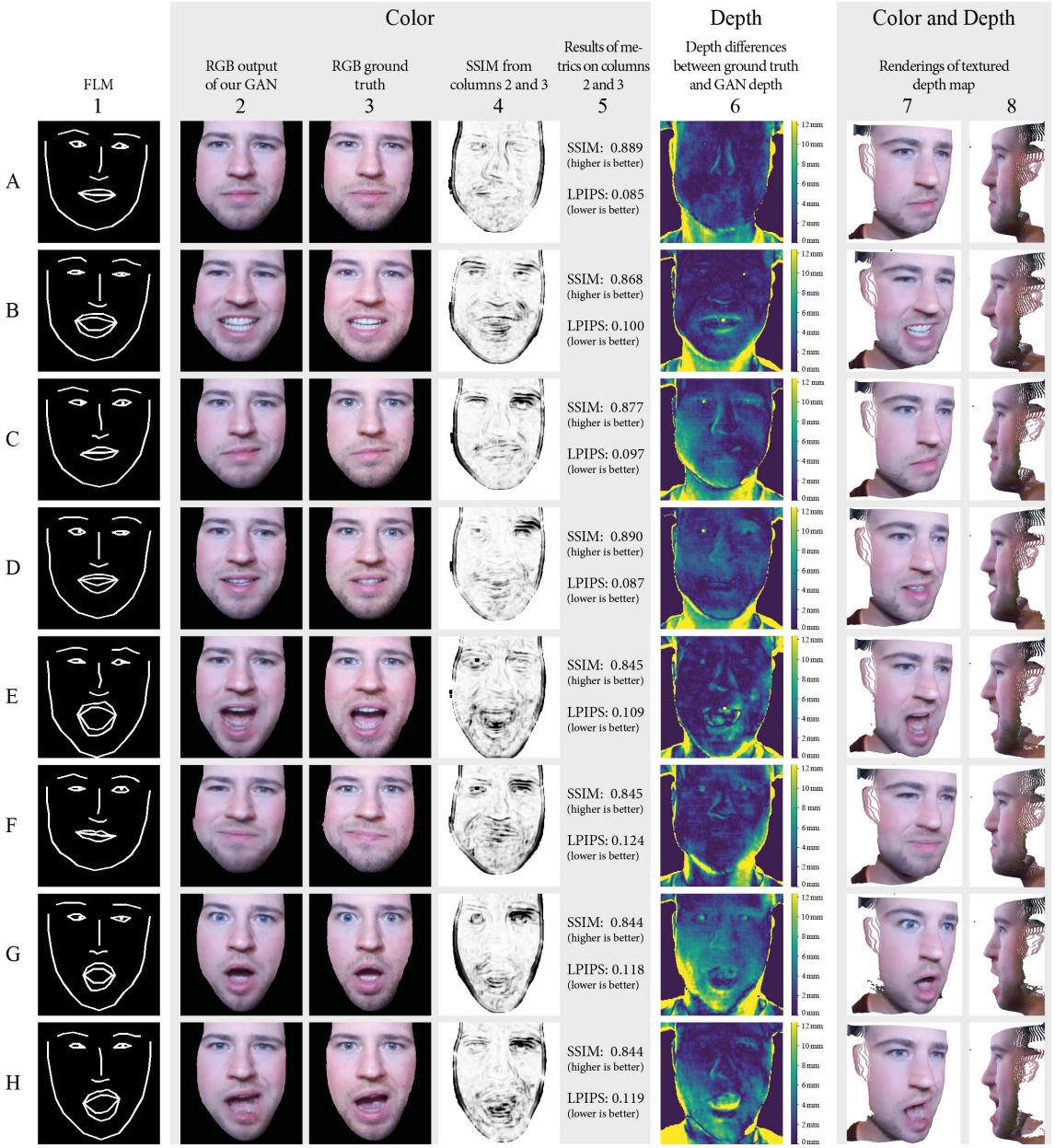


Figure 7.11.: *Results: Please zoom in for details. Column 1 shows the Facial Landmark Maps (FLM) provided to our GAN, which were not included in the training dataset and were generated using the face-tracking HMD. The second column shows the images produced by our GAN based on these FLMs. The third column contains the actual images captured by the helmet-mounted RGB-D camera, which represent the actual expressions. The participant maintained these expressions while the face-tracking HMD was removed and replaced with the helmet mount (Fig. 7.3). This method yields comparable results because the helmet mount allows RGB-D images to be captured under identical conditions as the training dataset. The fourth column shows the differences, measured by SSIM [Zho+04], between the images from columns 2 and 3, with darker areas indicating larger differences. Column 5 lists the computed values for SSIM [Zho+04] and LPIPS [Zha+18], using version 0.1 from GitHub. Column 6 compares the depth information between the ground truth (helmet-mounted camera) and the synthesized images. Columns 7 and 8 show renderings of the textured depth maps generated by our GAN. Image from [Lad+20b].*

To evaluate the differences between the images in columns 2 and 3, we employed the "structural similarity" (SSIM) [Zho+04] and "learned perceptual image patch similarity" (LPIPS) [Zha+18] metrics. We isolated the face region for these comparisons by using depth values to define this region and ignoring all other pixels.

As shown in Fig. 7.11, the individual's identity remains clearly recognizable and authentic. Although the images generated by our GAN are less sharp and detailed compared to the original, they still prominently display subtle personal features, such as the birthmark on the left cheek near the mouth and nose (please zoom in for details). The beard area is noticeably less sharp. The SSIM results are similar to what one would expect from a JPEG compressed to a quarter of the original file size of the images in the third column.

Depth discrepancies in the face area generally remain below 5 mm, as shown in column 6. We present both the unaltered depth image from the Kinect and the raw output from our GAN, without any filtering or smoothing. In columns 7 and 8, we applied erosion and clipping techniques to remove the background.

7.5.4. Quality of Expressions

Our system is capable of recognizing and reconstructing a wide range of expressions. However, human sensitivity to small variations in facial expressions means that even small tracking and reconstruction errors can result in slightly different expressions being conveyed. For example, a small change in eyebrow height in row E appears to create a more negative or critical expression than originally captured by our system. In addition, the mouth animation during speech is not accurate enough for reliable reconstruction, falling short of other SOTA approaches such as those of Olszewski et al. [Ols+16] and Wei et al. [Wei+19]. In addition, expressions not included in the training set lead to blurred results and noticeable degradation, especially around the mouth in row H.

7.5.5. Scalability

As shown in Fig. 7.12, our prototype was tested on 5 individuals and did not require any specific user adjustments to our data pipeline, except for the need to collect and process a new dataset for each person. The data processing is end-to-end and largely automated, requiring only a few console commands. Furthermore, each individual was able to manipulate their avatar with a similar range of expressions.

Figure 7.12.: *Results from the use of a face-tracking HMD on five individuals displaying different facial expressions. Our system is effective regardless of facial structure or individual expression, and does not require additional manual adjustments for new users. The images show different erosion and clipping settings that can either reveal or hide the neck and upper chest areas. One notable limitation we identified is that highly reflective surfaces, such as eyeglasses, lead to reconstruction errors, visible as black areas in the two rightmost images. Image from [Lad+20b].*

7.5.6. Performance

All measured training and execution times were performed in PyTorch without any further acceleration such as half/mixed presicion (Nvidia APEX or PyTorch's AMP/autocast) or "traced models" with TorchScript's just-in-time compilation. Training a dataset of 600 samples takes about 19 hours on an Nvidia GeForce RTX2080 using PyTorch 1.6 and Python 3.7. For inference and rendering tasks, LibTorch 1.6 is used along with an OpenGL-based C++ rendering environment based on OpenSceneGraph [OSG24]. The generation of an RGB-D image (8 bit per channel, 256 by 256 pixels) from an FLM (8 bit, one channel, 256 by 256 pixels) is completed in 1.05 ms. An additional 1.3 ms is required for stereoscopic rendering of the textured point cloud, allowing a frame rate of over 90 fps (equivalent to 11.1 ms per frame) for the HMD. In contrast, processing the facial tracking on the HMD takes longer. However, the frame reconstruction rate for the face is limited to 30 fps (equivalent to 33.3 ms per frame) due to the frame rate limitations of the miniature cameras built into the face tracking HMD.

7.6. Second Prototype: Experiments to Improve Visual Quality

In this subsection, we conduct experiments aimed at improving the visual quality of our framework, focusing on both resolution and overall image quality while maintaining speed. We incorporate SOTA methods into the architecture, mainly the discriminator network, and use a variety of metrics and losses to obtain and evaluate improvements in image quality. We conducted several experiments using as inspiration the Pix2PixHD framework [Wan+18b], an advanced version of the original Pix2Pix GAN [Iso+17]. While it delivers images of much higher resolution and quality, its inference speed is slower compared to Pix2Pix because its architecture is more complex. The generator of Pix2PixHD is capable of real-time processing only on high-end hardware from 2020. Even then, the GPU is nearly maxed out, leaving minimal capacity for additional tasks like face tracking and rendering the rasterizer. As a result, we kept the Pix2Pix framework as our foundation, and gradually incorporated features from the Pix2PixHD framework along with additional improvements from various works until we achieved a balance between image quality, training time, and inference speed suitable for interactive applications.

The generation of the RGB-D image is real-time, and the rendering by the hardware-accelerated rasterizer did not show significant latencies, as the first prototype in the last chapter proved. However, the visual quality of the results required improvements such as higher resolution and more detail. We tried adding more outermost layers to the network of our first prototype to scale the output to 512×512 pixel, but we experienced inauthentic reconstruction results, especially in image areas with high frequencies such as facial hair or teeth. The current network architecture tends to generalize and over-smooth these details, resulting in an unrealistic and uncanny appearance. To address these issues, we build a test environment to test different conditions such as network architecture and loss function.

7.6.1. Losses as Conditions of the Experiment

Several techniques are available from recent research published in papers after the release of Pix2Pix. In the following, we list seven losses and three different architectures with

upstream mapping networks that we have incorporated into the objective function of our GAN training in 18 different experiments to achieve better visual quality while maintaining real-time speed.

The losses are:

1. *Generative Adversarial Network Loss (GAN-Loss)*

   This is the typical loss used in GANs, formulated to handle the adversarial nature of the training process between the generator and the discriminator. This loss, used also by Isola et al. [Iso+17], is usually a Binary Cross Entropy Loss inside a sigmoid function (called BCEWithLogitsLoss() in PyTorch). The generator is trained to minimize this loss by trying to fool the discriminator into "thinking" that the generated images are real. The discriminator is trained to maximize this loss by getting better at distinguishing real images from generated ones. The adversarial loss can be expressed as:

   $$\mathcal{L}_{\text{GAN}}(G, D) = \mathbb{E}_{(\mathbf{x},\mathbf{y})}[\log D(\mathbf{x}, \mathbf{y})] + \mathbb{E}_{\mathbf{x},\mathbf{z}}[\log(1 - D(G(\mathbf{x}, \mathbf{z})))] \tag{7.5}$$

   where $\mathbf{x}$ is the feature map (in our case the FLM), $\mathbf{y}$ is the corresponding natural image, and $\mathbf{z}$ is a noise vector. While in traditional conditional GANs the noise is injected as input into the network, in our architecture the noise is injected by the dropouts on multiple layers during training and testing time [Iso+17]. Note that we use instance normalization instead of batch normalization because we only backprop each step over a generated image, which makes the formula shorter.

2. *Least Squares Generative Adversarial Network Loss (LSGAN-Loss, MSE-Loss or, most common, L2 Loss)*

   As shown above, traditional GANs used the Binary Cross Entropy Loss and sigmoid. Mao et al.[Mao+17] found that a Least Squares Loss is more stable, avoids mode collapse (the generator learns to produce a limited variety of outputs), and helps avoid vanishing gradients, leading to better results overall. Compared to the GAN loss above, the LSGAN loss does not use a sigmoid layer. According to Isola et al. [Iso+17], a combination of L2 loss and GAN loss tends to produce blurry results compared to using an L1 loss. However, follow-up work has shown that L2 in combination with other losses leads to good and sharp results, as we will show in the following experiment. This could be due to the fact that images with higher resolution are generated in our experiment. In this context, an L2 loss might work better compared to Binary Cross Entropy Loss. According to Mao et al., the least squares loss function may penalize samples far from the decision boundary more than the binary cross-entropy loss. The objective of an LSGAN is:

   $$\begin{aligned} \min_D V_{LSGAN}(D) &= \mathcal{L}_{LSGAN\_D}(D, G) \\ \min_G V_{LSGAN}(G) &= \mathcal{L}_{LSGAN\_G}(D, G) \end{aligned} \tag{7.6}$$

   with the loss function as follows:

   $$\mathcal{L}_{LSGAN\_D}(D, G) = \frac{1}{2}\mathbb{E}_{\mathbf{x},\mathbf{y}}\left[(D(\mathbf{x}, \mathbf{y}) - 1)^2\right] + \frac{1}{2}\mathbb{E}_{\mathbf{y},\mathbf{z}}\left[(D(G(\mathbf{y}, \mathbf{z}), \mathbf{y}))^2\right] \tag{7.7}$$

   $$\mathcal{L}_{LSGAN\_G}(D, G) = \frac{1}{2}\mathbb{E}_{\mathbf{y},\mathbf{z}}\left[(D(G(\mathbf{y}, \mathbf{z}), \mathbf{y}) - 1)^2\right] \tag{7.8}$$

where $\mathbf{x}$ is the feature map (in our case the FLM), $\mathbf{y}$ is the corresponding natural image, and $\mathbf{z}$ is a noise vector. While in traditional conditional GANs the noise is fed into the network as input, in our architecture the noise is injected by the dropouts on multiple layers during training and testing time [Iso+17].

3. *Adversarial Loss (Discriminator Loss)*

   In our following study, this loss indicates that a discriminator evaluates the generator's output and is backpropagated. This can be either the traditional binary cross entropy loss or the LSGAN loss. If there is no cross for LSGAN in Tab. 7.1, but one for Adversarial Loss, then this means that the Binary Cross Entropy Loss is used. In this case, it is comparable to the training of an autoencoder.

4. *Mean Absolute Error (MAE, more common: L1 Loss)*

   It is calculated between the real images from the training dataset and the synthesized images.

$$\mathcal{L}_1 = |x - y| \tag{7.9}$$

5. *Multiscale Discriminators*

   This technique uses multiple discriminators instead of a single one to calculate the GAN loss. These discriminators operate at different resolutions of the input images, allowing both local and global structures in the patch to be evaluated. The method is typically applied to GAN loss and Feature Matching Loss [Wan+18b]:

$$\min_G \max_{D_1,D_2,D_3} \sum_{k=1,2,3} \mathcal{L}_{\text{GAN}}(G, D_k) + \sum_{k=1,2,3} \mathcal{L}_{\text{FM}}(G, D_k) \tag{7.10}$$

   Note that such a multi-resolution pipeline is a well-established method in computer vision and is often used for image compression, generation, or analysis [BA83].

6. *Feature Matching Loss*

   This loss was introduced by Wang et al. [Wan+18b]. It improves image quality and increases the stability of the training process for G by extracting features across different layers of the discriminators and aligning the features of the real training images with those of the generated images. The resulting structural difference is penalized, ensuring that G produces images that induce identical features in the discriminators as those found in the images of the training dataset. The computation of this loss is described as follows:

$$\mathcal{L}_{FM}(D_k, G) = \mathbb{E}_{(x,y)} \sum_{i=1}^{T} \frac{1}{N_i} [||D_k^{(i)}(x|y) - D_k^{(i)}(x|G(x))||_1] \tag{7.11}$$

   where $D_k^{(i)}$ is the feature of discriminator D_k in layer i, T is the total number of layers of D_k, and N_i is the number of elements in layer i. $||\cdot||_1$ is the L1 loss.

7. *Learned Perceptual Image Patch Similarity Loss (LPIPS Loss)*

   The Learned Perceptual Image Patch Similarity (LPIPS) loss, introduced by Zhang et al. [Zha+18], is computed using a pre-trained CNN. This loss differs from traditional methods by using the deep features extracted from pre-trained CNNs to assess the perceptual similarity between images.

   Unlike traditional metrics that often rely on pixel-wise comparisons, LPIPS more effectively captures the intricacies of the human visual system by considering high-level features and patterns. As a result, it more closely aligns with human judgments of visual similarity, providing a more nuanced and accurate measure for tasks such as image compression, super-resolution, and synthesis quality assessment. Unlike the previous losses, this loss is unique because it cannot be explicitly mathematically formulated, but is instead implicitly learned by a neural network.

   Please note that Pix2PixHD and other works of this time use the VGG-perceptual loss of Johnson et al. [JAF16a] to increase the visual fidelity of the generated images. We did not use the VGG loss because there is strong evidence that LPIPS leads to better results in several deep learning vision tasks [JYK20; Gru+23].

8. *Cycle Consistency Loss (Cycle-Loss)*

   Cycle loss was originally introduced by Zhu et al. [Zhu+17] with the *Cycle GAN*. It realizes an unsupervised learning of a mapping between two distributions of images. The goal of these tasks is to learn a mapping between two different domains without having explicit paired examples showing how certain elements from one domain should translate to the other.

   There are two mappings: $F : A \to B$ and $G : B \to A$, where A and B are two different domains. The goal is to make F and G learn to translate images from A to B and from B to A, respectively, in such a way that an image translated from its original domain to the target domain and back is as close as possible to the original image. It can be expressed as the sum of two components: For an image $x \in A$, the cycle consistency loss ensures that the image translated to B by F and then back to A by G should be similar to the original image x. This is represented as

$$\mathcal{L}_{\text{cycle}}^{A \to B \to A} = \mathbb{E}_{x \sim p_{\text{data}}(x)} \left[\| G(F(x)) - x \|_1 \right] \tag{7.12}$$

   Similarly, for an image $y \in B$, the cycle consistency loss ensures that the image translated to A by G and then back to B by F should be similar to the original image y. This is represented as

$$\mathcal{L}_{\text{cycle}}^{B \to A \to B} = \mathbb{E}_{y \sim p_{\text{data}}(y)} \left[\| F(G(y)) - y \|_1 \right] \tag{7.13}$$

   where $\mathbb{E}$ denotes the expectation over the distribution of data samples (p_{data}). Combining these, the total Cycle Consistency Loss is

$$\mathcal{L}_{\text{cycle}} = \mathcal{L}_{\text{cycle}}^{A \to B \to A} + \mathcal{L}_{\text{cycle}}^{B \to A \to B}. \tag{7.14}$$

7.6.2. Dataset

In the following, the same dataset was used for all experiments. It comprises 1500 pairs of data, each including RGB-D images, feature maps, and a list of facial landmark positions.

This dataset for the second prototype has more than twice as many data pairs as the dataset used for the first prototype. The hypothesis was that more targeted losses would shorten the training time, and thus there should be the potential to process more data with acceptable training times. In general, in the field of neural networks and especially with GANs, results tend to improve with more data [Kar+17; KLA18; Iso+17]. The images in this dataset have a resolution of 512×512 pixel, which is four times higher than in the previous prototype. All other preprocessing steps for the captured data explained above remain unchanged, except that we have increased the margin around the face landmark bounding box from 15% to 20%, as can be seen in Fig. 7.13.

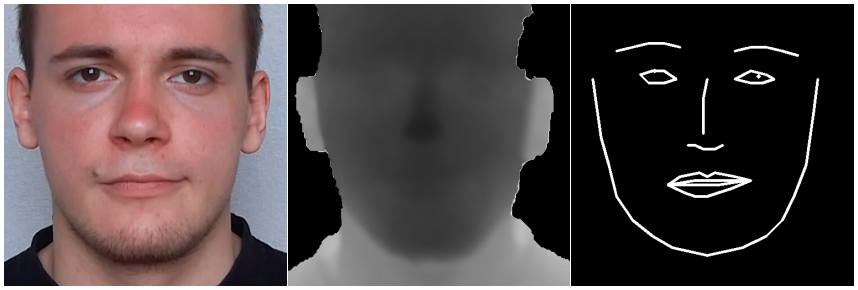


Figure 7.13.: *We increased the margin of the face landmark bounding box from 15% to 20% to better reconstruct expressive facial play. Aside from this change, the dataset processing remained unchanged from the first prototype pipeline described above in section 7.5. Images by René Ebertowski.*

.

7.6.3. Mapping Networks

The idea of a mapping network is inspired by *StyleGAN* [KLA18], *Ganverse3D/ StyleGAN3D* [Zha+20] and *Pi-GAN* [Cha+21]. The basic idea is to transform input data (or noise vectors) into a more structured and interpretable latent space. This transformation allows for more nuanced control over the attributes and features of the generated images. The goal is to " disentangle" the latent code. Essentially, the mapping network acts as an intermediary, learning a complex function that maps random input vectors to a latent space where different dimensions correspond to meaningful variations in the output data.

While the mapping networks of *FilM* [Per+18], StyleGAN [KLA18], and their successors mainly interact with intermediate layers of the network, Ganverse3D/StyleGAN3D [Zha+20] used a mapping network before the generator network to gain more control over the output and improve visual quality. Our idea is to use a similar architecture to see if the visual quality improves. To verify our hypothesis, we test three different mapping network architectures while keeping the generator architecture unchanged. Our three architectures are:

1. *Multilayer Perceptron (MLP)*

   In this method, a multilayer perceptron (MLP) accompanied by convolutional layers is placed in front of the generator. It processes a list containing the positions of all landmarks. The MLP consists of two fully-connected layers using the ReLU activation function. The output of these layers is transformed into a 1024-dimensional vector. This vector is then converted into four 16x16 pixel feature maps. These feature maps are further processed through five convolutional layers with a stride

of 2, culminating in a resolution of 512x512 pixels and increasing the number of feature maps to 64. Finally, these feature maps are passed through an additional convolutional layer before being used as input to the generator.

2. *Residual*

   The Residual Mapping Network uses Residual Blocks [He+16] to handle the FLMs extracted from the dataset. First, each FLM is fed into a convolutional layer. Next, two Residual Blocks are applied to the resulting output, and then it passes through another convolutional layer to complete the processing. The 64 resulting feature maps are then used as inputs to the generator.

3. *Multilayer Perceptron (MLP) and Residual*

   This mapping network combines the two previous networks to generate 64 feature maps. It computes 32 feature maps from both the MLP and the Residual Mapping Network, and then concatenates them. The aggregated feature maps are then used as input to the generator.

7.6.4. Network Architecture

The network architecture remains the same as in the previous prototype shown, except that we add two additional outermost conv layers to the generator. This allows us to increase the resolution from 256 px ×256 px to 512 px ×512 px, which means that the $\mathbf{T}$ and $\mathbf{D}$ from the equation 7.2 and 7.3 change to $\mathbf{T} \in \mathbb{R}^{512x512}$ and $\mathbf{D} \in \mathbb{R}^{512\times512}$. We also add another skip connection between these two layers. The tensor size in the bottleneck remains the same. The new architecture of the generator has 67 million parameters with a file size of the generator of 260 MB.

One of our initial hypotheses was that the original generator probably has enough parameters, since it can generate relatively complex and versatile images, as can be seen from various projects and demonstrations of the Pix2Pix framework [Iso+17]. Many application examples are aimed at a high variation of certain objects, such as different house facades, clothes or handbags. It is often only a matter of capturing "high level" structures. When viewed from a distance, these images often look real to the viewer. However, a closer look reveals errors, especially in high frequency areas of the image. In our use case of snythesizing believable faces and conveying nonverbal information, fine structures such as facial hair, blood vessels, and dimples are important. In our experiments, we therefore focused on more sophisticated losses and a more complex architecture of the discriminator. Note that we did not change the generator beyond adding two outermost layers, although there are various recommendations in the literature. Wang et al. [Wan+18b] suggest using a multiscale generator similar to the approach of the progressively growing GAN proposed by Karras et al. [Kar+17] to improve quality, stability, and variation. However, we deliberately chose not to pursue this option in order to avoid developing an overly complex generator and to maintain real-time frame rates.

7.6.5. Evaluation Metrics

For evaluation, we use five quantitative metrics to assess the results of different experiments. Similar to the training dataset, the evaluation dataset consists of RGB-D images, feature maps, and an array of the locations of all landmarks. Each metric is applied to an evaluation dataset containing 250 elements. The data from this evaluation set was not used in the training process, so it can accurately demonstrate the performance of the network during inference.

The following metrics have been used:

1. *Time in milliseconds (ms)*

   This metric represents the time in milliseconds it takes for a network to perform a forward operation during inference. The data presented in this study was obtained using a system equipped with an NVIDIA RTX 3090 GPU, an Intel Core i9-9900K CPU, and 32 GB of RAM.

2. *Peak Signal to Noise Ratio (PSNR)*

   PSNR is widely used as a measure of the quality of reconstruction of lossy compression codecs (e.g., for images, video, and audio) by comparing the original signal with the compressed version. This metric quantifies the ratio of the maximum possible power of a signal to the maximum possible power of the distorting noise within that signal. It is calculated using the Mean Squared Error (MSE) as follows:

   $$PSNR = 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE) \tag{7.15}$$

   with

   $$MSE = \frac{1}{w \cdot h} \sum_{x=0}^{w-1} \sum_{y=0}^{h-1} [I_r(x,y) - I_s(x,y)]^2 \tag{7.16}$$

   Here, MAX_I is the maximum possible pixel value, w and h are the width and height of an image, respectively, I_r is the real image from the data set, and I_s is the synthesized image from the network. The higher this metric is, the more similar the two original images are. Due to the logarithmic nature of PSNR, this metric is not proportional and can be misleading when comparing metrics directly. While PSNR is often used for image compression, the human eye, for example, can clearly distinguish between a compressed image with a PSNR of 31dB and 37dB, while there is hardly any difference between 37dB and 45dB.

3. *Structural Similarity Index Measure (SSIM)*

   The Structural Similarity Index Measure (SSIM) [Zho+04] metric measures the perceived similarity between two images, taking into account that each pixel has a structural dependence on its spatial neighbors. The metric is computed as follows:

   $$SSIM = \frac{(2\mu_r\mu_s + c_1)(2\sigma_{rs} + c_2)}{(\mu_r^2 + \mu_s^2 + c_1)(\sigma_r^2 + \sigma_s^2 + c_2)} \tag{7.17}$$

   where μ_r and μ_s are the pixel sample means of the real and synthesized images, σ_r and σ_s are the variances of the real and synthesized images, σ_{rs} is the covariance

of the real and synthesized images, $c_1 = (k_1 \cdot L)^2$, $c_2 = (k_2 \cdot L)^2$ where we use the default values of $k_1 = 0.01$ and $k_2 = 0.03$, and L is the dynamic range of the pixel values. Typically, L is set to $(2^{\text{nbitpix}} - 1)$, where *nbitpix* is the number of bits per pixel value. The higher the value of the SSIM metric, the more similar the two images are. The maximum value of SSIM is 1, which means that both images are identical.

4. *Fréchet Inception Distance (FID)*

   The FID is a metric that can be used to evaluate the quality of images generated by a GAN. It measures the similarity between the distribution of synthesized images and the distribution of real images. Each set of images is modeled as a multivariate Gaussian distribution with mean and covariance. If we denote the mean and covariance of the real images as (μ_r, Σ_r) and those of the synthesized images as (μ_s, Σ_s), the FID score is given by the formula:

$$\text{FID} = \|\mu_r - \mu_s\|_2^2 + \text{Tr}(\Sigma_r + \Sigma_s - 2(\Sigma_r\Sigma_s)^{\frac{1}{2}}) \tag{7.18}$$

   where $\|\mu_r - \mu_s\|_2^2$ is the squared Euclidean distance between the means of the real and synthesized distributions, and Tr denotes the trace of a matrix, capturing the sum of its diagonal elements. The term $\text{Tr}(\Sigma_r + \Sigma_s - 2(\Sigma_r\Sigma_s)^{\frac{1}{2}})$ computes a measure of similarity between the covariances of the two distributions. A lower FID score indicates that the generated images are more similar to the real images, implying better quality of the generated images.

5. *Learned Perceptual Image Patch Similarity (LPIPS)*

   This metric has already been discussed in detail above under section 7.6.1 "Losses as Conditions of the Experiment". Please note that we did not consider PSNR, SSIM, or FID as loss functions due to their unsuitability for these applications, stemming from issues related to differentiability, perceptual quality, training stability and computational efficiency. LPIPS is unique in this sense because it can be used not only as a loss, but also as a quantitative and descriptive metric that can be read by humans. Alternative losses such as the VGG perceptual loss of Johnson et al. [JAF16a] or the feature matching loss described above work on neural layers and latent codes, which is difficult to quantify as a human readable metric.

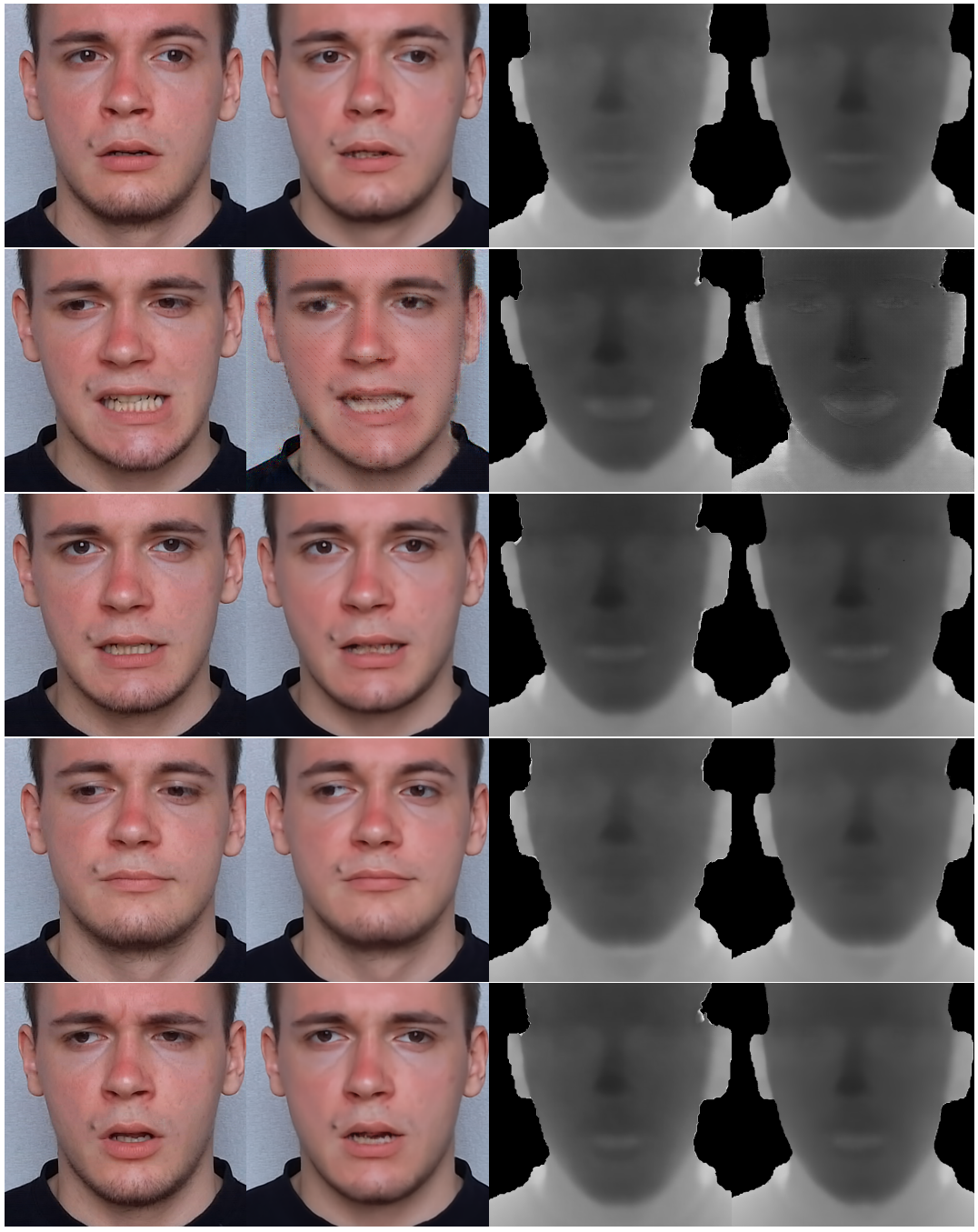
7.6.6. Results

Due to the large number of different test conditions, such as different losses or loss weights, we would reach several thousand permutations, which would require a training time of several months in total, with an average training time of about 10 hours per condition, which is not feasible in practice. Therefore, we conducted preliminary experiments and tested only the most promising combinations. For this purpose, we chose common weightings for the following losses, which have been frequently used in related work.

The results of the experiments are shown in the table below:

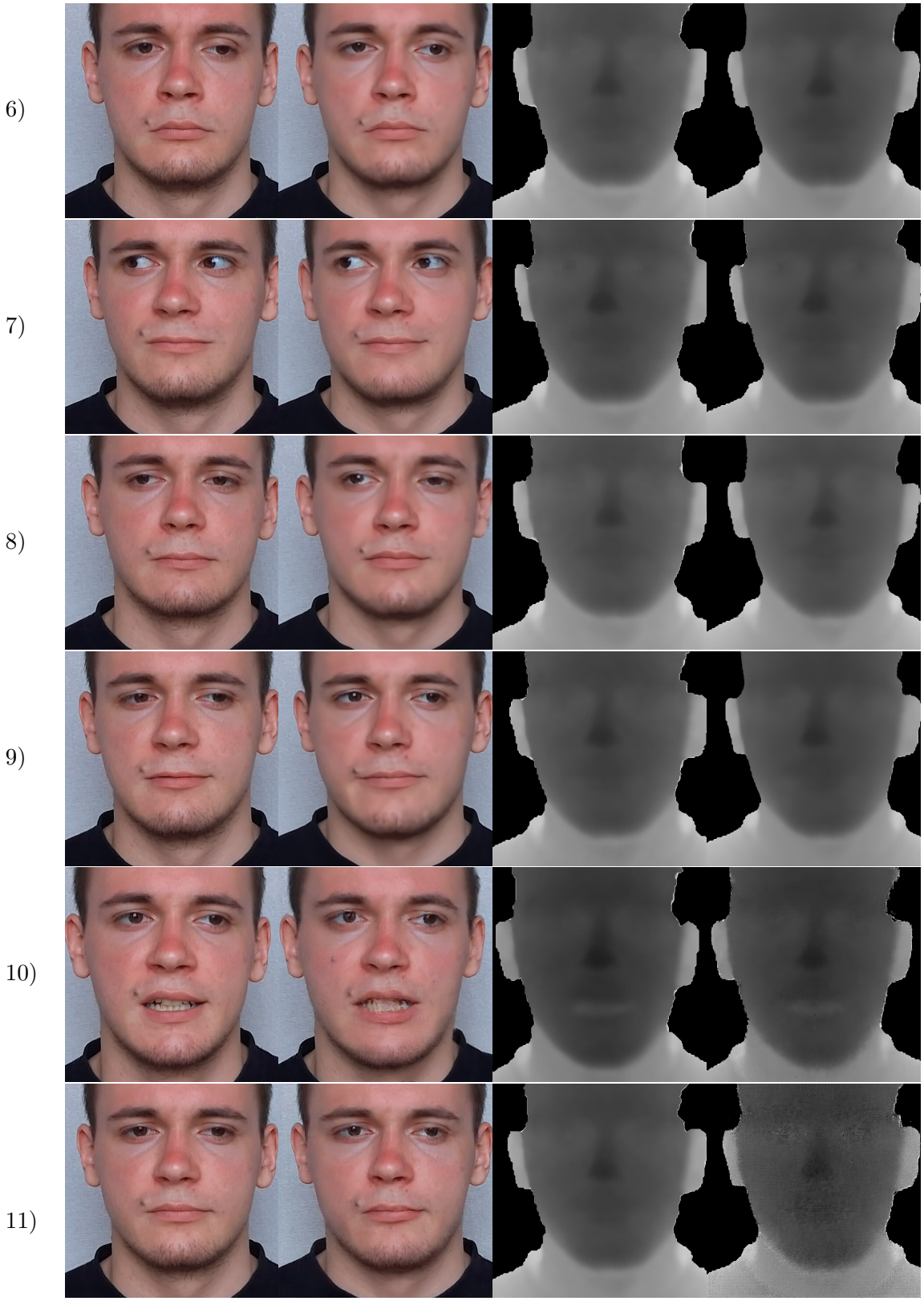
	Losses							Mapping Nets			Evaluation Metrics				
	LSGAN	L1	Advers.	Mul.D	FM	LPIPS	Cycle	MLP	Res.	Both	Time ↓	PSNR ↑	SSIM ↑	FID ↓	LPIPS ↓
1)	-	X	-	-	-	-	-	-	-	-	3.376	23.885	0.870	0.668	0.141
2)	-	X	X	-	-	-	-	-	-	-	3.417	21.102	0.670	0.740	0.421
3)	X	X	X	-	-	-	-	-	-	-	3.411	23.912	0.869	0.661	0.136
4)	X	X	X	-	-	-	-	X	-	-	4.611	23.690	0.865	0.658	0.134
5)	X	X	X	-	-	-	-	-	X	-	4.964	23.810	0.869	0.661	0.136
6)	X	X	X	-	-	-	-	-	-	X	5.997	23.658	0.867	0.657	0.136
7)	X	X	X	-	X	-	-	-	-	-	3.439	23.576	0.857	0.643	0.100
8)	X	X	X	-	-	X	-	-	-	-	3.426	23.828	0.869	0.664	0.136
9)	X	X	X	-	-	-	X	-	-	-	3.426	23.820	0.869	0.663	0.137
10)	X	X	X	X	-	-	-	-	-	-	3.473	23.631	0.843	0.652	0.118
11)	X	-	X	X	-	-	-	-	-	-	3.554	22.236	0.508	0.683	0.528
12)	X	-	X	X	X	X	X	-	-	-	3.661	23.358	0.815	0.668	0.226
13)	X	X	X	X	X	-	-	-	-	-	3.394	23.547	0.842	0.648	0.123
14)	X	-	X	X	X	-	-	-	-	-	3.394	23.332	0.821	0.663	0.207
15)	X	X	X	X	X	X	-	-	-	-	3.410	23.541	0.844	0.653	0.115
16)	X	X	X	X	X	X	X	-	-	-	3.401	23.629	0.853	0.655	0.111

Table 7.1.: *All experiments with their evaluation metrics. The color descriptions are: worst, second worst, third worst, third best, second best, best. The abbreviation "Mul.D" stands for Multiscale Discriminator, "FM" for Feature Matching and "Both" for the combination of an MLP and a Residual network. Stable epochs denote the number of epochs with a constant learning rate. Decaying epochs denote the number of epochs with a linearly decreasing learning rate. An X indicates that the loss function or mapping network was used in the experiment. The three horizontal lines, separating the results in four groups, are inserted only for better readability. The average of a 250-pair evaluation data set is shown, except for the FID metric, where the distribution of all evaluation data is compared to the distribution of all synthesized data.*



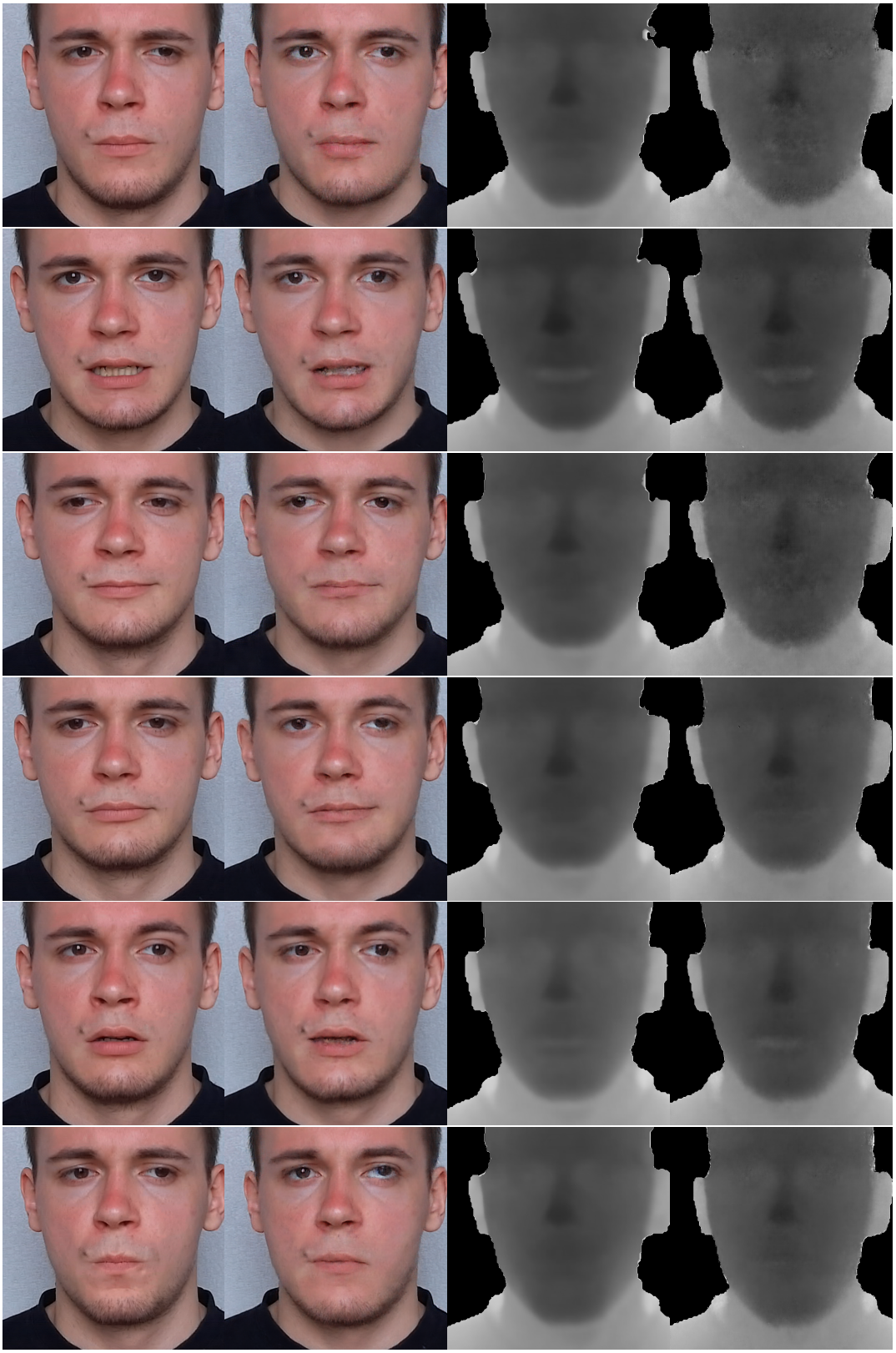
(a) *Ground Truth Color* **(b)** *Synthesized Color* **(c)** *Ground Truth Depth* **(d)** *Synthesized Depth*

Figure 7.14.: *Images from experiments 1-5. The RGB and depth ground truth images are shown next to the synthetic images for comparison with the images from the evaluation dataset. The images of 2) show the results of adding two outermost layers to the vanilla Pix2Pix network for upscaling. Images by René Ebertowski.*



(a) *Ground Truth Color* **(b)** *Synthesized Color* **(c)** *Ground Truth Depth* **(d)** *Synthesized Depth*

Figure 7.15.: *Images from experiments 6-11. For comparison with the images from the evaluation dataset, the RGB and depth ground truth images are shown next to the synthetic images. Images by René Ebertowski.*



(a) *Ground Truth Color* **(b)** *Synthesized Color* **(c)** *Ground Truth Depth* **(d)** *Synthesized Depth*

Figure 7.16.: *Images from experiments 12-17. The RGB and depth ground truth images are shown next to the synthetic images for comparison with the images from the evaluation dataset. Images by René Ebertowski.*

18)

(a) *Ground Truth Color* **(b)** *Synthesized Color* **(c)** *Ground Truth Depth* **(d)** *Synthesized Depth*

Figure 7.17.: *Images from Experiment 18: The RGB and depth ground truth images are shown next to the synthetic images for comparison with the images from the evaluation dataset. Images by René Ebertowski.*

7.6.7. Observations and Discussion

In Tab. 7.1 the different losses, architectures and results of the evaluation metrics are compared for each experiment. It's evident that the four metrics (PSNR, SSIM, FID, and LPIPS) differ in their evaluations. The PSNR and SSIM results are often similar, as are the values for FID and LPIPS. When results are significantly poor, all metrics agree, but this consensus does not always extend to the best results. This is evident in experiment 1, for example, which performs well on PSNR and SSIM, but has a significantly more blurred image than the other results. Notably, the introduction of LSGAN leads to sharper images, as shown in table 7.1.

Experiment 2, which used only an L1 and an adversarial loss, produced the worst results. The adversarial loss means that a binary cross entropy loss discriminator was used in this experiment. Thus, this training structure is similar to the vanilla structure of the Pix2Pix framework and shows the results of adding two outermost layers to the vanilla Pix2Pix network for upscaling from 256 pixels to 512 pixels edge length.

All further experiments after experiment 2 use an LSGAN loss with a discriminator. An early observation in our experiments was that the LSGAN loss is superior to the cross-entropy loss. Experiments 1 and 2 using cross-entropy loss result in much more blurred images, which cannot be observed with LSGAN loss. However, LSGAN loss alone is not sufficient to reconstruct high-frequency image information. Although experiment 7 achieves the best results according to FID and LPIPS, its visual results are still too blurred. There's a slight shift in skin color towards a pinker tone, and the skin appears more reflective. In general, we see that the use of a Multiscale Discriminator (MultiD) seems to improve realism by better preserving high-frequency structures (compare Fig. 7 and 15). Note that PSNR and SSIM tend to rate blurred results higher than those that preserve high-frequency image details, as mentioned above in experiment 1. In particular, the activation of MultiD leads to a deterioration of the PSNR and SSIM results.

The proposed mapping networks lead to longer inference and training times, and the visual results are satisfactory but not exceptional compared to the absence of a mapping network. This does not justify the additional inference time of about 2 ms. The size of the networks in all other experiments remains the same, indicating that the variations are within the measurement error of about 0.2ms and can be ignored. The different losses do not affect the inference time of the generator network.

Only a few experiments were able to accurately reconstruct facial hair. Experiment 7, despite a good metric score, still produces surfaces that are too soft. Removing the L1 loss preserves details better, but introduces other artifacts. L1 loss appears to be essential.

Teeth remain a consistent problem across experiments. Related work has produced better teeth reconstruction with a teeth proxy added to a 3DMM [Gar+15; Thi+15]. The problem with a good oral cavity reconstruction is that this area of the face is very dynamic, and facial landmarks that only track the lips but not the teeth or tongue can lead to ambiguous results.

Although we did not record the training time, and therefore it is not shown in the table, it was noticeable that the cycle loss doubled it from about 10 to 20 hours. Although the results of experiment 16 are very good, they do not surpass those of experiment 15, which was evaluated by an unstructured interview with 6 people from the local computer science department. Therefore, we rejected experiment 16 (with cycle loss) as the experiment with the best visual quality, because the long training time limits further hyperparameter tuning and also the general applicability of the system.

We see experiment 15 as the experiment with the most potential for further hyperparameter tuning with even better image quality, laying the groundwork for further insights into our final network architecture and training procedure in the next chapter. Although the PSNR and SSIM results for this experiment are not outstanding, experiment 15 is one of the experiments that achieved the best FID and LPIPS values and was also rated by humans as the most authentic result. Therefore, we decided to gain more insight into experiment 15 because we saw in the training plots that a more targeted loss leads to better results much earlier in the training process.

7.6.8. Improved Network Model Architecture

Based on our study, we choose experiment 15 as the experiment with the best visual quality, acceptable training time, and potential for further improvement. Based on our findings, we propose the following major changes to the Pix2Pix architecture of Isola et al. [Iso+17] framework for our application domain, which was the basis for the first prototype:

1. Introduction of multi-scale discriminators that process three different resolutions of the input image, coupled with an additional *Feature Matching Loss* as described in Pix2PixHD [Wan+18b].
2. Replace the sigmoid cross entropy loss of the Pix2Pix discriminator (in Pytorch, the *BCEWithLogitsLoss* loss) with the *Least-Squares Loss* from LSGAN [Mao+17] as the L2 loss, following the recommendations of Wang et al. [Wan+18b].
3. Replacement of the *Perceptual-VGG Loss* [JAF16b] originally proposed by Wang et al. [Wan+18b] with the more effective *Learned Perceptual Image Patch Similarity* (LPIPS) developed by Zhang et al. [Zha+18].

We extended the discriminator side with three multiscale networks and obtained the following objective:

$$\min_{D_1,D_2,D_3} V_{GAN}(D) = \sum_{k=1,2,3} \mathcal{L}_{cLSGAN\_D}(D_k, G) \tag{7.19}$$

where D_1, D_2, and D_3 represent the three resolutions of the input image. The objective

function for the generator is defined as:

$$\begin{aligned}&\min_G V_{GAN}(G) = \\ &\sum_{k=1,2,3} \Big[\mathcal{L}_{cLSGAN\_G}(D_k, G) + \lambda_{FM}\mathcal{L}_{FM}(D_k, G)\Big] \\ &+ \lambda_{L1}\mathcal{L}_{L1}(G) + \lambda_{LPIPS}\mathcal{L}_{LPIPS}(y, G(x))\end{aligned} \tag{7.20}$$

with the hyperparameters set as follows: $\lambda FM = 10$, $\lambda_{L1} = 100$, $\lambda_{LPIPS} = 10$. To ensure faster inference times than Pix2PixHD, we chose not to apply the coarse-to-fine strategy suggested for the generator by Wang et al. [Wan+18b]. This decision trades the high-resolution image generation of Wang et al. for improved computational efficiency.

These improvements preserve high-frequency details such as facial hair, as shown in 7.21, thereby substantially improving quality and resolution. Our modifications focused primarily on the loss function and discriminator, eliminating the need to modify the generator. This allows maintaining high frame rates during test/inference time, which is critical for telepresence or also live broadcast scenarios where only the generator is active. A drawback is the increased memory requirement for training, although the total training time is significantly reduced by over 50% compared to the Pix2Pix-only method (from approximately 19 hours down to 8 hours). This efficiency gain is due to the newly introduced loss term, which is more effective for our purposes and allows us to achieve better results in less time. Due to the increased memory requirements for training, training can no longer be performed on a GPU with only 11GB of VRAM (such as an Nvidia RTX 2080 Ti), but must be performed on a GPU with more memory, such as an Nvidia RTX 3090 with 24 GB of VRAM.

7.6.9. Further Hyperparameter Tuning and Training

In the first experiments we tried to change the dropout rate. Since the variance of the generated images is very high in many application scenarios of Pix2Pix and also Pix2PixHD, e.g. different house facades or images of streets from a moving car, our hypothesis was that we are actually overfitting the network with a single face and its expressions. Conditional GANs typically add a noise vector z to the network. In the Pix2Pix architecture, the noise is substituted by drop-outs and is not given as input, but is generated at the level between layers. In our tests, we tried to reduce the dropout rate to see if the results were better. We found that inputting data that is very similar to the training data (in our case, the FLMs) tends to produce better results, but the network's ability to interpolate and extrapolate facial landmarks decreases significantly. However, this is counterproductive for our specific use case with a face-tracking HMD, since the FT-HMD is not able to provide reliable enough face landmark data to match the training data almost one-to-one. Therefore, the neural network requires a certain error tolerance and thus a corresponding dropout rate to compensate for tracking errors. In the end, we keep the initial dropout rate of 50% as originally recommended by Isola et al. [Iso+17].

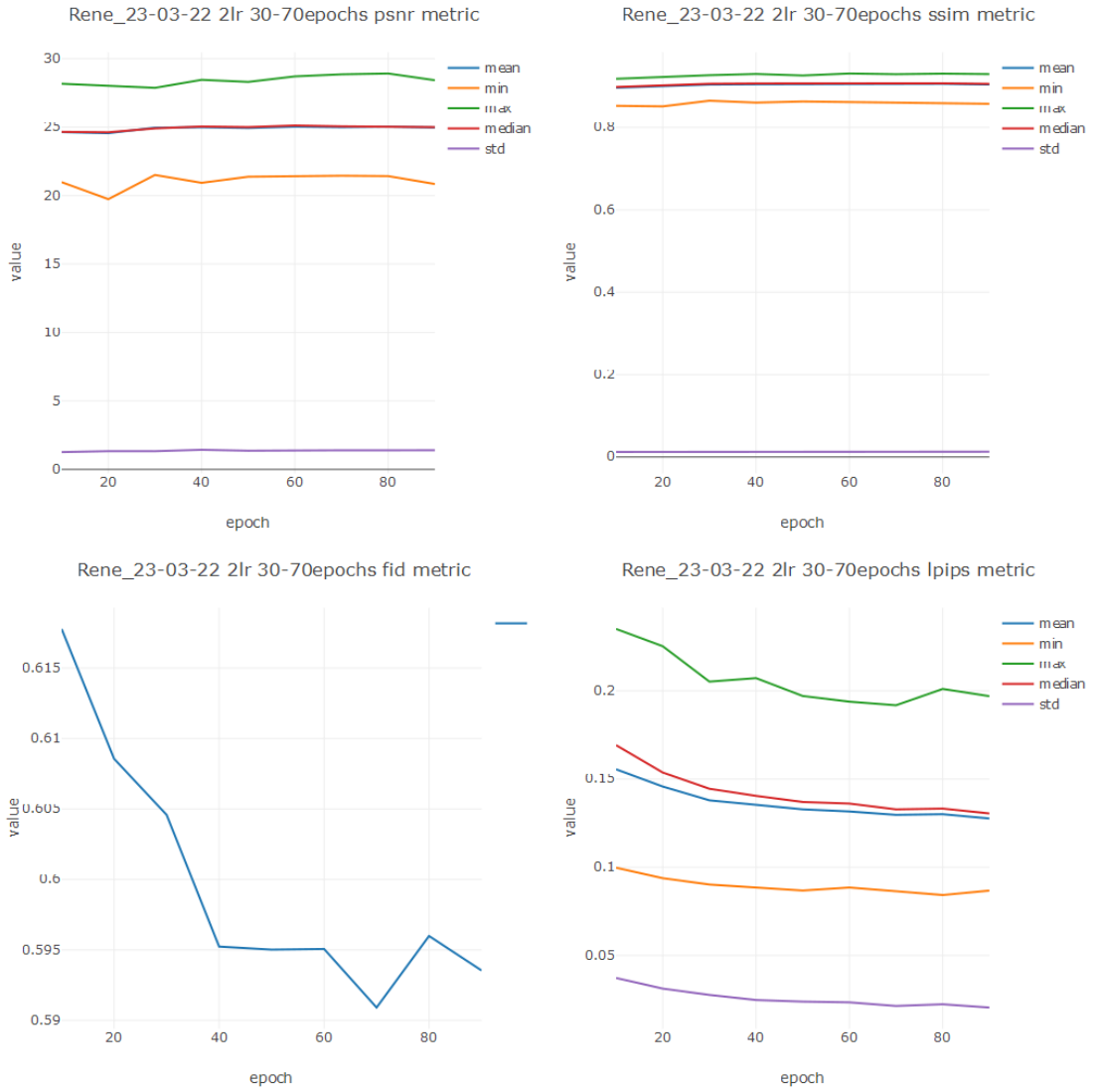


Figure 7.18.: *A training run over 100 epochs with the more goal-oriented losses from equation 7.19 and 7.20. Number of the epochs is displayed on the x axis. It can be seen that PSNR and SSIM reports good results already after the first epoch. While FID reaches a plateau after 40 epochs, LPIPS seems to reach a plateau only after 60-80 epochs.*

In the training protocols of the 18 experiments above, we found that good visual results were achieved much faster with the more goal-oriented loss functions from equation 7.19 and 7.20. The original Pix2Pix, Pix2PixHD, and CycleGAN pipelines recommend 200 epochs, but this is highly dependent on the dataset. In our tests, we found that good results can be obtained with less than 100 epochs, as shown in Fig. 7.18. With this in mind, we investigated a further hyperparameter tuning process: First, we adjust the learning rate, which could lead to better results even faster, and second, we test different combinations of different numbers of stable and decaying epochs, which could be useful to further reduce the training time without affecting the visual results. The results of the hyperparameter training are shown in the Tab. 7.2.

Epoch Count			Metrics				Time
Stable	Decaying	Learning Rate	PSNR ↑	SSIM ↑	FID ↓	LPIPS ↓	Hours
100	100	0.0002	24.8418	0.9013	0.5932	0.1273	10.22
30	70		**25.0188**	**0.9049**	0.5933	0.1281	5.08
20	65		25.0152	0.9046	0.5938	0.1265	4.75
15	60		25.0094	0.9045	0.5939	0.1284	3.61
100	100	0.0003	24.8895	0.9021	0.5973	0.1251	9.83
30	70		24.8993	0.9041	0.5986	0.1257	4.75
20	65		24.9664	0.9039	0.5926	0.1270	4.06
15	60		24.9742	0.9040	0.5953	0.1281	3.63
100	100	0.0004	24.8233	0.9015	**0.5908**	**0.1217**	10.10
30	70		24.9859	0.9043	0.5961	0.1262	5.15
20	65		24.9932	0.9041	0.5935	0.1261	4.30
15	60		24.9819	0.9043	0.5955	0.1286	3.70

Table 7.2.: *Results of experiments aimed at reducing the training time required. The best values are in bold. The table shows different learning rates combined with different numbers of stable and decaying epochs. Stable epochs are epochs during which the learning rate remains constant at its initial value, while decaying epochs are epochs during which the learning rate decreases linearly to zero. The quality of the images is represented by four metrics. These metrics are averaged over an evaluation dataset consisting of 250 data pairs. The exception is the FID metric, where the distribution of all evaluation data is compared to the distribution of all synthesized data. The time required for the training process, measured in hours, is given in the last column. Measured on a PC with an Intel i9 9900K (4.68GHz) CPU, NVIDIA GeForce RTX 3090 GPU and 32GB DDR4 RAM at 4266 MHz.*

Note that the preprocessing pipeline and the network architecture are the same as in Tab. 7.1, but we choose a different dataset. The main difference is that the images have a background behind the person's head with less structure in the new dataset, and therefore the metrics are slightly better because there is less noise in the background of the new dataset.

As Tab. 7.2 shows, it is possible to reduce the training time from around 10 hours initially to less than 4 hours without any significant difference in the metrics. The variations in the metrics are so small that they are not visible to the naked eye. Furthermore, changing the learning rate did not significantly change the final visual quality metrics. The learning rate recommended by Isola et al. [Iso+17] is 0.0002. In summary, the generator was able to maintain image quality despite a lower number of epochs.

7.6.10. Results

This iteration of the prototype has enhanced image quality and further reduced the amount of the uncanny valley effect. All results and metrics shown in Fig. 7.19 and 7.20 were generated using FLMs from the test set without any backpropagation performed on these datasets. Please note that these results are not directly comparable to the results in Fig. 7.11 because we have removed the face-tracking HMD and we take FLMs directly from the test dataset that were generated on the ground truth images. These images are more stable than the outputs of the face-tracking HMD.

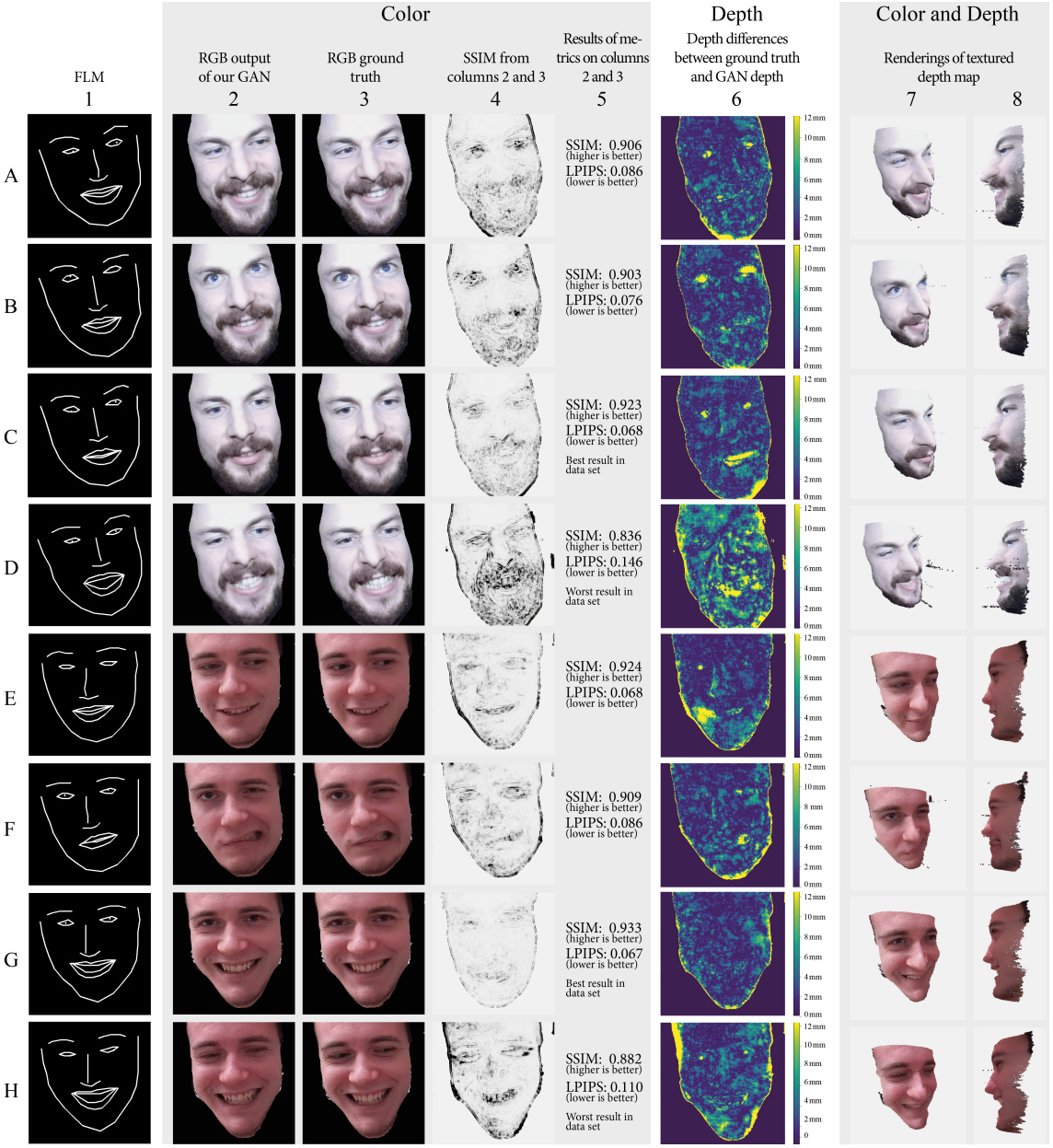


Figure 7.19.: *Results 1/2 of the second prototype: Please zoom in for details. This summary shows the FLMs in the first column from our test datasets and their results compared to the ground truth. The FLMs in the third column are from images that the neural network has never "seen" before. The second column shows the results of our trained generator, which produced images in the second column after receiving the FLMs from the first column. The fourth column illustrates the SSIM differences, with darker shades representing greater discrepancies between the images in the second and third columns. Column six shows the discrepancies between the GAN generated depth and the ground truth depth. Columns seven and eight show the integration of the generated depth and color data viewed at 30 and 90 degrees. Image from [Lad+25]*

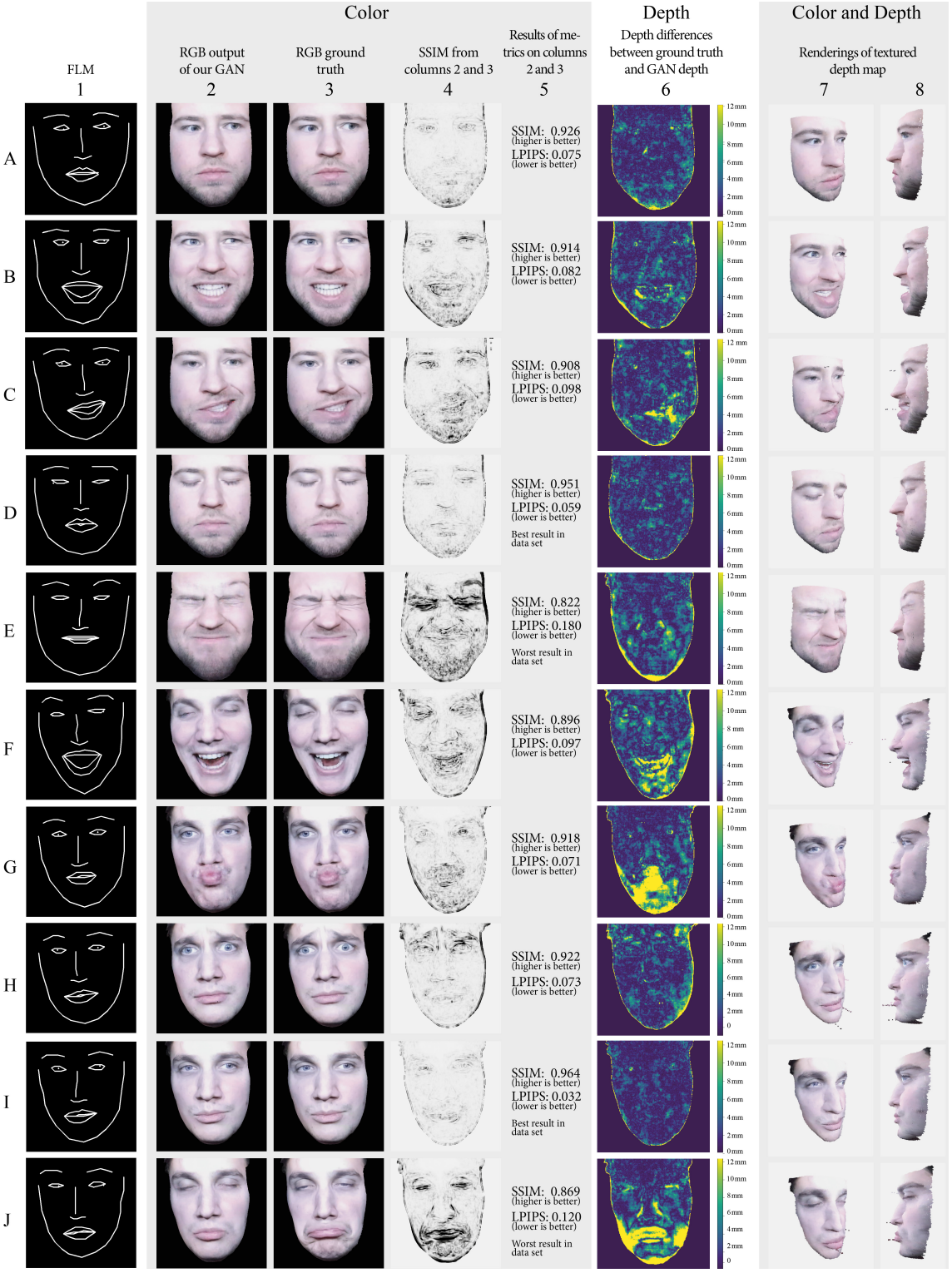


Figure 7.20.: *Results 2/2 of the second prototype: Please zoom in for details. Two further subjects are shown. The last two samples of each person combine the best and the worst images (measured by SSIM and LPIPS) and reflect the range of reconstruction quality. Image from [Lad+25]*

The data splits for the participants were as follows: the first was split into 1238/207, the second into 1500/250, the third into 1620/271, and the last into 2413/403. Notably, the dataset for the last participant is about 1000 items larger than the others, yielding slightly better quantitative results in terms of SSIM and LPIPS. This implies that larger datasets improve image quality in our tests.

The primary concern with our earlier prototype was image sharpness, as shown in Fig. 7.21a. The updated architecture and loss functions improve the level of detail in the images, even successfully reconstructing features such as skin pores, particularly noticeable on the user's forehead in Fig. 7.19, rows E through H. In addition, there is improved reconstruction of high-frequency areas such as facial hair, as shown in Fig. 7.21b. The system also shows enhanced temporal consistency. For more details, please refer to the supplementary video: `https://youtu.be/fBofqRfvoiM`

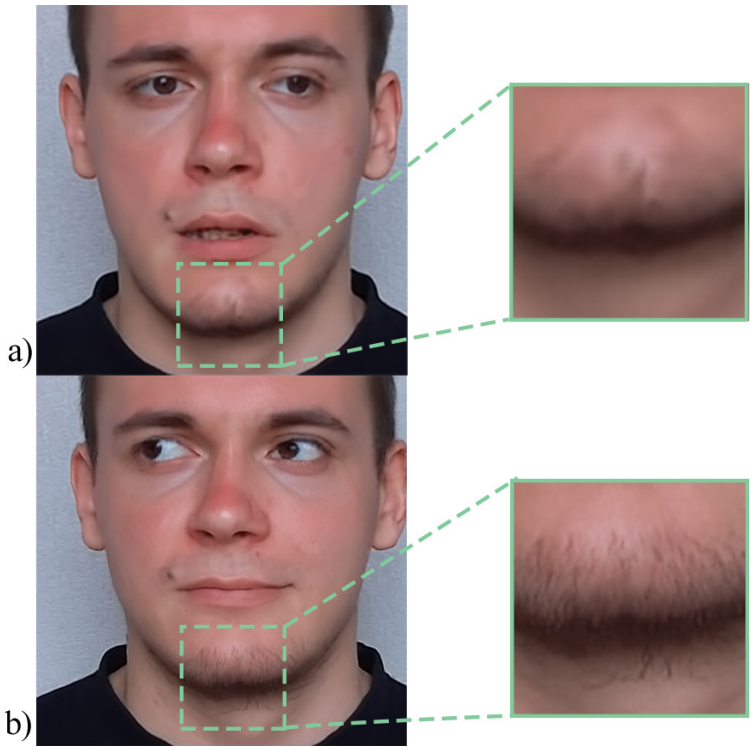


Figure 7.21.: *Our updated pipeline of our second prototype, including network design and loss functions, has significantly enhanced the quality. Image a) shows an example from the first prototype system, while image b) demonstrates the improved resolution (increased from 256x256 to 512x512 pixels) and better preservation of fine details. In particular, high-frequency local structures such as skin pores, scars, freckles, blood vessels, or facial hair can be synthesized in greater detail. Image from [Lad+25].*

The discrepancy between the actual and estimated depth values is typically less than 4 mm, as shown in column 6. Notable deviations occur in the reconstruction results that exhibit the worst SSIM and LPIPS values for each dataset of an individual, as shown in Fig. 7.19, rows D and H, and in Fig. 7.20, rows E and J. In addition, anomalies are observed in column 8. Please note that we are using the unprocessed depth image from the Kinect and the direct output from our GAN, without applying any filtering or smoothing to the images. Therefore, we hypothesize that the network has learned to replicate the depth noise from the sensor, thus contributing to further depth inaccuracies.

In order to analyze the face regions in the images from columns 2 and 3 without including background changes, we segmented the face region using the depth data and rejected all other pixels. There are significant discrepancies in the SSIM and depth variation visualization at the intersection between the face and background regions. These discrepancies result from slight misalignments between the cropped sections of the actual and synthesized images due to small variations in the generated images. In addition, we implement erosion on the face region to eliminate the background and convert it to black pixels, which can cause these slight discrepancies.

Despite the improved results in our quantitative evaluations, our system still faces challenges in accurately reconstructing features such as the eyes, lips, and oral cavity. In particular, teeth and tongue often exhibit noisy artifacts, as shown in Fig. 7.19, row D, column 2. These artifacts are even more pronounced when viewing the final video stream. The reconstruction error tends to increase when the facial expression deviates significantly from a neutral expression. The eyes show fewer artifacts than the mouth, but even minimal image perturbations can trigger the uncanny valley effect, as shown in Fig. 7.19, row B, column 2. Please enlarge the image to see more details.

In our experimental setup, we used PyTorch 1.8 and Python 3.7 on a Windows 10 system. All recorded training and processing times were without any speed-enhancing techniques such as half/mixed precision (Nvidia APEX or Pytorch's AMP/autocast) or TorchScript's "traced models" for just-in-time compilation. The forward pass time of the generator module for an image resolution of 512×512 pixels ranges from 3 to 4 ms (333-250 fps) on an Nvidia RTX3090 and from 6 to 7 ms (167-143 fps) on an Nvidia RTX 2080. While our earlier prototype was faster (between 1 and 3 ms) and produced smaller images of 256×256,pixels, the current timings are still sufficient for VR applications, where typical frame rates range from 75 to 120 fps. It's worth noting, however, that many face tracking systems are limited to 60 or even 30 fps, which can limit the pipeline's frame rate.

7.7. Third Prototype: Data Set Capture Without Helmet Mount

The third iteration removes the cumbersome helmet mount from Fig. 7.3 introduced in Sec. 7.5 from the capture pipeline. Instead, several algorithms check the orientation and distance of the face to the RGB-D sensor in real time. This makes the system more user friendly, more hardware agnostic, our results are easier to reproduce, and the system is potentially more usable in everyday contexts using off-the-shelf hardware, thus eliminating the need for intermediate laboratory hardware. We use the Azure Kinect because the programming interface is easily accessible and well documented, but the majority of high-end smartphones today have a built-in RGB-D sensor that can be used instead of the Azure Kinect to capture the data set.

7.7.1. Capture Pipeline

In this setup, the RGB-D sensor is stationary on a table in front of the user. The pipeline is divided into 1.) a capture process with real-time user feedback using a graphical user interface to ensure specific image parameters as well as quality, and 2.) an offline processing step similar to Subsec. 7.5.1, where landmark detection is performed, but with additional steps to improve image quality. The graphical user interface is built in the Motion Hub

of Chap. 4. A third-party library (pybind11) connects the Python side of neural network training and processing to the C++-based MotionHub.

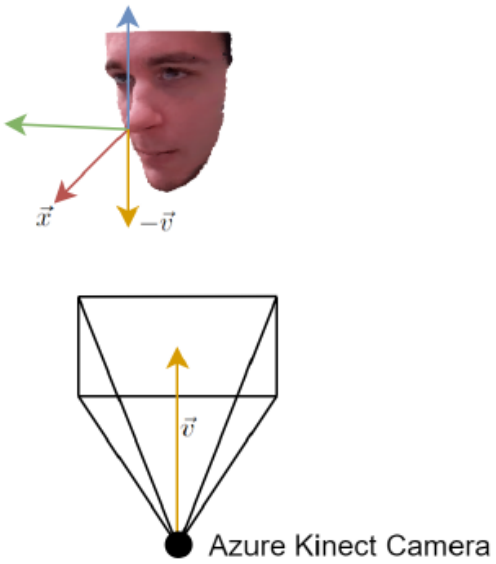


Figure 7.22.: *We verify the position and rotation of the captured face with the SynergyNet of Wu et al. [WXN21] in real time. If the head is rotated too much in one direction, these images are not used for training. The same applies to the distance of the face to the sensor. If the face is too close or too far away, the user will be informed and no further images will be taken. Image by René Ebertowski.*

The first part, the real-time user-assisted acquisition process, requires several image quality and parameter checks, since the degrees of freedom and the possibilities for the user to make mistakes during the acquisition process are much greater than in the helmet-mounted approach. It is structured as follows: 1.) The SynergyNet by Wu et al. [WXN21] is used to detect a face in the RGB image. If more than one face is detected, the user is warned and the images are not used for training. 2.) The Azure Kinect has non-overlapping regions in the border areas between the color and depth maps. We check whether the face is inside or outside the area where RGB and D data overlap. If the face is outside, the frame is rejected. 3.) A Laplace operator, applied to the face region, verifies that the image is sharp and does not contain significant motion blur, as this would reduce the quality of the reconstruction later. 4.) SynergyNet also provides a transformation and rotation matrix for the roll, pitch, and yaw angles of the face. We apply a dot product between the front-facing vector of the face and the inverted sensor's Z axis, as shown in Fig. 7.22. If the face is rotated further than a certain predefined threshold, we discard the image, warn the user, and start processing the next image. This process is supported by a user interface as shown in Fig. 7.23 and 7.24. In our experiments, we limited the face rotation to 10 degrees as a threshold, but we believe that a smaller value will further improve the final image quality, as our study and final results show slightly worse visual reconstruction quality compared to the wooden helmet mount of prototypes one and two. 5.) In the last step, we check if the face is within a certain distance from the sensor. If the face is too far away from the sensor, the resolution is not sufficient, the user is informed and the image is also discarded.

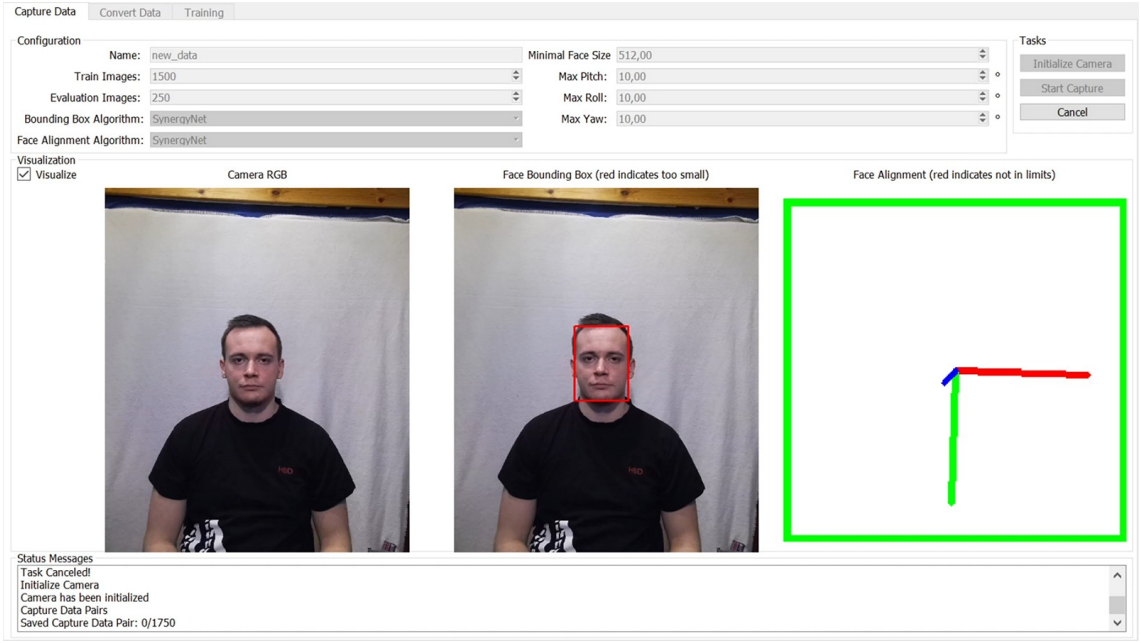


Figure 7.23.: *If the user moves too far away from the sensor, the recording stops, the user is notified, and the bounding box around the face turns red in the center image of the graphical interface, indicating that the face is too small and falls below the specified minimum size. Images by René Ebertowski.*

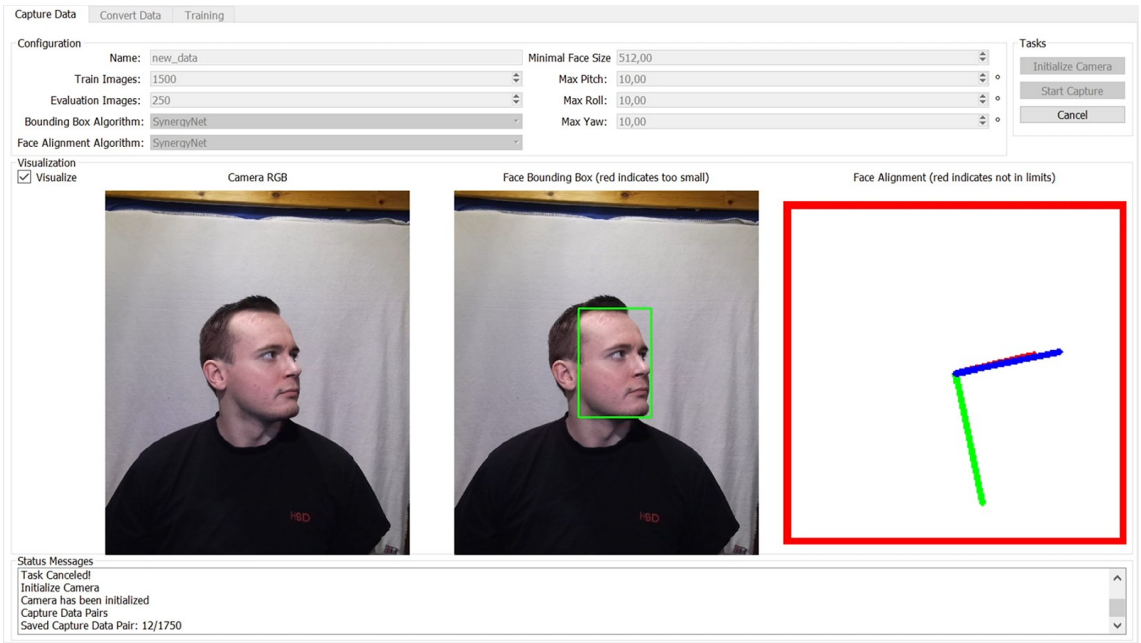


Figure 7.24.: *If the user rotates the head too much, he/she will be warned. In this case, the frame on the right side of the graphic interface, which shows the head rotation with an RGB cross, turns from green to red. Images by René Ebertowski.*

When the acquisition process reaches a certain number of images (1750 in our setup), the user is notified, the acquisition process is stopped, and offline processing can begin. The following section describes the same steps as in Subsec. 7.5.1, but uses improved

algorithms, adds more procedures, and provides a graphical user interface. The user interface for setting up the processing is shown in Fig. 7.25 below:

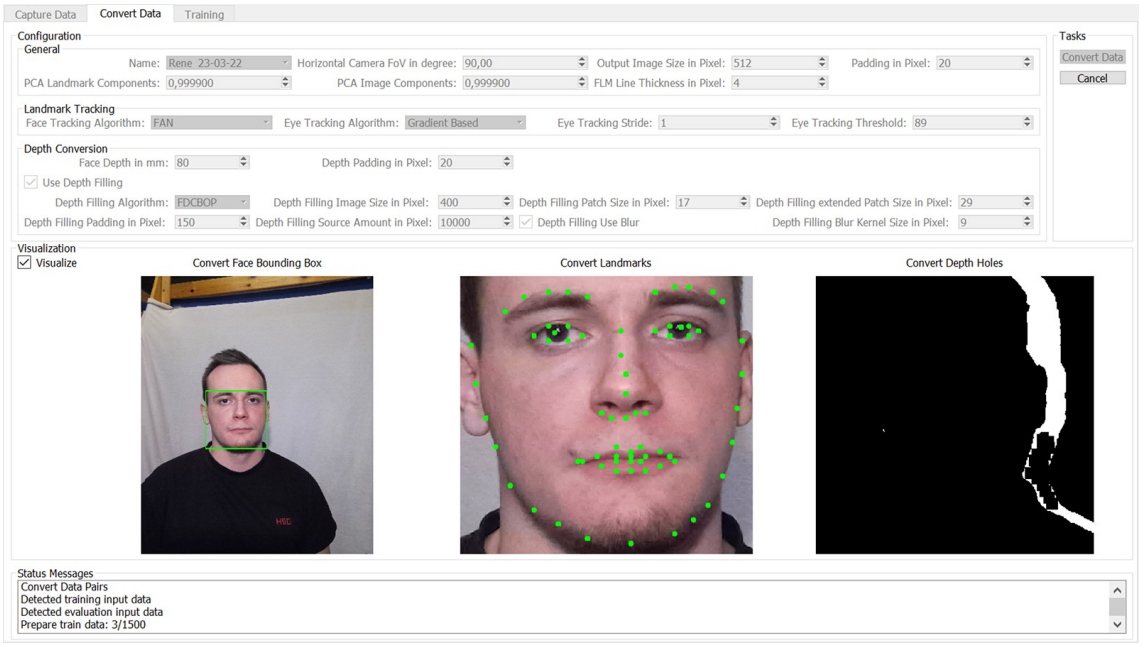


Figure 7.25.: *The graphical user interface for the data preparation phase. The visualization shows the axis-aligned bounding box (left), the 70 facial landmarks (center), and the hole region of the depth filling algorithm that still needs to be filled (right). Images by René Ebertowski.*

1.) Facial Landmark Detection

The first and second prototypes used the Facial Alignment Network (FAN) by Bulat and Tzimiropoulos [BT17]. The described pipeline was created back in 2020 and 2021 as the FAN was the SOTA solution back then, but Google's free MediaPipe landmark detection pipeline [Gri+20] became the preferred solution today in 2024. It outperforms other solutions, such as FAN, in speed and accuracy, and is used in many other SOTA projects [DBB22; ZBT22a; Gra+22].

2.) Gaze Tracking

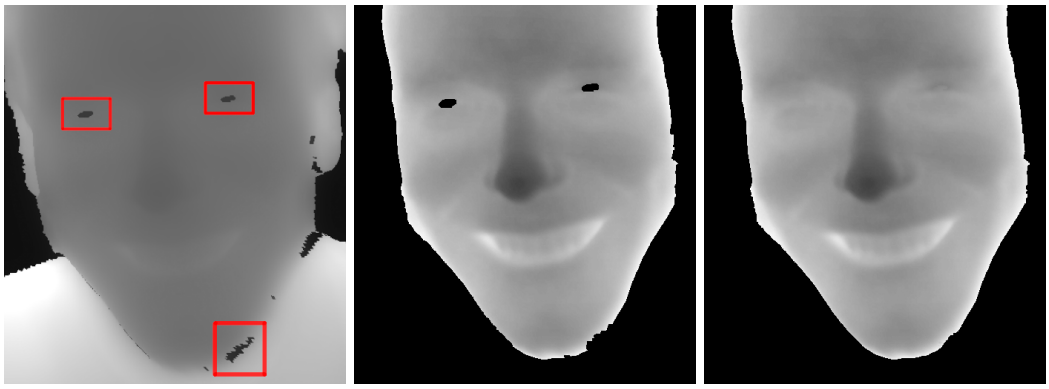
For our third prototype, it was no longer possible to work with an external eye tracker because the sensor and head rotation and position were independent. Synchronizing and spatially calibrating the Kinect and Tobii eye trackers proved to be difficult. Our solution was to implement a different eye tracking method based on a work by Timm and Barth [TB11; Jon18]. It uses a grayscale-converted RGB image as input to detect the position of the pupil based on image gradients. This method has been shown to be more robust under natural lighting conditions than techniques that use infrared images to locate the center of the eye. To use this tracking algorithm, we must crop the eyes from the original full-face RGB image using the bounding boxes of the facial landmarks for the eyes. Using this crop and converting to a grayscale image, the iris center could then be determined in 2D pixel coordinates using the iris tracking method of Timm and Barth [TB11]. These determined pixel coordinates were transferred back to the original image using the original section of the eye. This gave us two more landmarks to encode the gaze direction for the GAN in the FLM. In practice, however, this approach had drawbacks and needed further adaptation. When the subject blinked, there was a lot of tracking noise as the

algorithm searched for an iris and temporarily switched between different local minima on the closed eyelid. This problem could be minimized by freezing the last iris position when the eye landmarks of the upper and lower eyelid fell below a certain distance threshold indicating that the person was blinking. This resulted in better visual results for the GAN reconstruction.

As mentioned above, we used Google's Mediapipe for facial landmark tracking. Mediapipe also added iris detection and gaze tracking during the development of our system, which was not only faster than Timm and Barth's approach [TB11; Jon18], but also gave better results for blinking.

3.) Crop
The crop remains unchanged from the second prototype with an additional 20% of the bounding box edge length around the landmarks.

4.) Depth filling
It has been shown that there are a few depth pixels in every image that can lead to errors in depth measurement. These errors are often caused by reflections, e.g. from glasses or parts of the eyeball, at edges where occlusion occurs due to the technical design of RGB and D sensors, or by dark surfaces that are not sufficiently reflective. This is called depth invalidation [Tes24] and will set the depth pixel to the distance of 0, resulting in holes in the final depth maps, as shown in Fig. 7.26a and b. In our experiments, we found that our trained neural network also reconstructs these errors in the final images. In a three-dimensional immersive MR environment, this can be disruptive and trigger the uncanny valley effect. Therefore, we use a depth filling algorithm that is able to reduce these errors.



(a) *Image crop of a 16-bit depth image after data acquisition.*

(b) *8-bit depth image without depth filling.*

(c) *Final 8-bit depth image with depth filling.*

Figure 7.26.: *The depth images captured by the Azure Kinect may contain holes due to errors during acquisition. a) A depth image of a face contains holes in both eyes and on the chin after data acquisition. b) When the image is processed without depth filling and used for training, the holes are transferred to the final depth image generated by the GAN. c) With our depth filling algorithm, the holes are filled. Images by René Ebertowski.*

For this purpose, we used the algorithm of Nam et al. [Nam+16] for our system and adapted it to our specific use case. The main difference from Nam et al.'s implementation to ours is that we reduced the number of source pixels to reduce the runtime of the algorithm by 80%. Reducing the number of source pixels is comparable to reducing the image resolution

and comes at the cost of a lower final visual quality, which is acceptable for our application. However, since the depth fill correction process was originally implemented on the CPU and not on the GPU, it would have taken several days, or even weeks for large datasets what implies, we needed to find a faster solution. With the source pixel reduction, the depth fill steps take about 48 hours to compute the entire dataset of about 1750 images. The result of the depth filling algorithm is shown in Fig. 7.26c. Since the algorithm is highly parallelizable, we expect a significant speedup using a CUDA implementation. An additional depth filling time of 48 hr is a significant increase compared to the time for dataset acquisition (15 min) and network training (4 hours), but may be a possible option for datasets that require high quality reconstruction.

5.) Depth and Resolution Normalization

The next step is to determine the scaling factors necessary to normalize the depth and final resolution of the face in the image. Since we allow the user to move freely in front of the sensor, RGB-D images will be captured with the face at different distances from the sensor and therefore at different RGB and D image resolutions. The depth scaling factor, denoted as s_{depth}, is used to adjust for variations in the apparent size of faces due to their different distances from the camera. This adjustment ensures that the face sizes in each data pair are scaled appropriately. To achieve this, a target depth value, $d_{desired}$, is set as the reference point for scaling all images. The goal is to find a scaling factor that can geometrically change the position p of a pixel k in response to changes in depth. This calculation is directly derived from the first intersection theorem, as shown in the formula 7.21. Since a face in an image appears smaller with increasing distance from the camera, the scaling factor derived from the intercept theorem must be inverted. Consequently, if $d_i > d_{desired}$, then it follows that $s_{depth} > 1$.

$$\frac{\hat{p}_i^k}{p_i^k} = \frac{d_{\text{desired}}}{d_i} \Rightarrow \hat{p}_i^k = p_i^k \cdot \frac{d_{\text{desired}}}{d_i} = p_i^k \cdot \frac{1}{s_{depth}} \Rightarrow s_{\text{depth}} = \frac{d_i}{d_{\text{desired}}} \tag{7.21}$$

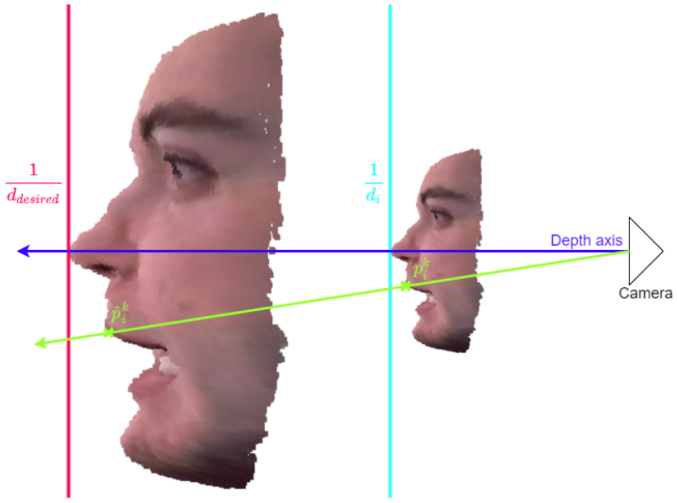


Figure 7.27.: *Visualization of the depth scaling factor. The inverse of the depth scaling factor is shown in this illustration by the reciprocal values of the depth values. The calculation is based on the intercept theorem. Images by René Ebertowski.*

After normalization, we set all images to a uniform resolution of 512×512 pixels. Note that during the capture process, SynergyNet checks in real time whether the face has a

minimum resolution (i.e. a maximum distance to the sensor) when the data set is captured. If this is not the case, the user is informed. This means that we ensure that all captured and processed images have a higher resolution than the target resolution of the GAN. We use bicubic interpolation to resize the images.

6.) Histogram normalization, mask and sort
Similar to step 4 of Sec. 7.5, the histogram is normalized, the background behind the user's head is rejected, and the data is randomly selected for 85% for the training set and 15% for the test set. The training process can also be started from the GUI, as shown below:

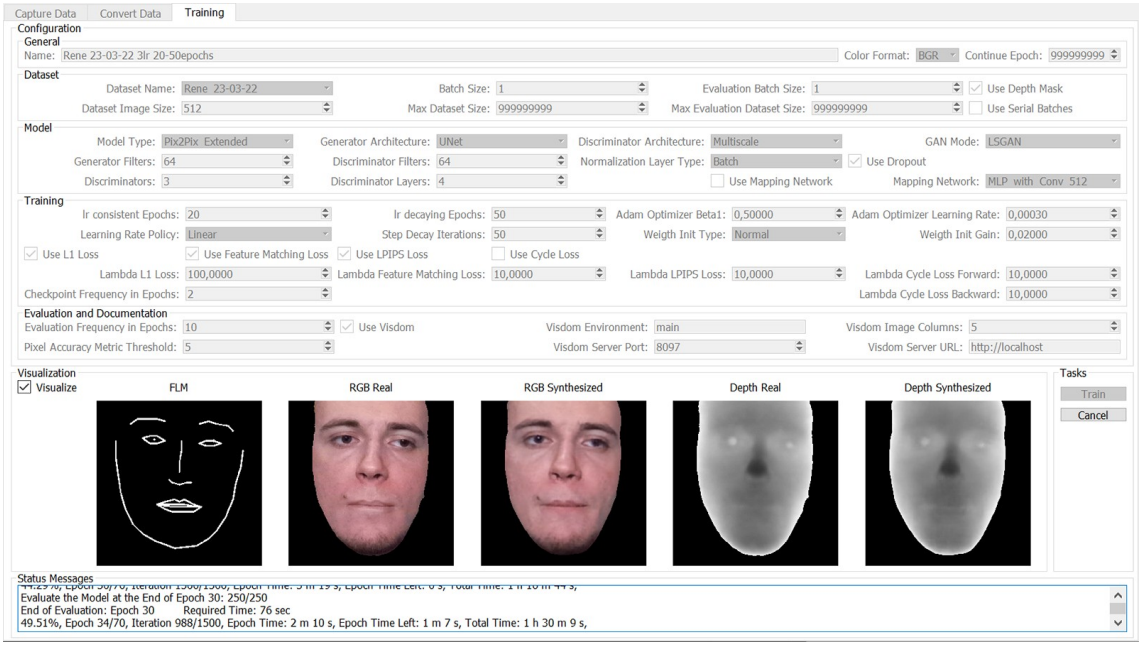


Figure 7.28.: *The graphical user interface helps to create and monitor the training process. Various parameters for the model and the training process can be specified in the configuration area. The visualization area displays the inputs and outputs of the neural network and the corresponding images from the data set for comparison. Images by René Ebertowski.*

7.7.2. Results

Fig. 7.29 shows the results of a training session using a dataset processed through the data pipeline of our third iteration. Compared to the datasets from the previous prototype, the image quality is inferior. Although the previous prototype achieves similar results in terms of SSIM, LPIPS reports inferior values. While the best values of the current prototype are around 0.120, the worst results of the previous prototype end up in this range. In terms of quality, a more blurred and less detailed representation of the face can be seen, as shown in Fig. 7.29. Please compare the results directly with Fig. 7.19 and 7.20.

We believe this dataset underperforms because it contains a greater variety of pan and tilt rotations compared to the images captured with the helmet mount. The helmet mount of the previous prototype made it impossible to rotate the head relative to the sensor. However, the new capture process allowed a maximum pan/tilt/yaw angle of 10° , supervised by the SynergyNet. We suspect that this threshold was set too high, resulting

in too much variance in the data set. This, in turn, leads to poorer final image quality because the training set contains several minimally rotated faces that are associated with very similar FLMs. This means that a presumably 1-to-1 relationship between an FLM and an RGB-D image tends to become a 1-to-n relationship, degrading the final image quality.

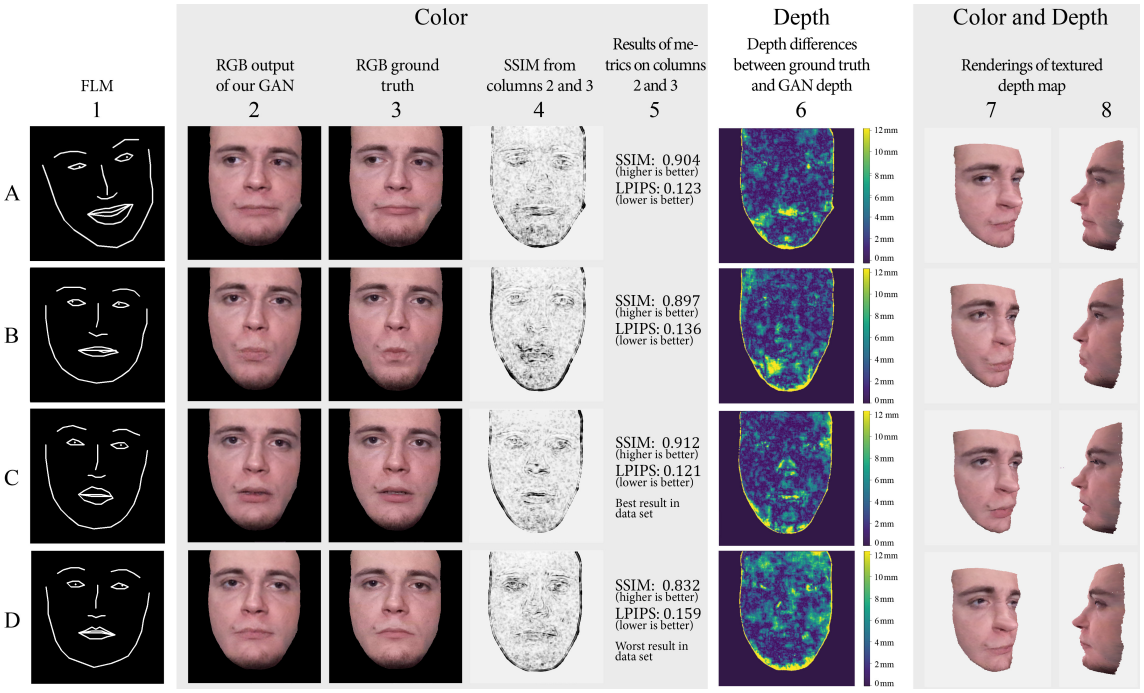


Figure 7.29.: *Although the third prototype does not require a helmet mount (Fig. 7.3), it produces a lower image quality. This is probably due to the fact that the dataset we capture using a stationary RGB-D sensor allows minimal head rotation, which increases the variance in the dataset and reduces the resulting image quality. Images by René Ebertowski.*

7.8. Discussion, Limitations and Future Work

The prototypes presented show solutions and some potential for transmitting NVC between physically separated individuals. Compared to video-based telephony, our GAN-based solution is characterized by the fact that only a small amount of bandwidth is required during operation. Once the generator module is transmitted to the remote side (it always has a size of 202MB for the 512×512 pixel version, regardless of the shape or expression of the face in the training set), the network bandwidth required to drive the avatar is only 67 kbit/s. Video telephony typically requires bandwidths of 0.3 to 3 Mbit/s, depending on the resolution, what is around 4 to 40 times less data to transmit. In addition to the usual RGB information, we also provide a depth channel, which would increase the required bandwidth by an additional 33%. The low bandwidth for our system is achieved because we do not send raw image data, but only the image positions of 70 landmarks as integers. Although compression is not the focus of our work, it is interesting to note that the required bandwidth of an RGB-D stream is orders of magnitude higher. On the receiver side, the FLM is reconstructed as an image again, which can be considered as a drawback because the receiver system has to provide the computing power for the image

synthesis inference for the GAN. In our experiments, the required computational power is provided by a desktop computer with a powerful GPU. It is currently still difficult to provide this performance on a standalone MR device.

Although we are able to create an almost photorealistic avatar of a person, there are three areas where there is a lot of room for improvement: First, it is important to replace the cumbersome helmet mount with a user-friendly algorithm so that no specialized hardware is needed to create an avatar. Since 2022, sufficiently accurate (open source) face tracking methods have been available, making it much easier to solve the tracking problem to create an accurate training dataset.

The second important area is accurate face tracking under an HMD, which was discussed previously in Chap. 5. Although a solution has been presented in this thesis, it cannot capture the full range of expressions with all the details of a human face, which is usually required to reproduce the experience of a real face-to-face conversation.

The third area of future interest is improving the speed of execution. As described, our system requires a desktop computer, and while the GAN achieves interactive frame rates, this computing power is not yet available for mobile MR devices. The current trend in academic research shows the potential of Implicit Neural Representations (INR) such as NeRFs or alternatives, that does not even use a neural network such as Gaussian splatting. In particular, the latter technique offers enormous potential in terms of execution speed, as it applies the neural network training method (Stocastic Gradient Descent) to 3D scenes by a lot of 3D points (Gaussians) and then uses only hardware-accelerated 3D visualization during the subsequent execution time. Avatars rendered using Gaussian splatting and similar methods may have great potential for display even on mobile devices.

7.9. Conclusion

In this chapter, we focussed on improvements of Generative Adversarial Networks (GAN) to answer the 6th research question "How to transfer the face in a photorealistic appearance with authentic movement in real time despite wearing an HMD?". We presented techniques for conveying nonverbal facial communication cues in a VR environments and realized face tracking with the face-tracking HMD introduced in the last chapter.

Through three iterative prototypes, we presented a pipeline that captures, processes, and renders identity-preserving photorealistic facial avatars in real time. The first prototype established a basic framework using the Pix2Pix GAN [Iso+17] architecture. We have extended Pix2Pix with an additional channel for depth data and demonstrated the feasibility of real-time facial avatar generation from RGB-D data with about 950 fps with a resolution of 256^2 pixel.

The second prototype incorporated advanced loss functions and architectural enhancements, including multiscale discriminators and feature matching losses. These enhancements increased the final resolution to 512^2 pixel. In related work, GANs have shown that higher resolutions can lead to poorer image quality, but we were able to successfully stabilize the training process and even further increase the visual quality in regard to higher image sharpness and more details compare to the first prototype. In order to maintain the high frame rates, the generator was changed only to a minimal extent and still delivers refresh rates of around 250-333 fps what is sufficient for telepresence. The training pipeline and the discriminator module were changed significantly compared to the foundation code,

the Pix2Pix [Iso+17]. Moreover, we have also reduced the training time to a quarter of the original time of the first prototype through hyperparameter tuning.

The third prototype aimed to improve usability by eliminating the need for a helmet mount during data acquisition. By implementing real-time face orientation and distance verification algorithms, we created a more user-friendly solution. Although this iteration was more comfortable, it highlighted the challenges associated with maintaining image quality, as the new capture process introduced more variability into the training data set, which in turn led to slightly more blurred results compared to the second prototype.

In summary, our research contributes to the ongoing effort to create immersive telepresence environments. By enhancing the realism and expressiveness of digital avatars, we move closer to bridging the gap between virtual and real-world interactions. The next chapter explores the same research question, but from a different technical perspective based on Implicit Neural Representations (INR).

8. Face Rendering with Implicit Neural Representations

Various researchers have already suspected that deep neural networks with ReLU activation functions tend to learn low-frequency components of functions more easily than high-frequency components. This was demonstrated and proved by Rahaman et al. [Rah+19] and the authors called this phenomenon "spectral bias". This is an important finding for reconstructing and generating data such as text, images, videos, audio, and 3D environments with neural networks.

Vaswani et al. [Vas+17] were already aware of the existence of spectral bias and presented the Transformer architecture two years before Rahaman and his team. Vaswani et al. introduced the simple but effective *Positional Encoding*, which allows Transformer to better encode the order of data elements and, therefore, can reconstruct (or "know") the relative position of a respective data element in its context much better compared without positional encoding. Positional encoding is a function that adds further dimensions to the input data using Trigonometric functions. For example, a 1D position vector for a word in a sentence (e.g. "the 5th word in a sentence") becomes a 10-dimensional vector, i.e. a single value is encoded by 10 values. With this encoding, a ReLU-based network can reference and reproduce the positions of data elements relative to other elements much better.

Positional encoding was applied three years later by Mildenhall et al. [Mil+21] to volumetric data in combination with ray tracing and produced astonishingly good results, which until then had not been achieved in the context of neural view synthesis (NVP). Sitzmann et al. [Sit+20] achieved similarly good results for volumetric data, especially for signed distance functions (SDF), a little earlier. However, instead of positional encoding, the researchers used sinusoidal activation functions instead of ReLU within the entire network. They were one of the first to successfully initialize and train a neural network with sinusoidal activation functions. The success of positional encoding and sinusoidal functions is based on similar circumstances and reasons why neural networks can better store high-frequency data/functions with the help of circular functions [Tan+20].

Implicit Neural Representations (INR) or also called coordinate-based neural networks got their name from the ability to use input encoding methods, such as positional encoding, lookup or hash tables, to send precise multidimensional coordinate queries to the neural network. The network can then respond with accurate data at those specific coordinates that is implicitly stored in the weights of the network. To illustrate this, consider an RGB image such as the image from the fox below (Fig. 8.1). When a traditional fully-connected MLP is trained on an image, such that it receives pixel coordinates in integer x and y, and the network is supposed to output the corresponding RGB data at that location of the image, it can be observed that the network only stores and reproduces low-frequency signals, as seen in the middle of Fig. 8.1. However, when positional encoding is used at the input, encoding the x and y data, the neural network is able to reproduce the individual color information of the learned image with high precision (right in Fig. 8.1).

Figure 8.1.: *To overcome the spectral bias of neural networks, Fourier features can be used as a higher dimensional input signal. On the left, the supervision image can be seen, which is the ground truth reference that an MLP is supposed to learn. In the middle, the output of an MLP without Fourier features is shown. Although the network has more parameters than pixels in the image, it is not able to reproduce the signal sharply. On the right, the same MLP with Fourier feature as input is used, showing only a minimal difference from the reference image in on the left. Images from Tancik et al. [Tan+20].*

In the previous chapter, GANs were discussed in detail. The networks presented in the previous chapter also use ReLU activation functions, but are able to store high-frequency data. This can be explained by the multi-layer architecture and the associated hierarchical feature extraction of CNNs. This means that different layers are responsible for different resolutions and therefore also image frequencies. Spectral bias will also be present in these architectures, but it is circumvented by a kind of scaling of the low-frequency signals stored by the ReLU functions due to storing features at different image resolutions. The disadvantage of CNNs is that the layers have to be run through consecutively and the calculations build on each other. However, if coordinate-based neural networks are used, it is possible to reduce the amount of consecutive and sequential queries by making certain adjustments and sorting input signals beforehand and can obtain image data much faster.

In this chapter, we explore the advantages of INRs, particularly coordinate-based neural networks, and tailor them specifically for real-time face animation. Our contribution is the reduction of execution time of an INR-based approach while keeping the visual rendering quality of animated human faces in telepresence applications, thereby enhancing the conveyance of nonverbal communication cues. This chapter presents another approach that shows a real-time capable synthesis of facial animations. We show that, in contrast to GANs, INRs have a lower generalization capability and thus a lower interpolation and extrapolation capability between, for example, facial expressions. Therefore, our approach does not manage to outperform GANs in quality and speed, but INRs still have some open research areas and could have the potential to become significantly better in quality and speed. The advantage of INRs lies in the potential to perform fewer consecutive computational steps and thus significantly speed up the process of facial animation.

8.1. Related Work

8.1.1. Implicit Neural Representations

Prior to the discovery of NeRFs, there was some preliminary work that attempted to implicitly store signed distance fields or also called signed distance functions (in both cases SDF) in neural networks. The first publications in the field of INR were Park et al. [Par+19] and Mescheder et al. [Mes+19]. Both works successfully store SDFs in the network and attempt to implicitly learn a three-dimensional decision boundary according to the simple formulas $o : \mathbb{R}^3 \rightarrow \{0, 1\}$. While Mescheder et al. [Mes+19] still use encoder architectures, Park et al. [Par+19] rely on fully-connected ReLU networks with a skip connection, which will prevail in the later course of implicit neural representations. Mildenhall et al. [Mil+21] used an 8-layer, 1024-neuron fully-connected architecture for NeRFs. However, the special feature of NeRFs was not the architecture, but the positional encoding, which, as mentioned above, adapts the input data in such a way that neural networks are able to reproduce certain color values and the density at a certain coordinate in the volume much more accurately. At the time, however, it was not known exactly why positional encoding was effective. Sitzmann et al. achieved similar results with high precision using the so-called SIREN [Sit+20]. Instead of positional encoding, Sitzmann et al. used sine instead of ReLU as the activation function, and were able to outperform ReLU networks with positional encoding. Sitzmann et al. showed that SIRENs can store various media with high quality, such as images, videos, audio, volumetric data, mathematical functions, and other modalities. It should be noted that the size of the neural network was generally larger than the media file. This means that the data was not compressed.

Building on NeRFs, the properties and storage potential of neural networks have been further explored. Different positional, coordinate or parameter encoding approaches have been investigated, for example to sample INRs with different spatial resolutions with less anti-aliasing artifacts, as demonstrated by Barron et al. [Bar+21]. Martel et al. introduced ACORN [Mar+21], which follows an adaptive spatial partitioning approach that samples high-frequency portions of a medium at higher resolution and low-frequency portions at lower resolution. The authors call their approach a hybrid implicit-explicit network architecture. ACORN allows the available capacity of the neural network to be focused on the areas where information relevant to the user is stored. This saves training and inference time and streamlines the network in a way that complex structures can be captured much faster and more efficiently. A special feature of ACORN is that the adaptive ability to subdivide the space independently and effectively is learned and optimized by backpropagation. This means that the training process adapts automatically the sampling frequency in areas with more details. Two ReLU MLPs with positional encoding are used, where the first network acts as a coordinate encoder and the second MLP is much smaller and acts as a feature decoder.

Müller et al. presented a further improvement for INR with Instant-NGP [Mül+22] and achieved not only faster training and inference times than ACORN, but also required less memory for the INR. While ACORN used an MLP with several million parameters as a coordinate encoder, Müller relied on a hash table. The advantage of hash tables is the constant query time of $O(1)$, which is much faster than the query by a multi-layer MLP, which requires matrix multiplications and the execution of activation functions. Müller et al. retained the feature encoder as a small MLP of two fully-connected layers with 64

neurons each. Surprisingly, hash collisions were automatically detected and eliminated by training.

Subsequent research has shown that in the original architecture of NeRFs, a large part of the capacity of the neural network was used for storing coordinates and only a small part was used for storing features such as color or density. With DINER [Xie+23], Xie et al. provide deeper insights into how exactly neural networks use positional encoding. The following Fig. 8.2 below shows how an hash table before a 2x64 fully connected MLP helps to learn a signal. The experiment includes three signals, as shown in column (a) "Ground Truth (GT)". The original image of a baboon is shown at the top. In the middle, the pixels are sorted by color from left to right, and in the bottom image, the pixel positions of the original image are randomly chosen. In all three images, the histogram of the images is identical, but the frequency greatly differs. Column (b) shows the Fourier spectrum. Column (c) shows the reconstruction of a 2x64 fully connected MLP with positional encoding and without a hash table. Column (d) shows the hash-mapped input, which is automatically sorted by color from the training process and thus shows a low frequency. Column (e) shows the same MLP as in column (c), but with a hash table and without the positional encoding. The hash table consists of learnable parameters and is trained jointly with the MLP. Note the very high reconstruction quality of the original signal, independent of the frequencies present in the ground image of column (a). Column (f) shows the output of the MLP when grid coordinates are fed directly into the MLP. Note that the MLP learns a low frequency representation what is congruent with the theory of the spectral bias. This means that the hash table stores the high-frequency position data and the MLP stores the available features, which in this sense represent the colors.

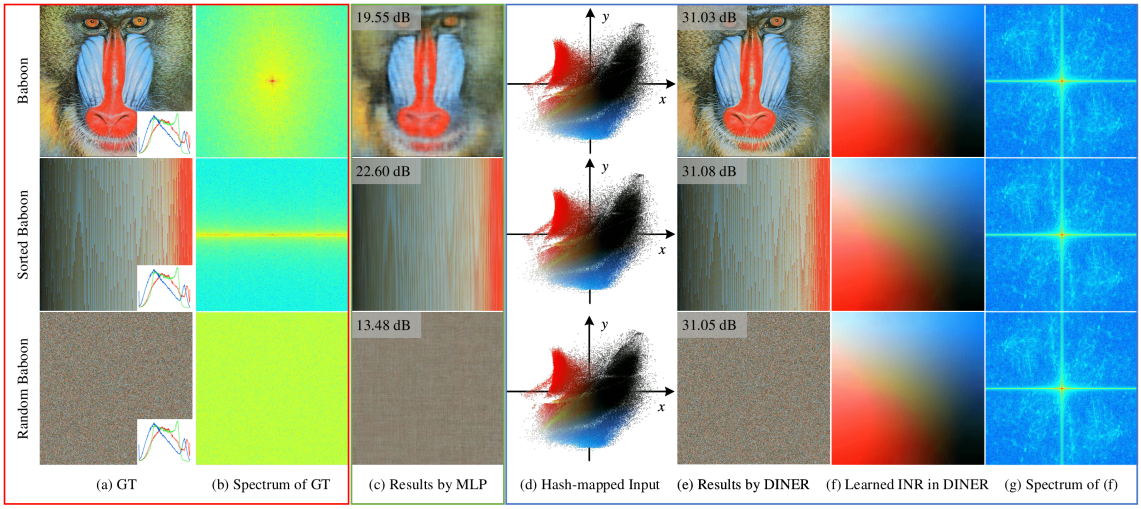


Figure 8.2.: *This Fig. explains, why the combination of a jointly-optimized lookup table with learnable parameters and an MLP successfully learns disorder-invariant INRs. The lookup table "learns" high-frequency positional information, while the MLP stores a low-frequency feature distribution. For a detailed explanation, see the text above. Image from Xie et al. [Xie+23].*

8.1.2. Face Animation with Implicit Neural Representations

A large number of different works have shown the potential for rendering and animating photorealistic faces with INRs. With NeRFace, Gafni et al. [Gaf+21] was one of the first

to use NeRFs for facial animation with INRs. Compared to the original NeRF work by Mildenhall et al. [Mil+21], Gafni et al. only require a monocular RGB video sequence. To realize this, the use of a 3DMM and corresponding face tracking is necessary, which can accurately determine the rotation and position of the head. For this purpose, the method of Thies et al. [Thi+18a] was used. By moving and tracking the rotation and position of the head, the multi-view data set required for NeRF, is replaced by inversely transferring the head movement to the camera position. This causes the camera to move around the head and serves as an alternative form of an multi-view data set. Furthermore, it allows creating a "canonical face space". This is essential to isolate the head position and rotation parameters from the expression vectors, otherwise the INR would learn specific expressions correlated to specific head positions or rotations. The results require several hours of training, are not real-time capable and, therefore, cannot be used for telepresence applications.

Zielonka et al. (INSTA) [ZBT22a] also use a canonical space compared to Gafni et al., but relies on the improved concept of Instant-NGP by Müller et al. [Mül+22]. This way, Zielonka et al. not only shorten the training times from hours (NeRFace) to only 10 minutes but also enables real-time rendering, which makes the use in MR telepresence applications theoretically possible, but was not explicitly demonstrated. Similar to Martel et al. with ACORN [Mar+21], Zielonka et al. uses a hierarchical tree structure that splits the space and can avoid calculations for ray computations, which further improves the speed of the system. However, NeRFace and INSTA have the weakness that the systems generalize poorly for unseen expressions.

Zheng et al. [Zhe+22] introduced "I M Avatar" and solves the problem of lack of generalization of unseen expressions by an architecture of 3 large fully-connected coordinate-based MLPs. This approach allows not only to interpolate expressions and poses based on the blendshapes of the parametric face model FLAME [Li+17], but also to extrapolate them further than NeRFace and INSTA. A major disadvantage, especially for the application of real-time telepresence, is the lack of real-time capability and interactivity of the system. The inference time is not reported, but based on the usual execution time for ray tracing and based on the documented large neural architecture in the paper, we assume that the system is computationally intensive and comparable with NeRFace [Gaf+21].

Grassal et al. [Gra+22] do not rely on ray tracing and instead use the classic rasterization pipeline with their system called "Neural Head Avatars". This makes their approach significantly faster than, for example, that of Zheng et al. (I M Avatar), but in this work, as in many others, the focus is not on a short execution time. Current work rather focuses on the visual quality of the results. The work of Grassal et al. has similarities to Zheng et al. (I M Avatar) in that Grassal et al. also uses FLAME, adapts the head geometry including hair with a coordinate-based MLP and uses another MLP for the texture. The special feature, however, is the architecture of the MLPs, which is strongly oriented towards the work of Sitzmann et al. [Sit+20]. Sitzmann et al. report on an experiment in which they use a mapping network based on FiLM by Perez et al. [Per+18]. FiLM is similar to the idea behind StyleGAN by Karras et al. [KLA18], which uses AdaIn to better control different styles and structures. Grassal et al. uses FiLM to condition expression and pose parameters of FLAME in the SIREN architecture. Compared to previous work, their approach is interactive and fast, making it compatible with telepresence applications on high-end hardware. In addition, it would be easy to integrate into existing applications since the system is based on a rasterization pipeline instead of ray tracing.

Lombardi et al. [Lom+21] introduces "Mixture of Volumentric Primitives". It is a hybrid

between polygonal mesh-based and volumetric-based approaches. Key contribution is an optimized end-to-end learning system, that creates small volumes, that are ray marched, along the mesh surface of the avatar. Creating small volume along the actual avatar avoids sending rays through empty space. This system has some similarities with Gaussian Splatting by Kerbl et al. [Ker+23]. Similar to Gaussian Splatting, Lombardi et al. optimize their system for positions, rotation and scaling of the individual volume primitives. A Variational Autoencoder (VAE) [KW19] generates the content, called payload, of these volumes. Gaussian Splatting, on the other hand, uses simple primitives (Gaussians) which are significantly smaller than the volumes of Lombardi et al. but can also occur in significantly higher numbers in the scene, as the pipeline of a rasterizer is hardware accelerated. Due to the VAE, however, Lombardi's system may offer a better illumination reconstruction than an approach based on Gaussian splatting.

8.2. System

This chapter describes our method for creating a controllable head model using a 3-5 minute monocular RGB video, e.g. created with a smartphone. By leveraging the FLAME 3D Morphable Model (3DMM) [Li+17], we extract and track facial identity and expression parameters. The process involves dividing the video into "expressive" and "silent" segments to evaluate the face synthesis independently from telepresence applications. We use a 2.5D approach with so-called "rainbow encoding" to encode the face surface in RGB coordinates, optimizing the neural network input for real-time, accurate rendering.

8.2.1. Capture Process and Training Dataset

Similar to our GAN-based approach, our goal is to build a controllable head model. For this we need a 3-5 min video of a person as in Chap. 7. Compared to Chap. 7, however, we only need RGB instead of RGB-D data. Depth data is no longer necessary with the use of a 3DMM in our use case. Although it would tend to speed up and stabilize face tracking, as more data would be available and it would be easier to fit the 3DMM using the Iterative Closest Point algorithm (ICP), for example, RGB-based face tracking solutions are now available that provide solid tracking results based on RGB data.

With the help of the work of Zielonka et al. [ZBT22b] we determine from a single still image from the video, in which a neutral expression is shown, the identity parameters $\beta \in \mathbb{R}^{300}$ of the target person. We then use these identity parameters and track further face parameters in the full training video, as shown in Fig. 8.3. Of each image I_i of this monocular video, we determine the intrinsic camera parameters $K \in \mathbb{R}^{3\times3}$ as well as the FLAME mesh $M_i \in \mathbb{R}^{5023}$ with corresponding facial expression coefficients $\Psi_i \in \mathbb{R}^{50}$, eye gaze parameters $G_i \in \mathbb{R}^{12}$ and jaw pose parameters $J_i \in \mathbb{R}^{6}$.

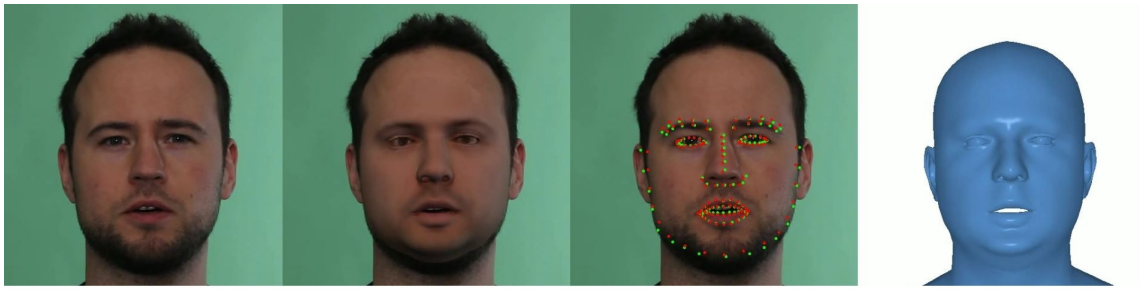


Figure 8.3.: *We used "MICA" for determining the face identity parameters and the "Metrical Tracker" for estimating face expression parameters in each frame of our training, validation and silent video. MICA and Metrical Tracker are works by Zielonka et al. [ZBT22b]. Image a) shows the monocular video frame. In b) the fitted 3DMM is shown as overlay over the input frame. The texture is used for optimization and is based on the Basel face model [BV99]. c) shows facial landmark detection of MediaPipe [Lug+19]. d) shows the rendered FLAME model without texture.*

In contrast to the pipeline we describe in Chap. 7, we divide the video into two parts. The first part comprises a 3-4min recording while the person reads out a predefined text (a 2-pages long random story created by ChatGPT) and shows various expressions such as laughing or raising eyebrows. We call the second part "silent video" and is 1-2 minutes long. In this video, the person does not speak and only moves their head slightly. The person nods from time to time, as if showing that they are listening to someone. In our experiments, we decided to overlay the generated face in the silent video in order to be able to better subjectively evaluate later how well the synthesis of the face harmonizes with the rest of the head. Therefore, we skipped the direct integration into a telepresence application, as we have learned from previous experiments that face tracking in the HMD is a variable that can strongly influence the final visual result. Therefore, we decided to decouple face reconstruction from face tracking in the HMD in this chapter.

8.2.2. Rainbow Encoding

One of the main goals of our work is real-time capability, so that this can be used in a telepresence context. Similar to Chap. 7, we have therefore again opted for a 2.5D approach and rely on hardware-accelerated processes of a rasterizer instead of computationally expensive ray casting, as used in NeRFs [Mil+21] and the follow-up work. Therefore, our approach is to render a 3DMM in three dimensions, but only reconstruct a projected texture from the viewer's point of view in screen space with an INR. We sacrifice a complete 3D reconstruction and higher quality for speed. While in NeRFs a ray is shot through the 3D volume for each pixel of a generated frame and the neural network is traversed several hundred times along this ray, in our system the neural network is traversed much less. This is because we only sample the visible pixels of the current face crop of the current frame and use those pixels as input to the neural network. Thus, the network is only iterated as often as there are pixels in the face. We do this using a kind of color coding, which we will call "rainbow encoding".

Our idea is to encode the surface of the face in a similar way to the UV coordinates commonly used in computer graphics, which serve as input for the INR. This means that every point on the face is uniquely encoded. The idea of encoding using visualization based on vertex positions comes from Doukas et al. [DZS21]. We use this encoding instead of

the UV data in order to run the INR networks in screen space and not camera space. The difference between screen space and camera space means that, for performance reasons, we only have to generate the currently visible pixels in the final image using the INR networks, instead of generating a texture in camera space on the entire 3D FLAME mesh, and we also have to take the Nyquist-Shannon sampling theorem into account. This means that we would ideally have to generate the texture in double resolution in camera space in order to avoid image artifacts in screen space. Please note that our encoding also transforms the input data in the range between -1 and 1, which is necessary before feeding the data into the network for optimal learning.

The original FLAME template does not have any faces in the oral cavity. We have closed this area manually and inserted faces, as otherwise there would be no rainbow encoding inside the mouth. We generate the RGB "rainbow" data once in advance of the training and leave it unchanged in later stages. To do this, we encode the 3D position of each vertex in the face crop area on the neutral FLAME mesh (identity and expression parameters are 0) as RGB data. This means that we pass the position data of each vertex from the vertex shader through the fragment shader, where the XYZ data is converted to RGB and rendered. The surface colors of the faces are interpolated between the colors of individual vertices. This ensures that there is a continuous transition between the edges and faces in such a way that each pixel in the final "rainbow image" has a unique color. The rainbow images are used as input to our INR networks and have an alpha mask that allows the INR networks to be executed only in the face area. This ensures that computing time is only used for essential areas and that no unnecessary calculations take place, such as in many NeRF papers where rays are sent through the empty space.

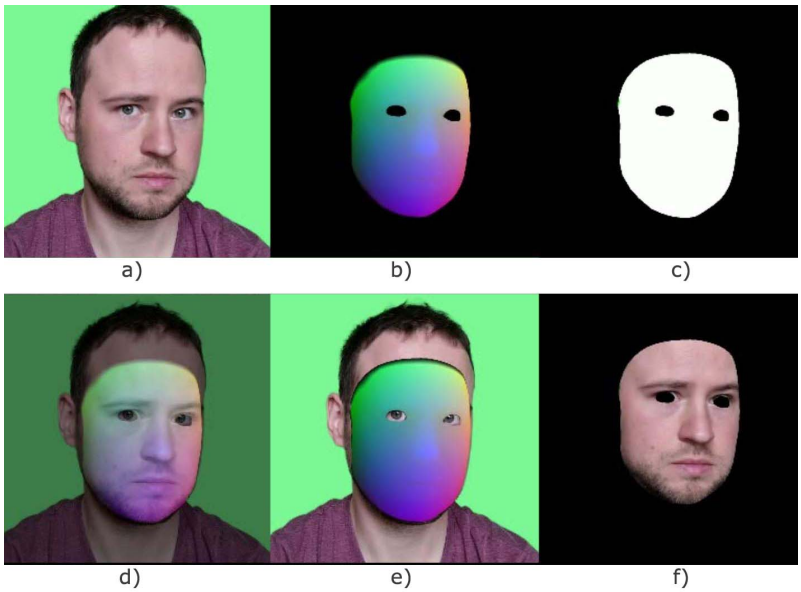


Figure 8.4.: *Combination of the tracking results of the Metrical Tracker (Zielonka et al. [ZBT22b]) with our "rainbow encoding". a) shows a frame from the "silent" video part. b) shows the according rainbow encoding to this frame. c) is the alpha mask. d) is a transparent and e) a non-transparent overlay. f) shows the applied alpha mask on the original frame from a). Note, that our INR optimizes on this face texture.*

8.2.3. Render Pipeline

The rainbow images are the base for the rendering process for the photorealistic synthesis and serve as input to the neural networks. The following image illustrates the role of the rainbow images and the networks in the entire render process:

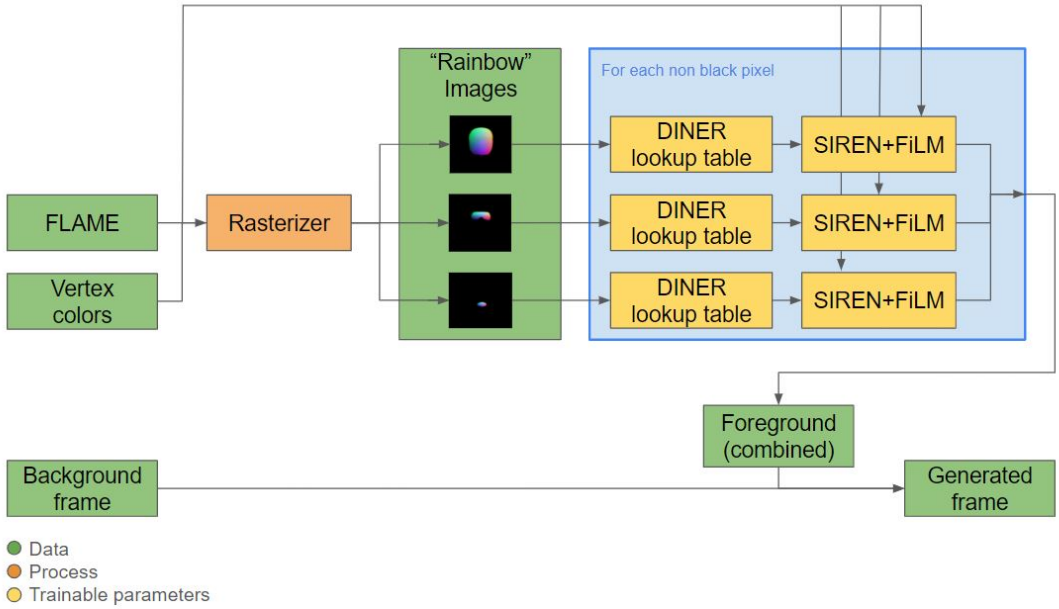


Figure 8.5.: *The render pipeline starts with the input of FLAME parameter. The FLAME mesh is provided with the previously defined rainbow encoding colors for the vertices and three variants are rendered by a rasterizer. These three rainbow images include a crop of the eye area, the mouth and the entire face, as shown in Fig. 8.6.*

As already mentioned in related work, Mescheder et al. [Mes+19] and Park et al. [Par+19] as well as SIRENs [Sit+20] and especially NeRFs [Mil+21] for novel view synthesis were the foundational work in the field of INRs. Follow-up papers have continuously developed new optimization methods for INRs, which require less memory and less computational effort. Xie et al. [Xie+23] has shown that a simple lookup table, which consists only of learnable parameters and is placed in front of a comparatively small fully-connected MLP, achieves very good results for INRs. Xie et al. [Xie+23] found the concept of disorder-invariant implicit neural representation, DINER for short. Xie et al. showed that DINER in combination with MLPs that use sinusoidal activation functions (SIRENs) [Sit+20] achieve better results for image data than MLPs with ReLU activation.

In our approach, we therefore use the concept of Xie et al. [Xie+23] (DINER) for our INR. We use a lookup table with 256^3 learnable parameters and instead of an ordinary ReLU MLP as presented in Instant-NGP, a SIREN MLP [Sit+20]. In addition, we extend the SIREN network with a FiLM mapping network [Per+18], which was successfully used for face rendering by Chan et al. (pi-GAN) [Cha+21] and Grassal et al. (Neural Head Avatars) [Gra+22]. The mapping network receives expressions and pose parameters from FLAME and is able to realize a kind of "pre-sorting" of the input signals. The mapping network is trained jointly with the lookup table and the SIREN MLP.

Our experiments have shown that the visual quality increases if we train individual net-

works for certain areas of the face. Instead of training the entire face with one network, we therefore use three separate smaller networks. One network that reconstructs the eye area, one for the mouth and one that represents the rest of the face. We superimpose the three results for the final image, as shown in the following Fig.:

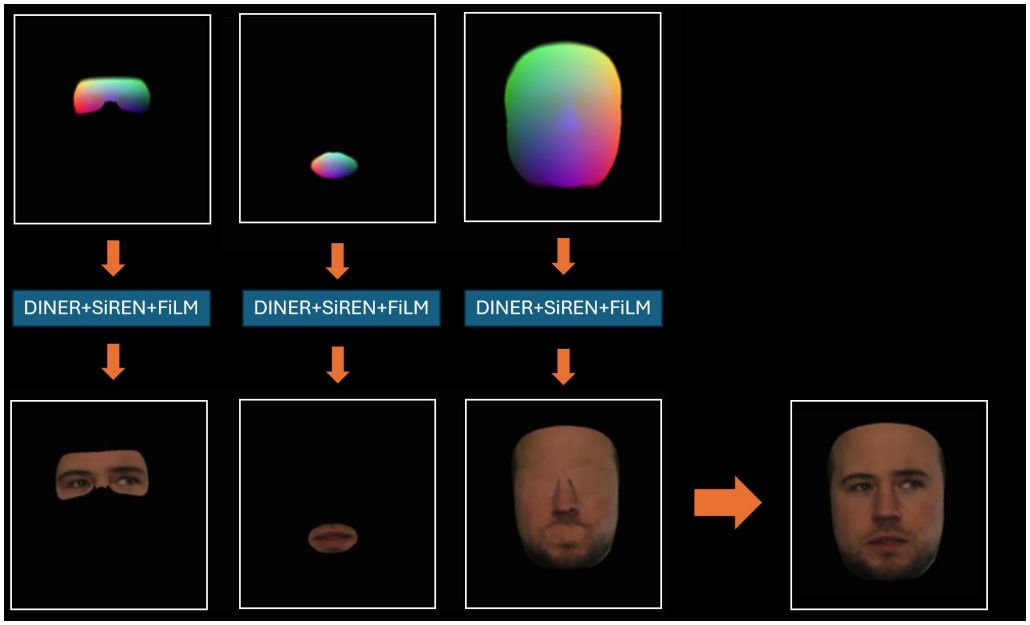


Figure 8.6.: *We use three separate DINER+SIREN networks to generate specific areas of the face. This approach delivers better results than a single DINER+SIREN network. We obtain the crops of the individual areas through face tracking and alpha blending with predefined FLAME face areas.*

The blue rectangle at the top right in Fig. 8.5 shows the combination of three strands of DINER+SIREN+FiLM networks. Each strand is specialized on eyes, mouth or the background of the face and receives two data as input: first, a single pixel of the according rainbow image generated by the rasterizer based on certain FLAME parameters and second, the FiLM mapping network receives 68 FLAME parameters. Among them are $\Psi_i \in \mathbb{R}^{50}$, eye gaze parameters $G_i \in \mathbb{R}^{12}$ and jaw pose parameters $J_i \in \mathbb{R}^6$. The following Fig. 8.7 shows the architecture in detail.

The FiLM architecture is inspired by Sitzmann et al. [Sit+20]. Please note that it is important how the SIREN layers need to be initialized before training. We have also used the initialization scheme of Sitzmann et al.. The size of the linear layer in the SIREN part is inspired by the color MLP of Instant-NGP [Mül+22] and DINER [Xie+23]. We use Mean Squared Error (MSE) as the loss function and employ the Adam optimizer [KB17] with a learning rate of 0.002. Our implementation utilizes PyTorch 2.0.1, Pytorch3D 0.7.4, and Kornia 0.6.12 [Rib+20].

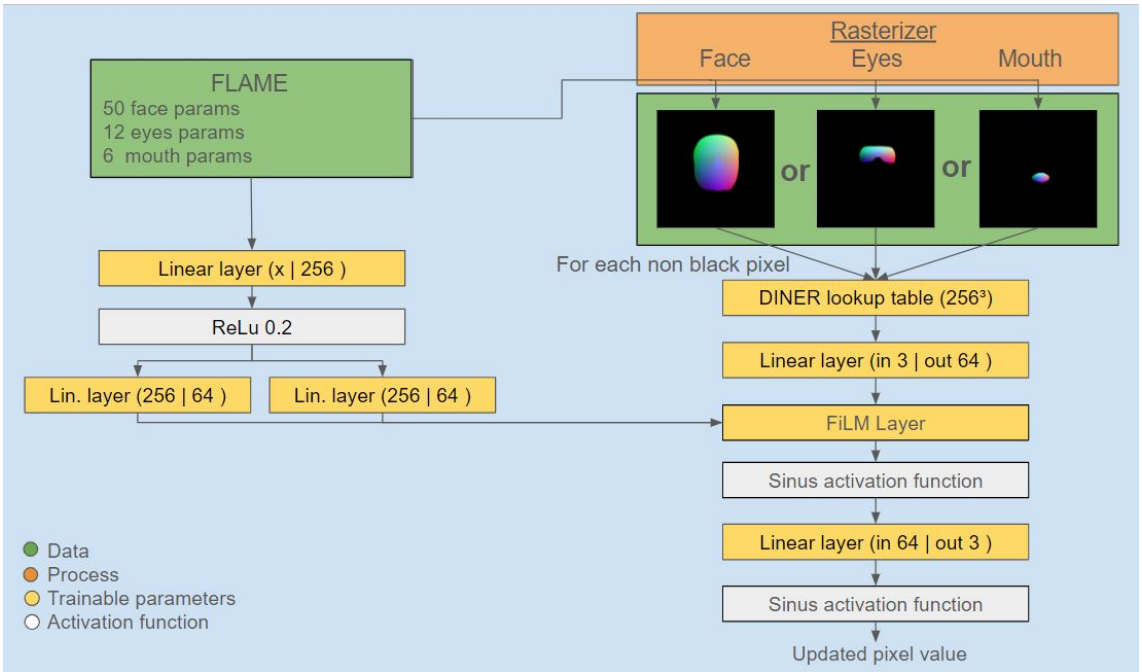


Figure 8.7.: *This illustration shows the blue rectangle from Fig. 8.5 in detail. The FLAME parameters are the foundation. This data is passed to the rasterizer, which generates three rainbow images for the eyes, mouth, and background. The FLAME parameters are also passed to the mapping network, which injects data (latent code from the FiLM mapping network) into the DINER+SIREN pipeline. This way, not only are the rasterizer images used as input, but also the FLAME parameters are used directly.*

8.2.4. Lib Sync

In 2024, off-the-shelf face tracking hardware for HMDs still struggle to authentically capture the mouth region when speaking. As an alternative to face tracking, we have integrated an audio-to-lip movement pipeline. This way, the HMD's microphone can be used to control the avatar's lip movements. To accomplish this, we use FaceFormer by Fan et al. [Fan+22]. FaceFormer uses Wav2Vec2 [Bae+20] for the audio analysis and receives an audio track with speech, analyzes it based on the MEL spectrograms with a CNN and forwards the local features as latent space to a transformer network, which generates the contextualized representations. The dataset of VOCA [Cud+19] is used to build a connection between audio and a vertex offset in the FLAME mesh, which is the core component of FaceFormer. These vertex offsets move the mouth according to the audio. However, a challenge for using FaceFormer in our application is that we need blendshape parameters for the expressions of the FLAME mesh for our pipeline in Sec. 8.2.3 instead of vertex offsets, because the DINER+SIREN+FiLM render pipeline learns to generate the final texture based on the blendshape parameters. For our case, we did not pursue a new training of the FaceFormer pipeline, as the authors of FaceFormer had already stated in GitHub issue (No.35) that the original data set of VOCA [Cud+19] for training does not contain any blendshape information as information, but only vertex offsets. Thus, we have decided to develop our own neural network, which we have named "FafeFormer-2-Expressions-Net", short "FF2EXP-Net". It accepts vertex offsets and returns 50 blendshape parameters and one jaw pose parameter of FLAME. The following Fig. 8.8 shows the results:

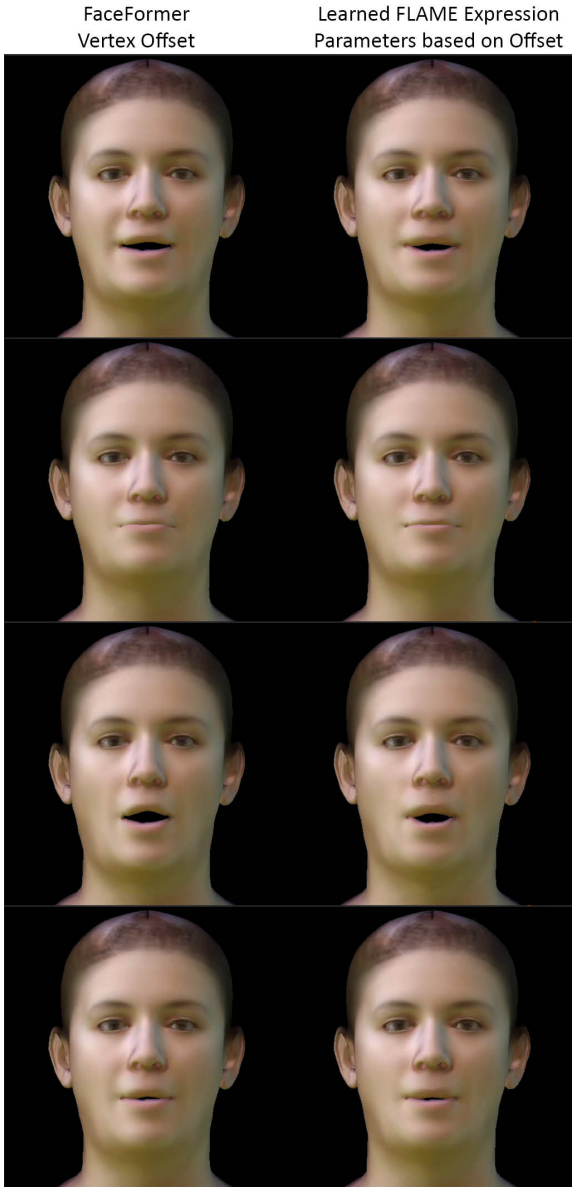


Figure 8.8.: *Results of the FaceFormer-to-Expression-Net (FF2EXP-Net) for driving the mouth of final avatar over the microphone of an HMD. FF2EXP-Net takes as input the generated FaceFormer [Fan+22] vertex offset data (left column) and infers 51 FLAME expression parameters from it (right column). This way, we can drive the avatar face with according FLAME parameters, and not only offsets. That is important, because the DINER+SIREN+FiLM pipeline needs as input FLAME parameters. The renderings do not show a generated avatar of our pipeline. They show the Basel Face Model textures [BV99] onto the FLAME geometry, only for supervision of the training.*

The following Fig. 8.9 illustrate the relationship between the different processes and data in our pipeline.

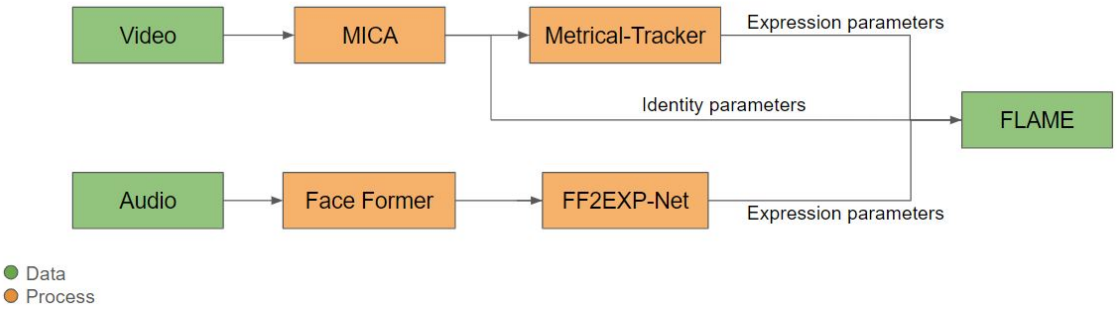


Figure 8.9.: *With MICA and the metrical tracker by Zielonka et al. [ZBT22b], we determine identity and expression parameters of the training video. With FaceFormer we drive the avatar based on audio input and can also use our pipeline in context of an interactive virtual assistant in a video call.*

Our network takes 15069 vertex offset values, which is the output of FaceFormer. This is followed by 8 fully-connected layers, each with 64 neurons. The input data is expanded to 16 bands using positional encoding. At the end of the network consists of 51 output values, which correspond to 50 expression parameters and one jaw parameter. As activation function we use LeakyReLU with 0.2. We train with a learning rate of 0.0001 with Adam over four epochs. The data set is about 10 min of audio. As loss we use a combination of 4 metrics. The first value is the accumulated L2 distance between the vertices of FaceFormer and the vertex positions based on the predicted 51 blendshape parameters of our FF2EXP-Net. The second loss is also a MSELoss, but only on two vertices of the inside of the lips. This better backpropagates the error of the mouth opening into the network. The third loss is a photometric MSELoss of the entire face and the fourth is also a photometric MSELoss, but with a focus on the mouth area. As texture, we used the Basel face model [BV99] texture with all PCA values set to zero. The weights of the individual losses were set to the following values by hyper-parameter tuning in the order in which they are mentioned above: 1, 1, 0.001 and 0.01. Also the architecture (number of layers and neurons), learning rate and epochs are determined by a hyperparamter tuning procedure using Ray Tune 2.0. A forward pass for a frame takes one millisecond on an Nvidia GeForce RTX 3090 laptop version. FaceFormer is also real-time capable and makes the entire audio-to-lip-movement pipeline interactively applicable in a telepresence environment. Please note that we do not use any optimization strategies such as Pytorch's Autocast (formerly known as Nvidia APEX) or tracing the model with a TorchScript JIT compiler.

8.3. Results

8.3.1. Self-driven Avatar

We show results of self driving the avatar, which is similar to a telepresence setting, and we show that our system can also be used in other areas. Our system is capable of authentically capture the identity of the target person. Below we show the visual results of our pipeline in Fig. 8.10:

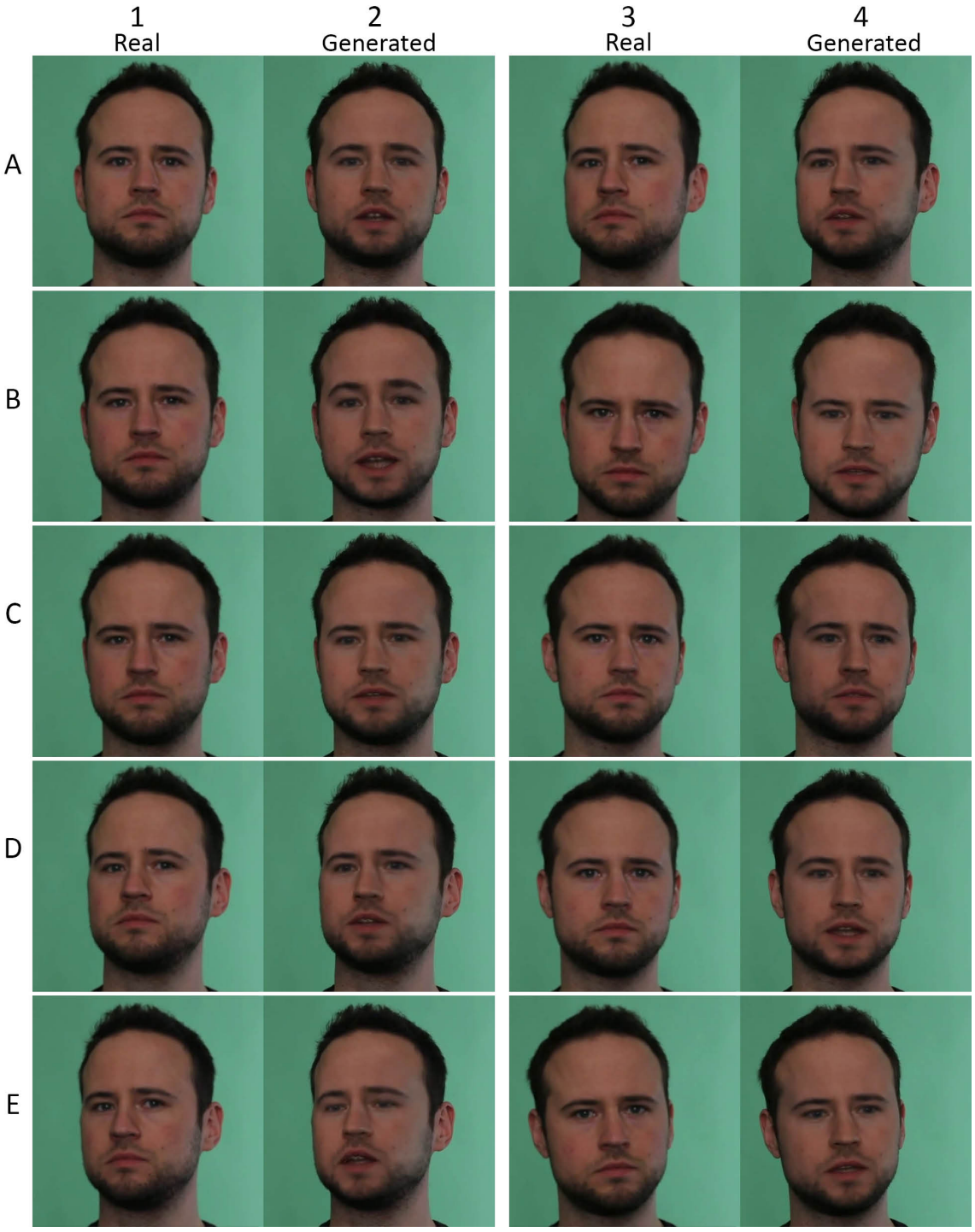


Figure 8.10.: *Results of self-driven reenactment: Columns 1 and 3 are frames from the "silent video" were the person did not speak and only moves and rotates the head. Columns 2 and 4 show generated results. We have taken tracking expression information from the eval dataset and overlays it onto the frames from columns 1 and 3. Note that we have used the head position and rotation from columns 1 and 3. Only the inner part of the face has been replaced by our system in real time by transferring the expression parameters of the eval set. The amount of the face (or face crop), what is rerenderen, is depicted in Fig. 8.6. The generated images have a edge length of 350 pixel. Results in motion:* `https://youtu.be/aXQh6xvscko`.

8.3.2. Interactive Text-to-Speech Driven Avatar

We demonstrate that it is possible to combine our system with a chatbot and a text-to-speech system to realize an interactive virtual assistant. A interactive virtual assistance with a photorealistic face is actually not a goal of this dissertation, but it demonstrates the broader impact of our invention. We use a simple text-to-speech synthesis to let an avatar say given phrases through the described pipeline of this chapter. Current text-to-video synthesis does also works in real time what maintains the interactivity of the system. At the time of development, ChatGPT has also demonstrated its interactive potential and offered an API. Therefore, we have combined ChatGPT with Microsoft Azure's text-to-speech system to test an interactive virtual assistant with a face. It is similar to have a video-conference call with a chatbot. The following figures show the user interface of the system as well as different individuals as avatars. For details in motion see: `https://youtu.be/yZQ5jmdExsE`

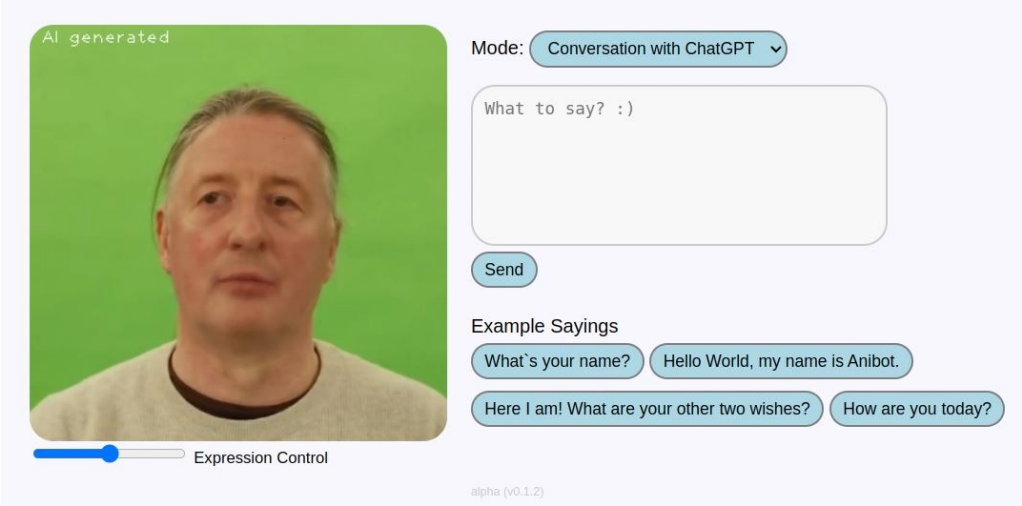


Figure 8.11.: *Broader impact and use cases of our pipeline: With the use of ChatGPTs API and the combination with a Text-to-Speech service, our system is able to provide an interactive virtual assistance with a face. The right side of the image shows the GUI of our experiment in a web browser.*

Figure 8.12.: *As the avatar creation process is almost completely automatic, additional avatars can easily be created and inserted into the GUI.*

Figure 8.13.: *All images are generated by using the silent video as background and driver for head position and rotation. With the Microsoft text-to-speech service, we create audio that we feed into our lip-sync pipeline based on FaceFormer by Fan et al. [Fan+22]. The entire face is reenacted such as Fig. 8.4 shows. The generated images have a edge length of 350 pixel. We can run two avatars on a Nvidia GeForce RTX3090 (desktop version).*

8.3.3. Timings

Our system needs 24 ms to generate a frame of 350 px edge length which leds to 41.6 fps. This was measured on a Lenovo Legion Notebook with AMD Ryzen 7 5800H and Nvidia RTX 3090 (notebook version). The framerate is sufficient to realize a smooth rendering in a video stream of 30 fps. The 24 ms per frame is measured on an average of 99 frames in

a row and are divided as follows: 20 ms for the rasterization of the 3 rainbow images with PyTorch3D, 2.3 ms for running through the DINER+SIREN+FiLM strands and the remaining 1.7 ms are used for combining the images with Kornia, which is GPU-accelerated. We do not use any optimization strategies such as Pytorch's Autocast (formerly known as Nvidia APEX) or tracing the model with a TorchScript JIT compiler.

Please note, that implementing the DINER lookup table significantly accelerates the forward passes for the neural network. Without DINER, we need a neural network with a capacity comparable to that presented in NeRFs [Mil+21] or Neural Head Avatars by Grassal et al. [Gra+22]. For example, Grassal et al. use 8 consecutive fully-connected layers with 256 neurons and an additional last fully-connected layer with 128 neurons for texture generation. In our experiments, a SIREN+FiLM architecture without DINER takes about 10 to 20 times longer for a forward pass. With the help of DINER, we can make our architecture much smaller and therefore faster, as shown in Fig. 8.7. This can be explained by the fact that accessing a lookup table has only a constant time complexity of $\mathcal{O}(1)$ while a neural network has a time complexity depending of the number of layers, weights and biases. Compared to other volume-based INR systems, our image-based INR approach is therefore much faster than other systems that use INRs for face reconstruction, such as NeRFace by Gafni et al. [Gaf+21], that requires minutes for generating a single frame.

8.3.4. Ablation Study

The use of a mapping network in the context of face reconstruction has yielded good results to disentangle input parameters and styles in several related works such as Sitzmann et al. [Sit+20], Karras et al. [KLA18] or Chan et al. [Cha+21]. In our final architecture, we use DINER with two SIREN+FiLM layers of 64 neurons each to speed up the generation process, but we want to report the results of a study in a different architecture that may be relevant to scientific discovery and later justify the decision with the DINER architecture.

The authors of Neural Head Avatars [Gra+22] have chosen an 8-layer 256 neuron SIREN+FiLM architecture with an additional 9th layer of 128 neurons and a small ReLU mapping net. The mapping net receives the pose and expression parameters from FLAME as input. Inspired by this work, we also chose a SIREN+FiLM architecture with a mapping net, but instead of one large net for the entire face, we use three smaller nets. These nets generate the areas around the eyes, mouth, and the rest of the face. Our architecture is 6 layers of 256 neurons each. In contrast to other works such as Sitzmann et al. [Sit+20] or Grassal et al. [Gra+22], we found during hyperparameter tuning that instead of initializing the untrained network with $w_o = 30$, the value $w_o = 40$ leads to better results. The following images show a direct comparison of an 8-layer SIREN with 256 neurons to our approach with 3 separate networks with only 6 layers and 256 neurons each.

Figure 8.14.: *Comparison of a single large SIREN full face network with 8 layers of 256 neurons each (left) versus 3 smaller SIREN+FiLM networks for eyes, mouth, and facial background with 6 layers of 256 neurons each (right). Please zoom in for details. The networks right and left received the same driving parameters. Reconstruction of eyes and mouth is much better with separate networks and FiLM on the right. Instead of* $w_o = 30$ *for initialization, we use* $w_o = 40$*.*

In our final architecture, in combination with DINER, we therefore also used three separate nets for eyes, mouth and background and used $w_o = 40$ for initialization. However, our nets are only 2 layers with 64 neurons each. In the following Sec. 8.4, the tradeoff between a small network with DINER and a large network without DINER is discussed again in more detail.

8.3.5. Bottleneck, Failure Cases and Limitations

In Sec. 8.3.3, we have found that the time required for rasterization is much longer compared to the network passes and the blending of the individual images. It was 20 ms for the rasterization, 2.3 ms for running through all three DINER+SIREN+FiLM strands and 1.7 ms for blending the final images together. The bottleneck in our case is the rasterization. This is due to the fact that we use the PyTorch3D renderer, which is not optimized for speed. The calculations do not take place in the hardware-accelerated rasterization pipeline of the GPU, but are executed as differentiable steps in CUDA. Initial tests have shown that we only need approximately 1 ms to render the 3 rainbow images in a Vulkan environment, which would represent an acceleration by a factor of 20. If we were to remove this bottleneck and use a hardware-accelerated rasterizer, we would achieve a calculation time of only 5 ms per frame, which would correspond to 200 fps.

As described in the ablation study (Sec. 8.3.4), we used a FiLM-based mapping network and three smaller networks in order to improve visual quality. But in our experiments, we saw that DINER seems to introduce a less dynamic texture. Please compare the oral cavities of the results in Fig. 8.13 with the images of the ablation study, that does not use DINER in Fig. 8.14. Both avatars have either a rather dark oral cavity (Fig. 8.13 column 1 and 2) or a rather light one (Fig. 8.13 column 3 and 4).

In the avatar of Fig. 8.10 the teeth were captured in higher quality, however, the problem occurs that a unrealistic morphing of the teeth became visible. This problem is hard to see in still images. Please see the video for more details: `https://youtu.be/aXQh6xvscko`. We do not use a geometric teeth proxy because we assumed that the neural architecture would be able to reproduce the teeth and the oral cavity reliably. However, the rather static texture that DINER seems to introduce makes the teeth look "soft".

As shown at the beginning of this section (8.3), we have seen in previous experiments that a DINER lookup table can significantly reduce the calculation time between factor 10 to 20. However, we also noticed that this makes the texture much less dynamic. It seems that our system with DINER tends to rather learn a static texture instead of reacting dynamically to FLAME expression parameters. This significantly reduces the generation of wrinkles, pits and corresponding shadows. We see similar problems in work such as that of Zielonka et al. [ZBT22a].

Another problem, which rarely occurs, but severely impairs immersion, is the fact that INRs react significantly worse to inputs that were not part of the training dataset compared to GANs. We have not measured the influence of DINER for this effect, but suspect that it has a rather negative influence. GANs show a better graceful degradation than our INR approach with DINER. The following Fig. 8.15 and 8.16 show an error caused by a FLAME parameter combination that the neural network did not see during training. This suggests that the inputs of the mapping network are probably not completely ignored, as the rainbow encoding does not show any anomalies with the parameter combination shown. The problem could be solved by training on a broader dataset with more variations.

Furthermore, it can be seen in the following images that the results are slightly blurred and specular points, for example in the eyes, are displayed less strongly than the reference images (Columns 1 and 3 with title "real" in Fig. 8.10). In related work, this is often realized by tracking the lighting parameters at texture level with a 9-value representation based on spherical harmonics [ZBT22a; Gra+22]. The illumination is usually also feed into the network. For reasons of performance, we have refrained from considering lighting and have

recorded our training images without strong highlights. We used a diffuse illumination in the recording room as well as facial skin powder.

Figure 8.15.: *Limitation: A typical artifact of our system, when FLAME parameters were not part of the training dataset, is image noise. This gaze direction of the avatar face was not previously trained.*

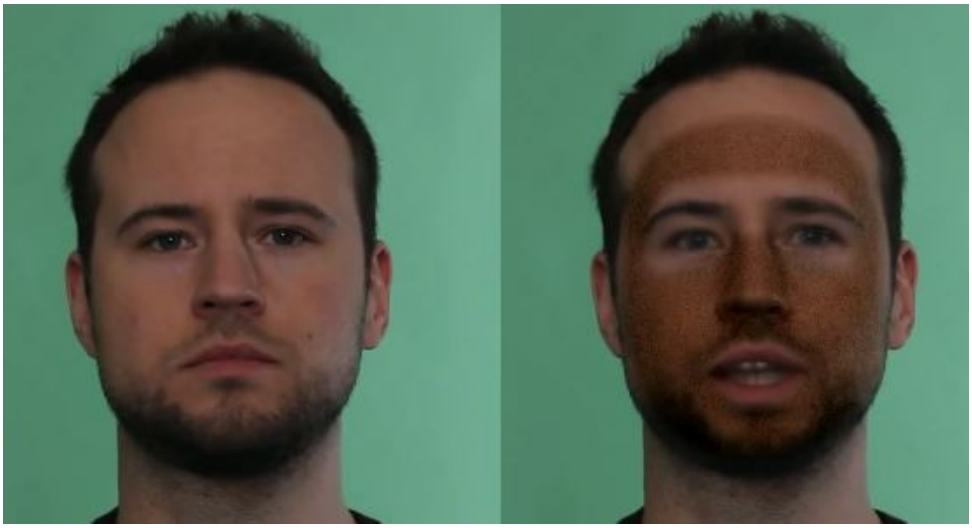


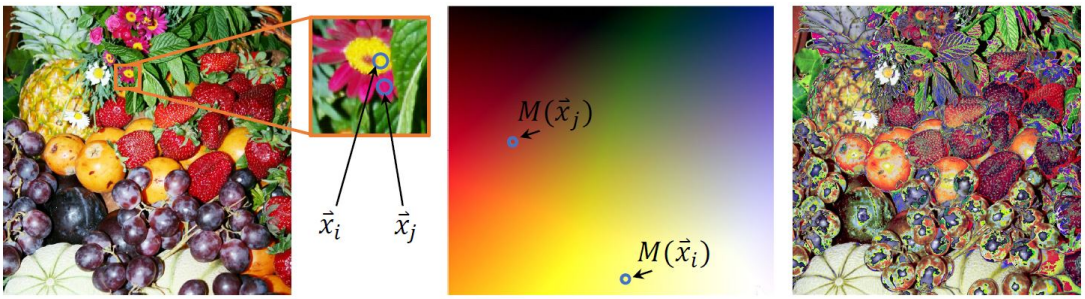
Figure 8.16.: *Limitation: Left is ground truth. Our architecture does not show a graceful degradation if input parameters were not part of the training (right). Network fails, when specific expression parameters are not part of the training data set. On the right, you can see that output of the face-background network is not only noisy, but also changes color.*

8.4. Discussion and Future Work

As already explained, the integration of DINER has significantly accelerated the whole process between factor 10 and 20, but on the one hand we have the problem with a less dynamic texture and also a lower generalization capability towards new input, as shown in Fig. 8.15 and 8.16. This has probably caused by DINER. The use of lookup tables or hash tables seems to be a good inductive bias for ray tracing of static scenes and thus for novel view syntheses, as demonstrated by Instant-NGP [Mül+22]. However, we suspect that a new inductive bias needs to be found for dynamic scenes, such as human faces. Similar to our approach, other systems that rely on INRs for face reconstruction also struggle with low generalization ability. This can be seen in published work such as Neural Head Avatars by Grassal et al. [Gra+22], NeRFace by Gafni et al. [Gaf+21] or INSTA by Zielonka et al. [ZBT22a], where the quality remains high when staying within the range of the training data, but drops off sharply when leaving it. A canonical space that decouples head position and rotation from expression does not always seem to be sufficient. While we did not perform a direct comparison between the GAN and INR approach in this dissertation, GANs seem to degrade more gracefully. Conditional GANs typically add a noise vector z to the network. In the Pix2Pix architecture, the noise is substituted by dropouts and is feed into the network between layers. Current INRs lack such stochastic variances and are therefore probably less able to find a solid generalization. This would represent a possible future improvement and area of research.

Another explanation for the lack of generalization and in particular the often poor interpolation ability between facial expressions in our system (see Fig. 8.15 and 8.16) could be provided by the paper on DINER by Xie et al. [Xie+23] itself. As already mentioned in Sec. 8.1, when combining INRs with lookup tables, the neural networks learn a low-frequency feature distribution and the lookup or hash tables learn a high-frequency position distribution of these features. The following Fig. 8.17 shows that interpolations in the original signal between $\vec{x}_i$ and $\vec{y}_i$ in the learned feature distribution are not possible without errors, at least in the sense of reconstructing the original image in an acceptable way for a human eye. If the positions of the original color values are interpolated in the neural network with $M(\vec{x}_i)$ and $M(\vec{y}_i)$, the result shows unacceptable quality to the human eye. A comparable phenomenon can occur if we interpolate between expression parameters of the FLAME 3DMM instead of between pixels in the output signal as in Fig. 8.17c.

Another option for our approach to further accelerate execution speed would be to use a pure hash table instead of a lookup table with predefined dimensions. A hash table would only allocate the memory that is necessary and would therefore be more memory-optimized than our approach. The work of Müller et al. [Mül+22] shows the advantages of the hash table. In many cases, hash tables are small enough that they fit into the L2 cache of the GPU. Due to the size of our lookup table, for example in our proposed system, it must remain in the VRAM, but the L2 cache is much faster with lower access latencies than the VRAM.



(a) Neighboring pixels in the original space

(b) Neighboring pixels in the mapped space

(c) Interpolated results

Figure 8.17.: *Xie et al. [Xie+23] shows that INRs, especially in combination with a lookup or hash tables, "sort" and store the signal in different way and order than the original. Therefore, an interpolation between two points in the original signal does not necessarily lead to the same intermediate results as if one would interpolate between these two equal points in the neural feature distribution. Interpolating in the neural feature distribution leads to "unpleasant" results for the human eye. Interpolation of face expressions in this space could lead to the same problem and could be the reason, why interpolation fails in many cases with our system. Fig. by Xie et al. [Xie+23].*

8.5. Conclusion

In this chapter, we have explored the application of Implicit Neural Representations (INRs) in real-time face rendering, specifically tailored for facial animation in telepresence applications. Our approach focused on leveraging coordinate-based neural networks to efficiently synthesize high-quality facial animations, aiming to enhance the realism and expressiveness of avatars in virtual environments.

We introduced the "rainbow encoding" to input positional data into the INR system inspired by Doukas et al. [DZS21]. This method allowed us to encode the surface of the face in screen space. This way, we maintain interactive framerates, because we make sure that we only generate the currently visible and necessary pixels in the image space through the neural networks, instead of requiring many queries in the camera space or possibly through many more queries through ray tracing with a NeRF-based volumetric approach.

Our system utilized a combination of the DINER (disorder-invariant INRs) [Xie+23], SIREN (Sinusoidal Representation Networks) [Sit+20], and FiLM (Feature-wise Linear Modulation) [Per+18] to achieve fast and detailed facial rendering. This architecture allowed us to maintain high visual fidelity while significantly reducing computational overhead, making real-time applications feasible.

Our implementation achieved real-time performance, rendering with over 40 frames per second on an Nvidia GeForce 3090 laptop version GPU. This capability was crucial for practical use in telepresence scenarios, where responsiveness and natural interaction are paramount. Additionally, we showcased the potential of our system in broader applications, such as interactive virtual assistants driven by text-to-speech and chatbot technologies.

Despite these advancements, our system faced limitations in generalization and interpolation capabilities. The reliance on specific training data led to artifacts when encountering

new expressions or facial configurations, highlighting a common challenge in INR-based face rendering systems. Future work could explore new inductive biases that enhance generalization capabilities.

9. Conclusion

This final chapter highlights our accomplishments in addressing key challenges in telepresence to convey nonverbal communication (NVC). In total, we have tested more than 200 different neural network architectures and spent more than 10,000 GPU hours training and tuning hyperparameters. We extensively explore deep learning architectures from the areas of Convolutional Neural Networks (CNNs), Generative Adversarial Networks (GANs), fully-connected MLPs, and various Implicit Neural Representations (INRs) with and without input encoding. We have extensively tested and optimized these architectures for the specific use case of face-to-face telepresence applications. In total, more than 15 different hardware prototypes for face capture and face tracking systems were developed. Two user studies were conducted to further substantiate the basis and motivation of this work. The following is a selection of the highlights of our findings and contributions:

1. **Generating Photorealistic Avatar Faces with Real-Time Expressions**
   For decades, computer graphics researchers have been trying to render human faces authentically. When computational time is not a constraint, this can be done with considerable manual effort with 3D modeling, e.g. in Hollywood movies. We have presented two end-to-end differentiable optimization pipelines, based on GANs and INRs, respectively, that have not only proven to render photorealistic faces in real time, but also to be fully automatic with no user intervention by also requiring only a fraction of the creation time that manual 3D modeling would need. In combination with a face-tracking HMD, these novel systems enable conveying NVC through authentic facial avatars within virtual environments. In Chap. 7 and 8 we have thus answered the research question: "How to transfer the face in a photorealistic appearance with authentic movement in real time despite wearing an HMD?".

2. **Face Tracking Head-Mounted Display**
   A major problem with immersive telepresence is that the face is obscured by the HMD. In Chap. 5 we have answered the research question RQ5: "How to track a face beneath an HMD?". Therefore, we developed a face-tracking HMD system that captures facial expressions in real time using deep learning approaches in a combination of our own lower face and eyebrow tracking modules. We merged the tracking data with an off-the-shelf eye-tracking module into 70 facial landmarks. Our system achieves high accuracy and minimal computational load through optimized CNNs, providing a robust solution for real-time facial expression tracking and enabling to drive the neural avatar faces presented in this thesis.

3. **Standardizing Body Tracking**
   There are many different body tracking systems available. Many of the systems use different basic technologies, such as IMU-, RGB-, RGB-D- or IR-sensor solutions. The combination of several systems lends itself to transmitting expressive NVC, but merging the data of these systems is difficult. Also the representation of movements can differ between the systems. In Chap. 4, we answered the 4th research question: "How can different body tracking systems and protocols be standardized to ensure

that the representation of nonverbal communication in a telepresence application looks as identical as possible, even with the use of different tracking systems?". We introduced MotionHub, an open-source platform that integrates tracking data from various body tracking systems into a unified skeleton structure, coordinate space via calibration and a standarized network protocol. We created a game engine plugin for the Unity game engine and demonstrated a simple integration. MotionHub standardizes NVC cues across different systems, adding minimal delay and allowing also seamless switching between tracking systems during runtime.

4. **Literature Review and Design Guidelines**
   In a literature review, we analyzed remote collaboration systems and concluded that the transfer of more information generally leads to better collaboration in terms of task effectiveness and efficiency. However, when designing such systems, researchers face technical limitations such as hardware performance and difficulties in creating believable avatar animations. We have derived requirements from the literature and created six design guidelines that are important when creating remote collaboration software. These guidelines also formed the foundation and motivation for this dissertation and are presented in Chap. 2.

5. **Impact of Shared Virtual Task Spaces on Remote Collaboration**
   Immersive telepresence systems can transmit and display spatial information. This makes it possible to transmit deictic gestures as NVC during telepresence. The 2nd research question was "How does the availability of a shared virtual task space, and in particular a referencing tool, affect task efficiency and error rates in remote collaboration?". Our study in Chap. 3 showed that the availability of a shared task space with a spatial referencing tool significantly improves task efficiency and reduces error rates in remote collaboration. The availability of deictic gestures can save about 30% of task completion time and reduce errors by 90%.

6. **Personalized Face Avatars Increase Social Presence**
   Creating authentic avatars is time-consuming, and rendering them in real time is technically challenging. Previous immersive collaboration applications often use stylized, comic-like avatars. Is there a justification for researching the creation of authentic personalized avatars and their use in telepresence applications? With the third research question, "Does a personalized avatar increase copresence and social presence compared to a non-personalized?", we try to better understand the influence of personalized avatars using questionnaires that measure copresence and social presence. Our study in Chap. 6 found that personalized avatars increased the sense of social presence in MR-based telepresence applications. While copresence did not show a significant difference between the two groups of generic and personalized avatar faces, social presence was significantly higher with personalized avatars, indicating their value in remote collaboration.

Overall, this dissertation has addressed the broad research question (RQ1) "How to technically support the transmission of nonverbal communication in Mixed Reality-based telepresence systems?". It has presented several approaches to overcome these challenges and contribute to the understanding and development of richer, more immersive telepresence applications. We believe that our work has made an important contribution to the creation of such applications, which could potentially serve as a viable substitute for face-to-face interactions in the future, and help save resources and CO2 in the long term by reducing physical travel.

Bibliography

[Ach+17] Jascha Achenbach, Thomas Waltemate, Marc Erich Latoschik, and Mario Botsch. "Fast generation of realistic virtual humans". In: *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*. VRST '17. Gothenburg, Sweden: Association for Computing Machinery, 2017. ISBN: 9781450355483. DOI: 10.1145/3139131.3139154.

[AD65] Michael Argyle and Janet Dean. "Eye-contact, distance and affiliation". In: *Sociometry*. Vol. 28. 3. American Sociological Association, 1965, pp. 289–304.

[AL11] Andreas Aristidou and Joan Lasenby. "FABRIK: A fast, iterative solver for the Inverse Kinematics problem". In: *Graph. Models*. Vol. 73. 5. San Diego, CA, USA: Academic Press Professional, Inc., 2011, pp. 243–260. DOI: 10.1016/j.gmod.2011.05.003.

[Asa+17] Nao Asano, Katsutoshi Masai, Yuta Sugiura, and Maki Sugimoto. "Facial performance capture by embedded photo reflective sensors on a smart eyewear". In: *Proceedings of the 27th International Conference on Artificial Reality and Telexistence and 22nd Eurographics Symposium on Virtual Environments*. ICAT-EGVE '17. Adelaide, Australia: Eurographics Association, 2017, pp. 21–28.

[AKB12] Laura Aymerich-Franch, Cody Karutz, and Jeremy N Bailenson. "Effects of Facial and Voice Similarity on Presence in a Public Speaking Virtual Environment". In: *Proceedings of the International Society for Presence Research Annual Conference*. 2012.

[Azu97] Ronald T. Azuma. "A Survey of Augmented Reality". In: *Presence: Teleoperators and Virtual Environments*. Vol. 6. 4. 1997, pp. 355–385. DOI: 10.1162/pres.1997.6.4.355.

[BVa] XSens - Movella Technologies B.V. *Multi Actor Recording - Best Practices*. Accessed: 24.02.2020. URL: https://base.xsens.com/s/article/Multi-Actor-Recording-Best-Practices.

[BVb] XSens - Movella Technologies B.V. *Position Aiding: HTC Vive*. Accessed: 28.12.2023. URL: https://base.xsens.com/s/article/Position-Aiding-HTC-Vive.

[Bae+20] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. "wav2vec 2.0: a framework for self-supervised learning of speech representations". In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS '20. Vancouver, BC, Canada: Curran Associates Inc., 2020. ISBN: 9781713829546.

[Bag+21] Timur Bagautdinov, Chenglei Wu, Tomas Simon, Fabián Prada, Takaaki Shiratori, Shih-En Wei, Weipeng Xu, Yaser Sheikh, and Jason Saragih. "Driving-signal aware full-body avatars". In: *ACM Trans. Graph.* Vol. 40. 4. New York, NY, USA: Association for Computing Machinery, 2021. DOI: 10.1145/3450626.3459850.

[Bai+03] Jeremy Bailenson, Jim Blascovich, Andrew Beall, and Jack Loomis. "Interpersonal Distance in Immersive Virtual Environments". In: *Personality & social psychology bulletin*. Vol. 29. 2003, pp. 819–33. DOI: 10.1177/0146167203029007002.

[Bai+06] Jeremy N. Bailenson, Nick Yee, Dan Merget, and Ralph Schroeder. "The Effect of Behavioral Realism and Form Realism of Real-Time Avatar Faces on Verbal Disclosure, Nonverbal Disclosure, Emotion Recognition, and Copresence in Dyadic Interaction". In: *Presence: Teleoper. Virtual Environ.* Vol. 15. 4. Cambridge, MA, USA: MIT Press, 2006, pp. 359–372. DOI: 10.1162/pres.15.4.359.

[BB99] K. M. Baird and W. Barfield. "Evaluating the effectiveness of augmented reality displays for a manual assembly task". In: *Virtual Reality*. Vol. 4. 4. Springer-Verlag, 1999, pp. 250–259.

[Bal+16] Marina Ballester Ripoll, Jens Herder, **Ladwig, Philipp**, and Kai Vermeegen. "Comparison of two Gesture Recognition Sensors for Virtual TV Studios". In: *GI-VRAR, Workshop Proceedings / Tagungsband: Virtuelle und Erweiterte Realität - 13. Workshop der GI-Fachgruppe VR/AR,* ed. by Thies Pfeiffer, Julia Fröhlich, and Rolf Kruse. 2016.

[Bar+21] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. "Mip-NeRF: A Multiscale Representation for Anti-Aliasing Neural Radiance Fields". In: *Proceedings of the ICCV*. 2021.

[BR03] Patrick Baudisch and Ruth Rosenholtz. "Halo: a technique for visualizing off-screen objects". In: *Proceedings of CHI*. 5. 2003, pp. 481–488. ISBN: 1581134533.

[Baz+20] Valentin Bazarevsky, Ivan Grishchenko, Karthik Raveendran, Tyler Zhu, Fan Zhang, and Matthias Grundmann. *BlazePose: On-device Real-time Body Pose tracking*. 2020.

[BF17] Stephan Beck and Bernd Froehlich. "Sweeping-based volumetric calibration and registration of multiple RGBD-sensors for 3D capturing systems". In: *IEEE Virtual Reality (VR)*. 2017. DOI: 10.1109/VR.2017.7892244.

[Bec+13] Stephan Beck, Andŕe Kunert, Alexander Kulik, and Bernd Froehlich. "Immersive group-to-group telepresence". In: *IEEE Transactions on Visualization and Computer Graphics*. 2013, pp. 616–625. DOI: 10.1109/TVCG.2013.33.

[BB12] James Bergstra and Yoshua Bengio. "Random Search for Hyper-Parameter Optimization". In: *J. Mach. Learn. Res.* Vol. 13. JMLR.org, 2012, pp. 281–305.

[Ber+22] Guillermo Bernal, Nelson Hidalgo, Conor Russomanno, and Pattie Maes. "Galea: A physiological sensing system for behavioral research in Virtual Environments". In: *IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 2022, pp. 66–76. DOI: 10.1109/VR51125.2022.00024.

[Ber+18] Guillermo Bernal, Tao Yang, Abhinandan Jain, and Pattie Maes. "PhysioHMD: a conformable, modular toolkit for collecting physiological data from head-mounted displays". In: ISWC '18. Singapore, Singapore: Association for Computing Machinery, 2018, pp. 160–167. ISBN: 9781450359672. DOI: 10.1145/3267242.3267268.

[BCL15] Mark Billinghurst, Adrian Clark, and Gun Lee. "A Survey of Augmented Reality". In: *Foundations and Trends in Human-Computer Interaction*. Vol. 8. 2-3. 2015, pp. 73–272. ISBN: 1551-3955.

[BK99] Mark Billinghurst and Hirokazu Kato. "Collaborative Mixed Reality". In: *Mixed Reality*. Springer Berlin Heidelberg, 1999, pp. 261–284.

[BK02a] Mark Billinghurst and Hirokazu Kato. "Collaborative augmented reality". In: *Communications of the ACM*. 2002. ISBN: 9781409285984. DOI: 10.1145/514236.514265.

[BK02b] Mark Billinghurst and Hirokazu Kato. "Collaborative augmented reality". In: *Communications of the ACM*. Vol. 45. 7. 2002. ISBN: 9781409285984. DOI: 10.1145/514236.514265.

[BinVR19] *BinaryVR - Face tracking for Virtual Reality*. https://binaryvr.com. Accessed: 2019-05-24 - No longer online. BinaryVR was acquired by Epic Games. 2019.

[Bio97] Frank Biocca. "The Cyborg's Dilemma: Progressive Embodiment in Virtual Environments". In: *Journal of Computer-Mediated Communication*. 1997. DOI: 10.1111/j.1083-6101.1997.tb00070.x.

[BC02] Frank Biocca and Harms Chad. "Defining and measuring social presence: Contribution to the networked minds theory and measure". In: *Proceedings of PRESENCE*. 2002.

[BHB03] Frank Biocca, Chad Harms, and Judee K. Burgoon. "Toward a More Robust Theory and Measure of Social Presence: Review and Suggested Criteria". In: *Presence: Teleoper. Virtual Environ.* Cambridge, MA, USA: MIT Press, 2003. DOI: 10.1162/105474603322761270.

[Bir52] R.L. Birdwhistell. *Introduction to Kinesics: An Annotation System for Analysis of Body Motion and Gesture*. University of Louisville, 1952.

[BV99] Volker Blanz and Thomas Vetter. "A Morphable Model for the Synthesis of 3D Faces". In: *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH. 1999. ISBN: 0201485605.

[Bla+02] Jim Blascovich, Andrew C. Beall, Jack M Loomis, and Jeremy N. Bailenson. "Equilibrium Theory Revisited: Mutual Gaze and Personal Space in Virtual Environments". In: *Presence: Teleoperators & Virtual Environments*. 2002. DOI: 10.1162/105474601753272844.

[BMF24] BMFVR. *Face and eye tracking on the Quest Pro is NUTS!* https://youtu.be/UiORIYYra1w?si=hHuIWfFQDB5IsFYp&t=317. Accessed: 2024-01-27. 2024.

[Bog+17] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael Black. "Dynamic FAUST: Registering Human Bodies in Motion". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017. DOI: 10.1109/CVPR.2017.591.

[BCJ10] Sébastien Bottecchia, Jean-Marc Cieutat, and Jean-Pierre Jessel. "T.A.C: Augmented Reality System for Collaborative Tele-assistance in the Field of Maintenance Through Internet". In: *Proceedings of the 1st Augmented Human International Conference*. AH '10. ACM, 2010. ISBN: 978-1-60558-825-4.

[Bre] Brekel. *Affordable tools for Motion Capture & Volumetric Video*. Accessed: July 27, 2024.

[BM19] Caio José Dos Santos Brito and Kenny Mitchell. "Recycling a Landmark Dataset for Real-Time Facial Capture and Animation with Low Cost HMD Integrated Cameras". In: *The 17th International Conference on Virtual-Reality Continuum and Its Applications in Industry*. VRCAI '19. Brisbane, QLD, Australia: Association for Computing Machinery, 2019. ISBN: 9781450370028. DOI: 10.1145/3359997.3365690.

[Bru06] Don Brutzman. *Humanoid Animation (H-Anim)*. 2006. URL: https://x3dgraphics.com/slidesets/X3dForAdvancedModeling/HumanoidAnimation.pdf.

[BT17] Adrian Bulat and Georgios Tzimiropoulos. "How far are we from solving the 2D & 3D Face Alignment problem? (and a dataset of 230,000 3D facial landmarks)". In: *International Conference on Computer Vision (ICCV)*. 2017.

[BH87] Judee Burgoon and Jerold Hale. "Validation of measurement of the fundamental themes of relational communication". In: *Communication Monographs*. Vol. 54. 1987, pp. 19–41. DOI: 10.1080/03637758709390214.

[BA83] P. Burt and E. Adelson. "The Laplacian Pyramid as a Compact Image Code". In: *IEEE Transactions on Communications*. Vol. 31. 4. 1983, pp. 532–540. DOI: 10.1109/TCOM.1983.1095851.

[Bux09] Bill Buxton. "Mediaspace – Meaningspace – Meetingspace". In: *Media Space 20 + Years of Mediated Life*. 2009, pp. 217–231.

[Bux92] William A. S. Buxton. "Telepresence: Integrating Shared Task and Person Spaces". In: *Proceedings of the Conference on Graphics Interface (GI'92)*. 1992, pp. 123–129. ISBN: 0-9695338-1-0.

[Cae+19] Carlos Caetano, Jessica Sena, François Brémond, Jefersson A Dos Santos, and William Robson Schwartz. "Skelemotion: A new representation of skeleton joint sequences based on motion information for 3d action recognition". In: *16th IEEE international conference on advanced video and signal based surveillance (AVSS)*. IEEE. 2019, pp. 1–8.

[Can+23] Alberto Cannavò, Filippo Gabriele Pratticò, Alberto Bruno, and Fabrizio Lamberti. "AR-MoCap: Using Augmented Reality to Support Motion Capture Acting". In: *IEEE Conference Virtual Reality and 3D User Interfaces (VR)*. IEEE. 2023, pp. 318–327.

[CD03] Adrian A. Canutescu and Roland L. Jr. Dunbrack. "Cyclic coordinate descent: A robotics algorithm for protein loop closure". In: *Protein Science* 12.5 (2003), pp. 963–972. DOI: 10.1110/ps.0242703.

[Cao+15] Chen Cao, Derek Bradley, Kun Zhou, and Thabo Beeler. "Real-Time High-Fidelity Facial Performance Capture". In: *ACM Trans. Graph.* Vol. 34. 4. New York, NY, USA: Association for Computing Machinery, 2015. DOI: 10.1145/2766943.

[Cao+22] Chen Cao, Tomas Simon, Jin Kyu Kim, Gabe Schwartz, Michael Zollhoefer, Shun-Suke Saito, Stephen Lombardi, Shih-En Wei, Danielle Belko, Shoou-I Yu, Yaser Sheikh, and Jason Saragih. "Authentic volumetric avatars from a phone scan". In: *ACM Trans. Graph.* Vol. 41. 4. New York, NY, USA: Association for Computing Machinery, 2022. DOI: 10.1145/3528223.3530143.

[Cao+18] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. "OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields". In: *arXiv preprint arXiv:1812.08008*. 2018.

[Cao+17] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. "Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[Car69] Hjortsjö Carl-Herman. "Man's face and mimic language". In: *Studen litteratur, Sweden*. 1969.

[Cas+15] Dan Casas, Oleg Alexander, Andrew W. Feng, Graham Fyffe, Ryosuke Ichikari, Paul Debevec, Rhuizhe Wang, Evan Suma, and Ari Shapiro. "Blendshapes from commodity RGB-D sensors". In: *ACM SIGGRAPH 2015 Talks*. SIGGRAPH '15. Los Angeles, California: Association for Computing Machinery, 2015. ISBN: 9781450336369. DOI: 10.1145/2775280.2792540.

[Cas+16] Dan Casas, Andrew Feng, Oleg Alexander, Graham Fyffe, Paul Debevec, Ryosuke Ichikari, Hao Li, Kyle Olszewski, Evan Suma, and Ari Shapiro. "Rapid Photorealistic Blendshape Modeling from RGB-D Sensors". In: *Computer Animation and Social Agents*. CASA '16. Geneva, Switzerland, 2016, pp. 121–129. ISBN: 9781450347457. DOI: 10.1145/2915926.2915936.

[CRV12] Géry Casiez, Nicolas Roussel, and Daniel Vogel. "1 € filter: a simple speed-based low-pass filter for noisy input in interactive systems". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '12. Austin, Texas, USA: Association for Computing Machinery, 2012, pp. 2527–2530. ISBN: 9781450310154. DOI: 10.1145/2207676.2208639.

[Cha+21] Eric R. Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. *pi-GAN: Periodic Implicit Generative Adversarial Networks for 3D-Aware Image Synthesis*. 2021.

[Cho+14] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, 2014, pp. 1724–1734. DOI: 10.3115/v1/D14-1179.

[Cho+22] Vasileios Choutas, Lea Müller, Chun-Hao P. Huang, Siyu Tang, Dimitrios Tzionas, and Michael J. Black. "Accurate 3D Body Shape Regression using Metric and Semantic Attributes". In: *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2022.

[Chu+20] Hang Chu, Shugao Ma, Fernando De la Torre, Sanja Fidler, and Yaser Sheikh. "Expressive Telepresence via Modular Codec Avatars". In: *European Conference on Computer Vision - ECCV*. 2020. ISBN: 978-3-030-58610-2.

[Qt20] The QT Company. *Qt 5*. 2020. URL: https://www.qt.io/.

[Coz+19] Davide Cozzolino, Justus Thies, Andreas Rössler, Christian Riess, Matthias Nießner, and Luisa Verdoliva. *ForensicTransfer: Weakly-supervised Domain Adaptation for Forgery Detection*. 2019.

[CC06] David Cristinacce and Timothy Cootes. "Feature Detection and Tracking with Constrained Local Models". In: *Pattern Recognition*. Vol. 41. 2006, pp. 929–938. DOI: 10.5244/C.20.95.

[Cud+19] Daniel Cudeiro, Timo Bolkart, Cassidy Laidlaw, Anurag Ranjan, and Michael Black. "Capture, Learning, and Synthesis of 3D Speaking Styles". In: *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 10101–10111.

[Dae+16] Jeff Daemen, Jens Herder, Cornelius Koch, **Ladwig, Philipp**, Roman Wiche, and Kai Wilgen. "Semi-Automatic Camera and Switcher Control for Live Broadcast". In: *Proceedings of the ACM International Conference on Interactive Experiences for TV and Online Video*. TVX '16. Chicago, Illinois, USA: Association for Computing Machinery, 2016, pp. 129–134. ISBN: 9781450340670. DOI: 10.1145/2932206.2933559.

[Dae+17] Jeff Daemen, Jens Herder, Cornelius Koch, **Ladwig, Philipp**, Roman Wiche, and Kai Wilgen. "Halbautomatische Steuerung von Kamera und Bildmischer bei Live-Übertragungen". In: *Fachzeitschrift für Fernsehen, Film und Elektronische Medien*. 11. 2017, pp. 501–505.

[DL84] R. L. Daft and R. H. Lengel. "Information Richness: A New Approach to Managerial Behavior and Organization Design". In: *Research in Organizational Behavior*. Vol. 6. 1984, pp. 191–233.

[DT05] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection". In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. 2005, 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177.

[DCL13] E. F. Damasceno, A. Cardoso, and E. A. Lamounier. "An middleware for motion capture devices applied to virtual rehab". In: *IEEE Virtual Reality (VR)*. 2013, pp. 171–172. DOI: 10.1109/VR.2013.6549417.

[DBB22] Radek Danecek, Michael J. Black, and Timo Bolkart. "EMOCA: Emotion Driven Monocular Face Capture and Animation". In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 20311–20322.

[Dar72] Charles Darwin. *The expression of the emotions in man and animals*. John Murray, Albemarle Street, 1872.

[Dat+14] Dragos Datcu, Marina Cidota, Heide Lukosch, and Stephan Lukosch. "On the usability of augmented reality for information exchange in teams from the security domain". In: *Proceedings - 2014 IEEE Joint Intelligence and Security Informatics Conference, JISIC 2014*. IEEE, 2014, pp. 160–167. ISBN: 9781479963645.

[Den+19] Yu Deng, Jiaolong Yang, Sicheng Xu, Dong Chen, Yunde Jia, and Xin Tong. "Accurate 3D Face Reconstruction with Weakly-Supervised Learning: From Single Image to Image Set". In: *IEEE Computer Vision and Pattern Recognition Workshops*. 2019.

[Dew+18] Bastian Dewitz, **Ladwig, Philipp**, Frank Steinicke, and Christian Geiger. "Classification of Beyond-Reality Interaction Techniques in Spatial Human-Computer Interaction". In: *Proceedings of the 2018 ACM Symposium on Spatial User Interaction*. SUI '18. Berlin, Germany: Association for Computing Machinery, 2018, p. 185. ISBN: 9781450357081. DOI: 10.1145/3267782.3274680.

[DZS21] Michail Christos Doukas, Stefanos Zafeiriou, and Viktoriia Sharmanska. "HeadGAN: One-Shot Neural Head Synthesis and Editing". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 14398–14407.

[Eck+16] M. Eckert, I. Gómez-Martinho, J. Meneses, and J. F. M. Ortega. "A modular middleware approach for exergaming". In: *IEEE 6th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*. 2016, pp. 169–173. DOI: 10.1109/ICCE-Berlin.2016.7684747.

[EOT64] Charles Egerton, George J Suci Osgood, and Percy Tannenbaum. "The measurement of meaning". In: *Proceedings of the Seventh International Conference on Motion in Games*. 1964.

[Egg+20] Bernhard Egger, William A. P. Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhoefer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, Christian Theobalt, Volker Blanz, and Thomas Vetter. "3D Morphable Face Models—Past, Present, and Future". In: *ACM Trans. Graph.* Vol. 39. 5. New York, NY, USA: Association for Computing Machinery, 2020. DOI: 10.1145/3395208.

[Eig20] Eigen. *C++ template library for linear algebra*. 2020. URL: http://eigen.tuxfamily.org.

[EF78] P. Ekman and W.V. Friesen. *Facial action coding system: A technique for the measurement of facial movement*. Palo Alto, CA: Consulting Psychologists Press, 1978.

[EF69] Paul Ekman and Wallace V Friesen. "The repertoire of nonverbal behavior: Categories, origins, usage, and coding". In: *semiotica*. Vol. 1. 1. De Gruyter, 1969, pp. 49–98.

[Ele23] Rokoko Electronics. *Smartsuit Pro II - Motion Capture Suit*. May 24, 2023. 2023.

[Elg+20] Mohamed Elgharib, Mohit Mendiratta, Justus Thies, Matthias Nießner, Hans-Peter Seidel, Ayush Tewari, Vladislav Golyanik, and Christian Theobalt. "Egocentric Videoconferencing". In: *ACM Transactions on Graphics*. Vol. 39. 6. ACM, 2020.

[Emt24] Emteq Ltd. *Emteq Labs*. www.emteqlabs.com. Accessed: 2024-01-21. 2024.

[End95] Mica R. Endsley. "Toward a Theory of Situation Awareness in Dynamic Systems". In: *Human Factors: The Journal of the Human Factors and Ergonomics Society*. Vol. 37. 1. 1995, pp. 32–64.

[Our19] Eric Ourcell. *Farecard Machine Maintenance*. License: https://creativecommons.org/licenses/by-nc/2.0/legalcode - Changes: Image are cropped, cap of the man changed to a helmet and text removed from blue, yellow and green label. 2019. URL: https://www.flickr.com/photos/ep_jhu/8294215082.

[Ess00] I. A. Essa. "Ubiquitous sensing for smart and aware environments". In: *IEEE Personal Communications*. Vol. 7. 5. 2000, pp. 47–49. DOI: 10.1109/98.878538.

[Eur06] European Parliament. *EU Directive 2006 25*. https://www.eumonitor.eu/9353000/1/j4nvk6yhcbpeywk_j9vvik7m1c3gyxp/vitgbgijjlzv. Accessed: 2024-01-04. 2006.

[FaceGen24] *FaceGen - 3D Human Faces*. https://facegen.com/. Accessed: 2024-03-12. 2024.

[Fan+22] Yingruo Fan, Zhaojiang Lin, Jun Saito, Wenping Wang, and Taku Komura. "FaceFormer: Speech-Driven 3D Facial Animation with Transformers". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022.

[Fen+14] Andrew Feng, Gale Lucas, Stacy Marsella, Evan Suma, Chung-Cheng Chiu, Dan Casas, and Ari Shapiro. "Acting the Part: The Role of Gesture on Avatar Identity". In: *Proceedings of the Seventh International Conference on Motion in Games*. 2014. ISBN: 9781450326230. DOI: 10.1145/2668064.2668102.

[Fen+21] Yao Feng, Haiwen Feng, Michael J. Black, and Timo Bolkart. "Learning an Animatable Detailed 3D Face Model from In-The-Wild Images". In: *ACM Transactions on Graphics, (Proc. SIGGRAPH)*. Vol. 40. 8. 2021.

[FSK17] Christian Frueh, Avneesh Sud, and Vivek Kwatra. "Headset Removal for Virtual and Mixed Reality". In: *ACM SIGGRAPH 2017 Talks*. SIGGRAPH '17. Los Angeles, California: Association for Computing Machinery, 2017. ISBN: 9781450350082. DOI: 10.1145/3084363.3085083.

[Fuc+94] Henry Fuchs, Gary Bishop, Kevin Arthur, Leonard Mcmillan, Ruzena Bajcsy, Sangwook Lee, Hany Farid, and Takeo Kanade. "Virtual Space Teleconferencing using a Sea of Cameras". In: vol. 2. 1994.

[Gaf+21] Guy Gafni, Justus Thies, Michael Zollhöfer, and Matthias Nießner. "Dynamic Neural Radiance Fields for Monocular 4D Facial Avatar Reconstruction". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 8649–8658.

[Gam+20] Guillaume Gamelin, Amine Chellali, Samia Cheikh, Aylen Ricca, Cedric Dumas, and Samir Otmane. "Point-cloud avatars to improve spatial communication in immersive collaborative virtual environments". In: *Personal Ubiquitous Comput.* Vol. 25. 3. Berlin, Heidelberg: Springer-Verlag, 2020, pp. 467–484. DOI: 10.1007/s00779-020-01431-1.

[Gar03] M. Garau. "The Impact of Avatar Fidelity on Social Interaction in Virtual Environments". In: PhD thesis. 2003.

[Gar+15] P. Garrido, L. Valgaerts, H. Sarmadi, I. Steiner, K. Varanasi, P. Pérez, and C. Theobalt. "VDub: Modifying Face Video of Actors for Plausible Visual Alignment to a Dubbed Audio Track". In: *Computer Graphics Forum*. Vol. 34. 2. 2015, pp. 193–204. DOI: https://doi.org/10.1111/cgf.12552.

[Gar+13] Pablo Garrido, Levi Valgaerts, Chenglei Wu, and Christian Theobalt. "Reconstructing Detailed Dynamic Face Geometry from Monocular Video". In: *ACM Trans. Graph. (Proceedings of SIGGRAPH Asia 2013)*. Vol. 32. 6. 2013, 158:1–158:10. DOI: 10.1145/2508363.2508380.

[GGS23] Chris Geiger, Emil Gerhard, and Mitja Säger. "persona. fractalis II-A trialog between artist, user and algorithm". In: *Proceedings of the 11th International Conference on Digital and Interactive Arts*. 2023, pp. 1–4.

[GSG17] Travis Gesslein, Daniel Scherer, and Jens Grubert. "BodyDigitizer: An Open Source Photogrammetry-based 3D Body Scanner". In: 2017. DOI: 10.48550/arXiv.1710.01370.

[Gmb20] Advanced Realtime Tracking GmbH. *ART Human*. February 24, 2020. 2020.

[Goe+23] Shubham Goel, Georgios Pavlakos, Jathushan Rajasegaran, Angjoo Kanazawa, and Jitendra Malik. "Humans in 4d: Reconstructing and tracking humans with transformers". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 14783–14794.

[Gof63] Erving Goffman. *Behavior in Public Places*. New York: The Free Press, 1963.

[Goo+14] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. *Generative Adversarial Networks*. 2014.

[Gou20] Isaac Gouy. *The Computer Language Benchmarks Game*. 2020. URL: http://benchmarksgame.wildervanck.eu/which-programs-are-fastest.html.

[Gra+22] P. Grassal, M. Prinzler, T. Leistner, C. Rother, M. Niebner, and J. Thies. "Neural Head Avatars from Monocular RGB Videos". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, 2022, pp. 18632–18643. DOI: 10.1109/CVPR52688.2022.01810.

[Gre+22] Daniel Greve, Marcel Tiator, Christian Kreischer, and Christian Geiger. "Personalized Motion Analysis with Consideration of Body Segment Shapes". In: *Proceedings of Mensch und Computer 2022*. 2022, pp. 467–471.

[Gri+20] Ivan Grishchenko, Artsiom Ablavatski, Yury Kartynnik, Karthik Raveendran, and Matthias Grundmann. "Attention mesh: High-fidelity face mesh prediction in real-time". In: *arXiv preprint arXiv:2006.10962*. 2020.

[Gro+10] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. "Multi-PIE". In: *Image and Vision Computing*. Vol. 28. 5. Best of Automatic Face and Gesture Recognition 2008. 2010, pp. 807–813. DOI: https://doi.org/10.1016/j.imavis.2009.08.002.

[Gru+17] Uwe Gruenefeld, Dag Ennenga, Abdallah El Ali, Wilko Heuten, and Susanne Boll. "EyeSee360: designing a visualization technique for out-of-view objects in head-mounted augmented reality". In: *Proceedings of the 5th Symposium on Spatial User Interaction*. SUI '17. Brighton, United Kingdom: Association for Computing Machinery, 2017, pp. 109–118. ISBN: 9781450354868. DOI: 10.1145/3131277.3132175.

[Gru+23] Gustav Grund Pihlgren, Konstantina Nikolaidou, Prakash Chandra Chhipa, Nosheen Abid, Rajkumar Saini, Fredrik Sandin, and Marcus Liwicki. *A Systematic Performance Analysis of Deep Perceptual Loss Networks Breaks Transfer Learning Conventions*. 2023. DOI: 10.48550/arXiv.2302.04032.

[Guo+20] Jianzhu Guo, Xiangyu Zhu, Yang Yang, Fan Yang, Zhen Lei, and Stan Z Li. "Towards fast, accurate and stable 3d dense face alignment". In: *European Conference on Computer Vision*. Springer. 2020, pp. 152–168.

[GLC15] Pavel Gurevich, Joel Lanir, and Benjamin Cohen. "Design and Implementation of TeleAdvisor: a Projection-Based Augmented Reality System for Remote Collaboration". In: *Computer Supported Cooperative Work: CSCW: An International Journal*. Vol. 24. 6. Springer Netherlands, 2015, pp. 527–562. ISBN: 09259724 (ISSN). DOI: 10.1007/s10606-015-9232-7.

[Hal14] Werner Halbritter. *TROS IOS - Nutzung von Herstellerangaben zur Gefährdungsbeurteilung*. https://www.baua.de/DE/Angebote/Veranstaltungen/Dokumentationen/Optische-Strahlung/pdf/InfoTROS-2014-6-Halbritter.pdf. Accessed: 2024-05-26. 2014.

[Hal+68] Edward T Hall, Ray L Birdwhistell, Bernhard Bock, Paul Bohannan, A Richard Diebold Jr, Marshall Durbin, Munro S Edmonson, JL Fischer, Dell Hymes, Solon T Kimball, et al. "Proxemics [and comments and replies]". In: *Current anthropology*. Vol. 9. 2/3. 1968, pp. 83–108.

[HKS14] Juho Hamari, Jonna Koivisto, and Harri Sarsa. "Does Gamification Work? - A Literature Review of Empirical Studies on Gamification". In: *Proceedings of the Annual Hawaii International Conference on System Sciences*. 2014, pp. 3025–3034. ISBN: 9781479925049. DOI: 10.1109/HICSS.2014.377.

[Han+20] Shangchen Han, Beibei Liu, Randi Cabezas, Christopher D. Twigg, Peizhao Zhang, Jeff Petkau, Tsz-Ho Yu, Chun-Jung Tai, Muzaffer Akbay, Zheng Wang, Asaf Nitzan, Gang Dong, Yuting Ye, Lingling Tao, Chengde Wan, and Robert Wang. "MEgATrack: Monochrome Egocentric Articulated Hand-Tracking for Virtual Reality". In: *ACM Trans. Graph.* Vol. 39. 4. New York, NY, USA: Association for Computing Machinery, 2020. DOI: 10.1145/3386569.3392452.

[He+16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[HTS04] Julie Heiser, Barbara Tversky, and Mia Silverman. "Sketches for and from collaboration". In: *Visual and Spatial Reasoning in Design III*. 2004, pp. 69–78.

[HF09] Steven J. Henderson and Steven Feiner. "Evaluating the benefits of augmented reality for task localization in maintenance of an armored personnel carrier turret". In: *Science and Technology Proceedings - IEEE 2009 International Symposium on Mixed and Augmented Reality, ISMAR 2009*. 2009, pp. 135–144. ISBN: 9781424453900. DOI: 10.1109/ISMAR.2009.5336486.

[Her+18a] Jens Herder, **Ladwig, Philipp**, Kai Vermeegen, Dennis Hergert, Florian Busch, Kevin Klever, Sebastian Holthausen, and Bektur Ryskeldiev. "Mixed Reality Experience - How to Use a Virtual (TV) Studio for Demonstration of Virtual Reality Applications". In: *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2018) - GRAPP*. INSTICC. SciTePress, 2018, pp. 281–287. ISBN: 978-989-758-287-5. DOI: 10.5220/0006637502810287.

[Her+18b] Jens Herder, **Ladwig, Philipp**, Kai Vermeegen, Dennis Hergert, Florian Busch, Kevin Klever, Sebastian Holthausen, and Bektur Ryskeldiev. "Mixed Reality Experience - How to Use a Virtual (TV) Studio for Demonstration of Virtual Reality Applications". In: *GRAPP 2018 - 13th International Conference on Computer Graphics Theory and Applications*. 2018, pp. 281–287. DOI: 10.5220/0006637502810287.

[HS97a] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation*. Vol. 9. 8. 1997, pp. 1735–1780.

[HS97b] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation*. Vol. 9. 8. MIT press, 1997, pp. 1735–1780.

[Hod+10] Jessica Hodgins, Sophie Jörg, Carol O'Sullivan, Sang Il Park, and Moshe Mahler. "The saliency of anomalies in animated human characters". In: *ACM Trans. Appl. Percept.* Vol. 7. 4. New York, NY, USA: Association for Computing Machinery, 2010. DOI: 10.1145/1823738.1823740.

[HTC24a] HTC Corporation. *Vive Focus 3 Eye Tracker™*. https://business.vive.com/de/product/vive-focus-3-eye-tracker/. Accessed: 2024-01-03. 2024.

[HTC24b] HTC Corporation. *Vive Focus 3 Facial Tracker™*. https://business.vive.com/de/product/vive-focus-3-facial-tracker/. Accessed: 2024-01-03. 2024.

[HTC24c] HTC Corporation. *Vive XR Elite Full Face Tracker*. https://blog.vive.com/us/introducing-the-new-vive-full-face-tracker-for-vive-xr-elite-developer-and-enterprise-details/. Accessed: 2024-01-10. 2024.

[IKH06] Wijnand A. IJsselsteijn, Yvonne A. W. de Kort, and Antal Haans. "Is This My Hand I See Before Me? The Rubber Hand Illusion in Reality, Virtual Reality, and Mixed Reality". In: *Presence: Teleoper. Virtual Environ.* Cambridge, MA, USA: MIT Press, 2006. DOI: 10.1162/pres.15.4.455.

[IKi20] IKinema. *IKinema Orion*. February 25, 2020. 2020. URL: https://youtu.be/Khoer5DpQkE.

[Inc] 3DiVi Inc. *Nuitrack Full Body Skeletal Tracking Software*. Accessed: 24.02.2020.

[Inc20a] The Khronos Group Inc. *OpenXR*. February 25, 2020. 2020. URL: https://www.khronos.org/openxr/.

[Inc20b] Kitware Inc. *CMake*. 2020. URL: https://cmake.org/.

[Inc20c] Motion Analysis Inc. *Cortex Software*. February 24, 2020. 2020. URL: https://www.motionanalysis.com/software/cortex-software/.

[Inc20d] Reallusion Inc. *iClone Motion LIVE*. 2020. URL: https://mocap.reallusion.com/iclone-motion-live-mocap/default.html.

[Ish90] H. Ishii. "TeamWorkStation: Towards a Seamless Shared Workspace". In: *Proceedings of the 1990 ACM Conference on Computer-supported Cooperative Work*. CSCW '90. Los Angeles, California, USA: ACM, 1990, pp. 13–26. ISBN: 0-89791-402-3. DOI: 10.1145/99332.99337.

[IKG93] Hiroshi Ishii, Minoru Kobayashi, and Jonathan Grudin. "Integration of Interpersonal Space and Shared Workspace: ClearBoard Design and Experiments". In: *ACM Trans. Inf. Syst.* Vol. 11. 4. New York, NY, USA: ACM, 1993, pp. 349–375. DOI: 10.1145/159764.159762.

[IM91] Hiroshi Ishii and Naomi Miyake. "Toward an open shared workspace: computer and video fusion approach of TeamWorkStation". In: *ACM*. Vol. 34. 12. ACM, 1991, pp. 37–50.

[ILC19] Kumi Ishii, Mary Madison Lyons, and Sabrina A Carr. "Revisiting media richness theory for today and future". In: *Human Behavior and Emerging Technologies*. Vol. 1. 2. Wiley Online Library, 2019, pp. 124–131.

[Iso+17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. "Image-to-Image Translation with Conditional Adversarial Networks". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[JS24] Claus Jaeger and Alfons Siedersbeck. *Eye safety of IREDs used in lamp applications*. https://dammedia.osram.info/media/resource/hires/osram-dam-2496541/EyeSafetyofIREDsusedinLampApplications.pdf. Accessed: 2024-05-25. 2024.

[JHH05] Jilin Tu, T. Huang, and Hai Tao. "Face as mouse through visual face tracking". In: *The 2nd Canadian Conference on Computer and Robot Vision (CRV'05)*. 2005, pp. 339–346. DOI: 10.1109/CRV.2005.39.

[JYK20] Younghyun Jo, Sejong Yang, and Seon Joo Kim. "Investigating Loss Functions for Extreme Super-Resolution". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2020, pp. 1705–1712. DOI: 10.1109/CVPRW50498.2020.00220.

[Joh89] Robert Johansen. "User approaches to computer-supported teams". In: *Technological Support for Work Group Collaboration*. USA: L. Erlbaum Associates Inc., 1989, pp. 1–31. ISBN: 0805803041.

[Joh19] Greg Walters John Hany. *Hands-On Generative Adversarial Networks with PyTorch 1.x*. Packt Publishing Ltd., December 2019.

[JAF16a] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. "Perceptual Losses for Real-Time Style Transfer and Super-Resolution". In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling. Cham: Springer International Publishing, 2016, pp. 694–711. ISBN: 978-3-319-46475-6.

[JAF16b] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. "Perceptual Losses for Real-Time Style Transfer and Super-Resolution". In: *ECCV*. 2016.

[Jon18] Jonnedtc on github. *Locating eye centers using means of gradients*. https://github.com/jonnedtc/PupilDetector. Accessed: 2021-12-28. 2018.

[Kai+12] Bernhard Kainz, Stefan Hauswiesner, Gerhard Reitmayr, Markus Steinberger, Raphael Grasset, Lukas Gruber, Eduardo Veas, Denis Kalkofen, Hartmut Seichter, and Dieter Schmalstieg. "OmniKinect: Real-Time Dense Volumetric Data Acquisition and Applications". In: *Proceedings of the 18th ACM Symposium on Virtual Reality Software and Technology*. VRST '12. Toronto, Ontario, Canada: Association for Computing Machinery, 2012, pp. 25–32. ISBN: 9781450314695. DOI: 10.1145/2407336.2407342.

[Kar+17] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. *Progressive Growing of GANs for Improved Quality, Stability, and Variation*. 2017.

[KLA18] Tero Karras, Samuli Laine, and Timo Aila. *A Style-Based Generator Architecture for Generative Adversarial Networks*. 2018.

[Kar+19] Yury Kartynnik, Artsiom Ablavatski, Ivan Grishchenko, and Matthias Grundmann. *Real-time Facial Surface Geometry from Monocular Video on Mobile GPUs*. 2019.

[KB99] Hirokazu Kato and Mark Billinghurst. "Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System". In: *Proceedings of the 2Nd IEEE and ACM International Workshop on Augmented Reality*. IWAR '99. Washington, DC, USA: IEEE Computer Society, 1999. ISBN: 0-7695-0359-4.

[KS14] Vahid Kazemi and Josephine Sullivan. "One millisecond face alignment with an ensemble of regression trees". In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1867–1874. DOI: 10.1109/CVPR.2014.241.

[Ken89] David G. Kendall. "A Survey of the Statistical Theory of Shape". In: *Statistical Science*. Vol. 4. 2. Institute of Mathematical Statistics, 1989, pp. 87–99.

[Ker+23] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. "3D Gaussian Splatting for Real-Time Radiance Field Rendering". In: *ACM Trans. Graph.* Vol. 42. 4. New York, NY, USA: Association for Computing Machinery, 2023. DOI: 10.1145/3592433.

[Kim+16] Kangsoo Kim, Gerd Bruder, Divine Maloney, and Greg Welch. "The Influence of Real Human Personality on Social Presence with a Virtual Human in Augmented Reality". In: *ICAT-EGVE 2016 - International Conference on Artificial Reality and Telexistence and Eurographics Symposium on Virtual Environments*. Ed. by Dirk Reiners, Daisuke Iwai, and Frank Steinicke. The Eurographics Association, 2016. ISBN: 978-3-03868-012-3. DOI: 10.2312/egve.20161443.

[Kin09] Davis E. King. "Dlib-ml: A Machine Learning Toolkit". In: *Journal of Machine Learning Research*. Vol. 10. 2009, pp. 1755–1758.

[KB17] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017.

[KW19] Diederik P. Kingma and Max Welling. "An Introduction to Variational Autoencoders". In: *Foundations and Trends® in Machine Learning*. Vol. 12. 4. Now Publishers, 2019, pp. 307–392. DOI: 10.1561/2200000056.

[KF05] D.S. Kirk and D.S. Fraser. *The effects of remote gesturing on distance instruction*. Lawrence Erlbaum Associates, 2005, pp. 301–310. ISBN: 0805857826.

[Kli+02] Gudrun Klinker, Allen H. Dutoit, Martin Bauer, Johannes Bayer, Vinko Novak, and Dietmar Matzke. "Fata Morgana - A presentation system for product design". In: *Proceedings - International Symposium on Mixed and Augmented Reality, ISMAR 2002*. IEEE Comput. Soc, 2002, pp. 76–85. ISBN: 0769517811.

[KAB20] Muhammed Kocabas, Nikos Athanasiou, and Michael J. Black. "VIBE: Video Inference for Human Body Pose and Shape Estimation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2020)*. Piscataway, NJ: IEEE, 2020, pp. 5252–5262. DOI: 10.1109/CVPR42600.2020.00530.

[Koc04] Ned Kock. "The Psychobiological Model: Towards a New Theory of Computer-Mediated Communication Based on Darwinian Evolution". In: *Organization Science*. Vol. 15. 3. 2004, pp. 327–348. DOI: 10.1287/orsc.1040.0071.

[KLG16] Okan Sadik Köse, **Ladwig, Philipp**, and Christian Geiger. "Fractal2Mesh: From Implicit 3D Fractal Volumes to 3D Polygonal Geometry". In: *GI-VRAR, Workshop Proceedings / Tagungsband: Virtuelle und Erweiterte Realität - 13. Workshop der GI-Fachgruppe VR/AR,* ed. by Thies Pfeiffer, Julia Fröhlich, and Rolf Kruse. 2016.

[KND15] Marek Kowalski, Jacek Naruniec, and Michal Daniluk. "Livescan3D: A Fast and Inexpensive 3D Data Acquisition System for Multiple Kinect v2 Sensors". In: *2015 International Conference on 3D Vision*. 2015, pp. 318–325. DOI: 10.1109/3DV.2015.43.

[KBF22] Adrian Kreskowski, Stephan Beck, and Bernd Froehlich. "Output-Sensitive Avatar Representations for Immersive Telepresence". In: *IEEE Transactions on Visualization and Computer Graphics*. Vol. 28. 7. 2022, pp. 2697–2709. DOI: 10.1109/TVCG.2020.3037360.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger. Vol. 25. Curran Associates, Inc., 2012.

[Kul+11] Philipp Kulms, Nicole C. Krämer, Jonathan Gratch, and Sin-Hwa Kang. "It's in Their Eyes: A Study on Female and Male Virtual Humans' Gaze". In: *Intelligent Virtual Agents*. Ed. by Hannes Högni Vilhjálmsson, Stefan Kopp, Stacy Marsella, and Kristinn R. Thórisson. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 80–92.

[Lad24a] **Ladwig, Philipp**. *Deepfakes: Technische Hintergründe und Trends*. https://www.bpb.de/lernen/bewegtbild-und-politische-bildung/556238/deepfakes-technische-hintergruende-und-trends/. In Dossier: KI, Deepfakes, Soziale Medien. Bundeszentrale für politische Bildung. 2024.

[Lad24b] **Ladwig, Philipp**. *Was ist KI und welche Formen von KI gibt es?* https://www.bpb.de/lernen/bewegtbild-und-politische-bildung/555997/was-ist-ki-und-welche-formen-von-ki-gibt-es/. In Dossier: KI, Deepfakes, Soziale Medien. Bundeszentrale für politische Bildung. 2024.

[Lad+19] **Ladwig, Philipp**, Bastian Dewitz, Hendrik Preu, and Mitja Säger. "Remote Guidance for Machine Maintenance Supported by Physical LEDs and Virtual Reality". In: *Proceedings of Mensch Und Computer 2019*. Ed. by Florian Alt, Andreas Bulling, and Tanja Döring. MuC'19. Hamburg, Germany: Association for Computing Machinery, 2019, pp. 255–262. ISBN: 9781450371988. DOI: 10.1145/3340764.3340780.

[Lad+25] **Ladwig, Philipp**, Rene Ebertowski, Alexander Pech, Ralf Dörner, and Christian Geiger. "Towards a Pipeline for Real-Time Visualization of Faces for VR-based Telepresence and Live Broadcasting Utilizing Neural Rendering". In: *Journal of Virtual Reality and Broadcasting (JVRB)*. Vol. 18 (2024) 2024.1. Section: GI VR/AR 2020. 2025. DOI: 10.48663/1860-2037/18.2024.1.

[Lad+20a] **Ladwig, Philipp**, Kester Evers, Eric J. Jansen, Ben Fischer, David Nowottnik, and Christian Geiger. "MotionHub: Middleware for Unification of Multiple Body Tracking Systems". In: *Proceedings of the 7th International Conference on Movement and Computing*. MOCO '20. Jersey City/Virtual, NJ, USA: Association for Computing Machinery, 2020. ISBN: 9781450375054. DOI: 10.1145/3401956.3404185.

[LF16] **Ladwig, Philipp** and Jannik Fiedler. "Demo: Mesh Modellierung in Virtual Reality". In: *GI-VRAR, Workshop Proceedings / Tagungsband: Virtuelle und Erweiterte Realität - 13. Workshop der GI-Fachgruppe VR/AR,* ed. by Thies Pfeiffer, Julia Fröhlich, and Rolf Kruse. **Best Demo Award**. Aachen: Shaker Verlag, 2016. ISBN: 9783844047189.

[LG19a] **Ladwig, Philipp** and Christian Geiger. "A Literature Review on Collaboration in Mixed Reality". In: *Smart Industry & Smart Education*. Ed. by Michael E. Auer and Reinhard Langmann. Cham: Springer International Publishing, 2019, pp. 591–600. ISBN: 978-3-319-95678-7. DOI: 10.1007/978-3-319-95678-7_65.

[LG19b] **Ladwig, Philipp** and Christian Geiger. "The Effects on Presence of Personalized and Generic Avatar Faces". In: *Virtuelle und Erweiterte Realität - GI VR/AR Workshop*. Ed. by Paul Grimm, Yvonne Jung, Ralf Dörner, and Christian Geiger. Berichte aus der Informatik. Shaker Verlag, 2019. ISBN: 9783844068870.

[LHG17] **Ladwig, Philipp**, Jens Herder, and Christian Geiger. "Towards Precise, Fast and Comfortable Immersive Polygon Mesh Modelling: Capitalising the Results of Past Research and Analysing the Needs of Professionals". In: *ICAT-EGVE 2017 - International Conference on Artificial Reality and Telexistence and Eurographics Symposium on Virtual Environments*. Ed. by Robert W. Lindeman, Gerd Bruder, and Daisuke Iwai. The Eurographics Association, 2017. ISBN: 978-3-03868-038-3. DOI: 10.2312/egve.20171360.

[LL16] **Ladwig, Philipp** and Birgit Lohmann. *Reflexion über die Bedeutung virtueller Welten für den Menschen*. Philosophy & Technology. PHILOTEC.de. 2016.

[Lad+20b] **Ladwig, Philipp**, Alexander Pech, Ralf Dörner, and Christian Geiger. "Unmasking Communication Partners: A Low-Cost AI Solution for Digitally Removing Head-Mounted Displays in VR-Based Telepresence". In: *IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*. 2020, pp. 82–90. DOI: 10.1109/AIVR50618.2020.00025.

[LPG20] **Ladwig, Philipp**, Alexander Pech, and Christian Geiger. "Auf dem Weg zu Face-to-Face-Telepräsenzanwendungen in Virtual Reality mit generativen neuronalen Netzen". In: *17. GI VR / AR Workshop*. Ed. by Benjamin Weyers, Christoph Lürig, and Daniel Zielasko. **Best Paper Award**. Gesellschaft für Informatik e.V., 2020. DOI: 10.18420/vrar2020_15.

[Lad+21] **Ladwig, Philipp**, Damian Zohlen, Manuel Zohlen, and Christian Geiger. "Towards 3D Scanning with Multiple RGB-D Sensors in Virtual Reality". In: *18. GI VR / AR Workshop*. Ed. by Martin Weier, Matthias Bues, and Reto Wechner. Gesellschaft für Informatik e.V., 2021. DOI: 10.18420/vrar2021_15.

[Lan+13] Joel Lanir, Ran Stone, Benjamin Cohen, and Pavel Gurevich. "Ownership and control of point of view in remote assistance". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*. ACM Press, 2013. ISBN: 9781450318990. DOI: 10.1145/2470654.2481309.

[Lat+17a] Marc Erich Latoschik, Daniel Roth, Dominik Gall, Jascha Achenbach, Thomas Waltemate, and Mario Botsch. "The effect of avatar realism in immersive social virtual realities". In: *Proceedings of the 23rd ACM symposium on virtual reality software and technology*. 2017, pp. 1–10.

[Lat+17b] Marc Erich Latoschik, Daniel Roth, Dominik Gall, Jascha Achenbach, Thomas Waltemate, and Mario Botsch. "The effect of avatar realism in immersive social virtual realities". In: *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*. VRST '17. Gothenburg, Sweden: Association for Computing Machinery, 2017. ISBN: 9781450355483. DOI: 10.1145/3139131.3139156.

[LeC+89] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. "Backpropagation Applied to Handwritten Zip Code Recognition". In: *Neural Computation*. Vol. 1. 4. 1989, pp. 541–551. DOI: 10.1162/neco.1989.1.4.541.

[Lee06] Kwan Min Lee. "Presence, Explicated". In: *Communication Theory*. Vol. 14. 1. 2006, pp. 27–50. DOI: 10.1111/j.1468-2885.2004.tb00302.x.

[Li+15a] Hao Li, Laura Trutoiu, Kyle Olszewski, Lingyu Wei, Tristan Trutna, Pei-Lun Hsieh, Aaron Nicholls, and Chongyang Ma. "Facial Performance Sensing Head-Mounted Display". In: *ACM Trans. Graph.* Vol. 34. 4. New York, NY, USA: Association for Computing Machinery, 2015.

[Li+15b] Hao Li, Laura Trutoiu, Kyle Olszewski, Lingyu Wei, Tristan Trutna, Pei-Lun Hsieh, Aaron Nicholls, and Chongyang Ma. "Facial performance sensing head-mounted display". In: *ACM Transactions on Graphics*. 2015. ISBN: 9781450333313. DOI: 10.1145/2766939.

[Li+20] Ruilong Li, Karl Bladin, Yajie Zhao, Chinmay Chinara, Owen Ingraham, Pengda Xiang, Xinglei Ren, Pratusha Prasad, Bipin Kishore, Jun Xing, and Hao Li. *Learning Formation of Physically-Based Face Attributes*. 2020.

[Li+17] Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. "Learning a model of facial shape and expression from 4D scans". In: *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*. Vol. 36. 6. 2017, 194:1–194:17.

[LYJ09] Zhenbo Li, Jun Yue, and David Antonio Gomez Jauregui. "A new virtual reality environment used for e-Learning". In: *IEEE International Symposium on IT in Medicine & Education*. IEEE, 2009, pp. 445–449. ISBN: 978-1-4244-3928-7.

[Lom+18] Stephen Lombardi, Jason Saragih, Tomas Simon, and Yaser Sheikh. "Deep Appearance Models for Face Rendering". In: *ACM Trans. Graph.* Vol. 37. 4. New York, NY, USA: Association for Computing Machinery, 2018. DOI: 10.1145/3197517.3201401.

[Lom+21] Stephen Lombardi, Tomas Simon, Gabriel Schwartz, Michael Zollhoefer, Yaser Sheikh, and Jason Saragih. "Mixture of volumetric primitives for efficient neural rendering". In: *ACM Trans. Graph.* Vol. 40. 4. New York, NY, USA: Association for Computing Machinery, 2021. DOI: 10.1145/3450626.3459863.

[Lub+16] Paul Lubos, Gerd Bruder, Oscar Ariza, and Frank Steinicke. "Touching the Sphere: Leveraging Joint-Centered Kinespheres for Spatial User Interaction". In: *Proceedings of the 2016 Symposium on Spatial User Interaction*. SUI '16. Tokyo, Japan: ACM, 2016, pp. 13–22. ISBN: 978-1-4503-4068-7.

[Lud20] Joe Ludwig. *OpenVR*. February 25, 2020. 2020. URL: https://github.com/ValveSoftware/openvr.

[Lug+19] Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Yong, Juhyun Lee, Wan-Teh Chang, Wei Hua, Manfred Georg, and Matthias Grundmann. "MediaPipe: A Framework for Perceiving and Processing Reality". In: *Third Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR) 2019*. 2019.

[LLL15] J. Lugrin, J. Latt, and M. E. Latoschik. "Avatar anthropomorphism and illusion of body ownership in VR". In: *IEEE Virtual Reality (VR)*. 2015, pp. 229–230. DOI: 10.1109/VR.2015.7223379.

[Luk+15a] Stephan Lukosch, Mark Billinghurst, Leila Alem, and Kiyoshi Kiyokawa. "Collaboration in Augmented Reality". In: *Computer Supported Cooperative Work: CSCW: An International Journal*. Vol. 24. 6. 2015, pp. 515–525.

[Luk+15b] Stephan Lukosch, Heide Lukosch, Dragoş Datcu, and Marina Cidota. "Providing Information on the Spot: Using Augmented Reality for Situational Awareness in the Security Domain". In: *Computer Supported Cooperative Work (CSCW)*. Vol. 24. 6. Springer Netherlands, 2015, pp. 613–664.

[Lv+17] Jiangjing Lv, Xiaohu Shao, Junliang Xing, Cheng Cheng, and Xi Zhou. "A Deep Regression Architecture with Two-Stage Re-initialization for High Performance Facial Landmark Detection". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 3691–3700. DOI: 10.1109/CVPR.2017.393.

[LHT17] Michael J. Lyons, Michael Hähnel, and Nobuji Tetsutani. "Designing, Playing, and Performing with a Vision-Based Mouth Interface". In: *A NIME Reader: Fifteen Years of New Interfaces for Musical Expression*. Ed. by Alexander Refsum Jensenius and Michael J. Lyons. Cham: Springer International Publishing, 2017, pp. 107–124. ISBN: 978-3-319-47214-0. DOI: 10.1007/978-3-319-47214-0_8.

[MC16] Karl F. MacDorman and Debaleena Chattopadhyay. "Reducing consistency in human realism increases the uncanny valley effect; increasing category uncertainty does not". In: *Cognition*. Vol. 146. 2016, pp. 190–205. DOI: https://doi.org/10.1016/j.cognition.2015.09.019.

[MF11] A. Maimone and H. Fuchs. "Encumbrance-free telepresence system with real-time 3D capture and display using commodity depth cameras". In: *10th IEEE International Symposium on Mixed and Augmented Reality*. 2011, pp. 137–146.

[Mai+13] A. Maimone, X. Yang, N. Dierk, A. State, M. Dou, and H. Fuchs. "General-purpose telepresence with head-worn optical see-through displays and projector-based lighting". In: *IEEE Virtual Reality (VR)*. 2013, pp. 23–26.

[Mal87] Vera Maletic. *The Development of Rudolf Laban's Movement and Dance Concepts*. Berlin, Boston: De Gruyter Mouton, 1987. ISBN: 9783110861839. DOI: doi:10.1515/9783110861839.

[Mao+17] X. Mao, Q. Li, H. Xie, R. K. Lau, Z. Wang, and S. Smolley. "Least Squares Generative Adversarial Networks". In: *IEEE International Conference on Computer Vision (ICCV)*. Los Alamitos, CA, USA, 2017. DOI: 10.1109/ICCV.2017.304.

[MFJ16] G. E. Marai, A. G. Forbes, and A. Johnson. "Interdisciplinary immersive analytics at the electronic visualization laboratory: Lessons learned and upcoming challenges". In: *Immersive Analytics (IA), 2016 Workshop on*. IEEE, 2016, pp. 54–59.

[Mar+21] Julien N. P. Martel, David B. Lindell, Connor Z. Lin, Eric R. Chan, Marco Monteiro, and Gordon Wetzstein. "ACORN: Adaptive coordinate networks for neural scene representation". In: *ACM Trans. Graph. (SIGGRAPH)*. Vol. 40. 4. 2021.

[Mas+16a] Katsutoshi Masai, Kai Kunze, Maki Sugimoto, and Mark Billinghurst. "Empathy Glasses". In: *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems - CHI EA '16*. New York, New York, USA: ACM Press, 2016, pp. 1257–1263. ISBN: 9781450340823.

[Mas+16b] Katsutoshi Masai, Kai Kunze, Maki Sugimoto, and Mark Billinghurst. "Empathy Glasses". In: *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems - CHI EA '16*. New York, New York, USA: ACM Press, 2016, pp. 1257–1263. ISBN: 9781450340823. DOI: 10.1145/2851581.2892370.

[McD+08] Rachel McDonnell, Sophie Jörg, Joanna McHugh, Fiona Newell, and Carol O'Sullivan. "Evaluating the Emotional Content of Human Motions on Real and Virtual Characters". In: *Proceedings of the 5th Symposium on Applied Perception in Graphics and Visualization*. APGV '08. Los Angeles, California: ACM, 2008, pp. 67–74. ISBN: 978-1-59593-981-4. DOI: 10.1145/1394281.1394294.

[Meh72] Albert Mehrabian. *Nonverbal Communication*. Chicago: Aldine-Atherton, 1972.

[MF67] Albert Mehrabian and Susan R. Ferris. "Inference of Attitudes from Nonverbal Communication in Two Channels". In: *Journal of Consulting Psychology*. 1967, pp. 248–252. DOI: 10.1037/h0024648.

[MW67] Albert Mehrabian and Morton Wiener. "Decoding of Inconsistent Communications". In: *Journal of Personality and Social Psychology*. 1967, pp. 109–114. DOI: 10.1037/h0024532.

[MP04] Roberto Merletti and Philip Parker. *Electromyography: Physiology, Engineering, and Non-Invasive Applications*. John Wiley & Sons, 2004. ISBN: 978-0-471-67580-8.

[Mes+19] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. "Occupancy Networks: Learning 3D Reconstruction in Function Space". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 4455–4465. DOI: 10.1109/CVPR.2019.00459.

[Met22] Meta. *Meta Horizon Worlds*. https://www.oculus.com/horizon-worlds/?locale=de_DE. Accessed: 2022-03-04. 2022.

[Mic22] Microsoft Corporation. *AltspaceVR*. https://altvr.com/. Accessed: 2022-03-04. 2022.

[Mic20a] Microsoft Inc. *Azure Kinect Body Tracking SDK*. https://docs.microsoft.com/en-us/azure/kinect-dk/body-sdk-download accessed at July 28, 2020. 2020.

[Mic20b] Microsoft Inc. *Azure Kinect Sensor SDK*. https://github.com/microsoft/Azure-Kinect-Sensor-SDK accessed at July 28, 2020. 2020.

[Mid20] MiddleVR. *MiddleVR SDK*. 2020. URL: https://www.middlevr.com/middlevr-sdk/.

[Mil+21] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis". In: *Commun. ACM*. Vol. 65. 1. New York, NY, USA: Association for Computing Machinery, 2021, pp. 99–106. DOI: 10.1145/3503250.

[MK94] Paul Milgram and Fumio Kishino. "A Taxonomy of Mixed Reality Visual Displays". In: *IEICE Transactions on Information Systems*. 12. 1994.

[Min80] Marvin Minsky. "Telepresence". In: *OMNI*. 1980, pp. 44–52.

[MO14] Mehdi Mirza and Simon Osindero. *Conditional Generative Adversarial Nets*. 2014.

[Mix24] Mixed.de. *Meta Quest 3: Eye-Tracking-Zubehör laut Metas CTO "schwierig"*. https://mixed.de/meta-quest-eye-tracking-zubehoer/. Accessed: 2024-01-03. 2024.

[MHK06] Thomas B. Moeslund, Adrian Hilton, and Volker Krüger. "A survey of advances in vision-based human motion capture and analysis". In: *Computer Vision and Image Understanding*. Vol. 104. 2. Special Issue on Modeling People: Vision-based understanding of a person's shape, appearance, movement and behaviour. 2006, pp. 90–126. DOI: https://doi.org/10.1016/j.cviu.2006.08.002.

[MMB08] Teresa Monahan, Gavin McArdle, and Michela Bertolotto. "Virtual reality for collaborative e-learning". In: *Computers and Education*. Vol. 50. 4. Pergamon, 2008, pp. 1339–1353. ISBN: 0360-1315.

[MMK12] M. Mori, K. F. MacDorman, and N. Kageki. "The Uncanny Valley [From the Field]". In: *IEEE Robotics Automation Magazine*. 2012, pp. 98–100. DOI: 10.1109/MRA.2012.2192811.

[MRR17] Jens Müller, Roman Rädle, and Harald Reiterer. "Remote Collaboration With Mixed Reality Displays". In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*. ACM Press, 2017, pp. 6481–6486. ISBN: 9781450346559.

[Mül+22] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. "Instant Neural Graphics Primitives with a Multiresolution Hash Encoding". In: *ACM Trans. Graph.* Vol. 41. 4. New York, NY, USA: ACM, 2022, 102:1–102:15. DOI: 10.1145/3528223.3530127.

[MBM16] Matteo Munaro, Filippo Basso, and Emanuele Menegatti. "OpenPTrack". In: *Robot. Auton. Syst.* Vol. 75. PB. NLD: North-Holland Publishing Co., 2016, pp. 525–538. DOI: 10.1016/j.robot.2015.10.004.

[Nag+17] Koki Nagano, Jaewoo Seo, Jens Fursund, Iman Sadeghi, Carrie Sun, Yen-chun Chen, Liwen Hu, Shunsuke Saito, Lingyu Wei, Jaewoo Seo, Jens Fursund, Iman Sadeghi, Carrie Sun, Yen-Chun Chen, and Hao Li. "Avatar Digitization From a Single Image For Real-Time Rendering". In: *ACM Transactions on Graphics*. 2017. DOI: 10.1145/3130800.3130887.

[Nam+16] Seung-Woo Nam, Kyung-Ho Jang, Yun-Ji Ban, Hye-Sun Kim, and Sung-Il Chien. "Hole-Filling Methods Using Depth and Color Information for Generating Multiview Images". In: *ETRI Journal*. Vol. 38. 5. 2016, pp. 996–1007. DOI: https://doi.org/10.4218/etrij.16.0116.0062.

[NR96] Clifford Nass and Byron Reeves. *The Media Equation: How People Treat Computers, Television, and New Media Like Real People and Places*. Cambridge, UK: Cambridge University Press, 1996.

[Nat20] Natural Point Inc. *OptiTrack Motive:Body*. February 24, 2020. 2020.

[Nat23] Natural Point Inc. Accessed: 28.12.2023. 2023.

[NW22] Hadar Nesher Shoshan and Wilken Wehrt. "Understanding "Zoom fatigue": A mixed-method approach". In: *Applied Psychology*. Vol. 71. 3. 2022, pp. 827–852. DOI: https://doi.org/10.1111/apps.12360.

[New+11] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon. "KinectFusion: Real-time dense surface mapping and tracking". In: *10th IEEE International Symposium on Mixed and Augmented Reality*. 2011, pp. 127–136.

[NYD16] Alejandro Newell, Kaiyu Yang, and Jia Deng. "Stacked Hourglass Networks for Human Pose Estimation". In: *Computer Vision – ECCV 2016*. Ed. by Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling. Cham: Springer International Publishing, 2016, pp. 483–499. ISBN: 978-3-319-46484-8.

[New+04] J. Newman, M. Wagner, M. Bauer, A. MacWilliams, T. Pintaric, D. Beyer, D. Pustka, F. Strasser, D. Schmalstieg, and G. Klinker. "Ubiquitous tracking for augmented reality". In: *Third IEEE and ACM International Symposium on Mixed and Augmented Reality*. 2004, pp. 192–201. DOI: 10.1109/ISMAR.2004.62.

[Noi20] Noitom. *Perception Neuron Motion Capture*. February 24, 2020. 2020.

[NB03] Kristine L. Nowak and Frank Biocca. "The Effect of the Agency and Anthropomorphism on users' Sense of Telepresence, Copresence, and Social Presence in Virtual Environments". In: *Presence: Teleoperators and Virtual Environments*. 2003. DOI: 10.1162/105474603322761289.

[NC90] J.F. Nunamaker and M. Chen. "Systems development in information systems research". In: *Twenty-Third Annual Hawaii International Conference on System Sciences*. Vol. 3. 1990, 631–640 vol.3. DOI: 10.1109/HICSS.1990.205401.

[NVI19] NVIDIA. *NVIDIA CUDA® Deep Neural Network library*. 2019. URL: https://developer.nvidia.com/cudnn.

[NVI24] NVIDIA. *APEX: A PyTorch Extension: Tools for easy mixed precision and distributed training in Pytorch*. https://github.com/NVIDIA/apex. Accessed: 2024-01-10. 2024.

[Nvi] Nvidia. *Nvidia Holodeck*. https://www.nvidia.com/en-us/design-visualization/technologies/holodeck/ Accessed on 2018-01-25.

[Occ20] Inc. Occipital. *OpenNI 2 SDK*. 2020. URL: https://structure.io/openni.

[Oda+15] Ohan Oda, Carmine Elvezio, Mengu Sukan, Steven Feiner, and Barbara Tversky. "Virtual Replicas for Remote Assistance in Virtual and Augmented Reality". In: *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology - UIST '15*. ACM Press, 2015, pp. 405–415. ISBN: 9781450337793.

[OBW18a] Catherine Oh, Jeremy Bailenson, and Gregory Welch. "A Systematic Review of Social Presence: Definition, Antecedents, and Implications". In: *Frontiers in Robotics and AI*. Vol. 5. 2018. DOI: 10.3389/frobt.2018.00114.

[OBW18b] Catherine S. Oh, Jeremy N. Bailenson, and Gregory F. Welch. "A Systematic Review of Social Presence: Definition, Antecedents, and Implications". In: *Frontiers in Robotics and AI*. 2018. DOI: 10.3389/frobt.2018.00114.

[okr16] Oliver Kreylos (aka okreylos). http://doc-ok.org/?p=1478. 2016.

[OO00] Gary M. Olson and Judith S. Olson. "Distance Matters". In: *Human–Computer Interaction*. Vol. 15. 2-3. L. Erlbaum Associates Inc., 2000, pp. 139–178. DOI: 10.1207/S15327051HCI1523_4.

[OO14] Judith S. Olson and Gary M. Olson. "How to make distance work work". In: ACM, 2014.

[Ols+16] Kyle Olszewski, Joseph J. Lim, Shunsuke Saito, and Hao Li. "High-Fidelity Facial and Speech Animation for VR HMDs". In: *ACM Trans. Graph.* Vol. 35. 6. New York, NY, USA: Association for Computing Machinery, 2016. DOI: 10.1145/2980179.2980252.

[Ort+16] Sergio Orts-Escolano, Christoph Rhemann, Sean Fanello, Wayne Chang, Adarsh Kowdle, Yury Degtyarev, David Kim, Philip L. Davidson, Sameh Khamis, Mingsong Dou, Vladimir Tankovich, Charles Loop, Qin Cai, Philip A. Chou, Sarah Mennicken, Julien Valentin, Vivek Pradeep, Shenlong Wang, Sing Bing Kang, Pushmeet Kohli, Yuliya Lutchyn, Cem Keskin, and Shahram Izadi. "Holoportation: Virtual 3D Teleportation in Real-time". In: *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. UIST '16. Tokyo, Japan: ACM, 2016, pp. 741–754. ISBN: 978-1-4503-4189-9. DOI: 10.1145/2984511.2984517.

[OSG24] Robert Osfield. *OpenSceneGraph*. https://openscenegraph.github.io/openscenegraph.io/. Accessed: 2024-04-04. 2024.

[Pal+18] Jussi Palomäki, Anton Kunnari, Marianna Drosinou, Mika Koverola, Noora Lehtonen, Juho Halonen, Marko Repo, and Michael Laakasuo. "Evaluating the replicability of the uncanny valley effect". In: *Heliyon*. Vol. 4. 11. 2018, e00939. DOI: https://doi.org/10.1016/j.heliyon.2018.e00939.

[Par+19] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. "DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 165–174. DOI: 10.1109/CVPR.2019.00025.

[PSP93] R. Pausch, M. A. Shackelford, and D. Proffitt. "A user study comparing head-mounted and stationary displays". In: *Proceedings of 1993 IEEE Research Properties in Virtual Reality Symposium*. 1993, pp. 41–45. DOI: 10.1109/VRAIS.1993.378265.

[Paw22] Maria Pawelec. "Deepfakes and Democracy (Theory): How Synthetic Audio-Visual Media for Disinformation and Hate Speech Threaten Core Democratic Functions". In: *Digital Society*. Vol. 1. 2. 8, 2022, p. 19. DOI: 10.1007/s44206-022-00010-6.

[Pay+09] *A 3D Face Model for Pose and Illumination Invariant Face Recognition*. IEEE. Genova, Italy, 2009.

[Pej+16a] Tomislav Pejsa, Julian Kantor, Hrvoje Benko, Eyal Ofek, and Andrew Wilson. "Room2Room: Enabling Life-Size Telepresence in a Projected Augmented Reality Environment". In: *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*. CSCW '16. San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 1716–1725. ISBN: 9781450335928. DOI: 10.1145/2818048.2819965.

[Pej+16b] Tomislav Pejsa, Julian Kantor, Hrvoje Benko, Eyal Ofek, and Andrew D Wilson. "Room2Room: Enabling Life-Size Telepresence in a Projected Augmented Reality Environment". In: *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing - CSCW '16*. ACM Press, 2016, pp. 1714–1723. ISBN: 9781450335928.

[Per+18] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron C. Courville. "FiLM: Visual Reasoning with a General Conditioning Layer". In: *AAAI*. 2018.

[PKL12] Kalle A. Piirainen, Gwendolyn L. Kolfschoten, and Stephan Lukosch. "The Joint Struggle of Complex Engineering: A Study of the Challenges of Collaborative Design". In: *International Journal of Information Technology & Decision Making*. Vol. 11. 6. 2012, pp. 1087–1125.

[Pir+14] Ivelina Piryankova, Jeanine Stefanucci, Javier Romero, Stephan de la Rosa, Michael Black, and Betty Mohler. "Can I Recognize My Body's Weight? The Influence of Shape and Texture on the Perception of Self". In: *ACM Transactions on Applied Perception*. 2014. DOI: 10.1145/2641568.

[Piu+17a] Thammathip Piumsomboon, Arindam Day, Barrett Ens, Youngho Lee, Gun Lee, and Mark Billinghurst. "Exploring Enhancements for Remote Mixed Reality Collaboration". In: *SIGGRAPH Asia 2017 Mobile Graphics & Interactive Applications*. SA '17. Bangkok, Thailand: ACM, 2017, 16:1–16:5. ISBN: 978-1-4503-5410-3. DOI: 10.1145/3132787.3139200.

[Piu+17b] Thammathip Piumsomboon, Youngho Lee, Gun Lee, and Mark Billinghurst. "CoVAR: a collaborative virtual and augmented reality system for remote collaboration". In: *SIGGRAPH Asia 2017 Emerging Technologies on - SA '17*. ACM Press, 2017, pp. 1–2. ISBN: 9781450354042.

[PMD23] PMD PicoFlexx2. https://3d.pmdtec.com/en/3d-cameras/flexx2/. 2023.

[Poe+12] Ronald Poelman, Oytun Akman, Stephan Lukosch, and Pieter Jonker. "As if being there". In: *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work - CSCW '12*. 5. 2012, p. 1267. ISBN: 9781450310864. DOI: 10.1145/2145204.2145394.

[Pol24] Polhemus. *Polhemus Electro-magnetic Tracking*. https://polhemus.com/motion-tracking/applications/. Accessed: 2024-05-09. 2024.

[Pop07] Ronald Poppe. "Vision-based human motion analysis: An overview". In: *Computer Vision and Image Understanding*. Vol. 108. 2007, pp. 4–18. DOI: 10.1016/j.cviu.2006.10.016.

[Qia+23] Shenhan Qian, Tobias Kirschstein, Liam Schoneveld, Davide Davoli, Simon Giebenhain, and Matthias Nießner. "GaussianAvatars: Photorealistic Head Avatars with Rigged 3D Gaussians". In: *arXiv preprint arXiv:2312.02069*. 2023.

[Qua24] Qualisys AB. *Qualisys Track Manager (QTM)*. https://www.qualisys.com/software/qualisys-track-manager/. Accessed: 2024-05-04. 2024.

[Raa+79] Frederick H. Raab, Ernest B. Blood, Terry O. Steiner, and Herbert R. Jones. "Magnetic Position and Orientation Tracking System". In: *IEEE Transactions on Aerospace and Electronic Systems*. Vol. AES-15. 5. 1979, pp. 709–718. DOI: 10.1109/TAES.1979.308860.

[RMC16] Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks". In: *Submitted as conference paper at ICLR*. 2016.

[Rah+19] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. "On the spectral bias of neural networks". In: *International conference on machine learning*. PMLR. 2019, pp. 5301–5310.

[Raj+21] Amit Raj, Michael Zollhofer, Tomas Simon, Jason Saragih, Shunsuke Saito, James Hays, and Stephen Lombardi. "Pixel-Aligned Volumetric Avatars". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021.

[RS05] Gerhard Reitmayr and Dieter Schmalstieg. "OpenTracker: A Flexible Software Design for Three-Dimensional Interaction". In: *Virtual Real.* Vol. 9. 1. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 79–92.

[RKF23] Gareth Rendle, Adrian Kreskowski, and Bernd Froehlich. "Volumetric Avatar Reconstruction with Spatio-Temporally Offset RGBD Cameras". In: *IEEE Conference Virtual Reality and 3D User Interfaces (VR)*. 2023, pp. 72–82. DOI: 10.1109/VR55154.2023.00023.

[Rho+11] Gillian Rhodes, Emma Jaquet, Linda Jeffery, Emma Evangelista, Jill Keane, and Andrew J. Calder. "Sex-specific norms code face identity." In: *Journal of vision*. 2011.

[Rib+20] Edgar Riba, Dmytro Mishkin, Daniel Ponsa, Ethan Rublee, and Gary Bradski. "Kornia: an Open Source Differentiable Computer Vision Library for PyTorch". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2020.

[RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi. Cham: Springer International Publishing, 2015, pp. 234–241. ISBN: 978-3-319-24574-4.

[RKG09] Astrid Marieke Rosenthal-von der Pütten, Nicole Krämer, and Jonathan Gratch. "Who's there? Can a Virtual Agent Really Elicit Social Presence?" In: *Proceedings of 12th Annual International Workshop on Presence*. 2009.

[Rös+19] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Niessner. "FaceForensics++: Learning to Detect Manipulated Facial Images". In: *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019.

[Rot+18] Daniel Roth, David Mal, Christian Felix Purps, Peter Kullmann, and Marc Erich Latoschik. "Injecting Nonverbal Mimicry with Hybrid Avatar-Agent Technologies: A Naive Approach". In: *Proceedings of the 2018 ACM Symposium on Spatial User Interaction*. SUI '18. Berlin, Germany: Association for Computing Machinery, 2018, pp. 69–73. ISBN: 9781450357081. DOI: 10.1145/3267782.3267791.

[RW18] Daniel Roth and Carolin Wienrich. "Effects of Media Immersiveness on the Perception of Virtual Characters". In: *Virtuelle und Erweiterte Realität - GI VR/AR Workshop*. August. 2018.

[RL01] S. Rusinkiewicz and M. Levoy. "Efficient variants of the ICP algorithm". In: *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*. 2001, pp. 145–152.

[Sag+13] Christos Sagonas, Georgios Tzimiropoulos, S. Zafeiriou, and M. Pantic. "300 Faces in-the-Wild Challenge: The First Facial Landmark Localization Challenge". In: *IEEE International Conference on Computer Vision Workshops*. 2013, pp. 397–403.

[San+15] Paolo Sangregorio, Alberto Luigi Cologni, Franklin Caleb Owen, and Fabio Previdi. "An integrated system for supporting remote maintenance services". In: *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*. Vol. 2015-October. IEEE, 2015. ISBN: 9781467379298. DOI: 10.1109/ETFA.2015.7301569.

[Sas+21] Prasanth Sasikumar, Soumith Chittajallu, Navindd Raj, Huidong Bai, and Mark Billinghurst. "Spatial perception enhancement in assembly training using augmented volumetric playback". In: *Frontiers in Virtual Reality*. Vol. 2. Frontiers Media SA, 2021, p. 698523.

[Sch+02] Dieter Schmalstieg, Anton Fuhrmann, Gerd Hesina, Zsolt Szalavári, L. Miguel Encarnação, Michael Gervautz, and Werner Purgathofer. "The Studierstube Augmented Reality Project". In: *Presence: Teleoperators and Virtual Environments*. Vol. 11. 1. 2002, pp. 33–54. DOI: 10.1162/105474602317343640.

[Sch+24] Ephraim Schott, Tony Jan Zoeppig, Anton Benjamin Lammert, and Bernd Froehlich. "Excuse Me: Large Groups in Small Rooms". In: *IEEE Conference Virtual Reality and 3D User Interfaces (VR)*. 2024, pp. 72–82. DOI: 10.1109/VR58804.2024.00031.

[SC21] Muhammad Bilal Shaikh and Douglas Chai. "RGB-D data-based action recognition: a review". In: *Sensors*. Vol. 21. 12. MDPI, 2021, p. 4246.

[Sha+14] Ari Shapiro, Andrew Feng, Ruizhe Wang, Hao Li, Mark Bolas, Gerard Medioni, and Evan Suma. "Rapid Avatar Capture and Simulation Using Commodity Depth Sensors". In: *Comput. Animat. Virtual Worlds*. Chichester, UK: John Wiley and Sons Ltd., 2014. DOI: 10.1002/cav.1579.

[Sha+15] Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, Daniel Freedman, Eyal Krupka, Andrew Fitzgibbon, Shahram Izadi, and Pushmeet Kohli. "Accurate, Robust, and Flexible Real-time Hand Tracking". In: *CHI '15 Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. Best of CHI Honorable Mention Award. ACM, 2015, pp. 3633–3642. ISBN: 978-1-4503-3145-6.

[She+15] J. Shen, S. Zafeiriou, G. G. Chrysos, J. Kossaifi, G. Tzimiropoulos, and M. Pantic. "The First Facial Landmark Tracking in-the-Wild Challenge: Benchmark and Results". In: *IEEE International Conference on Computer Vision Workshop (ICCVW)*. 2015, pp. 1003–1011. DOI: 10.1109/ICCVW.2015.132.

[SWC76] John Short, Ederyn Williams, and Bruce Christie. *The Social Psychology of Telecommunications*. Wiley, 1976.

[Sit+20] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. "Implicit Neural Representations with Periodic Activation Functions". In: *Proc. NeurIPS*. 2020.

[Sla04] Mel Slater. "How colorful was your day? Why questionnaires cannot assess presence in virtual environments". In: *Presence: Teleoperators and Virtual Environments*. 2004. DOI: 10.1162/1054746041944849.

[SW97] Mel Slater and Sylvia Wilbur. "A framework for immersive virtual environments five: Speculations on the role of presence in virtual environments". In: vol. 6. 6. Cambridge, MA, USA: MIT Press, 1997, pp. 603–616. DOI: 10.1162/pres.1997.6.6.603.

[SN18] Harrison Jesse Smith and Michael Neff. "Communication Behavior in Embodied Virtual Reality". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*. 2018. ISBN: 9781450356206. DOI: 10.1145/3173574.3173863.

[Sod+13] Rajinder S. Sodhi, Brett R. Jones, David Forsyth, Brian P Bailey, and Giuliano Maciocci. "BeThere: 3D Mobile Collaboration with Spatial Input". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*. 2013, pp. 179–188. DOI: 10.1145/2470654.2470679.

[Sof24] iPi Soft LLC. *Markerless Motion Capture*. Accessed: July 27, 2024. 2024. URL: http://ipisoft.com/.

[Sta97] International Organization for Standardization. *ISO/IEC 14772-1:1997*. https://www.iso.org/standard/25508.html. 1997.

[Sta03] International Organization for Standardization. *ISO/IEC 14772-1:1997/AMD 1:2003*. https://www.iso.org/standard/30998.html. 2003.

[Sta06] International Organization for Standardization. *ISO/IEC 19774:2006*. https://www.iso.org/standard/33912.html. 2006.

[Ste24] Steel Crate Games. *Keep Talking and Nobody Explodes*. https://store.steampowered.com/app/341800/Keep_Talking_and_Nobody_Explodes/. Accessed: 2024-05-04. 2024.

[Ste06] Jonathan Steuer. "Defining Virtual Reality: Dimensions Determining Telepresence". In: *Journal of Communication*. Vol. 42. 4. 2006, pp. 73–93. DOI: 10.1111/j.1460-2466.1992.tb00812.x.

[SCP95] Richard Stoakley, Matthew J. Conway, and Randy Pausch. "Virtual reality on a WIM". In: *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '95*. New York, New York, USA: ACM Press, 1995, pp. 265–272. ISBN: 0201847051.

[Sto+11] Carsten Stoll, Nils Hasler, Juergen Gall, Hans-Peter Seidel, and Christian Theobalt. "Fast articulated motion tracking using a sums of Gaussians body model". In: *International Conference on Computer Vision*. 2011, pp. 951–958. DOI: 10.1109/ICCV.2011.6126338.

[Sto+19a] Patrick Stotko, Stefan Krumpen, Matthias B. Hullin, Michael Weinmann, and Reinhard Klein. "SLAMCast: Large-Scale, Real-Time 3D Reconstruction and Streaming for Immersive Multi-Client Live Telepresence". In: *IEEE Transactions on Visualization and Computer Graphics (TVCG)*. Vol. 25. 5. 2019, pp. 2102–2112. DOI: 10.1109/TVCG.2019.2899231.

[Sto+19b] Patrick Stotko, Stefan Krumpen, Michael Weinmann, and Reinhard Klein. "Efficient 3D Reconstruction and Streaming for Group-Scale Multi-Client Live Telepresence". In: *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. 2019, pp. 19–25. DOI: 10.1109/ISMAR.2019.00018.

[SK14] Jeremy Straub and Scott Kerlin. “Development of a Large, Low-Cost, Instant 3D Scanner”. In: *Technologies*. 2014. DOI: 10.3390/technologies2020076.

[Sum+11] E. A. Suma, B. Lange, A. S. Rizzo, D. M. Krum, and M. Bolas. “FAAST: The Flexible Action and Articulated Skeleton Toolkit”. In: *IEEE Virtual Reality Conference*. 2011, pp. 247–248. DOI: 10.1109/VR.2011.5759491.

[Sut66] Ivan E. Sutherland. “The Ultimate Display”. In: *Proceedings of the International Federation of Information Processing Congress*. Ed. by Wayne Kalenich. Vol. 2. Washington/London, 1966, pp. 506–508.

[SVL14] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. “Sequence to Sequence Learning with Neural Networks”. In: *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. NIPS’14. Montreal, Canada: MIT Press, 2014, pp. 3104–3112.

[SKS14] Supasorn Suwajanakorn, Ira Kemelmacher-Shlizerman, and Steven M. Seitz. “Total Moving Face Reconstruction”. In: *Computer Vision – ECCV 2014*. Ed. by David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars. Cham: Springer International Publishing, 2014, pp. 796–812. ISBN: 978-3-319-10593-2.

[TB15] Matthew Tait and Mark Billinghurst. “The Effect of View Independence in a Collaborative AR System”. In: *Computer Supported Cooperative Work: CSCW: An International Journal*. Vol. 24. 6. Springer Netherlands, 2015, pp. 563–589. ISBN: 0925-9724. DOI: 10.1007/s10606-015-9231-8.

[Tan+20] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. “Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains”. In: *NeurIPS*. 2020.

[TEN14] Joshua Tanenbaum, Magy Seif El-Nasr, and Michael Nixon. *Nonverbal Communication in Virtual Worlds*. ETC Press Pittsburgh, PA, 2014.

[Tan+03] Arthur Tang, Charles Owen, Frank Biocca, and Weimin Mou. “Comparative effectiveness of augmented reality in object assembly”. In: *Proceedings of the conference on Human factors in computing systems - CHI ’03*. 2003, p. 73. ISBN: 1581136307.

[Tar+23] Hugo Le Tarnec, Elisabetta Bevacqua, Olivier Augereau, and Pierre De Loor. “Effect of Avatar Facial Expressiveness on Team Collaboration in Virtual Reality”. In: *Proceedings of the 23rd ACM International Conference on Intelligent Virtual Agents*. IVA ’23. New York, NY, USA: Association for Computing Machinery, 2023. ISBN: 9781450399944. DOI: 10.1145/3570945.3607330.

[Tay+01] Russell M. Taylor, Thomas C. Hudson, Adam Seeger, Hans Weber, Jeffrey Juliano, and Aron T. Helser. “VRPN: A Device-Independent, Network-Transparent VR Peripheral System”. In: *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*. VRST ’01. Baniff, Alberta, Canada: Association for Computing Machinery, 2001, pp. 55–61. ISBN: 1581134274. DOI: 10.1145/505008.505019.

[Tea24] Teamviewer GmbH. *Teamviewer Assist AR*. https://www.teamviewer.com/de/products/remote/solutions/remote-ar-assistance/. Accessed: 2024-05-04. 2024.

[TAH12] Franco Tecchia, Leila Alem, and Weidong Huang. “3D Helping Hands : a Gesture Based MR System for Remote Collaboration”. In: *VRCAI - Virtual Reality Continuum and its Applications in Industry*. Vol. 1. 212. ACM Press, 2012, pp. 323–328. ISBN: 9781450318259.

[Tec19] Unity Technologies. *Preparing Humanoid Assets for export*. 2019. URL: https://docs.unity3d.com/Manual/UsingHumanoidChars.html.

[Tec20] Unity Technologies. *Unity - Manual: Humanoid Avatars*. 2020. URL: https://docs.unity3d.com/Manual/AvatarCreationandSetup.html.

[Tes24] Brent-A Tesych Philmea. *Azure Kinect DK depth camera*. https://learn.microsoft.com/en-us/azure/kinect-dk/depth-camera (Accessed at 16.03.2024). 2024.

[Tew+20] A. Tewari, O. Fried, J. Thies, V. Sitzmann, S. Lombardi, K. Sunkavalli, R. Martin-Brualla, T. Simon, J. Saragih, M. Nießner, R. Pandey, S. Fanello, G. Wetzstein, J.-Y. Zhu, C. Theobalt, M. Agrawala, E. Shechtman, D. B Goldman, and M. Zollhöfer. “State of the Art on Neural Rendering”. In: *Computer Graphics Forum*. 2020.

[Tew+22] Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul Srinivasan, Edgar Tretschk, Wang Yifan, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, et al. "Advances in neural rendering". In: *Computer Graphics Forum*. Vol. 41. 2. Wiley Online Library. 2022, pp. 703–735.

[Thi+15] J. Thies, M. Zollhöfer, M. Nießner, L. Valgaerts, M. Stamminger, and C. Theobalt. "Real-time Expression Transfer for Facial Reenactment". In: *ACM Transactions on Graphics (TOG)*. Vol. 34. 6. ACM, 2015.

[Thi+20] Justus Thies, Mohamed Elgharib, Ayush Tewari, Christian Theobalt, and Matthias Nießner. "Neural Voice Puppetry: Audio-driven Facial Reenactment". In: *ECCV 2020*. 2020.

[TZN19] Justus Thies, Michael Zollhöfer, and Matthias Nießner. "Deferred neural rendering: image synthesis using neural textures". In: *ACM Trans. Graph.* Vol. 38. 4. New York, NY, USA: Association for Computing Machinery, 2019. DOI: 10.1145/3306346.3323035.

[Thi+18a] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. "Face2Face: Real-Time Face Capture and Reenactment of RGB Videos". In: *Commun. ACM*. Vol. 62. 1. New York, NY, USA: Association for Computing Machinery, 2018, pp. 96–104. DOI: 10.1145/3292039.

[Thi+18b] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. "FaceVR: Real-Time Gaze-Aware Facial Reenactment in Virtual Reality". In: *ACM Trans. Graph.* Vol. 37. 2. New York, NY, USA: Association for Computing Machinery, 2018.

[TGG20] Marcel Tiator, Christian Geiger, and Paul Grimm. "Point cloud segmentation with deep reinforcement learning". In: *ECAI 2020*. IOS Press, 2020, pp. 2768–2775.

[TB11] Fabian Timm and Erhardt Barth. "Accurate Eye Centre Localisation by Means of Gradients". In: *Computer Vision Theory and Applications (VISAPP)*. 2011.

[Upl23] UploadVR.com. https://www.uploadvr.com/epic-games-hyprsense/. 2023.

[Val24] Valve Corp. *SteamVR on Steam*. Accessed: July 25, 2024. 2024. URL: https://store.steampowered.com/app/250820/SteamVR.

[Vas+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017.

[Vic20] Vicon Motion Systems Ltd. *Tracker - Delivery Precise Real-World Data | Motion Capture Software*. February 24, 2020. 2020. URL: https://www.vicon.com/software/tracker/.

[VJ01] P. Viola and M. Jones. "Rapid object detection using a boosted cascade of simple features". In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. Vol. 1. 2001, pp. I–I. DOI: 10.1109/CVPR.2001.990517.

[VRc20] VRchat Inc. *VRchat*. 2020. URL: https://vrchat.com/.

[Wag+04] M. Wagner, A. MacWilliams, M. Bauer, G. Klinker, J. Newman, T. Pintaric, and D. Schmalstieg. "Fundamentals of Ubiquitous Tracking". In: *IN ADVANCES IN PERVASIVE COMPUTING*. Austrian Computer Society, 2004, pp. 285–290.

[Wal+18a] Thomas Waltemate, Dominik Gall, Daniel Roth, Mario Botsch, and Marc Erich Latoschik. "The Impact of Avatar Personalization and Immersion on Virtual Body Ownership, Presence, and Emotional Response". In: *IEEE Transactions on Visualization and Computer Graphics*. Vol. 24. 4. 2018, pp. 1643–1652. DOI: 10.1109/TVCG.2018.2794629.

[Wal+18b] Thomas Waltemate, Dominik Gall, Daniel Roth, Mario Botsch, and Marc Erich Latoschik. "The impact of avatar personalization and immersion on virtual body ownership, presence, and emotional response". In: *IEEE Transactions on Visualization and Computer Graphics*. 2018. DOI: 10.1109/TVCG.2018.2794629.

[Wan+18a] Pichao Wang, Wanqing Li, Philip Ogunbona, Jun Wan, and Sergio Escalera. "RGB-D-based human motion recognition with deep learning: A survey". In: *Computer vision and image understanding*. Vol. 171. Elsevier, 2018, pp. 118–139.

[Wan+18b] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. "High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[WAB93] Colin Ware, Kevin Arthur, and Kellogg S. Booth. "Fish Tank Virtual Reality". In: *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*. CHI '93. Amsterdam, The Netherlands: ACM, 1993, pp. 37–42. ISBN: 0-89791-575-5. DOI: 10.1145/169059.169066.

[Wat+22] Ukrit Watchareeruetai, Benjaphan Sommana, Sanjana Jain, Pavit Noinongyao, Ankush Ganguly, Aubin Samacoits, Samuel W. F. Earp, and Nakarin Sritrakool. "LOTR: Face Landmark Localization Using Localization Transformer". In: *IEEE Access*. Vol. 10. 2022, pp. 16530–16543. DOI: 10.1109/ACCESS.2022.3149380.

[WBK07] Prof. Dr. Jürgen Wegge, Tanja Bipp, and Uwe Kleinbeck. "Goal setting via videoconferencing". In: *European Journal of Work and Organizational Psychology*. Vol. 16. 2. Routledge, 2007, pp. 169–194. DOI: 10.1080/13594320601125567.

[Wei+19] Shih-En Wei, Jason Saragih, Tomas Simon, Adam W. Harley, Stephen Lombardi, Michal Perdoch, Alexander Hypes, Dawei Wang, Hernan Badino, and Yaser Sheikh. "VR Facial Animation via Multiview Image Translation". In: *ACM Trans. Graph.* Vol. 38. 4. New York, NY, USA: Association for Computing Machinery, 2019.

[Wei+11] Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly. "Realtime performance-based facial animation". In: *ACM Trans. Graph.* Vol. 30. 4. New York, NY, USA: Association for Computing Machinery, 2011. DOI: 10.1145/2010324.1964972.

[Wei+09] Thibaut Weise, Hao Li, Luc Van Gool, and Mark Pauly. "Face/Off: live facial puppetry". In: *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. SCA '09. New Orleans, Louisiana: Association for Computing Machinery, 2009, pp. 7–16. ISBN: 9781605586106. DOI: 10.1145/1599470.1599472.

[Wel+96] Robert B. Welch, Theodore T. Blackmon, Andrew Liu, Barbara A. Mellers, and Lawrence W. Stark. "The effects of pictorial realism, delay of visual feedback, and observer interactivity on the subjective sense of presence". In: *Presence: Teleoperators and Virtual Environments*. Vol. 5. 3. 1996, pp. 263–273. DOI: 10.1162/pres.1996.5.3.263.

[Woo+21] Erroll Wood, Tadas Baltrušaitis, Charlie Hewitt, Sebastian Dziadzio, Matthew Johnson, Virginia Estellers, Thomas J. Cashman, and Jamie Shotton. *Fake It Till You Make It: Face analysis in the wild using synthetic data alone*. 2021.

[Wri05] Matthew Wright. "Open Sound Control: An Enabling Technology for Musical Networking". In: *Org. Sound*. Vol. 10. 3. USA: Cambridge University Press, 2005, pp. 193–200. DOI: 10.1017/S1355771805000932.

[Wu+16a] Chenglei Wu, Derek Bradley, Markus Gross, and Thabo Beeler. "An Anatomically-Constrained Local Deformation Model for Monocular Face Capture". In: *ACM Trans. Graph.* Vol. 35. 4. New York, NY, USA: Association for Computing Machinery, 2016. DOI: 10.1145/2897824.2925882.

[WXN21] Cho-Ying Wu, Qiangeng Xu, and Ulrich Neumann. "Synergy between 3dmm and 3d landmarks for accurate 3d facial geometry". In: *International Conference on 3D Vision (3DV)*. IEEE. 2021, pp. 453–463.

[Wu+16b] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T Freeman, and Joshua B Tenenbaum. "Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling". In: *Advances in Neural Information Processing Systems*. 2016, pp. 82–90.

[WY17] Wenyan Wu and Shuo Yang. "Leveraging Intra and Inter-Dataset Variations for Robust Face Alignment". In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2017, pp. 2096–2105. DOI: 10.1109/CVPRW.2017.261.

[Wu+21] Yuanjie Wu, Yu Wang, Sungchul Jung, Simon Hoermann, and Robert W Lindeman. "Using a fully expressive avatar to collaborate in virtual reality: Evaluation of task performance, presence, and attraction". In: *Frontiers in Virtual Reality*. Vol. 2. Frontiers Media SA, 2021, p. 641296.

[Xie+23] Shaowen Xie, Hao Zhu, Zhen Liu, Qi Zhang, You Zhou, Xun Cao, and Zhan Ma. "DINER: Disorder-Invariant Implicit Neural Representation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023.

[Xio+14] Xuehan Xiong, Zicheng Liu, Qin Cai, and Zhengyou Zhang. "Eye Gaze Tracking Using an RGBD Camera: A Comparison with a RGB Solution". In: UbiComp '14 Adjunct. Seattle, Washington, 2014. ISBN: 9781450330473.

[XSe] XSens. *MVN Animate*. Accessed: 24.02.2020. URL: https://www.xsens.com/products/mvn-animate.

[Yam+17] Koki Yamashita, Takashi Kikuchi, Katsutoshi Masai, Maki Sugimoto, Bruce H. Thomas, and Yuta Sugiura. "CheekInput: Turning Your Cheek into an Input Surface by Embedded Optical Sensors on a Head-Mounted Display". In: *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology*. VRST '17. Gothenburg, Sweden: Association for Computing Machinery, 2017. ISBN: 9781450355483. DOI: 10.1145/3139131.3139146.

[YS19] Alan Yates and Jeremy Selan. *Positional tracking systems and methods*. US Patent 10,338,186. 2019.

[You03] Christine Youngblut. "Experience of Presence in Virtual Environments". In: *Defense Technical Information Center*. 2003.

[Yu+21a] Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. "Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation". In: *International Journal of Computer Vision*. Vol. 129. Springer, 2021, pp. 3051–3068.

[Yu+21b] Kevin Yu, Gleb Gorbachev, Ulrich Eck, Frieder Pankratz, Nassir Navab, and Daniel Roth. "Avatars for Teleconsultation: Effects of Avatar Embodiment Techniques on User Perception in 3D Asymmetric Telepresence". In: *IEEE Transactions on Visualization and Computer Graphics*. 2021. DOI: 10.1109/TVCG.2021.3106480.

[Zha+18] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[Zha+20] Yuxuan Zhang, Chen Wenzheng, Huan Ling, Jun Gao, Yinan Zhang, Antonio Torralba, and Sanja Fidler. "Image GANs meet Differentiable Rendering for Inverse Graphics and Interpretable 3D Neural Rendering". In: 2020.

[Zha00] Z. Zhang. "A flexible new technique for camera calibration". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Vol. 22. 11. 2000, pp. 1330–1334. DOI: 10.1109/34.888718.

[Zhe+22] Yufeng Zheng, Victoria Fernández Abrevaya, Marcel C. Bühler, Xu Chen, Michael J. Black, and Otmar Hilliges. "I M Avatar: Implicit Morphable Head Avatars from Videos". In: *Computer Vision and Pattern Recognition (CVPR)*. 2022.

[Zhe+23] Yufeng Zheng, Wang Yifan, Gordon Wetzstein, Michael J. Black, and Otmar Hilliges. "PointAvatar: Deformable Point-based Head Avatars from Videos". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023.

[Zho+04] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. "Image quality assessment: from error visibility to structural similarity". In: *IEEE Transactions on Image Processing*. Vol. 13. 4. 2004. DOI: 10.1109/TIP.2003.819861.

[Zhu+17] J. Zhu, T. Park, P. Isola, and A. A. Efros. "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks". In: *IEEE International Conference on Computer Vision (ICCV)*. 2017.

[ZBT22a] Wojciech Zielonka, Timo Bolkart, and Justus Thies. "Instant Volumetric Head Avatars". In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 4574–4584.

[ZBT22b] Wojciech Zielonka, Timo Bolkart, and Justus Thies. "Towards Metrical Reconstruction of Human Faces". In: *European Conference on Computer Vision (ECCV)*. Springer International Publishing, 2022.

[Zol+18] M. Zollhöfer, Justus Thies, P. Garrido, D. Bradley, T. Beeler, Patrick Pérez, M. Stamminger, M. Nießner, and C. Theobalt. "State of the Art on Monocular 3D Face Reconstruction, Tracking, and Applications". In: *Computer Graphics Forum*. Vol. 37. 2018.

A. Source Code, Implementation Details and Videos

A.1. Code and Videos for the MotionHub from Chap. 4

Code:
https://github.com/Mirevi/MotionHub

Oral presentation:
https://youtu.be/GRZqkAN6I9k

Human Tetris Demo:
https://youtu.be/O_5hiweZQhE

A.2. Video of user study from Chap. 6

Demo video user study:
https://youtu.be/_SJYunw6kVU

A.3. Code for the Face-Tracking HMD from Chap. 5.4

https://github.com/Mirevi/UCP-Framework/tree/main/Lower-Face-CNN

A.4. Code and Videos for First GAN Prototype from Chap. 7.5

Code:
https://github.com/Alpe6825/RGBD-Face-Avatar-GAN

Oral presentation:
https://youtu.be/iL4Z3tg6qFs

A.5. Code and Videos for Second GAN Prototype from Chap. 7.6

Code and 3D print files:
https://github.com/Mirevi/UCP-Framework

Oral presentation:
https://youtu.be/Wa95qDPV8vk

A.6. Code and Videos for Third GAN Prototype from Chap. 7.7

Code:
https://github.com/Mirevi/face-synthesizer-JVRB

Oral presentation:
https://youtu.be/fBofqRfvoiM

A.7. Videos for Implicit Neural Representation (INR) Approach from Chap. 8

Video self-driven avatar (telepresence quality):
https://youtu.be/aXQh6xvscko

Interactive Virtual Assistance:
https://youtu.be/yZQ5jmdExsE

Architecture of the GAN of chapter 7 for image synthesis of 512px edge length

```
Layer (type:depth-idx)                                  Output Shape              Param #
==========================================================================================
├─UnetSkipConnectionBlock: 1-1                          [-1, 4, 512, 512]         --
|    └─Sequential: 2-1                                  [-1, 4, 512, 512]         --
|    |    └─Conv2d: 3-1                                 [-1, 64, 256, 256]        1,024
|    |    └─UnetSkipConnectionBlock: 3-2                [-1, 128, 256, 256]       --
|    |    |    └─Sequential: 4-1                        [-1, 64, 256, 256]        --
|    |    |    |    └─LeakyReLU: 5-1                    [-1, 64, 256, 256]        --
|    |    |    |    └─Conv2d: 5-2                       [-1, 128, 128, 128]       131,072
|    |    |    |    └─BatchNorm2d: 5-3                  [-1, 128, 128, 128]       256
|    |    |    |    └─UnetSkipConnectionBlock: 5-4      [-1, 256, 128, 128]       --
|    |    |    |    |    └─Sequential: 6-1              [-1, 128, 128, 128]       --
|    |    |    |    |    |    └─LeakyReLU: 7-1          [-1, 128, 128, 128]       --
|    |    |    |    |    |    └─Conv2d: 7-2             [-1, 256, 64, 64]         524,288
|    |    |    |    |    |    └─BatchNorm2d: 7-3        [-1, 256, 64, 64]         512
|    |    |    |    |    |    └─UnetSkipConnectionBlock: 7-4    [-1, 512, 64, 64]   --
|    |    |    |    |    |    |    └─Sequential: 8-1    [-1, 256, 64, 64]         --
|    |    |    |    |    |    |    |    └─LeakyReLU: 9-1    [-1, 256, 64, 64]      --
|    |    |    |    |    |    |    |    └─Conv2d: 9-2   [-1, 512, 32, 32]         2,097,152
|    |    |    |    |    |    |    |    └─BatchNorm2d: 9-3    [-1, 512, 32, 32]    1,024
|    |    |    |    |    |    |    |    └─UnetSkipConnectionBlock: 9-4    [-1, 1024, 32, 32]    --
|    |    |    |    |    |    |    |    |    └─Sequential: 10-1    [-1, 512, 32, 32]    --
|    |    |    |    |    |    |    |    |    |    └─LeakyReLU: 11-1    [-1, 512, 32, 32]    --
|    |    |    |    |    |    |    |    |    |    └─Conv2d: 11-2    [-1, 512, 16, 16]    4,194,304
|    |    |    |    |    |    |    |    |    |    └─BatchNorm2d: 11-3    [-1, 512, 16, 16]    1,024
|    |    |    |    |    |    |    |    |    |    └─UnetSkipConnectionBlock: 11-4    [-1, 1024, 16, 16]    --
|    |    |    |    |    |    |    |    |    |    |    └─Sequential: 12-1    [-1, 512, 16, 16]    --
|    |    |    |    |    |    |    |    |    |    |    |    └─LeakyReLU: 13-1    [-1, 512, 16, 16]    --
|    |    |    |    |    |    |    |    |    |    |    |    └─Conv2d: 13-2    [-1, 512, 8, 8]    4,194,304
|    |    |    |    |    |    |    |    |    |    |    |    └─BatchNorm2d: 13-3    [-1, 512, 8, 8]    1,024
|    |    |    |    |    |    |    |    |    |    |    |    └─UnetSkipConnectionBlock: 13-4    [-1, 1024, 8, 8]    --
|    |    |    |    |    |    |    |    |    |    |    |    |    └─Sequential: 14-1    [-1, 512, 8, 8]    --
|    |    |    |    |    |    |    |    |    |    |    |    |    |    └─LeakyReLU: 15-1    [-1, 512, 8, 8]    --
|    |    |    |    |    |    |    |    |    |    |    |    |    |    └─Conv2d: 15-2    [-1, 512, 4, 4]    4,194,304
|    |    |    |    |    |    |    |    |    |    |    |    |    |    └─BatchNorm2d: 15-3    [-1, 512, 4, 4]    1,024
|    |    |    |    |    |    |    |    |    |    |    |    |    |    └─UnetSkipConnectionBlock: 15-4    [-1, 1024, 4, 4]    --
|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    └─Sequential: 16-1    [-1, 512, 4, 4]    --
|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    └─LeakyReLU: 17-1    [-1, 512, 4, 4]    --
|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    └─Conv2d: 17-2    [-1, 512, 2, 2]    4,194,304
|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    └─BatchNorm2d: 17-3    [-1, 512, 2, 2]    1,024
|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    └─UnetSkipConnectionBlock: 17-4    [-1, 1024, 2, 2]    --
|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    └─Sequential: 18-1    [-1, 512, 2, 2]    --
|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    └─LeakyReLU: 19-1    [-1, 512, 2, 2]    --
|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    └─Conv2d: 19-2    [-1, 512, 1, 1]    4,194,304
|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    └─ReLU: 19-3    [-1, 512, 1, 1]    --
|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    └─ConvTranspose2d: 19-4    [-1, 512, 2, 2]    4,194,304
|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    └─BatchNorm2d: 19-5    [-1, 512, 2, 2]    1,024
|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    └─ReLU: 17-5    [-1, 1024, 2, 2]    --
|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    └─ConvTranspose2d: 17-6    [-1, 512, 4, 4]    8,388,608
|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    └─BatchNorm2d: 17-7    [-1, 512, 4, 4]    1,024
|    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    └─Dropout: 17-8    [-1, 512, 4, 4]    --
|    |    |    |    |    |    |    |    |    |    |    |    |    |    └─ReLU: 15-5    [-1, 1024, 4, 4]    --
|    |    |    |    |    |    |    |    |    |    |    |    |    |    └─ConvTranspose2d: 15-6    [-1, 512, 8, 8]    8,388,608
|    |    |    |    |    |    |    |    |    |    |    |    |    |    └─BatchNorm2d: 15-7    [-1, 512, 8, 8]    1,024
|    |    |    |    |    |    |    |    |    |    |    |    |    |    └─Dropout: 15-8    [-1, 512, 8, 8]    --
|    |    |    |    |    |    |    |    |    |    |    |    └─ReLU: 13-5    [-1, 1024, 8, 8]    --
|    |    |    |    |    |    |    |    |    |    |    |    └─ConvTranspose2d: 13-6    [-1, 512, 16, 16]    8,388,608
|    |    |    |    |    |    |    |    |    |    |    |    └─BatchNorm2d: 13-7    [-1, 512, 16, 16]    1,024
|    |    |    |    |    |    |    |    |    |    |    |    └─Dropout: 13-8    [-1, 512, 16, 16]    --
|    |    |    |    |    |    |    |    |    |    └─ReLU: 11-5    [-1, 1024, 16, 16]    --
|    |    |    |    |    |    |    |    |    |    └─ConvTranspose2d: 11-6    [-1, 512, 32, 32]    8,388,608
|    |    |    |    |    |    |    |    |    |    └─BatchNorm2d: 11-7    [-1, 512, 32, 32]    1,024
|    |    |    |    |    |    |    |    |    |    └─Dropout: 11-8    [-1, 512, 32, 32]    --
|    |    |    |    |    |    |    |    └─ReLU: 9-5    [-1, 1024, 32, 32]    --
|    |    |    |    |    |    |    |    └─ConvTranspose2d: 9-6    [-1, 256, 64, 64]    4,194,304
|    |    |    |    |    |    |    |    └─BatchNorm2d: 9-7    [-1, 256, 64, 64]    512
|    |    |    |    |    |    └─ReLU: 7-5                [-1, 512, 64, 64]         --
|    |    |    |    |    |    └─ConvTranspose2d: 7-6     [-1, 128, 128, 128]       1,048,576
|    |    |    |    |    |    └─BatchNorm2d: 7-7         [-1, 128, 128, 128]       256
|    |    |    |    └─ReLU: 5-5                          [-1, 256, 128, 128]       --
|    |    |    |    └─ConvTranspose2d: 5-6               [-1, 64, 256, 256]        262,144
|    |    |    |    └─BatchNorm2d: 5-7                   [-1, 64, 256, 256]        128
|    |    └─ReLU: 3-3                                    [-1, 128, 256, 256]       --
|    |    └─ConvTranspose2d: 3-4                         [-1, 4, 512, 512]         8,196
|    |    └─Tanh: 3-5                                    [-1, 4, 512, 512]         --
==========================================================================================
Total params: 66,998,916
Trainable params: 66,998,916
Non-trainable params: 0
Total mult-adds (G): 73.92
==========================================================================================
Input size (MB): 1.00
Forward/backward pass size (MB): 221.32
Params size (MB): 255.58
Estimated Total Size (MB): 477.90
==========================================================================================
```

B. Data of User Studies

The following pages report the raw study data that led to the scientific findings in the corresponding sections. The questionnaire is available at the following link: https://docs.google.com/forms/d/e/1FAIpQLSeHlLt6xNs3ER9GzXbNTCQVJPC39ROFBdLCXCIMZ5KFASZW8A/viewform

Data 1/2 for user study from Sec. 3.2

Session	Gesamtzeit	GameMode	ModuleDistance	ModuleLight	ModuleCablePin	ModulePoti	PotiFehler	ModuleFlipSwitch	FlipSwitchFehler	ModuleIR	Bold times	
1	0:07:31	mit Zeigen	0:00:56	**0:00:28**	**0:00:57**	**0:04:14**	**2**	**0:00:37**	**0**	0:00:21	0:05:48	348
2	0:04:45	mit Zeigen	0:00:04	**0:00:09**	**0:00:27**	**0:01:21**	**3**	**0:02:02**	**7**	0:00:42	0:03:50	230
3	0:15:19	ohne Zeigen	0:02:47	**0:00:55**	**0:04:16**	**0:03:05**	**3**	**0:03:36**	**1**	0:00:40	0:10:57	657
4	0:10:04	ohne Zeigen	0:01:24	**0:02:33**	**0:01:51**	**0:01:53**	**5**	**0:01:31**	**1**	0:00:51	0:05:15	315
5	0:08:54	mit Zeigen	0:01:53	**0:02:23**	**0:01:44**	**0:01:28**	**0**	**0:00:54**	**1**	0:00:32	0:04:06	246
6	0:16:20	ohne Zeigen	0:03:23	**0:00:47**	**0:03:50**	**0:02:41**	**0**	**0:05:00**	**15**	0:00:40	0:11:31	691
7	0:08:42	mit Zeigen	0:01:16	**0:00:38**	**0:01:56**	**0:02:28**	**0**	**0:01:38**	**0**	0:00:46	0:06:02	362
8	0:04:17	mit Zeigen	0:00:04	**0:00:43**	**0:00:58**	**0:01:07**	**0**	**0:01:01**	**0**	0:00:24	0:03:06	186
9	0:10:01	ohne Zeigen	0:01:07	**0:02:22**	**0:01:31**	**0:02:07**	**2**	**0:02:33**	**4**	0:00:22	0:06:11	371
10	0:09:35	ohne Zeigen	0:01:59	**0:01:18**	**0:01:13**	**0:01:52**	**1**	**0:02:31**	**3**	0:00:41	0:05:36	336
11	0:12:02	mit Zeigen	0:03:58	**0:01:35**	**0:01:22**	**0:01:13**	**0**	**0:03:14**	**2**	0:00:41	0:05:49	349
12	0:20:26	ohne Zeigen	0:05:41	**0:02:07**	**0:01:45**	**0:04:06**	**1**	**0:04:47**	**3**	0:01:14	0:10:38	638
13	0:09:46	mit Zeigen	0:00:58	**0:01:44**	**0:01:14**	**0:04:01**	**0**	**0:01:11**	**0**	0:00:40	0:06:26	386
14	0:14:31	ohne Zeigen	0:01:56	**0:02:16**	**0:03:29**	**0:03:06**	**3**	**0:03:05**	**0**	0:00:37	0:09:40	580
15	0:13:48	mit Zeigen	0:02:54	**0:02:25**	**0:04:04**	**0:01:49**	**0**	**0:02:00**	**0**	0:00:36	0:07:53	473
16	0:12:37	ohne Zeigen	0:03:32	**0:01:00**	**0:01:30**	**0:02:27**	**1**	**0:03:38**	**5**	0:00:30	0:07:35	455
17	0:16:09	ohne Zeigen	0:06:36	**0:02:15**	**0:01:32**	**0:01:58**	**0**	**0:03:02**	**1**	0:00:47	0:06:32	392
18	0:12:00	mit Zeigen	0:03:21	**0:01:30**	**0:02:07**	**0:02:52**	**0**	**0:01:46**	**0**	0:00:24	0:06:45	405
	Gesamtzeit	Mittelwerte >										
mit Zeigen	**1:21:45**	**0:09:05**		**0:01:43**	**0:01:17**	**0:01:39**	**0:02:17**	**0,56**	**0:01:36**	**1,11**	**0:00:34**	
ohne Zeigen	**2:05:02**	**0:13:54**		**0:03:09**	**0:01:44**	**0:02:20**	**0:02:35**	**1,78**	**0:03:18**	**3,67**	**0:00:42**	
Differenz	**0:43:17**	**0:04:49**		**0:01:27**	**0:00:26**	**0:00:41**	**0:00:18**	**1,22**	**0:01:42**	**2,56**	**0:00:08**	

9x mit Zeigen
9x ohne Zeigen

Fehler (Poti + FlipSwitch)

mit	ohne
2	4
10	6
1	15
0	6
0	4
2	4
0	3
0	6
0	1

Levene Test	bestanden
Varianz Mit	11,98214286
Varianz Ohne	17,41071429
Teststatistik	1,453055142
kritischer Wert der F-vert	3,178893104
***1**	
***2**	

Summe der Aufgaben -> nur mit Pointen <-

	mit	ohne
1	376	712
2	239	468
3	389	738
4	400	513
5	229	414
6	444	765
7	430	716
8	618	515
9	495	527
Summen:	3620	5368

Levene Test	bestanden
Varianz Mit	14399,94444
Varianz Ohne	18037,27778
Teststatistik	1,252593567
kritischer Wert der F-	3,178893104
***1**	
***2**	

Gesamtzeit des Tests

mit	ohne
451	919
285	604
534	980
522	601
257	575
722	1181
586	757
828	871
720	969
4905	7457

Übrig geblieben Zeiten ->Tasks die kein Pointen beinhalten, unabhängig von Condition)

mit	ohne
75	207
46	136
145	242
122	88
28	161
278	416
156	41
210	356
225	442
1285	2089

Levene Test	bestanden
Varianz Mit	7163,69444
Varianz Ohne	20691,3611
Teststatistik	2,88836455
kritischer Wert der	3,1788931
***1**	
***2**	

***1** Test sagt, dass der kritische Wert größer als die Teststastik und damit können wir die H0 nicht verwerfen. H0 sagt aus, dass eine Varianzen-Gleichheit vorliegt
***2** Heißt, dass man einen t-test durchführen darf.

Data 2/2 for user study from Sec. 3.2

Session	Teilnehmer	Geschlecht	Alter	Deiktische Ausdrücke	Fragen	Explikative Aussagen	Kopplungsgrad	Führungsrolle	Bitte bewerten Sie, wie viel Erfahrung Sie im Bereich der Virtuellen Realität (VR) haben	Bitte bewerten Sie, wie viel Erfahrung Sie im Bereich der Erweiterten Realität (AR) haben	Bitte bewerten Sie, wie hoch ist der Anteil an LEDs im Alltag der einen bewussten Einfluss auf Sie hat (z.B. Statusinformationen, Benachrichtigungen, Aktivitätsanzeige von elektronischen Geräten)	Bitte bewerten Sie, in wie weit kann eine LED-Anzeige der Bedienelemente am Rätselkoffer die Kommunikation zwischen beiden Spielern verbessern	Bitte bewerten Sie, ist Ihnen die Kommunikation während der Studie besonders leicht oder eher schwer gefallen	Bitte bewerten Sie, unter Berücksichtigung des Rätselspieles: wie gut war der Gesamtspielablauf
1				6	7	21	eng	Experte-Experte-Rolle						
	A (VR)	m	22						1	2	4	1	1	2
	B	m	29						1	1	2	1	1	1
2				11	15	36	eng	Experte-Experte-Rolle						
	A (VR)	m	24						1	3	4	2	1	1
	B	m	23						1	3	2	1	2	2
3				5	16	37	eng	Lehrer-Schüler-Rolle (VR > AR)						
	A (VR)	m	36						2	2	2	1	2	1
	B	m	26						2	2	4	1	3	1
4				2	7	36	eng	Experte-Experte-Rolle						
	A (VR)	m	28						6	4	4	2	1	1
	B	m	37						2	2	3	2	2	2
5				4	7	30	eng	Experte-Experte-Rolle						
	A (VR)	m	27						2	3	2	1	2	2
	B	m	27						1	1	1	1	2	2
6				6	9	47	lose	Lehrer-Schüler-Rolle (AR > VR)						
	A (VR)	w	29						1	1	4	3	2	2
	B	w	57						5	5	4	2	2	3
7				7	6	21	eng	Experte-Experte-Rolle						
	A (VR)	w	36						3	2	2	3	2	2
	B	w	30						5	4	2	2	2	1
8				6	1	13	eng	Experte-Experte-Rolle						
	A (VR)	m	31						2	3	2	2	2	1
	B	m	34						1	2	4	4	1	1
9				3	5	30	eng	Experte-Experte-Rolle						
	A (VR)	m	24						2	2	3	1	2	1
	B	w	30						6	5	2	1	1	1
10				6	23	27	lose	Lehrer-Schüler-Rolle (AR > VR)						
	A (VR)	m	24						2	3	1	2	1	2
	B	m	34						3	3	3	2	2	2
11				14	16	48	lose	Experte-Experte-Rolle						
	A (VR)	m	39						1	2	3	2	1	2
	B	m	29						2	1	2	1	4	3
12				10	30	48	eng	Experte-Experte-Rolle						
	A (VR)	w	65						5	6	4	1	1	2
	B	m	67						2	3	1	1	3	2
13				2	13	24	eng	Experte-Experte-Rolle						
	A (VR)	w	34						5	5	2	1	2	2
	B	w	41						5	6	3	2	2	2
14				1	11	45	eng	Experte-Experte-Rolle						
	A (VR)	m	25						6	6	3	1	2	2
	B	w	33						5	6	3	2	2	2
15				1	9	47	eng	Experte-Experte-Rolle						
	A (VR)	w	25						6	6	2	2	2	1
	B	w	22						6	5	2	1	1	1
16				1	4	18	lose	Lehrer-Schüler-Rolle (AR > VR)						
	A (VR)	w	29						5	5	2	1	1	1
	B	m	43						5	5	2	1	1	1
17				1	7	50	lose	Experte-Experte-Rolle						
	A (VR)	w	64						6	6	3	5	2	2
	B	w	59						6	6	3	5	2	2
18				4	10	31	eng	Experte-Experte-Rolle						
	A (VR)	m	37						5	3	2	2	2	1
	B	w	36						6	6	2	1	1	1
Durchschnitt			**34,89**	**5,00**	**10,89**	**33,83**			**3,47**	**3,61**	**2,61**	**1,78**	**1,75**	**1,61**
Min			**67**	**14**	**30**	**50**			**6**	**6**	**4**	**5**	**4**	**3**
Max			**22**	**1**	**1**	**13**			**1**	**1**	**1**	**1**	**1**	**1**
		21 Männer (58,33%)				13x eng (72,22%)		14x Experte-Experte-Rolle (77,78%)						
		15 Frauen (41,67%)				5x lose (27,78%)		4x Lehrer-Schüler-Rolle (22,22%)						
						davon in der Lehrer-Rolle immer der ältere Spieler								

Data for user study of Chap. 6.4

ID	Type	Scale		Condition 1 (Personal Avatar)											Condition 2 (Generic Avatar)										
Q1	Co-P.	L5	I did not want a deeper relationship with my interaction partner.	5	4	2	5	2	2	5	4	4	4	5	5	5	5	5	5	2	4	5	3	4
Q2	Co-P.	L5	I wanted to maintain a sense of distance between us.	4	4	4	4	2	4	5	3	5	3	5	3	4	5	5	4	5	2	3	5	2	3
Q3	Co-P.	L5	I was interested in talking to my interaction partner.	1	2	1	1	2	1	1	1	2	2	1	2	1	1	1	1	4	4	2	4	2
Q4	Co-P.	L5	My interaction partner was intensely involved in our interaction.	2	1	2	1	3	1	2	2	2	1	2	3	1	1	1	1	2	2	1	2	3	2
Q5	Co-P.	L5	My interaction partner seemed to find our interaction stimulating.	2	2	3	2	2	2	1	3	2	3	1	5	1	1	2	1	3	3	2	3	4	2
Q6	Co-P.	L5	My interaction partner communicated coldness rather than warmth.	3	4	2	3	4	5	5	4	4	4	5	3	3	5	5	5	2	4	5	5	1	2
Q7	Co-P.	L5	My interaction partner created a sense of distance between us.	4	4	2	5	4	5	3	3	5	4	5	5	4	5	4	5	5	4	5	5	2	3
Q8	Co-P.	L5	My interaction partner seemed detached during our interaction.	2	4	4	4	5	5	2	4	4	3	4	4	5	5	4	5	4	2	5	4	3	5
Q9	Co-P.	L5	My interaction partner acted bored by our conversation.	5	5	3	4	4	3	4	4	4	4	5	4	4	5	3	5	5	4	4	3	3	5
Q10	Co-P.	L5	My interaction partner was interested in talking to me.	2	1	3	1	2	2	1	2	2	2	2	2	2	1	2	1	1	4	2	2	4	2
Q11	Co-P.	L5	My interaction partner showed enthusiasm while talking to me.	1	1	3	3	3	2	1	3	2	2	2	2	3	1	2	1	3	4	2	2	4	3

Q12	SP.	L5	To what extent did you feel able to assess your partner's reactions to what you said?—Able to assess reactions, not able to assess reactions.	4	3	2	2	4	2	3	2	2	3	2	2	2	2	3	2	4	4	4	3	5	3
Q13	SP.	L5	To what extent was this like a face-to-face meeting?—A lot like face to face, not like face to face at all.	2	2	2	2	4	3	3	2	4	4	2	3	3	3	3	4	5	4	3	3	5	5
Q14	SP.	L5	To what extent was this like you were in the same room with your partner?—A lot like being in the same room, not like being in the same room at all.	2	1	2	3	2	3	1	2	2	3	1	4	4	3	3	3	3	4	2	4	5	4
Q15	SP.	Sliding 0-10	To what extent did your partner seem "real"?—Very real, not real at all.	5	5	4	5	4	4	3	6	5	8	4	7	8	5	4	5	8	8	3	5	9	9
Q16	SP.	Sliding 0-10	How likely is it that you would choose to use this system of interaction for a meeting in which you wanted to persuade others of something?— Very likely, not likely at all.	8	4	7	10	8	2	5	9	10	4	8	7	8	4	8	10	10	9	8	4	10	10
Q17	SP.	Sliding 0-10	To what extent did you feel you could get to know someone that you met only through this system?—Very well, not at all.	7	6	7	7	7	1	4	5	8	5	6	6	8	6	4	5	9	9	7	5	8	8

C. Curriculum Vitae

Aus Gründen des Datenschutzes ist der Lebenslauf in der Online-Version nicht enthalten.