
Erweiterung der gegenseitigen visuellen Kommunikation zwischen realen und virtuellen Umgebungen

Ein modularer, verteilter Mixed-Reality-Systementwurf zur Erkennung, Segmentierung und Visualisierung von menschlichen Texturen in unterschiedlich immersiven virtuellen Umgebungen

Master-Thesis im Studiengang
Human-Centred Computing
Fakultät Informatik
Hochschule Reutlingen
Alteburgstraße 150
72762 Reutlingen



Eingereicht von
Robin Horst
Freiherr-vom-Stein-Straße 12
66693 Mettlach
Matrikel Nr.: 761519
robinhorst@gmx.de

SPONSORED BY THE



1. Prüfer: Prof. Dr. rer. nat. Uwe Kloos
 2. Prüferin: Prof. Dr. rer. nat. Gabriela Tullius
-

INHALTSVERZEICHNIS

KURZFASSUNG	V
ABSTRACT	VI
STICHWÖRTER / KEY WORDS	VI
ABBILDUNGSVERZEICHNIS	VII
TABELLENVERZEICHNIS	X
FORMELVERZEICHNIS.....	XI
1. EINLEITUNG UND MOTIVATION	1
1.1 PROBLEMBeschreibung	1
1.1.1 <i>Ist-Situation</i>	1
1.1.2 <i>Soll-Situation</i>	2
1.1.3 <i>Definition der Problematik.</i>	2
1.2 ZIELSETZUNG.....	3
1.3 AUFBAU DER ARBEIT.....	4
1.4 VORGEHENSWEISE UND METHODIK	5
2. STAND DER WISSENSCHAFT UND TECHNIK	7
2.1 VIRTUELLE TECHNIKEN UND UMGEBUNGEN.....	7
2.2 KO-LOKALISIERTE MEHRBENUTZER-GEMISCHE REALITÄT (Co-Located Multi-User Mixed Reality)	10
2.2.1 <i>Verdeckungsproblematik in Gemischter Realität</i>	13
2.3 VISUELLE DARSTELLUNG VON NUTZERN IN VIRTUELLEN UMGEBUNGEN.....	14
2.3.1 <i>Technische Beispiele</i>	15
2.4 KOMMUNIKATION IM BEREICH DER VIRTUELLEN REALITÄT	20
2.4.1 <i>Visuelle Kommunikation</i>	20
2.4.2 <i>Visuelle Kommunikation in Virtuellen Umgebungen</i>	21
2.4.3 <i>Visuelle Kommunikation zwischen Realen und Virtuellen Umgebungen</i>	22
2.5 ERKENNUNG UND EXTRAKTION VON MENSCHLICHEN TEXTUREN AUS BILDDATEN	24
2.5.1 <i>Histogram of Oriented Gradients</i>	24
2.5.2 <i>Haar-Merkmal-basierter Cascade Classifier</i>	25
2.5.3 <i>Part Affinity Fields und Convolutional Pose Machines</i>	26
2.5.4 <i>GrabCut</i>	27
2.5.5 <i>Hintergrundsubtraktion</i>	27
2.5.6 <i>Deep Randomized Decision Forest mit Nutzung von Tiefeninformationen</i>	27
2.6 MODULARITÄT IN VIRTUELLEN TECHNOLOGIEN	28
2.6.1 <i>High-Level – Inhaltliche Ebene</i>	29
2.6.2 <i>Mid-Level – Middleware</i>	29
2.6.3 <i>Low Level – Systemarchitektur</i>	30
2.7 EVALUATION VON VIRTUELLE Realität	32
2.7.1 <i>Messung von Immersion</i>	32
2.7.2 <i>Messung von Präsenz</i>	33
2.7.3 <i>Messung der Akzeptanz von MR-Technologie</i>	33
3. DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP	35
3.1 KONZEPTION UND DESIGN DES KOMMUNIKATIONSKONZEPTE UND EINES VISUALISIERUNGSSYSTEMS.....	35
3.2 KONZEPTION UND DESIGN DES PROTOTYPs	39
3.3 TECHNISCHE DOKUMENTATION DES PROTOTYPs.....	42
3.3.1 <i>Modulare Architektur und Modulkommunikation</i>	43
3.3.2 <i>VR-Szene und Synchronisierte Non-VR-Szene</i>	58
3.3.3 <i>Kinect Module</i>	65
3.3.4 <i>Szenen-Übersicht-Compositor / Darstellung von Szene und Proband für den Experimentator</i> 68	68

INHALTSVERZEICHNIS

3.3.5 <i>Proband-POV-Composer-Modul / Darstellung des Experimentators für den Probanden</i>	72
3.3.6 <i>Nutzerverhältnis in Mehrbenutzerumgebungen</i>	81
3.4 ANWENDUNGSFALL	82
3.5 EVALUATION DES VORGESCHLAGENEN KONZEPTS MITHILFE DES PROTOTYPs	84
3.5.1 <i>Teilnehmer</i>	85
3.5.2 <i>Technik</i>	85
3.5.3 <i>Ablauf</i>	86
3.5.4 <i>Design des Experiments</i>	86
4. ANALYSE DER ERGEBNISSE.....	88
4.1 ANALYSE DER EVALUATION	88
5. DISKUSSION DER ERGEBNISSE	91
5.1 BEWERTUNG DER EVALUATION	91
5.2 BEWERTUNG DER METHODEN	93
6. ZUSAMMENFASSUNG	95
6.1 AUSBLICK	95
6.1.1 <i>Technische Verbesserungen</i>	96
6.1.2 <i>Zukünftige Arbeiten</i>	96
DANKSAGUNGEN	98
LITERATURVERZEICHNIS	99
ABKÜRZUNGSVERZEICHNIS.....	108
ANHANG	109

KURZFASSUNG

Im Zuge der anhaltenden Digitalisierung aller Bereiche des täglichen Lebens finden auch virtuelle Technologien vermehrt Einsatz in bisher branchenuntypischen Feldern. Technische Lösungen für den semiprofessionellen Markt lassen auch den Einsatz für Konsumenten zu. Aktuelle Anwendungsgebiete sind unter anderem psychologisch-orientierte Experimente, Vertriebsverhandlungen sowie soziale Medien. Einer der Kernaspekte der diese und weitere Bereiche miteinander verbindet ist die Kommunikation zwischen Menschen. Dabei spielt der visuelle Aspekt eine tragende Rolle, da die Wahrnehmung anderer Personen zum größten Teil visuell erfolgt.

In der vorliegenden Master-Thesis wird ein Konzept vorgestellt, das eine visuelle Darstellung von Personen innerhalb und außerhalb von virtuellen Umgebungen ermöglicht. Dabei wird ausschließlich semiprofessionelle Technik einbezogen. Diese Darstellungen werden dann in Relation zu den virtuellen Umgebungen visualisiert. Dabei wird weitgehend weder ein a priori festgelegtes Menschmodell noch ein animierter Avatar genutzt. Der echte Mensch wird als Textur in Echtzeit aufgenommen und durch verschiedene Transformationen realitätsnah zusammen mit der virtuellen Umgebung dargestellt. Diese vorgeschlagene Repräsentation gewährleistet die Wahrnehmung der Person als Individuum. Die visuellen Aspekte der Kommunikation, wie Körpersprache oder Kleidung, werden damit im Vergleich zu anderen Techniken verstärkt.

Bei diesen Mixed-Reality-Visualisierungen wird nach zwei Fällen unterschieden. Zum einen wird ein Außenstehender für einen Nutzer innerhalb einer immersiven virtuellen Umgebung dargestellt und zum anderen wird der Nutzer zusammen mit der virtuellen Umgebung für den Außenstehenden visualisiert. Dem Außenstehenden wird die Szene und sein Mitnutzer auf einem Bildschirm dargestellt. Dem Nutzer in der immersiven virtuellen Umgebung wird der Außenstehende innerhalb der virtuellen Szene visualisiert. Dabei wird eine neuartige Methode zur Erkennung, Segmentierung und Visualisierung der menschlichen Texturen in einer virtuellen Realität vorgeschlagen. Anhand einer prototypischen Implementierung und eines Anwendungsfalls wird gezeigt, wie das vorgeschlagene Konzept zur gegenseitigen visuellen Kommunikation zwischen realer und virtueller Umgebung unter ausschließlicher Nutzung von kostengünstiger, semi-professioneller Technologie umsetzbar ist.

Zusätzlich zur Darstellungsmethode und dem Prototyp wird der Aspekt der Modularität des entstandenen Systementwurfs thematisiert. In der Beschreibung der modularen Architektur werden besonders Aspekte der Kapselung von Funktionalität aufgegriffen. Somit kann das Grundkonzept der Kommunikation weitgehend unabhängig von Einsatzgebiet und Technik genutzt werden. Es wird außerdem ermöglicht einzelne Module des Systems auszutauschen, da die Schnittstellen klar definiert sind im Systementwurf.

Anhand von einschlägiger Literatur wird gezeigt, dass die visuelle Kommunikation zwischen realen und virtuellen Umgebungen durch das vorgeschlagene Konzept erweitert wird. Mithilfe eines Experiments wird anhand des Prototyps dann abschließend gezeigt, dass das vorgeschlagene Kommunikationskonzept die Präsenz der Nutzer von immersiven virtuellen Umgebungen nicht negativ beeinflusst, womit der Nutzen für ko-lokalierte Mehrbenutzerumgebungen verdeutlicht wird.

ABSTRACT

ABSTRACT

Over the course of digitization of all areas of daily life, Virtual Reality technologies find application in a vast amount of different research and industry sectors. Emerging solutions for the semi-professional sector involve even consumers to utilize them. Current domains of application are psychological oriented experiments, conference situations or social media among others. One of the core aspects which connects those and other areas is the mutual communication between humans. At that, the visual aspect is the most crucial one, because the major part of a communication is transported visually.

In this master's thesis we propose a concept which allows to visually represent individual persons for both the user which is within an immersive virtual Reality and stakeholders outside of the virtual environment. These representations are visualized with respect to the virtual world. We focus on augmenting the visual communication for persons which are located in the same physical room. For that we generally do not make use of animated avatars. Instead we combine different computer-vision based methods for extraction of the realistic human texture and concurrently propose a novel one that fits in very well into present consumer Virtual Reality technology. In comparison to other representation techniques we support by that a high degree of realism in body language and other visual communication media, like individual clothes.

In addition to the suggested theoretical concept of a realistic mutual representation, a prototype implementation shows the feasibility and usability of our work, as well as the integration into existing virtual environment. We also focused on a modular engineering architecture, so that distinct modules can easily be exchanged depending on the Virtual Reality setup. Altogether, the user outside the immersive virtual environment can view his immersed joint user on a screen, embedded in the virtual scenery. The immersed user however is enabled to see the non-immersed one realistically integrated within the virtual environment.

By reference to relevant literature, it is shown that visual communication between real and virtual environments is augmented by the proposed concept. With the aid of the elaborated prototype it is shown in an experiment, that this concept does not negatively affect the presence of users within virtual environments. As a result, the benefit for co-located multi-user environments is illustrated.

STICHWÖRTER / KEY WORDS

Virtual Reality, Avatar, Human-Computer Interaction, Computer Vision, Image Processing

ABBILDUNGSVERZEICHNIS

ABBILDUNGSVERZEICHNIS

ABBILDUNG 1.4-1: ABLAUF DES UCD FREI NACH ISO 9241-210.	6
ABBILDUNG 2.1-1: REALITY-VIRTUALITY-KONTINUUM (MILGRAM ET AL., 1995).....	8
ABBILDUNG 2.1-2: KLASSEKATIONSSCHEMA FÜR VR-AUSGABEGERÄTE (ANTHES ET AL., 2016). ZUSÄTZLICH WURDEN GERÄTE MARKIERT, WELCHE ALS KONSUMENTENGEIGNET ANGESEHEN WERDEN.	10
ABBILDUNG 2.2-1: LINKS: GEMEINSAMER GETEILTER RAUM. RECHTS: SICHTFELD EINES NUTZERS, WOBEI DIE VIRTUELLEN OBJEKTE DIE VISUELLE KOMMUNIKATION BEEINTRÄCHTIGT.	11
ABBILDUNG 2.2-2: a) REALITÄT: EIN BUCH, ALS REALES OBJEKT, KANN TRADITIONELL GELESEN WERDEN. b) MR: DURCH EIN TRAGBARES DISPLAY KÖNNEN SZENEN AUS DEM BUCH AUS VERSCHIEDENEN PERSPEKTIVEN BETRACHTET WERDEN, OHNE DABEI REALE RÄUMLICHE INFORMATIONEN ZU VERLIEREN. DIE SZENE KANN HIER EBENSO BETRACHTET WERDEN WIE BEISPIELSWEISE EIN ZWEITER NUTZER. c) VR: DA DIE NUTZER DIE PERSPEKTIYE SELBST VERÄNDERN KÖNNEN, KÖNNEN SIE EBENFALLS SO NAH AN DIE VIRTUELLE SZENE HERANGEHEN, BIS AUSSCHLIESSLICH VIRTUELLE INFORMATIONEN ZU SEHEN SIND.	12
ABBILDUNG 2.2-3: EIN MOCK-UP EINER MR-VULKANSIMULATION DIE MEHRERE NUTZER AUF VERSCHIEDENE WEISEN EINBINDET, AUS ROO UND HACHET (2017).	13
ABBILDUNG 2.3-1: EINORDNUNG VON NUTZERDARSTELLUNGEN IN VE ANHAND DER REALITÄTSNÄHE IN DAS REALISMUS-METAPHER-AVATARKONTINUUM.....	14
ABBILDUNG 2.3-2: AVATAR-REPRÄSENTATION VON NUTZERN IN EINEM MULTI-USER-ENVIRONMENT. (BILLINGHURST, KATO & POUPRYEV, 2001).....	16
ABBILDUNG 2.3-3: LINKS: BEISPIELHAFTER AVATAR NACH PANDZIC ET AL. (2001). RECHTS: ECHTZEIT-STEUERUNG EINES AVATARS NACH HAGEN (2017).....	17
ABBILDUNG 2.3-4: LINKS: REALISTISCHE 2D-ÜBERBLENDUNG UND SCHATTIERUNG NACH POMI UND SLUSALLEK (2005). RECHTS: REALISTISCHE 2D-VERDECKUNGSBERECHNUNG NACH ZHU ET AL. (2016).....	18
ABBILDUNG 2.3-5: LINKS: MESH-DARSTELLUNG EINES AUßenSTEHENDEN NUTZERS NACH NORDBY, GERNEZ UND BØRRESEN (2016). RECHTS: VOXEL-DARSTELLUNG EINES AUFGENOMMENEN AUßenSTEHENDEN UND EINES ECHTZEIT VR- NUTZERS NACH REGENBRECHT ET AL. (2017).....	19
ABBILDUNG 2.4-1: LINKS: TRADITIONELLE VISUELLE KOMMUNIKATION. RECHTS: SCHEMATISCHE VERÄNDERUNG DER VISUELLEN KOMMUNIKATION DURCH VR. (KRUEGER, 1991)	21
ABBILDUNG 2.4-2: KOLLABORATIVES KONFERENZ SYSTEM. EIN VR-NUTZER KOMMUNIZIERT DADURCH MIT AUßenSTEHENDEN NUTZERN, WELCHE IM VE REALISTISCH REPRÄSENTIERT WERDEN. (KATO & BILLINGHURST, 1999)	23
ABBILDUNG 2.5-1: HAAR-ÄHNLICHE MERKMALE DIE IN HAAR-CASCADE-CLASSIFIERN GENUTZT WERDEN. ((VIOLA & JONES, 2001)	25
ABBILDUNG 2.5-2: OBEN: ITERATIVE VORHERSAGE DES RECHTEN ELLENBOGENS EINER CONVOLUTIONAL POSE MACHINE. MITTE: VORHERSAGE DES RECHTEN UNTERARMS MIT PART AFFINITY FIELDS VON MEHREREN PERSONEN. UNTER: ERGEBNIS DER KOMBINATION VON PART AFFINITY FIELDS UND CONVOLUTIONAL POSE MACHINE WIE IN DER ANWENDUNG OPENPOSE. (WEI, RAMAKRISHNA, KANADE & SHEIKH, 2016) (CAO, SIMON, WEI, SHEIKH, 2017)	26
ABBILDUNG 2.5-3: GRABCUT BEISPIELE. (ROTHER, KOLMOGOROV & BLAKE (2004)	27
ABBILDUNG 2.5-4: LINKS: AUF BASIS EINES EINZELNEN TIEFENBILDES WERDEN PERSONEN ERKENNT, EINE BESTIMMUNG DER ZUGEHÖRIGEN KÖRPERTEILE WIRD VORGENOMMEN, UND ES WERDEN 3D-POSITIONEN DER GELENKE BESTIMMT. RECHTS: BEISPIELHAFTE DARSTELLUNG EINES ENTSCHEIDUNGSWALDES. (SHOTTON ET AL., 2013).....	28
ABBILDUNG 2.6-1: DARSTELLUNG DER MODULARITÄT EINES VR-ARBEITSPLATZES. (MULDER & BOSKER, 2004)	30
ABBILDUNG 3.1-1: DIE GEPLANTE OPTIK DER GEGENSEITIGEN DARSTELLUNG ZUR KOMMUNIKATION ZWISCHEN REALEN UND VIRTUELLEN UMGBÜNDUNGEN IST VON HOHEM REALISMUS GEPRÄGT. DURCH DIE ECHT-TEXTUR-REPRÄSENTATION WERDEN VISUELLE ASPEKTE, WIE DIE KÖRPERSPRACHE, BESONDERS REALISTISCH ABGEBILDET.....	37
ABBILDUNG 3.1-2: PROZESS DER ANFORDERUNGSERHEBUNG NACH KOTONYA UND SOMMERVILLE (1998).	38
ABBILDUNG 3.2-1: ABSTRAKTER AUFBAU DES SYSTEMS MIT DEM FOKUS AUF MODULARE KAPSELUNG DER FUNKTIONALITÄT. DIESES SUB-SYSTEM BILDET DIE DARSTELLUNG DES PROBANDEN UND DES VE FÜR DEN EXPERIMENTATOR (ITERATION 1) AB.	40
ABBILDUNG 3.2-2: SUB-SYSTEM, DAS DIE DARSTELLUNG DES EXPERIMENTATORS FÜR DEN PROBANDEN INNERHALB DES VE (ITERATION 2) BEREITSTELLT.....	41
ABBILDUNG 3.3-1: DEFINITION DER BEOBACHTUNGSRICHTUNGEN UND DEM VERHÄLTNIS VON BEOBACHTER/EXPERIMENTATOR ZU NUTZER/PROBAND IN EINEM IMMERSIVEN VE.....	43
ABBILDUNG 3.3-2: DIE MODELLIERUNG DER KOMPOSITIONELLEN STRUKTUR (NACH FMC) DER BEOBACHTUNGSRICHTUNG VON EXPERIMENTATOR ZU PROBAND ALS BLOCKDIAGRAMM. DAS SUB-SYSTEM SPIEGELT DAS ERGEBNIS DER ERSTEN ITERATION	

ABBILDUNGSVERZEICHNIS

WIEDER. DIE ROTE FORM IST DIE SYSTEMGRENZE. ALLES INNERHALB DIESER GRENZE BEINHALTET DIE FOKUSSIERTE FUNKTIONALITÄT UND IST UMFASSEND MODELLIERT. DIAGRAMMKOMPONENTEN AUßERHALB DIESER GRENZE BEINHALTEN ZU VIEL UNBEEINFLUSSBARE, EXTERNE FAKTOREN, ALS DAS DIESSE UMFASSEND MODELLIERT WERDEN KÖNNEN.	45
ABBLUDUNG 3.3-3: DAS SUB-SYSTEM DER ERSTEN ITERATION ALS PETRI-NETZ SPIEGELT DEN ABLAUF WIDER.	46
ABBLUDUNG 3.3-4: DAS ENTITY-RELATIONSHIP-DIAGRAMM, DAS DEN FUNKTIONSUMFANG FÜR DIE DARSTELLUNG DES PROBANDEN UND DES VE FÜR DEN EXPERIMENTATOR BESCHREIBT.	47
ABBLUDUNG 3.3-5: DAS STRUKTURDIAGRAMM DER ZWEITEN ITERATION. DIE FUNKTIONALITÄT DER BEOBSCHAUUNG VON PROBAND ZU EXPERIMENTATOR IST WEITGEHEND GETRENNNT VON DER UMGEKEHRTE BEOBSCHAUUNGSRICHTUNG AUS DER ERSTEN ITERATION. DIE SYSTEMGRENZE DES SU-SYSTEMS IST IN GRÜN DARGESTELLT.	48
ABBLUDUNG 3.3-6: DAS PETRI-NETZ FÜR DIE FUNKTIONALITÄT DIE BEI DER ZWEITEN ITERATION HINZUGEKOMMEN IST VERDEUTLICHT DIE ÄHNLICHKEIT ZUM SUB-SYSTEM DER ERSTEN ITERATION.	49
ABBLUDUNG 3.3-7: DAS ENTITY-RELATIONSHIP-DIAGRAMM FÜR DIE BEOBSCHAUUNGSRICHTUNG VON PROBAND ZU EXPERIMENTATOR ZEIGT DIE WERTEBEREICHSSSTRUKTUR AN.	50
ABBLUDUNG 3.3-8: DAS BLOCKDIAGRAMM DES GESAMTSYSTEMS UMFAßT SOWOHL DIE FUNKTIONALITÄT AUS DER ERSTEN, ALS AUCH AUS DER ZWEITEN ITERATION. HIER WIRD BESONDERS DIE KOMMUNIKATION ZWISCHEN DIESEN HERVORGEHOBEN.	51
ABBLUDUNG 3.3-9: <i>LINKS</i> : DIE ERSTE, NAIVE STRUKTUR DES SYSTEMS. <i>RECHTS</i> : DIE DURCH SYSTEMISCHE MODELLIERUNG BEGRÜNDETE NEUE ARCHITEKTUR.	52
ABBLUDUNG 3.3-10: ABLAUFDIAGRAMM DES GESAMTSYSTEMS, VISUALISIERT ALS FMC-PETRI-NETZ.	53
ABBLUDUNG 3.3-11: MODELLIERUNG DER WERTE-STRUKTUR DES GESAMTSYSTEMS.	56
ABBLUDUNG 3.3-12: DAS FLUSSDIAGRAMM BESCHREIBT DEN ABLAUF UND DAS ZUSAMMENSENDELN DES NATIVE PLUG-INS ZUM TEXTUR READBACK MIT DER UNITY INSTANZ.	61
ABBLUDUNG 3.3-13: DER UNITYSEITIGE ABLAUF ALS FLUSSDIAGRAMM ZEIGT DIE HANDHABUNG MEHRERER TEXTUREN, MIT DER UNSICHERHEIT WANN DIE JEWELIGE TEXTUR FERTIG FÜR DEN ZUGRIFF IN UNITY IST.	62
ABBLUDUNG 3.3-14: <i>A</i>) UNITYPROFILER FENSTER <i>B</i>) INTEGRIERTES STATISTICS FENSTER <i>C</i>) EIGENS IMPLEMENTIERTES STATISTICS 2.0 FENSTER, WELCHES INFORMATIONEN ZU DEN ALTERNATIVEN BILDRATEN UND RENDERVORGÄNGEN GIBT.	63
ABBLUDUNG 3.3-15: PROBLEMATIK DER EXTRAPOLATION VON TIEFENINFORMATIONEN ANHAND EINER TRIVIALEN, INJEKTIVEN ABBLUDUNG DER FARBPXEL AUF TIEFENPIXEL. FARBPXEL OHNE TIEFENWERT SIND ROT DARGESTELLT. FARBPXEL MIT TIEFENINFORMATIONEN SIND IN SCHWARZ-BLAU-ABSTUFUNGEN DARGESTELLT. GROÙE ROTE BEREICHE SIND ZURÜCKZUFÜHREN AUF NICHT SICHTBARE BEREICHE DER TIEFENKAMERA AUF GRUND MEHRERER ZENTIMETER VERSATZ DER PHYSISCHEN KAMERAS.	67
ABBLUDUNG 3.3-16: <i>LINKS</i> : MASKIERTES AUSGABEBILD DES SZENEN-ÜBERSICHT-KINECT-MODULS. <i>RECHTS</i> : VISUALISIERUNG VERGLEICHBARER TIEFENDATEN.	67
ABBLUDUNG 3.3-17: <i>LINKS</i> : UNITY-KAMERA MIT VERTIKALEM FOV VON 60°. <i>RECHTS</i> : SELBE KAMERAPosition MIT GEÄNDERTEM VERTIKALEM FOV VON 100°.	68
ABBLUDUNG 3.3-18: <i>OBEN LINKS</i> : FINALES COMPOSING DES SZENEN-ÜBERSICHT-COMPOSITOR-MODULS. <i>OBEN RECHTS</i> : FOKUS AUF PRÄZISE VERDECKUNGSBERECHNUNG DURCH EINBEZIEHUNG VON 3D-DATEN. <i>UNTEN</i> : EINORDNUNG DER AVATARREPRÄSENTATION IN DAS REALISMUS-METAPHER-AVATARKONTINUUM.	71
ABBLUDUNG 3.3-19: ABLAUF DER POV-VISUALISIERUNG UNTER NUTZUNG DES BOUNDING-BOX-ANSATZES UND DES TIEFENCOMPOSITINGS.	73
ABBLUDUNG 3.3-20: ÜBERBLICK ÜBER DAS DIY-RIG FÜR DIE KINECT ZUR BEFESTIGUNG DER KINECT AM KOPF DES PROBANDEN.	75
ABBLUDUNG 3.3-21: <i>LINKS</i> : ZUGRUNDELIEGENDES BILD DER KALIBRIERUNG. <i>MITTE</i> : HERVORGEHOBEN GRÜN: EINBILDUNG DER TEXTUR EINES REALEN TRACKERS, ROT: DARSTELLUNG EINES DIGITALEN MODELLS DES TRACKERS IN DER SZENE ALS GROUNDTRUTH. ZU BEOBACHTEN SIND INSBESTONERE UNTERSCHIEDE, DIE SICH AUF DIE VERSCHIEDENEN BETRACHUNGSWINKEL VON AUGEN ZU KINECT ZURÜCKFÜHREN LASSEN. <i>RECHTS</i> : DER POV VON EINEM (!) AUGE IN RELATION ZUM SICHTFELD, DAS DIE KINECT SIMULIEREN KANN (IN GRÜN).	76
ABBLUDUNG 3.3-22: BEISPIELFALL FÜR FALSCH-NEGATIVES RENDERING. DA DIE TEXTURIERTE EBENE RÄUMLICH HINTER DEM OBJKT PLATZIERT IST, KÖNNEN DIE PIXEL DES AUSGESTRECKTEN ARMS NICHT VOR DAS OBJKT GERENDERT WERDEN. <i>LINKS</i> : FALSCH NEGATIVES BEISPIEL. <i>RECHTS</i> : RICHTIG POSITIVES BEISPIEL.	77
ABBLUDUNG 3.3-23: <i>OBEN LINKS UND RECHTS</i> : DIE FINALE VISUALISIERUNG EINES EXPERIMENTATORS INNERHALB DES VE DES PROBANDEN. <i>UNTEN</i> : EINORDNUNG DER AVATAR-DARSTELLUNG IN DAS REALISMUS-METAPHER-AVATARKONTINUUM.	79
ABBLUDUNG 3.3-24: WEITERE GETESTETE DARSTELLUNGSMÖGLICHKEITEN FÜR REALISTISCHE MENSCHREPRÄSENTATIONEN IN VR. <i>OBEN</i> : PUNKTWOLKE (BILLBOARD-EFFEKT) (<i>LINKS</i>), REFERENZBILD (<i>RECHTS OBEN</i>) UND SKELETTSCHEITZUNG DER KINECT API (<i>RECHTS UNTER</i>). <i>UNTEN</i> : REFERENZBILD (<i>LINKS</i>) UND MESHREPRÄSENTATION EINES SELBSTAVATARS AUS DEM POV EINES NUTZERS (<i>RECHTS</i>).	80

ABBILDUNGSVERZEICHNIS

ABILDUNG 3.3-25: BEISPIELHAFTE SKALA FÜR DIE UNTERScheidung ZWISCHEN VERSCHIEDENEN NUTZERN ANHAND DES IMMERSIONSVERHÄLTNISSES. DIE GRENZE ZWISCHEN VERSCHIEDENEN IMMERSIONSGRADEN IST FLIEßEND.	82
ABILDUNG 3.4-1: <i>LINKS</i> : GETRIEBELAGERMODELL, <i>Mitte</i> : KAROSSERIEMODELL, <i>RECHTS</i> : ROBOTERMODELL.	83
ABILDUNG 3.4-2: DARSTELLUNG DES URSPRÜNGLICHEN ABLAUFES DER SIMULATIONSSZENE.....	83
ABILDUNG 3.4-3: VEREINFACHTE DARSTELLUNG DES FÜR VR ANGEPASSTEN ABLAUFES.	83
ABILDUNG 4.1-1: KASTENGRAFIKEN DER EINZELNEN UNTERKATEGORIEN SP, EXP, INV UND PRES.	90

TABELLENVERZEICHNIS

TABELLENVERZEICHNIS

TABELLE 1.2-1: ZIELE UND MAÙE DER THESIS.....	4
TABELLE 2.3-1: EINORDNUNG DER BESCHRIEBENEN ARBEITEN IN „DARSTELLUNG DER NUTZERGRUPPE“ UND „REALISMUS-METAPHER-AVATARKONTINUUM“	20
TABELLE 3.3-1: ZUGRIFF AUF RENDERING-DATEN NACH MICROSOFTS DIRECT3D DOKUMENTATION.	60
TABELLE 3.3-2: EINORDNUNG DES RENDERINGS NACH FALSCH/RICHTIG NEGATIV/POSITIV.....	77
TABELLE 4.1-1: ANALYSE DES MITTELWERTUNTERSCHIEDS DER BEIDEN POPULATIONEN BEZÙGLICH DER PRÄSENZ.	89
TABELLE 4.1-2: GEGENÜBERSTELLUNG DER FINALEN ERGEBNISSE DER ZWEI POPULATIONEN: EXPERIMENTALGRUPPE UND KONTROLLGRUPPE.	89

FORMELVERZEICHNIS

FORMEL 3.3-1: SYNCHRONISATIONSDATEN IN BYTE.....	52
FORMEL 3.3-2: POV-TEXTUR IN BYTE.....	52
FORMEL 3.3-3: DIE ABHÄNGIGKEIT DER UMSETZUNG EINER ABWEICHENDEN BILDRATE \mathbf{w} AN DIE RENDER- UND VERARBEITUNGSZEIT \mathbf{z} UND DIE GEWÜNSCHTE FPS ANZAHL \mathbf{x} WERDEN ANHAND DIESER FORMEL MIT DER COROUTINE UND <i>WAITFORSECONDSREALTIME</i> IMPLEMENTIERT.....	63
FORMEL 3.3-4: DIE ABWEICHENDE BILDRATE \mathbf{w} IST MIT DER <i>INVOKEREPEAT</i> IMPLEMENTIERUNG NUR VON EINER EINZIGEN VARIABLE \mathbf{x} ABHÄNGIG.....	64
FORMEL 3.3-5: 1D REPRÄSENTATION EINES 3D ARRAYS.....	66
FORMEL 3.3-6: BERECHNUNG DES HORIZONTALEN WERTES DES FOV \mathbf{H} ANHAND DES VERTIKALEN WERTES \mathbf{V} UND DES SEITENVERHÄLTNISSES, BZW. HÖHE \mathbf{h} UND BREITE \mathbf{b} DES VIEWPORTS DER KAMERA NACH AKTUELLER MICROSOFT DIRECTX DOKUMENTATION. \mathbf{V} UND \mathbf{H} SIND IN RADIANT ANGEgeben.....	69

1. EINLEITUNG UND MOTIVATION

Die jüngsten technologischen Entwicklungen machen es möglich, Techniken aus dem Fachgebiet „Virtual Reality“ (VR) in ein breites Spektrum von Anwendungsfällen anderer Forschungs- und Industriezweige zu übertragen. Psychologie, Wirtschaftswissenschaften und soziale Medien sind nur ein kleiner Auszug der großen Bandbreite an Möglichkeiten. Egal ob in Experimenten, in Verhandlungssituationen oder in gemeinsamen virtuellen Verabredungen – Menschen und deren Kommunikation sind hier klarer Mittelpunkt des Geschehens. Die gegenseitige Wahrnehmung und Kommunikation zwischen den Nutzern von VR-Systemen ist daher eine der grundlegendsten Funktionen, die zukünftige VR-Anwendungen bieten müssen. Dies gilt nicht nur für ganzheitlich-verteilte virtuelle Mehrbenutzerumgebungen, sondern fängt bereits bei fundamentalen Gegebenheiten an. Als Beispiel, welches in Kürze noch näher beschrieben wird, ist hier eine Experimentator-Proband-Beziehung zu nennen. Beide direkten und indirekten Nutzer des immersiven VR-Systems sind im selben physischen Raum, können jedoch nicht auf direktem Wege miteinander visuell kommunizieren.

Da die benötigten Hardware-Komponenten immer günstiger werden und zugleich freie proprietäre Software, wie Game Engines, Computer-Aided-Design-Programme (CAD) und kostenlose Software Development Kits (SDKs) angeboten werden, gibt es derzeit eine Fülle an Möglichkeiten, um solche virtuellen Umgebungen (virtual environments, VE) zu erstellen. Trotz kostengünstiger Systeme werden diese meist als Spezialsysteme angefertigt und bieten damit eine Lösung für genau einen spezifischen Anwendungsfall.

1.1 Problembeschreibung

Die Problematik, die von der vorliegenden Master-Thesis behandelt wird, wird vorerst anhand eines einfachen und konkreten Szenarios der Ist-Situation aufgezeigt. Danach wird eine darauf aufbauende Soll-Situation geschildert, welche die Ist-Situation des Szenarios verbessert. Anhand dieser beiden Beschreibungen wird die zugrundeliegende Problematik dann generalisiert. Dabei entsteht ein Kontext, der die wissenschaftliche Relevanz und Basis der Thesis aufzeigt. Anhand des Kontexts werden im Verlauf dieses Dokuments zu einem späteren Zeitpunkt ebenfalls die Ergebnisse gemessen.

1.1.1 Ist-Situation

Das folgende Anwendungsszenario wurde methodisch an Adolph, Cockburn und Bramble (2002) angelehnt. Das Szenario wurde inhaltlich aus beschriebenen Situationen aus durchgeföhrten Anforderungs-Interviews erstellt.

Experimentsituations

Anwendungsbereich: Wissenschaftliches Experiment, durchgeführt an Instituten wie einer Hochschule

Nebenumstand: Immersive virtuelle Umgebungen werden für das Experiment genutzt

Akteure: Ein Experimentator außerhalb und ein Proband innerhalb einer immersiven virtuellen Umgebung

Voraussetzung: Ein Problem tritt während des Experiments auf

EINLEITUNG UND MOTIVATION

Auslöser: Der Proband findet einen virtuellen Gegenstand nicht / kann sich in der virtuellen Umgebung nicht richtig orientieren

Problem-Szenario:

1. Der Proband erbittet Hilfe vom Experimentator und dreht dabei seine Blickrichtung zur geschätzten Position des Experimentators. Er zeigt dabei auf einen bestimmten Bereich innerhalb der virtuellen Umgebung. Der Experimentator, welcher in der Nähe des Probanden ist, kann die visuelle Geste nicht räumlich zuordnen und sieht ebenfalls nicht, wo genau sich der Proband in der virtuellen Umgebung befindet.
2. Nach kurzem Nachschauen auf dem VR-Computer versucht er den Ort des Gegenstands verbal zu beschreiben. Der Proband kann der Beschreibung nicht folgen, ist verunsichert und nimmt das Head-Mounted Display (HMD) ab.
3. Der Experimentator muss wieder zu ihm laufen, schaut kurz durch das HMD und zeigt ihm dann im realen Raum die Position.
4. Der Proband setzt das HMD wieder auf, muss sich erstmal wieder neu orientieren und findet durch die visuelle Hilfestellung des Experimentators letztendlich den Gegenstand. Das Experiment kann ab diesem Punkt wie gewohnt weitergeführt werden.

Stakeholder und Interessen:

Experimentator: Will Probleme lösen, die Immersion und die Präsenz als Vorteil von VR-Experimenten jedoch nicht stören.

Proband: Will sich auf die virtuelle Umgebung und das Experiment konzentrieren. Möchte gleichzeitig aber die Sicherheit haben Hilfestellungen vom Experimentator zu bekommen.

Weitere Experimentatoren die nicht mit der Probandenbetreuung beauftragt sind:
Dokumentieren der Ergebnisse und Bedienung der Technik

1.1.2 Soll-Situation

Abgesehen von der Tatsache, dass idealerweise kein Problem auftreten sollte, unterscheidet sich die Soll-Situation von der Ist-Situation in folgenden Aspekten:

1. Der Proband sollte um die Position des Experimentators im virtuellen und physischen Raum wissen, um nicht noch weiter verunsichert zu werden. Der Experimentator sollte ebenfalls um die Position des Probanden im Raum wissen, um so die Gesten direkt deuten zu können.
2. Der Experimentator sollte direkt auf die Fragen antworten können und nicht erst umherlaufen müssen. Der Experimentator sollte befähigt sein, das visuelle Problem ebenfalls visuell zu lösen (beispielsweise ebenfalls per Geste). Dafür sollte er wissen, wo er sich selbst in Relation zum virtuellen Raum befindet.
3. Die Immersion und die Präsenz sollten nicht gestört werden.
4. Der Proband sollte sich nicht neu orientieren müssen.

1.1.3 Definition der Problematik

Die Diskrepanzen zwischen der Ist- und der Soll-Situation des geschilderten Szenarios sind an dieser Stelle bereits klar zu erkennen. Die generelle Problematik die hier aufgezeigt wird entsteht vor allem auf Grund eines mangelhaften visuellen Kommunikations- und Darstellungskonzepts. Wegen des Aufkommens des Problems stellt der Proband eine Frage an den Experimentator. Dieser wird dadurch implizit auch zum aktiven Benutzer des Systems

EINLEITUNG UND MOTIVATION

und damit auch ein Teil dieser virtuellen Umgebung. Die virtuelle Umgebung des Experiments ist jedoch nur auf einen Nutzer ausgelegt, weswegen auch keine Möglichkeit zur gegenseitigen Kommunikation besteht.

Obwohl dieses konkrete Beispiel ein Spezialfall ist, da es mehrere Nutzer im selben Raum betrachtet, ist das zugrundeliegende Problem von genereller Natur. Jedwede Art von virtuellen Mehrbenutzerumgebungen ist angewiesen auf eine gegenseitige Repräsentation der Benutzer. Die Nachhaltigkeit eines solchen Kommunikations- und Darstellungskonzepts ist daher ein weiterer Teil dieser Problematik. Die Konzeption und Implementierung für eine einzige spezifische Mehrbenutzergabe hat zur Folge, dass für jede weitere Umgebung ebenfalls hohe Kosten entstehen.

Die zu behandelnden Hauptprobleme werden wie folgt definiert:

1. Es fehlt ein Konzept und eine Umsetzung für die gegenseitige visuelle Kommunikation und Darstellung in virtuellen Mehrbenutzergaben.
2. Durch die verschiedenen Anwendungsbereiche und -techniken von virtuellen Umgebungen ist eine Speziallösung nicht nachhaltig.

1.2 Zielsetzung

Die aus den beschriebenen Ist- und Soll-Situationen entstandene Diskrepanz wird genutzt, um die Ziele für die vorliegende Master-Thesis herzuleiten. Außerdem wird zu jedem Ziel auch die Art des Maßes angegeben, mit dem die Erfüllung des Ziels gemessen wird.

Das wichtigste Ziel der Thesis ist selbstredend die Erstellung eines Systems, das zur Verbesserung der Wahrnehmung und Kommunikation zwischen den verschiedenen Benutzern einer virtuellen Umgebung beiträgt. Dieses Konzept soll vor allem den visuellen Aspekt unterstützen, da etwa 55% der Wahrnehmung anderer Personen nonverbal stattfindet (Mehrabian & Wiener, 1967). Mögliche non-verbale Kommunikationsaspekte sind zum Beispiel die Körpersprache, der Gesichtsausdruck, Gesten oder auch die Körperhaltung und -orientierung (Mehrabian & Ferris, 1967).

Um das Konzept zu bestätigen und auf Machbarkeit zu überprüfen, wird dieses prototypisch umgesetzt. Aus Zeit- und Kostengründen wird das Konzept zwar unabhängig von spezifischer VR-Technik und Anwendungsbereich geplant, jedoch wird sich der Prototyp selbst auf einen konkreten Anwendungsfall beziehen. Ebenfalls aus Kostengründen wird sich der Mehrbenutzer-Aspekt auf die Beziehung zwischen Experimentator und Proband in einer Experimentsituation beschränken (ähnlich wie in Kapitel 1.1.1). Es ist also nur einer der Nutzer in der eigentlichen immersiven virtuellen Umgebung platziert. Der Kommunikationskanal wird damit zwischen realer und virtueller Umgebung stattfinden. Beide Nutzer befinden sich im selben physischen Raum. Insgesamt ist somit auch ein gewisser „low-budget“-Gedanke in dem Erstellungsprozess verankert. Dieser ist nicht nur pragmatisch im Rahmen der Master-Thesis von Vorteil, sondern auch auf dem universellen Einsatzgebiet von VR-Techniken begründet. Mögliche prosumentenorientierte (professionelle Konsumenten, semi-professionell) Anwendungsbereiche werden somit nicht ausgeschlossen.

Das nächste Ziel ist unter dem Gesichtspunkt der Modularität des entstehenden Systems zu betrachten. Um so wenig Anwendungsbereiche wie möglich auszuschließen, sollte das System technisch und inhaltlich unabhängig sein. Dies ist zwar der Idealfall, jedoch in der Realität nur begrenzt möglich. Deswegen wird in der prototypischen Implementierung darauf geachtet, dass Kapselung genutzt wird. So werden einzelne Komponenten des Prototyps für weitere Projekte wiederverwendbar. Dieses Ziel wird durch eine Modellierung der

EINLEITUNG UND MOTIVATION

Systemarchitektur auf hoher Ebene bestätigt. Insgesamt gilt für das Projekt: So generell wie möglich, so spezifisch wie nötig! Bestimmte Schwierigkeiten und Abhängigkeiten des Konzepts und der Implementierung werden herausgestellt, um in zukünftigen Arbeiten berücksichtigt zu werden.

Zusätzlich kann aus der Ist-Situation aus Kapitel 1.1.1 auch abgeleitet werden, dass die Präsenz der Nutzer insgesamt nicht gestört werden sollte. Die Kommunikation muss daher in irgendeiner Weise in Einklang mit der Präsenz, beziehungsweise der Immersion der virtuellen Umgebung sein. Dieses Ziel kann überprüft werden, indem beispielsweise das Absetzen des HMD während der Kommunikation nicht notwendig wird.

Nr.	Ziel	Maß
1	Erstellung eines Systems zur Verbesserung der Wahrnehmung und Kommunikation zwischen Benutzern von virtuellen Umgebungen.	Ein System wird vorgeschlagen (Konzeption, prototypische Umsetzung und Evaluation).
2	Visuelle Kommunikation ist möglich. (Konkret zwischen realer und virtueller Umgebung)	Ein Nutzer außerhalb der immersiven virtuellen Umgebung kann einen Nutzer innerhalb einer immersiven virtuellen Umgebung sehen – und umgekehrt.
3	System hat „low-budget“-Charakter.	Es wird vorwiegend Prosumenten-Technik genutzt.
4	System (-Komponenten) sind wiederverwendbar.	Die Kapselung der Funktionalitäten wird durch Modellierung einer modularen Architektur aufgezeigt. Schnittstellen werden durch Dokumentation des Datenflusses zwischen den einzelnen Modulen definiert.
5	Die Präsenz des Probanden in immersiver VR wird nicht gestört.	Der Proband muss zur visuellen Kommunikation nicht die virtuelle Szene verlassen (bzw. das HMD ablegen).

Tabelle 1.2-1: Ziele und Maße der Thesis.

1.3 Aufbau der Arbeit

Nachdem in der Einleitung die Motivation beschrieben, die Problematik anhand eines Anwendungsszenarios gezeigt und die Ziele für die Thesis klar definiert wurden, folgt im ersten Kapitel nun ein Überblick über den Aufbau des gesamten Thesis-Dokuments. Die Darlegung der Vorgehensweise und Methodik schließt das erste Kapitel dann ab.

In Kapitel 2 wird der Stand der Wissenschaft und der Stand der Technik bezüglich der relevanten Themengebiete detailliert beleuchtet. Grundkonzepte sowie aktuelle Projekte im Bereich der virtuellen Technologien und besonders der gemischten Realität mit mehreren Benutzern werden aufgeführt. Danach werden Darstellungsmöglichkeiten von Nutzern in virtuellen Umgebungen illustriert, gefolgt von Konzepten zur Kommunikation im Gebiet der VR. Außerdem werden aktuelle Arbeiten aus dem Gebiet der Bildsegmentierung bzw. Personenerkennung vorgestellt. Danach wird ein Überblick über Modularität und modulare

EINLEITUNG UND MOTIVATION

System-Architekturen bezüglich VR-naher Bereiche gegeben. Als letztes wird ein kurzer Einblick in mögliche Evaluationsformen von VR-Fragestellungen dargestellt.

Im Hauptteil der Thesis, Kapitel 3, werden erst konzeptionelle Aspekte beschrieben. Danach folgt das umfassendste Unterkapitel 3.3 – die technische Dokumentation des Prototyps. Auf die Beschreibung des speziellen Anwendungsfalls in dieser Thesis folgt dann die Evaluation des vorgeschlagenen Systems mittels eines Experiments als Pilot-Studie.

Im vierten Kapitel werden die Ergebnisse der Thesis analysiert und im fünften Kapitel dann anhand der festgelegten Ziele und Maße diskutiert. Eine Zusammenfassung mit Ausblick auf zukünftige Arbeiten schließt die vorliegende Master-Thesis inhaltlich ab.

1.4 Vorgehensweise und Methodik

Der gesamte Prozess der Master-Thesis ist geprägt von iterativem Vorgehen in verschiedener Granularität. Auf gröbster Ebene wird sich am Ablauf des nutzerzentrierten Designs (User-Centred Design, UCD) nach ISO 9241-210 orientiert (Abbildung 1.4-1). Außerdem wurden Teile aus de Groots empirischem Kreislauf (1969) sowie weitere ethnographische und designspezifische Verfahren (Charmaz & Belgrave, 2007; Gaver, Beaver & Benford, 2003; Findeli, 2010; Greene, Caracelli & Graham, 1989; Roberts, Headleand & Ritsos, 2016) berücksichtigt. Die Vorgehensweise könnte insgesamt als scrum-ähnlich (Schwaber & Beedle, 2002) beschrieben werden. Diese Arbeitsweise wurde besonders auf Grund der Kooperation mit dem Deutschen Forschungsinstitut für Künstliche Intelligenz (DFKI) gewählt. Direkte Nutzer für das geplante System sind hier sowohl vor Ort, als auch in weiteren Industriezweigen gut erreichbar. Die erste Iteration beinhaltet eine literaturgestützte Recherche zur möglichen Umsetzung des Systems. Dabei werden die infrage kommenden Nutzer am DFKI, in der Arbeitsgruppe Agenten und Simulierte Realität (ASR), regelmäßige miteingebunden, um den Nutzerkontext zu analysieren. Nachdem so erste Anforderungen abgeleitet werden, wird ein erster Systementwurf erarbeitet. Da sich der Kern der Arbeit um einen visuellen Aspekt dreht, wird der Entwurf dann direkt als eine Art Medium-Fidelity-Prototyp umgesetzt. So kann den Nutzern in einer kurzen Evaluationsphase ein direkter Eindruck vermittelt und frühes Feedback erhalten werden. Dieser Prototyp wird besonders in einer abschließenden Evaluation der ersten Iteration genutzt. Hier werden Interviews durchgeführt, die später qualitativ ausgewertet und analysiert werden. Somit wird nach der ersten großen Iteration auch bereits eine erste Validierung des Systems durchgeführt, um mögliche Schwierigkeiten bereits in der zweiten Iteration zu behandeln.

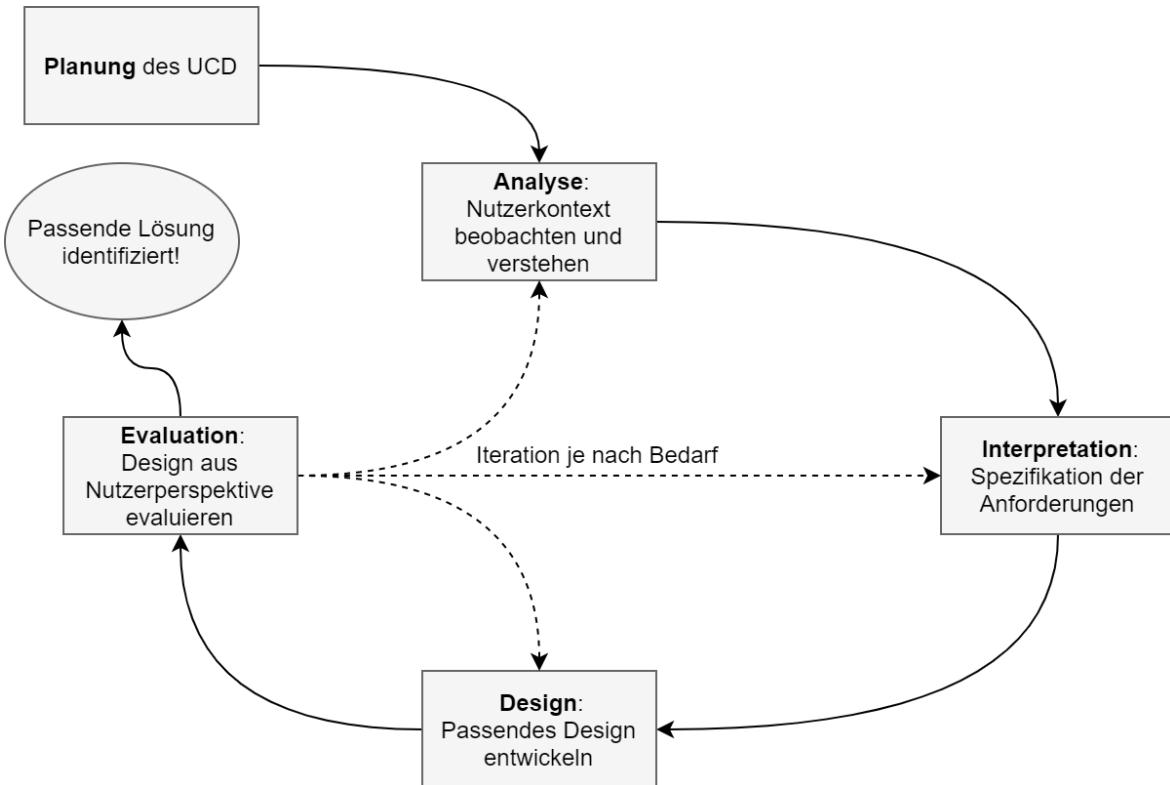


Abbildung 1.4-1: Ablauf des UCD frei nach ISO 9241-210.

Die daraus neu gewonnenen Spezifikationen werden in einer zweiten Iteration auf höchster Ebene dann im Design berücksichtigt. Die spezifischen Änderungen werden in Abschnitten aus Kapitel 3 genauer dargestellt. Das Design wird dann wiederum im bestehenden Prototyp geändert. Um detailliertere Ergebnisse aus einer ausführlicheren Evaluation am Ende des Thesis-Projekts zu bekommen, wird der Prototyp außerdem zu einem High-Fidelity-Prototyp ausgebaut. Diese Teilevaluation wird zum größten Teil mittels eines Experiments durchgeführt, zeigt aber ebenfalls technische Aspekte auf.

Ein weiterer, nicht unerheblicher Aufwand bezüglich der Vorgehensweise ist die Abstimmung mit allen anderen Stakeholdern des Projekts. Zu den potentiellen Nutzern sind hierbei auch die Betreuer am DFKI, die ASR-Gruppenleitung, aber auch die Projektleitung des Forschungsprojekts mit dem die Master-Thesis finanziert wurde zu nennen. Um eine möglichst einfache Kommunikationsstruktur zu schaffen, wird die Projektmanagement-Software Trello¹ und die Versionsverwaltungs-Software Git² zum verteilten Arbeiten genutzt.

Insgesamt ist auch zu sagen, dass in allen inhaltlichen Phasen der Thesis eine nebenläufige Literaturrecherche durchgeführt wird. Der Stand von Wissenschaft und Technik wird an dieser Stelle betrachtet, um den wissenschaftlichen Anspruch der Arbeit zu gewährleisten.

¹ www.trello.com (8/03/2018)

² www.git-scm.com (8/03/2018)

2. STAND DER WISSENSCHAFT UND TECHNIK

In diesem Kapitel wird der wissenschaftliche Kontext dargestellt, in den sich der darauffolgende Kern der Thesis einordnen lässt. Nach einer kurzen Einführung in den zeitgemäßen Stand virtueller Technologie werden besonders aktuelle Arbeiten aus verwandten Gebieten des Projekts näher beleuchtet. Anhand dieser Arbeiten wird außerdem die wissenschaftliche Abgrenzung des Projekts deutlich.

Zu den relevanten Themen wird daher der Stand der Wissenschaft und Technik betrachtet. Das in Kapitel 3 vorgestellte Konzept und der daraufhin entwickelte Prototyp bauen beide auf aktuellsten virtuellen Techniken und Umgebungen (2.1) auf. Es werden mehrere Nutzer berücksichtigt (2.2), die jeweils miteinander kommunizieren (2.4). Zur visuellen Kommunikation ist besonders die gegenseitige Darstellung der Nutzer (2.3) notwendig. Hierzu wiederum müssen durch Computer-Vision-Techniken (2.5) erst einmal die visuellen Repräsentationen der Personen erzeugt werden. Insgesamt wurde der Prototyp unter Berücksichtigung der Modularität des Systems (2.6) entwickelt, um so besonders die Wiederverwendbarkeit einzelner Komponenten zu fördern. Außerdem ist das Konzept anhand des Prototyps mit aktuellen Evaluationstechniken (2.7) evaluiert worden.

2.1 Virtuelle Techniken und Umgebungen

Virtuelle Techniken (VT) ist ein Oberbegriff und umfasst ein breites Spektrum an Fachdomänen, wie beispielsweise VE, 3D Visualisierung, Simulation, Tracking, Interaktion und viele mehr. Sie bilden die Grundlage für jede Art von VE und sind ebenfalls zu unterteilen in eine Vielzahl von einzelnen Komponenten. Diese Komponenten sind wiederum kombinierbar und ergeben in den verschiedenen Permutationen aus ihrer Menge jeweils Umgebungen mit anderen Ausprägungen. So schnell wie sich manche dieser Techniken, besonders im Bereich der VR-Hardware für den Konsumenten- und den semi-professionellen Prosumenten-Markt, in den letzten Jahren weiterentwickelt haben, so valide ist jedoch immer noch ihre Einordnung in ein bereits bestehendes Konzept – das Reality-Virtuality-Kontinuum (RV-Kontinuum) nach Milgram et al. (1995).

Die Einordnung einer vorerst abstrakten Umgebung in das RVC ist eine dieser Ausprägungen die durch VT bestimmt wird. Werden zu den reinen virtuellen Aspekten eines VE ebenfalls reale Aspekte hinzugezogen, so spricht man nicht mehr von VE, sondern von gemischten Umgebung (Mixed Environments, ME). ME sind abstufbar von Virtualität zur physischen Realität bis zu dem Punkt, an dem nur noch die reale Umgebung (Real Environment, RE) existiert. Im weiteren Verlauf wird der Begriff „Umgebungen“ sinngemäß für eine noch nicht eingeordnete reale, gemischte oder virtuelle Umgebung verwendet.

Der Zusammenhang zwischen den verschiedenen Umgebungsarten wird genauso auch auf das Verhältnis von Virtualität zu Realität selbst übertragen. Infolgedessen ist eine gemischte Realität (Mixed Reality, MR) die Realitätsform, die zwischen RE und VE existiert. In Abbildung 2.1-1 ist zudem die Unterteilung von MR in erweiterte Realität (Augmented Reality, AR) und erweiterte Virtualität (Augmented Virtuality, AV) ersichtlich, wobei in AR der reale und in AV der virtuelle Aspekt überwiegt. Die Grenzen zwischen diesen beiden Unterkategorien von MR sind fließend, sodass eine klare Einordnung in entweder die eine oder die andere Kategorie oft nicht möglich ist.

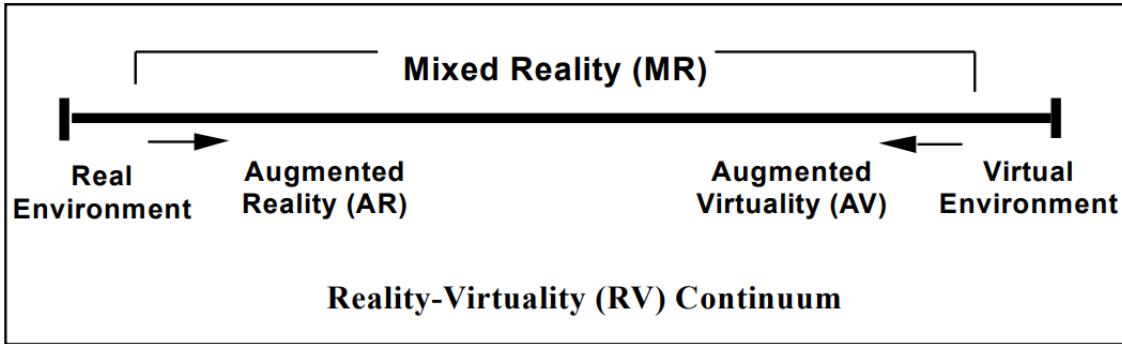


Abbildung 2.1-1: Reality-Virtuality-Kontinuum (Milgram et al., 1995)

Aufgrund der Anzahl an virtuellen Techniken und deren Unterkomponenten wird sich im weiteren Verlauf auf einen kurzen Auszug der Techniken beschränkt, die für den Bezug zu Mixed-Reality-Visualisierungssystemen und Kommunikation zwischen realen und virtuellen Umgebungen nötig sind. Hierzu zählen besonders VR-Hardware, virtuelle Objekte und virtuelle Welten. Für einen umfassenderen Einblick in das Thema besteht aktuelle Basisliteratur zum Thema VR, wie die Veröffentlichungen von Jerald (2015), Parisi (2015), Milgram und Kishino (1994), Biocca und Levy (2013) und Burdea und Coiffet (2003).

Um die Gegenwärtigkeit der anschließenden Auflistungen zu gewährleisten, wurden diese mit den aktuellen Arbeiten von Antakli et al. (2017) und Anthes et al. (2016) begründet. Mit der Auswahl dieser Literatur wurde sich nach Smith (1997) gezielt für den Aspekt der „Currency“, und zu eventuellen Lasten der „Authority“ entschieden.

VR-Hardware

Als eine der wichtigsten Hardware Komponenten wird meist das Ausgabegerät eines VR-Systems genannt. Die aktuelle Display-Technologie ist bezüglich Konsumenten-VR besonders durch HMDs repräsentiert. Haptische und multi-sensorische Geräte sind zwar ebenfalls existent, jedoch hauptsächlich auf dem Expertenmarkt von Relevanz. Die derzeit am weit verbreitetsten HMDs sind besonders HTC Vive³, Oculus – The Rift⁴ und mobile Lösungen wie Samsung Gear⁵ und Google Cardboard⁶ zu nennen. Eine Übersicht über aktuelle Ausgabegeräte ist in Abbildung 2.1-2 zu sehen. Hier ist neben der von Anthes et al. (2016) vorgeschlagenen Taxonomie für solche Technologien außerdem markiert, welche Geräte in dieser Arbeit als konsumentengeeignet angesehen werden.

Eine weitere notwendige Komponente sind Eingabegeräte zur Interaktion mit der Umgebung. Oft sind Eingabegeräte an entsprechende Ausgabehardware gebunden. Besonders häufig in Verbindung mit HMDs werden klassische Controller genutzt. Neben diskreten Eingabetechniken, wie Knöpfen, kommen auch kontinuierliche Techniken, wie Joysticks und Touchpads zum Einsatz. Zum Bereich der Eingabegeräte zählt ebenfalls Tracking in verschiedenen Formen. Dabei wird zwischen Körper- und Gestentracking unterschieden. Ersteres zielt meist auf die Schätzung der Körperpose ab, um die virtuelle Umgebung entsprechend der Körperausrichtung anzupassen. Gestentracking wird besonders zur Interaktion mit dem virtuellen Inhalt genutzt.

³ <http://www.htcvr.com/> (8/03/2018)

⁴ <https://www.oculus.com/en-us/rift/> (8/03/2018)

⁵ <https://www.oculus.com/en-us/gear-vr/> (8/03/2018)

⁶ <https://www.google.com/get/cardboard/> (8/03/2018)

Virtuelles Objekt

Virtuelle Objekte sind dreidimensionale Einzelobjekt-Repräsentationen mit denen der Nutzer eines VR-Systems interagieren kann. Da diese virtuell sind, sind es computergenerierte Objekte. Sowohl Software, wie CAD Programme (z.B. 3ds Max,⁷ Maya⁸, Blender⁹), aber auch Tiefenkameras (z.B. Kinect¹⁰, Realsens¹¹, Hololens¹²) sind Möglichkeiten für die Erzeugung von virtuellen Objekten. Es ist zu unterscheiden ob diese automatisch oder manuell generiert wurden. Außerdem kann man ein Objekt anhand seines Realitätsbezugs einordnen. Eine Fragestellung hierzu wäre beispielsweise: Wurde das 3D-Modell anhand von exakten Messungen einer realen Repräsentation erstellt, oder wurde das Objekt anhand einer skizzenhaften Zeichnung einer Idee umgesetzt?

Virtuelle Welt

Die virtuelle Welt, in der sich der Nutzer in VR wiederfindet, ist aus virtuellen Objekten aufgebaut. Infolgedessen sind virtuelle Welten ebenfalls dreidimensional und computergeneriert. Zusätzlich zur visuellen Repräsentation, die durch Bilderzeugungs- und Rendering-Verfahren auf Displays abgebildet wird, wird in einer virtuellen Welt außerdem das Verhalten und die gegenseitige Interaktion von virtuellen Objekten definiert. Hinzu kommen außerdem ein mögliches User-Interface (UI) sowie Avatare und weitere Komponenten. Zur Erstellung von virtuellen Welten für aktuelle Konsumententechnologie etablieren sich derzeit besonders Spiele-Engines. Unity3D¹³, Unreal Engine¹⁴ und CryEngine¹⁵ sind bezüglich dieser VR-Anwendungen die meistgenutzten. Der Grund ist, dass mit diesen eine einfache Integration von HMDs wie Oculus – The Rift, GearVR und HTC Vive gegeben ist, wie von Liwones (2015) und Whiting und Donaldson (2014) beschrieben.

Virtuelle Umgebungen sind infolge dieser Definitionen als eine Kombination von VR-Hardware, virtuellen Objekten, virtuellen Welten sowie deren Softwares und Anwendungen definiert.

⁷ <https://www.autodesk.de/products/3ds-max/overview> (8/03/2018)

⁸ <https://www.autodesk.de/products/maya/overview> (8/03/2018)

⁹ <https://www.blender.org/> (8/03/2018)

¹⁰ <https://developer.microsoft.com/de-de/windows/kinect> (8/03/2018)

¹¹ <https://www.intel.de/content/www/de/de/architecture-and-technology/realsense-overview.html> (8/03/2018)

¹² <https://www.microsoft.com/de-de/hololens> (8/03/2018)

¹³ <https://unity3d.com/de/> (8/03/2018)

¹⁴ <https://www.unrealengine.com> (8/03/2018)

¹⁵ <https://www.cryengine.com> (8/03/2018)

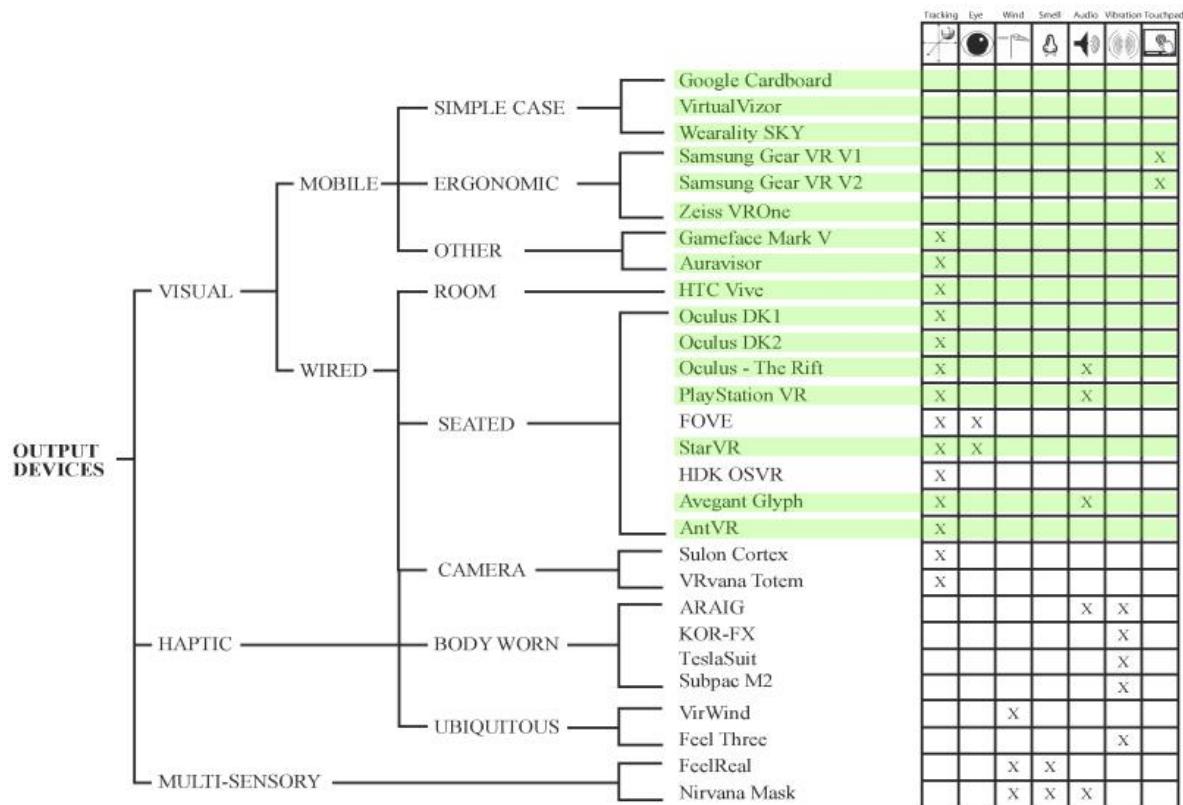


Abbildung 2.1-2: Klassifikationsschema für VR-Ausgabegeräte (Anthes et al., 2016). Zusätzlich wurden Geräte markiert, welche als konsumentengeeignet angesehen werden.

2.2 Ko-Lokalisierte Mehrbenutzer-Gemischte-Realität (Co-Located Multi-User Mixed Reality)

Für den weiteren Verlauf sind besonders aktuelle Arbeiten im Bereich der MR von hoher Relevanz, da der Fokus in Kapitel 3 dieser Thesis auf einem Mixed-Reality-Visualisierungssystem liegt. Besonders mehrere Benutzer spielen dabei eine große Rolle, sodass Grundkonzepte und Probleme einer Mehrbenutzer-Gemischten-Realität (Multi-User-Mixed Reality) betrachtet werden müssen.

Insgesamt wird bei MR-Systemen die gleiche grundlegende VR-Technik genutzt wie für gänzlich virtuelle Umgebungen. Diese werden um Komponenten erweitert, die realistische Aspekte mit in die VE einbeziehen. Hierbei ist die Einordnung von virtuellen Systemen nicht ganz trivial, da die Abgrenzung von AR zu AV fließend ist.

Eine besondere Herausforderung ist die Kategorisierung von virtuellen Systemen in das RV-Kontinuum, die mehrere Nutzer einbeziehen oder Kombinationen von verschiedenen MR-Techniken nutzen. Besonders auf Grund der rasanten Weiterentwicklung von VR-Hardware ist es durchaus sogar kostengünstig möglich, mehrere VR-Techniken zu vereinen, um so die Grenzen zwischen physischer und virtueller Welt auf mehreren Ebenen zu verblassen. Diese Thematik wird bereits in frühen Publikationen zu dem Thema behandelt, z.B. von Billinghurst und Kato (1999) und Billinghurst, Kato und Poupyrev (2001).

In ersterem wird MR als ein Mittel für kollaboratives Arbeiten betrachtet. Da in AR-nahen Systemen besonders der reale Aspekt überwiegt, wird die Kollaboration nicht durch eine zu hohe Virtualisierung der Umgebung gestört. Besonders im Bereich der „Co-Located

STAND DER WISSENSCHAFT UND TECHNIK

Collaboration“ wird ein Vorteil von MR-Systemen herausgestellt. Nutzer von VE, die sich im selben physischen Raum befinden, teilen sich hier sowohl den selben virtuellen, als auch den realen Informationsraum (Abbildung 2.2-1). Als große Bereicherung der Kollaboration wird besonders die Kommunikation genannt, da eine direkte visuelle Kommunikation der Nutzer unterstützt wird. Dieser Aspekt fällt jedoch zu Lasten der virtuellen Komponente der Umgebung. Nur wenige virtuelle Objekte können in das Sichtfeld der Nutzer eingeblendet werden. Andernfalls würde die reale Umgebung nicht mehr in Gänze wahrgenommen werden, was auch die gegenseitige reale Repräsentation der Nutzer einschließt.

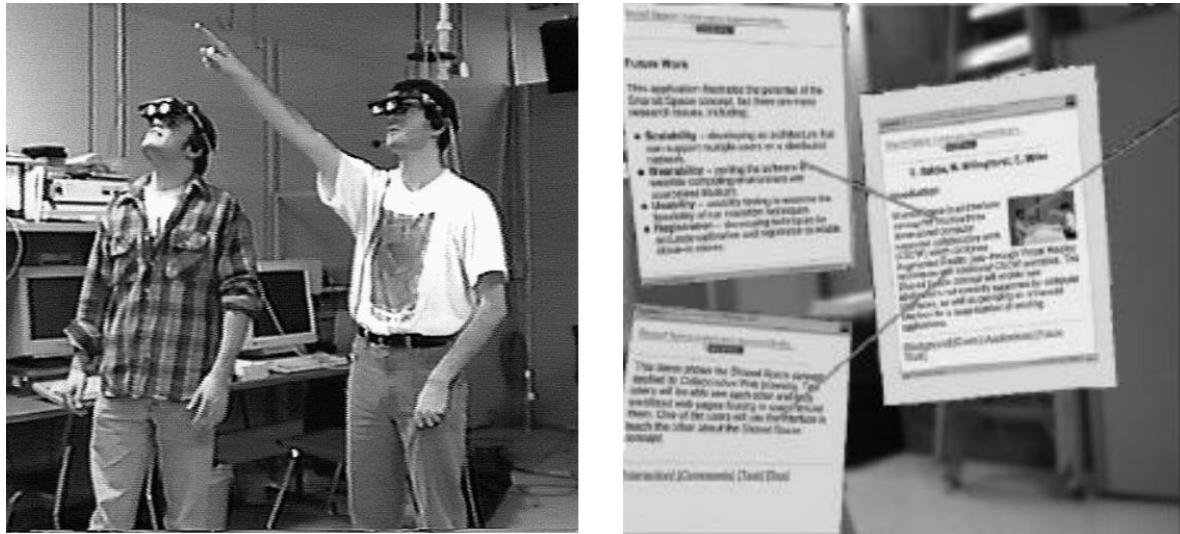


Abbildung 2.2-1: Links: Gemeinsamer geteilter Raum. Rechts: Sichtfeld eines Nutzers, wobei die virtuellen Objekte die visuelle Kommunikation beeinträchtigt.

In einer weiteren Publikation (Billinghurst, Kato & Poupyrev, 2001) wird eine MR-Anwendung – The MagicBook – präsentiert, die ebenfalls mehrere Nutzer einbezieht. Die Grenzen zwischen Realität und Virtualität sollen hier nahtlos ineinander übergehen, sodass die Nutzer die gesamte Spanne des RV-Kontinuum innerhalb einer Anwendung erfahren können. Die verschiedenen Ebenen dieses Kontinuums werden durch Interaktionen mit dem System erreicht (Abbildung 2.2-2):

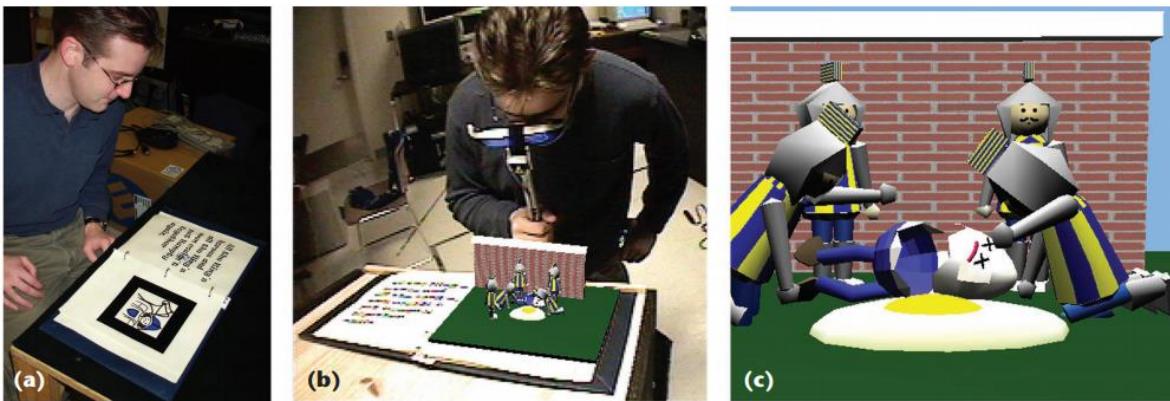


Abbildung 2.2-2: *a)* Realität: Ein Buch, als reales Objekt, kann traditionell gelesen werden. *b)* MR: Durch ein tragbares Display können Szenen aus dem Buch aus verschiedenen Perspektiven betrachtet werden, ohne dabei reale räumliche Informationen zu verlieren. Die Szene kann hier ebenso betrachtet werden wie beispielsweise ein zweiter Nutzer. *c)* VR: Da die Nutzer die Perspektive selbst verändern können, können sie ebenfalls so nah an die virtuelle Szene herangehen, bis ausschließlich virtuelle Informationen zu sehen sind.

Ein weiteres Projekt, welches von Roo und Hachet (2017) vorgestellt wird, ist One Reality. In dieser Arbeit wird ein Konzept vorgestellt, das Mehrbenutzerumgebungen (Multi-User-Environments) durch Fusion verschiedener VR-Techniken herstellt. Dabei werden für verschiedene Nutzergruppen jeweils verschiedene Ebenen des VR-Kontinuums abgebildet, um unterschiedliche Informationen bereitzustellen. Die Nutzer selbst können dabei auch im selben Raum sein. In einer exemplarischen Umsetzung des Konzepts wird neben Konsumententechnik wie einem Leap Motion-Controller¹⁶, einer Kinect-Tiefenkamera, einer HTC-Vive und mehreren Projektoren, auch professionelle Hardware wie ein OptiTrack¹⁷ Tracking-System benutzt. Die Grenzen von Realität zu Virtualität verlaufen hier besonders fließend, da das System nun nicht nur auf einer Art von Technik beruht, sondern gleich verschiedene VR-Hardware und -Technik genutzt wird. Im Anwendungsbeispiel ist eine Vulkansimulation dargestellt. Die Nutzer sind dabei am selben physischen Tisch und nehmen die Anwendung auf verschiedene Weisen wahr (Abbildung 2.2-3).

Die unterschiedliche Wahrnehmung der Nutzer bezüglich der Szene und der anderen Nutzer ist dabei besonders problematisch. Es könnten versehentliche Berührungen mit physischen Objekten und Nutzern entstehen, da diese (vor allem visuell) nicht für alle Nutzer dargestellt werden. In der immersivsten der unterstützten Darstellungsmethoden, mit einem HMD, wird dem Nutzer lediglich eine Warnmeldung angezeigt, wenn dieser in die Nähe anderer Nutzer oder Gegenstände kommt. Die Kommunikation mit anderen Nutzern ist durch das System daher stark beeinträchtigt.

¹⁶ <https://www.leapmotion.com/> (8/03/2018)

¹⁷ <http://optitrack.com/> (8/03/2018)

STAND DER WISSENSCHAFT UND TECHNIK

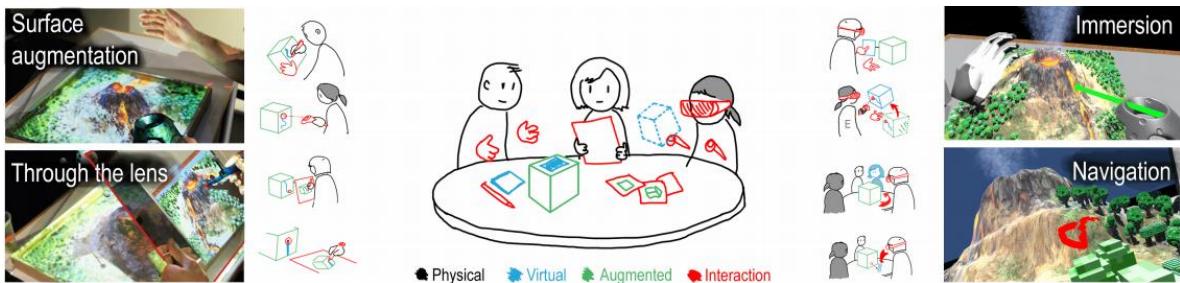


Abbildung 2.2-3: Ein Mock-Up einer MR-Vulkansimulation die mehrere Nutzer auf verschiedene Weisen einbindet, aus Roo und Hatchet (2017).

Neben der Problematik, dass ein Multi-User-MR-System mehrere Nutzer einbinden und darstellen muss (näheres in Kapitel 2.3), gibt es ebenfalls andere Schwierigkeiten eher genereller Natur.

2.2.1 Verdeckungsproblematik in Gemischter Realität

Eine der größten Problematiken bei der Synthese von virtuellen und realen Umgebungen ist die Verdeckung von Objekten. Da es sich bei MR-Anwendungen vorwiegend um Echtzeit-Anwendungen handelt, muss eine Verdeckung in beide Richtungen (Überlagerung von virtuellen Objekten über echte und Überlagerung von echten Objekten über virtuelle) ebenfalls in Echtzeit berechnet werden. Die Anzahl der aktuellen Publikationen die sich vermehrt mit diesem Problem beschäftigt zeigt die momentane Gegenwärtigkeit dieser Problematik. Arbeiten von Hebborn, Höhner und Müller (2017), Walton und Steed (2017), Regenbrecht et al. (2017), Collins, Regenbrecht und Langlotz, Collins (2016), Abate, Narducci und Ricciardi (2014), Gao, Lin und Hua (2012), Fischer, Bartz und Straßer (2004) und Kiyokawa, Billinghurst, Campbell und Woods (2003) sind nur ein kurzer Auszug der aufzeigt, dass mit unterschiedlichsten Arten von Lösungen geforscht wird.

Bei diesen Ansätzen wird grundsätzlich nach Art der genutzten HMDs unterschieden. Bei der Video See-Through-Methode (VST) (beispielsweise Oculus und Vive) nimmt eine Kamera ein Bild der Realität auf und fügt dieses Bild mit der virtuellen Umgebung zusammen. Erst nach dieser Bildfusion wird das Bild dann im HMD des Nutzers angezeigt. Bei der Optical See-Through-Methode (OST) (beispielsweise HoloLens) werden Realität und Virtualität kombiniert, indem die Szene optisch aus Sicht des Nutzers aufgenommen wird (nicht zwingend bildlich). Dieser optische Kombinierer berechnet dann Unterschiede auf Grund von diesen Aufnahmen, die unter anderem die Tiefendaten des RE sein können.

Eine der einfacheren Techniken im Einsatz bei VST HMDs ist Phantom Model Rendering nach Fischer et al. (2004). Der Einsatz der Methode ist ursprünglich im medizinischen Bereich angesiedelt. A priori Informationen des RE, wie prä-operative 3D-Datensätze (MRT- / CT-Scans), werden dabei genutzt, um das Volumen eines realen Gegenstands dann in die Berechnung der virtuellen Szene miteinzubeziehen. So kann eine präzise Berechnung der Verdeckung von realen Objekten in dem VE durchgeführt werden.

Die Publikationen von Gao et al. (2012) und Kiyokawa et al. (2004) fokussieren die Verdeckungsproblematik bei MR-Systemen wiederum bei OST HMDs. Zusätzlich zur Verdeckung wird bei Kiyokawa außerdem explizit auf MR Multi-User-Umgebungen eingegangen. Mehrere Nutzer im selben Raum stellen hier ein erhöhtes Potential für Unfälle dar. Besonders für Co-Located Collaboration sind die jeweils anderen Nutzer keine statischen oder a priori berechenbaren Objekte, müssen jedoch hinsichtlich Verdeckung mit einbezogen werden.

2.3 Visuelle Darstellung von Nutzern in virtuellen Umgebungen

Die visuelle Repräsentation von Nutzern in VEs – der Avatar –, insbesondere in Multi-User-Umgebungen, kann aus zwei unterschiedlichen, theoretischen Perspektiven betrachtet werden (Latoschik et al., 2017):

- 1) Die visuelle Selbst-Darstellung eines Nutzers
- 2) Die visuelle Darstellung anderer Nutzer

Im Zuge dieser Thesis ist besonders die visuelle Darstellung von anderen Nutzern von hoher Wichtigkeit, weswegen bereits an dieser Stelle eine Eigenleistung erbracht wird, sodass die Einordnung und der Vergleich folgender Arbeiten vereinfacht wird.

Ein weiteres Unterscheidungsmerkmal ist die Art der Darstellung selbst. Diese kann in ein ähnliches Kontinuum eingeordnet werden wie das RV-Kontinuum und ist in Abbildung 2.3-1 veranschaulicht. Hier kann ein Avatar anhand seiner Realitätsnähe in das Realismus-Metapher-Avatarkontinuum eingeordnet werden. Ein gänzlich realistischer Avatar ist dabei die realitätsgtreue Abbildung des individuellen Nutzers. Eine gänzlich metaphorische Darstellung ist eine weitgehend abstrahierte Form, die nur wenig bis keine Rückschlüsse auf den eigentlichen Nutzer zulässt. Zwischen diesen Extremen ist der Bereich der simulierten Avatare. Diese sind von hoher menschlicher und individueller Ähnlichkeit bis hin zu stark metaphorischer Darstellung eines generellen Avatars geprägt.



Abbildung 2.3-1: Einordnung von Nutzertdarstellungen in VE anhand der Realitätsnähe in das Realismus-Metapher-Avatarkontinuum.

Oft wird die Einordnung von Avataren anhand ihrer Realitätsnähe nur nach „weniger realistisch“ und „mehr-realistisch“ vorgenommen, wie beispielsweise nach Garau, Slater und Vinayagamoorthy (2003). Weitere Arbeiten, von Slater (1999b) und Bailenson, Yee, Merget und Schröder (2006) stellen dabei besonders heraus, dass Nutzer in VE insbesondere sozialer und natürlicher auf realistischere Darstellungsformen von anderen Nutzern reagieren.

Latoschick et al. (2017) haben außerdem bewiesen, dass das Aussehen von Anderen nicht nur soziale Interaktionen prägt, sondern auch die Selbstwahrnehmung bezüglich der eigenen Körper beeinflusst. Das zeigt, dass die Darstellung anderer Nutzer in Multi-User-Umgebungen nicht nur die offensichtliche Problematik der Wahrnehmung von Avataren und des damit verbundenen Uncanny Valley (Mori, 1970) (Mori, MacDorman & Kageki, 2012) tangiert, sondern auch Aspekte des virtuellen Körperbesitzes (Slater, Perez-Marcos, Ehrsson & Sanchez-Vives, 2009) und des Protheus-Effekts (Yee & Bailenson, 2007) beeinflusst.

Je nach Anwendungsgebiet haben visuelle Repräsentationen eine andere Wichtigkeit und Bedeutung. Von Phillips und Badler (1988) und Badler (2007) werden auf der einen Seite beispielsweise grundlegende technische Charakteristiken für die Erstellung von virtuellen oder digitalen Mensch-Modellen beschrieben. Diese kommen besonders in technischen

STAND DER WISSENSCHAFT UND TECHNIK

Bereichen wie Ergonomie- und Sicherheitsevaluationen oder -simulationen zum Einsatz. Auf der anderen Seite zielen besonders aktuelle Arbeiten auf die sozialen Komponenten von Avataren ab (Latoschick et al. 2017) (Argelaguet, Hoyet, Trico & Lécuyer, 2016) (Roth et al., 2016) (Bailenson, Yee, Merget und Schröder 2006) (Nowak & Biocca, 2003) (Taylor, 2002).

Dieser Trend zu Social VR ist durchaus auch durch aufkommende günstige VR Technik begründet. Die gegenseitige Darstellung von Nutzern virtueller Systeme ist hier eine elementare Funktion die gegeben sein muss, da soziale Interaktion und Kommunikation sonst nur schwer möglich ist. Das Einsatzgebiet von virtuellen Systeme weitet sich jedoch nicht auf den privaten Bereich, sondern auch auf andere vormals fachfremde Forschungs- und Industriezweige aus. Darunter sind unter anderem Themen wie Industrie 4.0 und psychologisch-orientierte Experimente. In diesen Gebieten sind jedoch nicht nur explizite Nutzer von VR Systemen wichtig, sondern auch implizite Nutzer des Systems, wie Mitarbeiter oder Experimentatoren im selben Raum. Nutzer, die in diese Beobachter-Situationen kommen, haben oft das Problem, dass die gesamte Szene nur unzureichend visualisiert wird. Meist ist es für die Außenstehenden nur mit Umständen möglich mitzuerleben oder wenigstens zu sehen, was der Nutzer eines immersiven VE gerade wahrnimmt (Nordby, Gernez & Børresen, 2016) (Zhu et al., 2016) (Biocca & Levy, 2013). Auch die Kommunikation zwischen der virtuellen und der realen Umgebung wird dadurch erschwert.

Ein weiterer wichtiger visueller Aspekt bezüglich Avataren ist die Anpassung oder auch Personalisierung. Ducheneaut, Wen, Yee und Wadley (2009) beschreiben den Einfluss der Möglichkeit der Nutzer zur Individualisierung ihrer Avatare als ambivalent. Zum einen legen Anwender großen Wert auf eine individuelle Repräsentation in VE, zum anderen führt eine aktive Gestaltungsmöglichkeit der Avatare dazu, dass genau diese Individualität verloren geht. Da sich bei der Anpassung meist nur auf die offensichtlicheren Körpermerkmale konzentriert wird, werden besonders Haare und Körperbau beeinflusst und im Vergleich zum Aussehen des echten Nutzers geschönt. Das führt nach Ducheneaut et al. dazu, dass die eigentlich beabsichtigte Vielfalt der Avatare eher in eine stereotypische Vision gepackt wird. Diese Vision spiegelt weniger die reelle Individualität der Nutzer wieder, sondern eher das was die Nutzer denken, wie ein menschlicher Körper aussehen sollte.

2.3.1 *Technische Beispiele*

Auf Grund der technischen Umsetzung eines Prototyps in Kapitel 3.3 werden im Folgenden verschiedene Darstellungsmethoden beispielhaft vorgestellt und in Kürze in den theoretischen Kontext aus Abschnitt 2.3 eingeordnet. Der Fokus wird hier nach einem Beispiel zu eher metaphorischen Avataren dann auf Arbeiten im Bereich realistischer Avatare zur Darstellung anderer Nutzer gesetzt.

Im bereits genannten Projekt von Billinghurst, Kato und Poupyrev (2001) wird ein stark metaphorischer Avatar genutzt, um sowohl sich selbst, aber auch andere Nutzer zu verkörpern (Abbildung 2.3-2). Ein menschlicher Körper ist zwar erkennbar, jedoch ohne jeglichen Realitätsanspruch, sodass etwaige a priori Simulation für die optische Modellierung des Charakters definitiv auszuschließen ist. Über die Simulation von Bewegungen kann keine Aussage gemacht werden, da Bewegungsabläufe des Avatars nicht weiter beschrieben werden und in dem von Billinghurst et al. beschriebenen Szenario auch keine Relevanz haben. Somit wird diese Avatar-Form nicht gänzlich als Metapher eingeordnet, jedoch sehr stark abstrahiert.



Abbildung 2.3-2: Avatar-Repräsentation von Nutzern in einem Multi-User-Environment. (Billinghurst, Kato & Poupyrev, 2001)

Von Pandzic et al. (2001) wird ein Konzept vorgeschlagen, bei dem Avatare zu Teilen realistisch repräsentiert werden. Der Schwerpunkt wird auf das Gesicht der Avatare gelegt. Dieses wird vor Benutzung am Avatar mit Kameratechnik gescannt und durch Verwendung von generischen Mensch-Modellen rekonstruiert (Abbildung 2.3-3 links). Durch das Mapping auf das generische Modell kann die vorerst statische Textur dann dynamisch animiert werden. Der Avatar-Kopf, der dem des realen Nutzers entspricht, wird dann auf einen eher metaphorisch modellierten Körper gesetzt. Auf die Anwendung für einen Selbstavatar oder die Darstellung anderer Nutzer wird hier nicht genauer eingegangen, wobei das Konzept auf beides anwendbar wäre.

Ein ähnlicher Ansatz wird von Hagen (2017) vorgeschlagen, wobei hier ein höherer Simulationsanteil und eine Echtzeit-Komponente miteingebunden werden. Ein 3D-Scanner wird genutzt, um ein physikalisch korrektes Abbild des individuellen Nutzers zu erstellen. Das 3D-Modell spiegelt ebenfalls Gesichtszüge und Kleidung wieder, da die Textur ebenfalls mitgescannt wird. Danach wird dieses Modell mit CAD-Software geriggt um später animiert werden zu können. Ein Perception Neuron¹⁸ Tracking-System wird dann genutzt (Abbildung 2.3-3 rechts), um das digitale Skelett daraufhin in Echtzeit zu animieren. Diese Darstellungsart ist beschränkt auf die Selbstdarstellung eines Nutzers. Durch die Ganzkörper-Virtualisierung und die Echtzeit synchronisierte Bewegung des Avatars ist dieses Konzept als sehr realistisch einzuführen. Es bestehen jedoch Abhängigkeiten an das zugrundeliegende Skelett, welches die realistischen Daten aus dem Perception Neuron verfälschen können. Auch die Deformationen des echten dynamischen Körpers des Nutzers können durch Deformationen des statischen, gescannten 3D-Modells entsprechend nicht realitätsnah dargestellt werden.

¹⁸ <https://neuronmocap.com/> (8/03/2018)

STAND DER WISSENSCHAFT UND TECHNIK

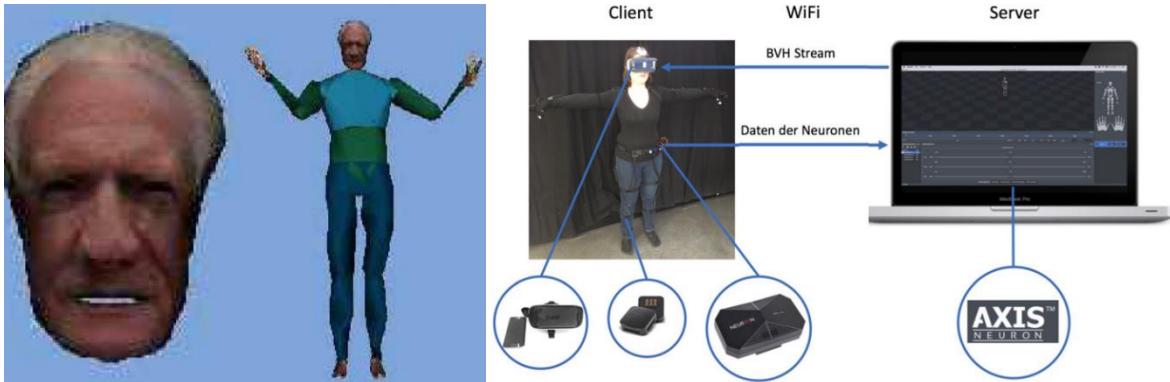


Abbildung 2.3-3: *Links: Beispielhafter Avatar nach Pandzic et al. (2001). Rechts: Echtzeit-Steuerung eines Avatars nach Hagen (2017).*

Eine weitere Möglichkeit Personen mit hohem Anteil an Realismus in virtuellen Welten darzustellen ist die Nutzung von Compositing-Techniken, wie einem Green-Screen. Dabei wird ein Nutzer mit einer Kamera aufgenommen und durch Bildverarbeitungsmethoden vom Hintergrund segmentiert. Pomi und Slusallek (2005) nutzen diese Technik, um die reale Textur von Personen so in Echtzeit auf eine digitale 2D-Ebene im VE zu projizieren. Durch Ray Tracing wird das entstehende Bild noch glaubhafter, da beispielsweise Schatten der Person so relativ realistisch abgebildet werden (Abbildung 2.3-4 links). Zhu et al. (2016) verwenden einen Green-Screen und verbinden diesen mit einer Tiefenkamera und einem Trackingsystem. So werden sowohl die digitale als auch die physische Kamera an der gleichen Stelle positioniert, um dann im finalen Compositing ebenfalls die Tiefeninformationen mit einzubinden. Besonders Verdeckungen werden damit sehr gut simuliert. Das Bild wird dann genutzt, um Zuschauern im selben Raum einen Eindruck des Nutzers in der Szene zu geben. Ein großer Nachteil bei der Verwendung von Green-Screen ist der Einfluss auf den physischen Raum. Zum einen ist die Kameraperspektive durch den Umfang des grünen Stoffs beschränkt und zum anderen hat diese unnatürliche Umgebung auch Einfluss auf die Optik der Personen-Textur. Ein grüner Rand und Reflektionen auf dem Körper sind nicht einfach auszuschließen (Abbildung 2.3-4 rechts). Insgesamt werden besonders die Bewegungen bei diesen beiden Arbeiten sehr realistisch dargestellt, da die echte Textur der Nutzer abgebildet wird. Der entstehende Avatar ist in höchstem Grad persönlich, sodass sogar präzise, realistische Körpersprache hier gut in die virtuelle Welt übertragen wird. Die Textur wird auf einem 2D-Objekt gerendert, womit die 3D-Information des Körpers weitgehend verworfen wird, auch wenn durch geschicktes Ray Casting oder Tiefeninformationen diesem Problem versucht wird entgegenzuwirken. Ein weiteres Problem ist außerdem, die Visualisierung für den VR-Nutzer. Im Fokus steht hier definitiv eine Art außenstehender Beobachter-Nutzer der Szene. Der VR-Nutzer selbst bekommt nur schwer eine Repräsentation der anderen Nutzer.



Abbildung 2.3-4: Links: Realistische 2D-Überblendung und Schattierung nach Pomi und Slusallek (2005). Rechts: Realistische 2D-Verdeckungsberechnung nach Zhu et al. (2016)

Aktuellste Veröffentlichungen, wie etwa von Nordby, Gernez und Børresen (2016) oder Regenbrecht et al. (2017) betrachten sowohl die Beziehung von Nutzern in VEs zu Nutzern in REs, als auch die komplexe Integration von 3D-Daten. Nordby et al. verwenden eine Kinect, um aus einer Kombination von Farb- und Tiefendaten ein texturiertes 3D Mesh zu generieren (Abbildung 2.3-5 links). Dabei werden die Informationen der Tiefe auf Pixel eines Farbbildes abgebildet. Das Farbbild wird tesseliert und die Farb-Pixel-Knoten werden entsprechend der Tiefe versetzt. Da ein Computervision-Back-End (CV-Back-End) vorher nur die Pixel extrahiert die zu einem Menschen gehören, wird der Hintergrund des Bildes vorher verworfen – nur der Nutzer wird in 3D dargestellt. Wie man in Abbildung 2.3-5 gut erkennen kann, reicht eine einzige Kamera (RGB+Tiefe, (RGBD)) nicht aus, um ein vollständiges abgeschlossenes Mesh zu generieren, da man nur Informationen von einer Perspektive verarbeiten kann. Im Anwendungsfall den Nordby beschreibt, kann ein VR-Nutzer mit HMD daher den außenstehenden Nutzer nur von einer Seite aus wirklich korrekt betrachten.

Bei Regenbrecht wird ein ähnlicher Ansatz verfolgt. CV wird genutzt, um die Person (oder auch andere Objekte) vom Hintergrund zu extrahieren. Danach wird eine Kinect-Kamera verwendet, um wieder RGBD-Daten einander zuzuordnen. Jedoch wird hier eine Untermenge der Pixel verwendet um eine Art Punktwolke zu generieren und anschließend zu rendern. Die Farb-Punkte, die auf den dreidimensionalen Raum eines Rasters abgebildet werden, sind vergleichbar einfach weiterzuverarbeiten. Die einzelnen Punkte werden dann mit einer vorgegebenen Größe und geometrischen Form in der zugehörigen Farbe gerendert, um so den eigentlichen Menschen virtuell zu repräsentieren. Um Nutzern in und vor allem außerhalb des eigentlichen VE einen räumlichen Eindruck zu geben, wird ein virtueller Spiegel genutzt, um sich selbst und die Umgebung zu sehen.

Diese beiden Visualisierungsformen sind auf der einen Seite etwas abstrakter als die vorherigen, aber gleichzeitig auch realistischer und weniger metaphorisch. Es wird kein vorher festgelegtes Mensch-Modell zur Darstellung verwendet und im Vergleich zur 2D-Textur werden die Tiefeninformationen genutzt, um eine dreidimensionale Form eines

STAND DER WISSENSCHAFT UND TECHNIK

individuellen Menschen in Echtzeit zu simulieren. Ähnliche Ansätze werden auch von weiteren Publikationen aufgegriffen (Kowalski, Naruniec & Daniluk, 2015) (Jones et al., 2014) (Alexiadis, Zarpalas & Daras, 2013). Meistens wird jedoch sehr kostenintensive Hardware benutzt, die nicht im Sinne von „low-budget“ ist, oder der physische Raum wird ähnlich stark eingeschränkt wie bei der Green-Screen-Methode.

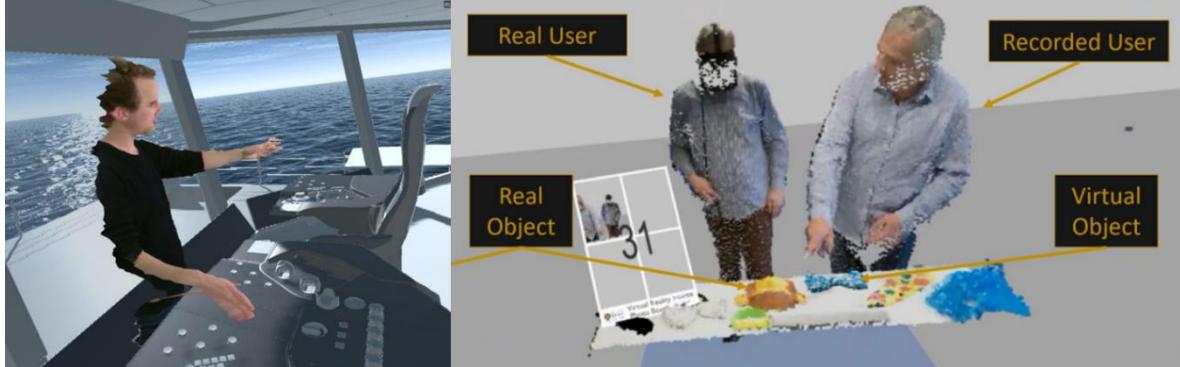


Abbildung 2.3-5: Links: Meshs-Darstellung eines außenstehenden Nutzers nach Nordby, Gernez und Børresen (2016). Rechts: Voxel-Darstellung eines aufgenommenen außenstehenden und eines Echtzeit VR- Nutzers nach Regenbrecht et al. (2017).

Die Kommunikation zwischen den beiden Realitäten ist trotz verschiedenster vorgestellter Methoden dennoch immer stark eingeschränkt, da man in den vorgestellten Ansätzen meist nur eine Kommunikationsrichtung erweitert – entweder sieht der VR-Nutzer den außenstehenden oder der außenstehende Nutzer den VR-Nutzer in seinem VE. Dies ist auch nur in starker Einschränkung möglich, sodass Repräsentationen nur aus einer fixen Kameraperspektive möglich sind oder an ein statisches virtuelles Objekt (Spiegel) gebunden sind. In Tabelle 2.3-1 sind die in diesem Abschnitt beschriebenen Arbeiten in die zwei Kategorien:

- Darstellung der Nutzergruppe
- Realismus-Metapher-Avatarkontinuum

ingeordnet.

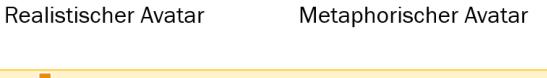
Veröffentlichung	Darstellung der Nutzergruppe	Einordnung in Realismus-Metapher-Avatarkontinuum
Billinghurst, Kato & Poupyrev, 2001	Selbstdarstellung + Darstellung Anderer	
Pandzic et al., 2001)	Selbstdarstellung + Darstellung Anderer	
Hagen, 2017	Selbstdarstellung	
Pomi & Slusallek (2005) Zhu et al. (2016)	Darstellung Anderer	
Nordby, Gernez & Børresen, 2016 Regenbrecht et al., 2017	Selbstdarstellung + Darstellung Anderer	

Tabelle 2.3-1: Einordnung der beschriebenen Arbeiten in „Darstellung der Nutzergruppe“ und „Realismus-Metapher-Avatarkontinuum“.

2.4 Kommunikation im Bereich der Virtuellen Realität

Nachdem nun ein kurzer theoretischer und praktischer Überblick über moderne Darstellungsarten von Nutzern in VE gegeben wurde, ist die Kommunikation im VR-Bereich ein relevantes Thema. In diesem Abschnitt wird ebenfalls erst eine generelle theoretische Basis dargestellt. Insbesondere der visuelle Aspekt wird hervorgehoben, sodass Grundkonzepte bekannt sind. Danach werden Publikationen betrachtet in denen es speziell um Kommunikation in virtuellen Welten geht, um danach einen Einblick in aktuelle Arbeiten zu geben, welche die Kommunikation zwischen realer und virtueller Umgebung thematisieren.

2.4.1 Visuelle Kommunikation

Die visuelle oder auch physiologische Komponente der Kommunikation zwischen Menschen ist der Teil der Kommunikation, der non-verbal und non-vokal stattfindet. Statt dem Sinn der gesprochenen Wörter oder der Art und Weise wie die Wörter gesagt werden, stehen Aspekte wie Kinesik (Körpersprache), Mimik (Gesichtsausdruck) und Motorik (Bewegungsverhalten) im Fokus (Fast, 1988) (Scheflen, 1972). Für Menschen ist es beispielsweise ein Leichtes in Sekundenbruchteilen sowohl Alter, Geschlecht, Gewicht und Gemütslage zu differenzieren. Dafür sind lediglich minimale visuelle Informationen nötig. Mit in etwa 55% macht die non-verbale, non-vokale Kommunikation nach Mehrabian (Mehrabian & Wiender, 1967) (Mehrabian & Ferris, 1967) somit den größten Teil der Wahrnehmung und Schlussfolgerung von Personen und deren Aktionen aus. 38% wird der vokalen Kommunikation zugeschrieben – nur 7% der verbalen. Nach Sagert-Müller (2008) ist auch die visuelle Individualität ein Aspekt der visuellen Kommunikation, wobei diese auch oft zur Körpersprache gezählt wird. Dazu werden sowohl Embleme und Handzeichen gezählt, aber auch visuelle Artefakte, wie die Wahl der Kleidung, Schmuck oder anderer oberflächlicher, optischer Merkmale.

Die Implikation aus den allgemein anerkannten Veröffentlichungen von Mehrabian sind daher, dass der eigentliche Sinn der gesprochenen Wörter im Vergleich zum visuellen Aspekt insgesamt deutlich an Wichtigkeit für die Kommunikation verloren. Es ist dadurch einfacher

STAND DER WISSENSCHAFT UND TECHNIK

für Menschen schwierige Sachverhalte misszuverstehen, wenn man den Gesprächspartner nicht sieht. Besonders bei inhaltlich komplizierten Begebenheiten gewinnen die non-verbalen Aspekte noch mehr Bedeutung.

Da dieses Thema bereits ganze Forschungszweige zu füllen vermag und gleichzeitig für das Verständnis und die Einschätzung Relevanz der vorliegenden Thesis nur ein Bruchteil dessen nötig ist, wird für tiefere inhaltliche Einblicke auf Fachliteratur psychologischer Art, wie von Rothwell (2010), Jamieson (2007) oder auch Sless (1981) verwiesen.

2.4.2 Visuelle Kommunikation in Virtuellen Umgebungen

VR-Technologie wurde bereits in frühen Jahren als maßgeblich für zukünftige Kommunikationsformen (Krueger, 1991) und Medium für Kommunikation und Perzeption angesehen (Biocca, 1992). Dabei wird besonders die visuelle Direktheit der Kommunikation durch die VR verändert (Abbildung 2.4-1). Auch aktuellere Publikationen, wie von Biocca und Levy (2013) vertreten noch die selbe Meinung, sodass generell gesagt werden kann, dass ein VR-Kommunikationssystem keine direkten Kommunikationsinformationen transportiert, jedoch diese heutzutage sehr gut encodiert und übermittelt.

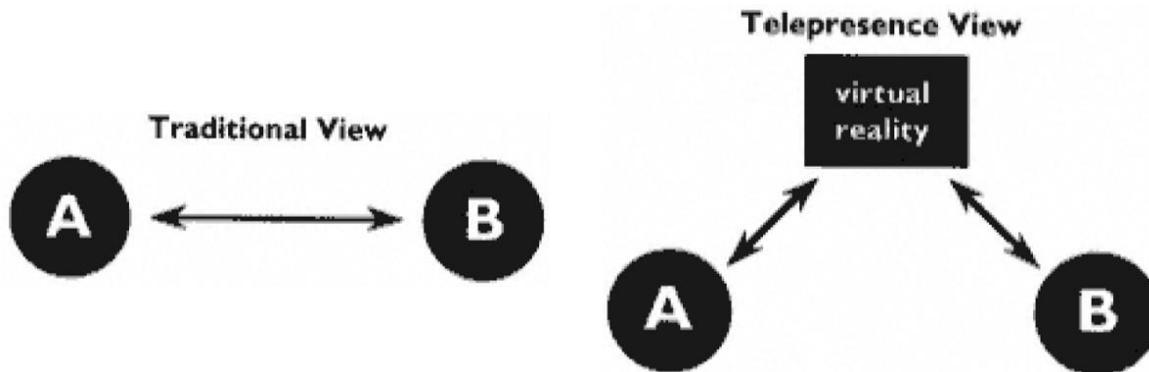


Abbildung 2.4-1: Links: Traditionelle visuelle Kommunikation. Rechts: Schematische Veränderung der visuellen Kommunikation durch VR. (Krueger, 1991)

Es besteht auch der Konsens, dass non-verbale, non-vokale Elemente in virtueller Kommunikation ebenfalls überwiegen und besonders durch Körpersprache geprägt werden (Mania & Chalmers, 1998). Die virtuelle Körpersprache wird in VE entsprechend durch Avatare geprägt und wird nach Mania und Chalmers (1998) und Ventrella (2011) in folgende Elemente unterteilt:

- Gesichtsausdruck
- Blick(-Richtung)
- Gesten
- Körperhaltung
- Selbstrepräsentation
- Körperkontakt

Viele dieser Punkte finden ein exaktes Pendant in der Literatur, die in Sektion 2.4.1 genannt wird. Ein auffallender Unterschied ist aber die explizite Nennung der Selbstrepräsentation. Da Darstellungsformen in VE nicht automatisch immer an eine Darstellung des eigenen Körpers geknüpft sind, gibt es einige Untersuchungen im Bereich der Kommunikation und Wahrnehmung, die sich derzeit mit dieser Thematik beschäftigen. Dodds et al. (2011) beschreiben die Wichtigkeit eines Selbstavatars für die Kommunikation, sowohl für einen Sprecher, als auch für den Zuhörer. Auch für das Empfinden von Entfernung (Mohler,

STAND DER WISSENSCHAFT UND TECHNIK

Creem-Regehr, Thompson & Bühlhoff, 2010) und Höhe (Gutekunst et al., 2014) spielt die Avatardarstellung des Nutzers eine Rolle. Insgesamt ist der positive Einfluss von Körpersprache realistischer Avatare auf die Kommunikation zwischen Nutzern einheitlich anerkannt (Dodds et al., 2011) (Dodds, Mohler & Bühlhoff, 2010) (Pandzic et al. 2001) (Daft & Lengel, 1986), wobei hier nicht nur ein Selbstavatar gemeint ist, sondern auch die gegenseitige Darstellung.

Hinsichtlich der hohen Relevanz von Avataren in Computerspielen, wird oft auch genau dort der Stellenwert von realistischem Aussehen und realistischen Bewegungen klar. In Meilensteinen wie Heavy Rain wurden State-of-the-Art-Techniken genutzt, um den animierten Avataren ein möglichst realistisches Aussehen zu geben und somit das Uncanny Valley zu durchschreiten (Doerr, 2007). Trotz qualitativ hochwertigen Hauttexturen und Gesichtsausdrücken wurde die Kommunikation mit den Figuren jedoch nicht als realistisch genug eingeordnet, da die Sprache noch deutlich realistischer war als der Gesichtsausdruck (Hoggins, 2010) (Gouskos, 2006). Nach Tinwell, Grimshaw, Nabi und Williams (2011) wurde auch in LA Noir ein ähnliches Fehlen an Realismus bei Avataren festgestellt. Nutzer konnten besonders bei der Kommunikation von Angesicht zu Angesicht nicht die subtilen aber essentiellen Körper- und Gesichtsbewegungen wahrnehmen und hatten damit Schwierigkeiten den Täter der Krimi-Story ausfindig zu machen.

Auch die optische Personalisierung von Avataren wirkt sich auf die Wahrnehmung und Kommunikation in VE aus. Nutzer geben dort gezielt weniger persönliche Informationen preis und verbergen, im Gegensatz zur direkten Kommunikation von Angesicht zu Angesicht, gezielt soziale Merkmale (Joinson, 2004). Das beeinflusst die visuellen Aspekte der Kommunikation maßgeblich. Diese Tendenzen werden nach Caplan und Turner (2007) und Wright (2002) durch erhöhte Anonymität und reduzierte non-verbale/demographische Merkmale verstärkt. Um eine natürliche und realistische Kommunikation in VE zu unterstützen, schlussfolgern Pandzic et al. (2001) auch die Notwendigkeit von natürlichen und realistischen Repräsentationen der Nutzer.

2.4.3 Visuelle Kommunikation zwischen Realen und Virtuellen Umgebungen

Um sich inhaltlich nun dem Kern der Thesis weiter zu nähern, wird die visuelle Kommunikation zwischen realen und virtuellen Umgebungen problematisiert. An dieser Stelle muss gesagt werden, dass die Anzahl an Publikationen die sich mit dieser Problematik befassen gering ist. Dies zeigt jedoch unter anderem auch die Relevanz der vorliegenden Thesis.

Auch wenn ein Nutzer außerhalb des eigentlichen VE kein direkter VR-Nutzer ist, so ist dieser außenstehende Nutzer oder Beobachter doch in vielen Bereichen auch Teil einer notwendigen Kommunikation. Diese tangiert sowohl die Realität, als auch die Virtualität, da in irgendeiner Art und Weise beide Nutzergruppen alle Informationen brauchen:

- Der VR-Nutzer benötigt visuelle Informationen aus der realen Umgebung, wie beispielsweise die realistische Repräsentation des Beobachter-Nutzers im realen oder virtuellen Raum.
- Der außenstehende Nutzer / Beobachter benötigt für eine erfolgreiche Kommunikation sowohl die realistische Repräsentation des VR-Nutzers, aber gleichzeitig auch die nötigen räumlichen Informationen aus dem VE. Nur so ist es beispielsweise möglich, dem Nutzer in einer immersiven VR relevante Hilfestellungen zu geben.

STAND DER WISSENSCHAFT UND TECHNIK

Mögliche Anwendungsgebiete für solch eine Kommunikation zwischen RE und VE sind durch aufkommende kostengünstige VR-Technik derzeit so mannigfaltig wie ihre Nutzer selbst. Experimente im Bereich der Psychologie und Ergonomie, Social VR-Anwendungen und der Einsatz in Management und Wirtschaft sind nur ein kurzer Auszug der Möglichkeiten.

Nordby et al. (2016) machen besonders die Bedeutsamkeit in Designfragestellungen und qualitativen Ergonomieevaluationen deutlich. Die vorher beschriebene Darstellung der Nutzer wird erfolgreich genutzt, um ein konzeptuelles Design von maritimen Arbeitsplätzen durchzuführen (Abbildung 2.3-5 links). Die direkte Kommunikation ist bezüglich eines gemeinsamen Kontextes und der dialogähnlichen Besprechung des Designs ein großer Vorteil. Außerdem wird die Immersion nur in sehr geringem Maße durch die Kommunikation mit der Außenwelt verringert. Ebenfalls wird dem Gefühl der physischen Isoliertheit, nach Slater und Wilbur (1997) entgegengewirkt, da außenstehende Personen realistisch virtualisiert werden.

In der Arbeit von Zhu et al. (2016) wird hingegen die soziale Komponente in den Vordergrund gerückt. Es wird auf die These von Biocca und Levy (2013) aufgebaut, dass es mehr Spaß macht virtuelle Momente in realistischer Qualität mit Freunden, Familie und Kollegen zu teilen, als sie alleine im HMD zu erleben. Derzeitige Konsumenten-VR-Technik ist normalerweise begrenzt auf ein einziges HMD und daher einen VR-Nutzer. Trotzdem möchte man als sozialen Aspekt auch außenstehende Personen teilhaben lassen. Dieser Aspekt zählt ebenfalls zur visuellen Kommunikation zwischen den Welten. Dafür eignet sich nach Ohta und Tamura (2014) besonders ein MR-System: Eine AV-Visualisierung auf der VR-Seite und eine AR-Visualisierung auf der Beobachter-Seite. Vorausgehend ist das auch die Meinung die auch Billinghurst und Kato (1999) implizit vertreten, indem sie MR als geeignetes Kommunikationsmedium für kollaboratives Arbeiten im selben Raum vorschlagen (Abbildung 2.2-1). Kato und Billinghurst (1999) schlagen hierzu eine Alternative vor, die die echte Welt und die virtuelle Welt über ein verteiltes System vereint (Abbildung 2.4-2).



Abbildung 2.4-2: Kollaboratives Konferenz System. Ein VR-Nutzer kommuniziert dadurch mit außenstehenden Nutzern, welche m VE realistisch repräsentiert werden. (Kato & Billinghurst, 1999)

Auch Szalavári, Schmalstieg, Fuhrmann und Gervautz (1998) sind der Meinung, dass MR eine geeignete Technik für kollaboratives Arbeiten ist – auch wegen der guten Kombination zwischen realen und virtuellen Aspekten. Sie nennen explizit fünf Vorteile gegenüber VR-Technik:

- Virtualität
- Erweiterung
- Kooperation

- Unabhängigkeit
- Individualität

Damit wird besonders der Unterschied zu Kommunikation und Kollaboration in gänzlich immersiven VE aufgezeigt. Nutzer können beispielsweise Notizen, Bücher oder andere reale Gegenstände nutzen, während sie trotzdem auch mit virtuellen Objekten interagieren können (Abbildung 2.2-1 rechts). Der wichtigste Unterschied ist jedoch nach Billinghurst und Kato (1999) der, dass die Nutzer sich gegenseitig in realer Repräsentation sehen können (Abbildung 2.2-1 links). Die Erkennung feingranularer Bewegungen, Gesichtsausdrücke und Körpersprache erhöht die Bandbreite der Kommunikation deutlich.

2.5 Erkennung und Extraktion von menschlichen Texturen aus Bilddaten

Die Erkennung und Extraktion, beziehungsweise Segmentierung von Objekten in Bilddaten ist ein weit verbreitetes Problem in vielen Bereichen der Informatik. Lösungsansätze werden vorwiegend im Fachgebiet der Computervision (CV), sowie angrenzenden Gebieten (Bildverarbeitung, Machine Learning (ML) und Human-Computer Interaction (HCI)) erforscht.

Der wichtigste Schritt dabei ist die eigentliche Objekterkennung – oder in diesem Fall die Erkennung des menschlichen Körpers in einem Bild. Das kann auf verschiedene Weisen geschehen, jedoch muss letztendlich eine Abbildung des menschlichen Körpers auf diskrete Bildpixel vorgenommen werden, um danach eine 2D-Textur generieren zu können. Dabei gibt es verschiedenste Ansätze zur menschlichen Körpererkennung (Human Body Detection) und Posenschätzung (Human Pose Estimation), wovon in dieser Thesis folgende in Kürze zusammengefasst werden:

- Histogram of Oriented Gradients (Dalal & Triggs, 2005)
- Haar Feature-basierter Cascade Classifier (Viola & Jones, 2001) (Lienhart & Maydt, 2002)
- Part Affinity Fields (Cao, Simon, Wei, Sheikh, 2017) und Convolutional Pose Machines (Wei, Ramakrishna, Kanade & Sheikh, 2016)
- GrabCut (Rother, Kolmogorov & Blake, 2004)
- Hintergrundsubtraktion (KaewTraKulPong & Bowden, 2002) (Godbehere, Matsukawa & Goldberg, 2012)
- Deep Randomized Decision Forest mit Nutzung von Tiefeninformationen (Shotton et al., 2013)

Es ist an dieser Stelle nicht der Anspruch der Thesis, diese Konzepte technisch aufzubereiten und vorzustellen. Viel eher werden anwendungsnahe Informationen dargestellt und elementare Vor- und Nachteile im Vergleich zu anderen Ansätzen aufgezeigt, da das Hauptkapitel nicht die technische Konzeption eines Algorithmus vorsieht, sondern diesen versucht geschickt anzuwenden und zu einem neuen Vorgehen zu kombinieren.

2.5.1 Histogram of Oriented Gradients

Die derzeit wohl bekannteste und etablierteste Methode zur Personenerkennung ist die Nutzung eines Histogram of Oriented Gradients (HOG) Detektors. Ein HOG ist nach Dalal und Triggs (2005) eine Sorte von Merkmaldeskriptoren. Ein Merkmalsdeskriptor wird als eine generalisierte Darstellung eines Objekts - in diesem Fall einer Person - genutzt. Die interne, bijektive Abbildung eines solchen Deskriptors in einem System ist ein einziger Vektor. Diese vergleichbar einfache und globale Repräsentation eines Objekts vereinfacht die eigentliche Klassifikation im späteren Verlauf der Anwendung. Ein Objekt wird daher ebenfalls als ein einziges Merkmal abgebildet, ganz im Kontrast zu komplexeren

Deskriptoren wie SIFT (Lowe, 1999). Um aus den HOG Deskriptoren einen Detektor zu erstellen, wird meist eine Support Vector Machine (SVM) (Vapnik & Chervonenkis, 1974) auf einem ganzen Datensatz von passenden Merkmalsvektoren trainiert.

Die größte Stärke eines HOG Detektors ist wohl ebenfalls seine größte Schwäche. Die globale Repräsentation eines Menschen ist zwar einfach zu handeln und mit einfachen binären Entscheidungen („ist Person“ oder „ist nicht Person“) durchführbar, jedoch sind Teilbasierte Lösungen deutlich flexibler. Wenn einzelne Körperteile als Deskriptoren genutzt werden, sind besonders sich bewegende Körperteile deutlich besser zu erkennen. Ebenso sind Teilabbildungen eines Körpers damit erkennbar (beispielsweise durch Verdeckung).

2.5.2 Haar-Merkmal-basierter Cascade Classifier

Ein Haar-Merkmal-basierter Cascade Classifier (Haar-Cascade Classifier) nach Viola und Jones (2001), verbessert von Lienhart und Maydt (2002), ist im Gegensatz zum globalen HOG Detektor ein auf lokale Merkmale trainierter Detektor. Der Detektor selbst besteht aus mehreren trainierten Klassifikatoren (Classifiern), welche wiederum je aus einer Kaskade von Klassifikatoren besteht. Diese Basis-Klassifikatoren sind Entscheidungsbäume, die haarähnliche Merkmale verarbeiten (Abbildung 2.5-1). Unter einer Cascade versteht man hier, dass es verschiedene Stufen gibt, in der die Classifier angeordnet sind. Diese Stufen werden dann nacheinander auf einen Bereich des Bildes angewandt, bis alle Stufen angewandt wurden oder der Bereich vernachlässigt werden kann. Es wird entweder „0“ oder „1“ zurückgegeben, um anzugeben, ob das vorher antrainierte Objekt hier zu sehen ist oder nicht. Dieses Verhalten wird immer für einen Bereich von fester Größe für alle Bereiche des Bildes angewandt (Sliding Window Approach).

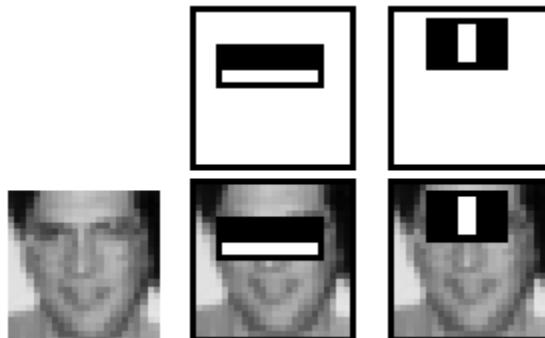


Abbildung 2.5-1: Haar-ähnliche Merkmale die in Haar-Cascade-Classifiern genutzt werden. ((Viola & Jones, 2001)

Beschriebene Vorteile sind besonders die Schnelligkeit und Hardware-Anforderungen. Der Detektor kann ohne Probleme auch auf schwächeren CPUs, wie in Kameras und Smartphones genutzt werden. Problematisch ist jedoch die Stabilität und Genauigkeit. Durch den Sliding Window Approach, ähnlich wie bei HOG, hängt die Genauigkeit in hohem Maße von den Voreinstellungen der Parameter ab, sodass zwar spezifische Szenarien (beispielsweise Gesicht in typischer Entfernung zu einem Smartphone) sehr gut voreingestellt werden können. Sobald die Variablen Entfernung und Skalierung der Objekte jedoch innerhalb eines Szenarios sehr divergieren, ist die Erkennung nicht mehr stabil.

2.5.3 Part Affinity Fields und Convolutional Pose Machines

Part Affinity Fields und Convolutional Pose Machines werden an dieser Stelle zusammen behandelt, da der Autorenkreis ein sehr ähnlicher ist und beide in der Anwendung OpenPose¹⁹ – welche derzeit sehr en vogue ist – vereint genutzt werden.

Convolutional Pose Machines, nach Wei, Ramakrishna, Kanade und Sheikh (2016) nutzen die Zusammenhänge von mehreren Körperteilen die erkannt werden, um eine Person zu erkennen und gleichzeitig Aussagen über ihre Pose zu machen. Damit ist das Vorgehen ähnlich zu Haar-Cascade Classifiern. Durch iterative Einbeziehung von erkannten Körperteilen wird die Schätzung der Körperteile immer genauer (Abbildung 2.5-2 oben).

Part Affinity Fields, nach Cao, Simon, Wei, Sheikh (2017), ist es hingegen sogar möglich, die Pose von mehreren Personen gleichzeitig abzuschätzen und dabei jede Person individuell zu betrachten. Dabei wird, ähnlich zu HOG, eine Ausrichtung der einzelnen Objekte miteinbezogen. Diese werden hier sogar Pixelweise als 2D-Vektor geschätzt (Abbildung 2.5-2 Mitte), was die Genauigkeit der Schätzung erhöht.



Abbildung 2.5-2: *Oben:* Iterative Vorhersage des rechten Ellenbogens einer Convolutional Pose Machine. *Mitte:* Vorhersage des rechten Unterarms mit Part Affinity Fields von mehreren Personen. *Unten:* Ergebnis der Kombination von Part Affinity Fields und Convolutional Pose Machine wie in der Anwendung OpenPose. (Wei, Ramakrishna, Kanade & Sheikh, 2016) (Cao, Simon, Wei, Sheikh, 2017)

Die enormen Vorteile der Kombination der beiden Vorgehensweisen zur Personenerkennung und Posenschätzung sind relativ klar. Durch Convolutional Pose Machines wird eine schnelle und akkurate Erkennung und erste Posenschätzung durchgeführt. Durch die Einbeziehung mehrerer Körperteile in den Gesamtkontext werden besonders Mehrdeutigkeiten verringert,

¹⁹ <https://github.com/CMU-Perceptual-Computing-Lab/openpose> (8/03/2018)

die beispielsweise durch symmetrische Körperteile hervorgerufen werden können. Die Part Affinity Fields werden dann genutzt, um mehrere Personen als Individuen zu betrachten, wodurch das ganze System sehr robust ist. Schwierigkeiten in der Erkennung und Posenschätzung, durch Interaktionen der Personen oder Teilverdeckungen, können gut entgegengewirkt werden. Durch die Kombination dieser zwei vielversprechenden Methoden ist zwar das Ergebnis statistisch sehr beeindruckend, jedoch sind die Hardware-Anforderungen dieses neuen Vorgehens noch immens hoch.

2.5.4 *GrabCut*

Als gänzlich anderer Ansatz für die Lösung der Personenerkennung und -segmentierung kann durchaus auch der GrabCut Algorithmus nach Rother, Kolmogorov und Blake (2004) verwendet werden. Hier wird nicht direkt eine Personenerkennung durchgeführt, jedoch wird eine interaktive Hintergrundsegmentierung vollzogen, die – je nach Interaktion – den (menschlichen) Vordergrund extrahiert. Dabei werden mehrere Segmentierungsverfahren aus der Bildverarbeitung vereint. Nachdem der Nutzer eine Bounding-Box um das zu extrahierende Objekt zieht, wird das Bild innerhalb der Box unter Berücksichtigung von Farbähnlichkeiten (Textur), Kontrastinformationen (Kanten) und „Border Matting“ (Weichzeichnung und Verblendung des Objektumrisses) segmentiert.

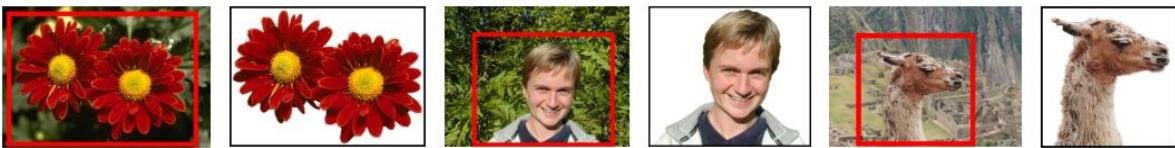


Abbildung 2.5-3: GrabCut Beispiele. (Rother, Kolmogorov & Blake (2004))

Die Ergebnisse sind optisch vielversprechend (Abbildung 2.5-3). Diese werden besonders durch iterative Interaktionen des Benutzers zu diesem Maß an Genauigkeit gebracht. Eine Echtzeit-Segmentierung ist mit GrabCut daher nicht vorgesehen.

2.5.5 *Hintergrundsubtraktion*

Ähnliche Ansätze wie GrabCut versuchen den Hintergrund eines Bildes zu subtrahieren. Dies kann auf Basis von Vorwissen durchgeführt werden (KaewTraKulPong & Bowden, 2002) oder aber der Hintergrund wird anhand einer Bildfolge a priori gelernt, um danach das Bild in Hintergrund (statisch) und Vordergrund (dynamisch) (Godbehere, Matsukawa & Goldberg, 2012) zu segmentieren. Auch diese Verfahren bieten eine Möglichkeit zur Extraktion von menschlichen Texturen aus Bilddaten. Insgesamt wird dabei der geschickte Einsatz von Vorwissen genutzt, was auch bei GrabCut der Fall ist. Bei GrabCut wird sich einer Nutzerinteraktion bedient, da dieser genau weiß, welches Objekt den Vordergrund darstellt und vom Hintergrund extrahiert werden soll. Bei der generellen Hintergrundsubtraktion nutzt man nun das Wissen um den Hintergrund, sodass später beispielweise mit einfacher Subtraktion des Hintergrundes auch automatisch der Vordergrund freigestellt ist.

2.5.6 *Deep Randomized Decision Forest mit Nutzung von Tiefeninformationen*

Der Ansatz zur Personenerkennung mit einem Deep Randomized Decision Forest und Tiefeninformationen ist im Bereich der Konsumenten-Technik wohl der bekannteste. Die Veröffentlichung von Shotton et al. (2013) ist nicht nur tausendfach zitiert, sondern die bahnbrechende Methode die hier vorgeschlagen wird, ist auch in Microsofts Kinect v2 verbaut.

Entscheidungswälder (Decision Forests), die aus einer Menge einfacher Entscheidungsbäume bestehen (Abbildung 2.5-4 rechts), werden als Mehr-Klassen-

Klassifikatoren genutzt. Jeder der Entscheidungsbäume repräsentiert dabei einen bestimmten Körperbereich und gibt eine Wahrscheinlichkeitsverteilung zurück, wenn er pro Pixel durchlaufen wird. Die gemittelte Verteilung aller Entscheidungsbäume ergibt dann letztendlich die Klassifikation des Pixels im menschlichen Kontext. Dieses Vorgehen wird Bildweise für jeden Pixel durchlaufen, womit keine temporale Kausalität besteht – es wird für jedes Bild unabhängig von vorherigen Bildern entschieden, ob und wo ein Mensch zu sehen ist. Auf Grund der Pixelklassifikation, einem darunterliegenden Mensch-Modell und den Tiefendaten kann somit auch ein sehr robustes 3D-Skelett mit Gelenken berechnet werden.

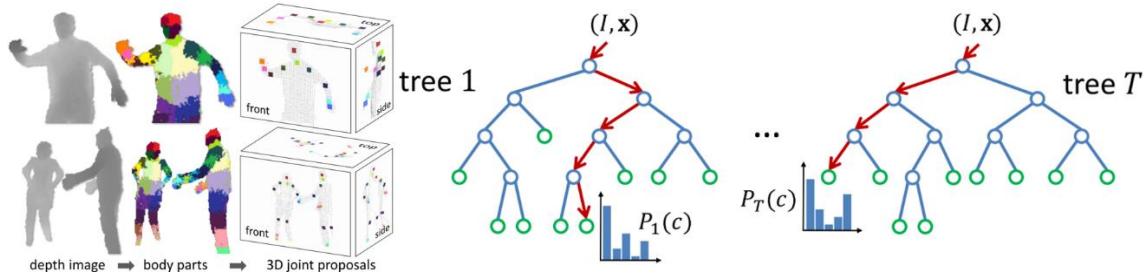


Abbildung 2.5-4: *Links: Auf Basis eines einzelnen Tiefenbildes werden Personen erkannt, eine Bestimmung der zugehörigen Körperteile wird vorgenommen, und es werden 3D-Positionen der Gelenke bestimmt. Rechts: Beispielhafte Darstellung eines Entscheidungswaldes. (Shotton et al., 2013)*

Da die hardwarelastige Posenschätzung auf einfach zu verarbeitende per-Pixel-Klassifikationsprobleme reduziert wird, kann der Algorithmus auf Konsumentenhardware, wie der Kinect, in Echtzeit mit bis zu 200 Frames pro Sekunde (FPS) verarbeitet werden. Diese pixelweise, surjektive Abbildung von Pixeln auf Körperteile lässt es außerdem als bisher einziger der hier vorgestellten Ansätze zu, direkt die menschen-zugehörige Textur zu extrahieren. Die Methode ist jedoch stark angewiesen auf die Kamera-Technik der Kinect, da sowohl Farb- als auch Tiefeninformationen benötigt werden. Außerdem kann die Methode auf Grund der Struktur der Kinect und der Synchronisierung der Bildströme nicht genutzt werden sobald sich die Kamera bewegt, womit point-of-view-mäßige Simulationen ausgeschlossen werden. Die Kalibrierung von zwei Kameras ist ebenfalls kein triviales Problem. Trotz der Dreidimensionalität der Schätzung ist es eine Schätzung aus nur einer Perspektive. Das bedeutet, dass trotz Tiefeninformationen nicht der gesamte Körper dargestellt werden kann, da beispielsweise keine Texturinformationen aus der gegenüberliegenden Richtung der Kinect aufgenommen werden können. Hierzu gibt es jedoch bereits aktuelle Arbeiten (Jones et al., 2014) (Rietzler, Geiselhart, Thomas & Rukzio, 2016) (Berger et al., 2011) (Asteriadis et al., 2013) (Kreylos, 2009), die sich damit befassen, Daten aus mehreren Kinect-Sensoren zu fusionieren.

2.6 Modularität in virtuellen Technologien

Eine erhöhte Modularität in Systemen wird bereits nach frühen Arbeiten von Parnas (1972) und Schoett (1986) als äußerst erstrebenswert angesehen und hat zwei große Vorteile:

1. Erhöhte Wiederverwendbarkeit
2. Reduzierte Komplexität

Im Bereich von VR-Technologien ist man ebenfalls dieser Meinung. Aktuelle fachspezifische Publikationen von Bär et al. (2017), Antakli et al. (2017), Behr et al. (2017), Alvarado et al. (2017), Allard, Lesage und Raffin (2010) und Bierbaum et al. (2001) unterstützen diese Aussagen. Beim Vergleich der Anwendung von Modularität dieser

STAND DER WISSENSCHAFT UND TECHNIK

Autoren geht gleichzeitig auch hervor, dass es unterschiedliche Ebenen von Modularität in VR-Technologien gibt. In der vorliegenden Thesis werden diese wie folgt untergliedert:

- High-Level – Inhaltliche Ebene
- Mid-Level – Middleware
- Low-level – Systemarchitektur

2.6.1 *High-Level – Inhaltliche Ebene*

Auf inhaltlicher Ebene einer VR-Anwendung können Nutzer virtuelle Gegenstände modular verwenden. Es wird an diesem Punkt als höchste Ebene der Modularität angesehen, da alle anderen Arten von Modularität hier bereits festgelegt worden sein müssen. VR ist dabei besonders für Design-Anwendungen prädestiniert. Beispiele sind Projekte von:

- Feretti et al. (1999), bei denen durch VR-Technologie Robotersysteme modular modelliert werden können,
- Peng, Wang, Liu und Yu (2010), die es durch virtuelle modulare Objekte Designern ermöglicht, verschiedenste Arten von Befestigungen zu erstellen,
- Nigolian et al. (2017), in deren Anwendung Einrichtungsgegenstände aus diversen robotischen Modulen zusammengesetzt werden können
- und Pai et al. (2016), in deren VE die Planung einer automatisierten Fabrik mit modularen Maschinen ermöglicht wird.

Bei allen diesen Projekten ist jedoch das eigentliche VE von vorneherein festgelegt.

2.6.2 *Mid-Level – Middleware*

Um auch Nicht-VR-Spezialisten zu befähigen, ganze VEs individualisiert zu erstellen, wird dedizierte Middleware erforscht. Diese funktioniert meist nach einem Baukastenprinzip, sodass nach einer Auswahl von VR-Modulen in der Middleware-Anwendung dann ein VE erschaffen wird. Diese mittlere Ebene ist daher softwarebasiert. Solche Anwendungskonzepte werden in folgenden Publikationen dargestellt:

- Bierbaum et al. (2001) stellen wohl eine der bekanntesten dieser Middleware-Plattformen – VR Juggler – vor. Mit dieser Softwareumgebung lassen sich VR-Anwendungen unabhängig von speziellen VR-Betriebssystemen erzeugen.
- Hashemipour, Manesh und Bal (2011) stellen ein Projekt dar, welches sich zwischen High- und Mid-Level einordnen lässt. Den Anwendern wird die Möglichkeit gegeben ein eigenes VR-System zu erstellen. Die zu Verfügung gestellten Module sind jedoch schon themenspezifisch auf den Bereich der Erstellung von technischen Laboren ausgerichtet.
- Tramberend (1999) beschreibt die Entwicklung seines verteilten VR Frameworks Avocado, in welchem er besonders die Vernetzung der Module und die Datenverteilung herausstellt. Der Datenfluss im VR-Bereich wird dort als nicht-triviales Problem dargestellt.
- Allard, Lesage und Raffin (2010) nehmen diesen Gedanken auf und fokussieren sich auf High-Performance-VR-Anwendungen. Im Zuge dessen nutzen sie Modularität in ihrer Middleware und gleichzeitig in der damit erstellten VR-Anwendung, um das entstehende VE ebenfalls zu verteilen. Damit umgehen sie VR-typische Performanzprobleme.

Mit einer modularen Middleware ist es also nicht nur möglich Einfachheit in die Erstellung von VR-Systemen zu bringen. Gleichzeitig ist eine modulare Aufteilung von VR-Komponenten auch nützlich bezüglich einer späteren Verteilung auf mehrere Sub-Systeme. Damit sind solche Middleware-Lösungen ein guter Kompromiss zwischen Konsumenten-

STAND DER WISSENSCHAFT UND TECHNIK

Systemen um VEs zu generieren und der Umgehung von Performanzproblemen von VR-Anwendungen. Durch die Verteilung der Module im Netzwerk wird das gesamte System in der Breite skaliert.

2.6.3 Low Level – Systemarchitektur

Die tiefste Ebene der Modularisierung von VR-Systemen beschreibt nach dieser Analogie die Grundkomponenten der Architektur eines VR-Systems. In diesem Bereich müssen jedoch nicht nur VR-spezifische Arbeiten betrachtet werden, sondern auch generelle Konzepte bezüglich modularer Architekturmödellierung. Hierzu zählen Themengebiete wie verteilte Systeme, Serviceorientierung, modulares Programmieren und Softwaresystemtechnik. Bezuglich verschiedener VR-spezifischer Ansätze können folgende drei genannt werden, um die Spanne an Möglichkeiten zu skizzieren:

- Sung und Ou (2001) beschreiben eine modularisierte und verteilte VR-Anwendung, die 3D Design von Gegenständen für mehrere Nutzer zulässt. Einzelne Module der VR-Pipeline sind dabei ausgelagert und werden von jedem Client als externer Service wieder zusammengeführt. Eine Komponente ist beispielsweise das Authoring des VE, welches als unabhängiger Service mit dem Gesamtsystem kommuniziert.
- Pavlik und Vance (2010) illustrieren eine modulare Implementierung eines Systems, bei der die Hardware der VR-Technologie selbst modular ist. Bei ihrem Projekt wird der Ansatz anhand von einer Nintendo Wii²⁰ Remote-Fernbedienung gezeigt, die modular in ein VR-System integriert und für Kopf-Tracking genutzt wird.
- Mulder und Boscker (2004) wiederum erweitern ein VR-System nicht durch neue kompatible Module, sondern haben ein System entworfen, welches mehrfach die gleiche Technik einbinden kann. So ist es möglich einen VR-Arbeitsplatz an mehrere andere VR-Arbeitsplätze der gleichen Art zu koppeln, wie in Abbildung 2.6-1 schematisch von den Autoren dargestellt.

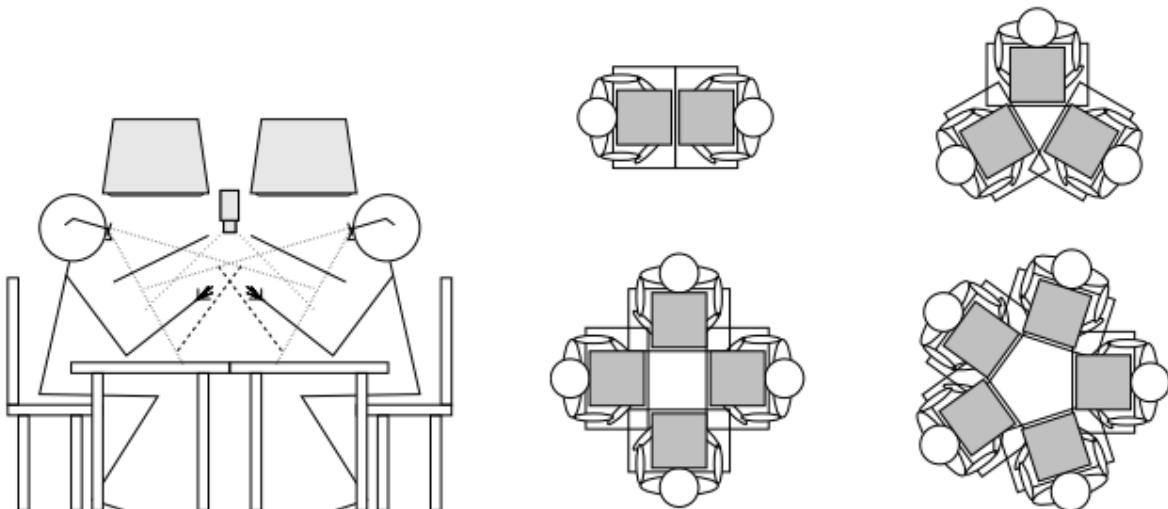


Abbildung 2.6-1: Darstellung der Modularität eines VR-Arbeitsplatzes. (Mulder & Boscker, 2004)

Insgesamt wird zwischen diesen Arbeiten also unterschieden zwischen:

- modularer Software-Architektur,

²⁰ <https://www.nintendo.de/Wii/Wii-94559.html> (8/03/2018)

STAND DER WISSENSCHAFT UND TECHNIK

- modularer Hardware-Architektur, im Sinne der Definition von Schnittstellen zur Anbindung neuer Hardware
- und modularer Hardware-Architektur, im Sinne von einem ganzen abgekapselten System, welches nach Belieben in Systeme gleicher Art integriert werden kann.

Da sich die Popularität dieser Arbeiten am ehesten als niedrig einzuordnen ist, ist dies ebenfalls ein Anzeichen dafür, dass der Low-Level-Bereich der Modularisierung auf Spezialwissen ausgerichtet ist. Damit kommen dann durchaus Konzepte aus den jeweiligen Spezialgebieten – wie Software-Engineering, Rechnernetze, Datenstrukturen und viele mehr – zur interdisziplinären Anwendung im VR-Bereich. Viele dieser Konzepte beruhen auf Design Patterns (Muster). Gängige Architekturmuster die verwendbar wären, um die Modularität eines Systems zu unterstützen, sind beispielsweise aus dem Bereich der verteilten Systeme, wie Client-Server-Architektur oder auch eine serviceorientierte Architektur. Andere Konzepte die sich nicht direkt auf konkrete Muster zurückführen lassen sind modulares Programmieren oder die systemische Modellierung des Systems in Module gemäß ihrer Funktionalität.

Bei **verteilten Systemen** handelt es sich nach Tanenbaum und Van Steen (2007) um Systeme, deren Komponenten auf verschiedenen Computern laufen können, aber für den Endnutzer als ein einziges System dargestellt werden. Dadurch können Prozesse unabhängig voneinander, also echt nebenläufig berechnet werden, womit ein scheinbar einziges System die Rechenlast auf mehrere Computer verteilt. Wie in der bereits genannten Arbeit von Sung et al. (2001) skaliert ein System dabei sehr gut in der Breite, wodurch wiederum die Rechenlast geschickt auf mehrere physische Computer aufgeteilt werden kann. Dies ist besonders bei aufwendigen Berechnungen, wie VR-Anwendungen, von großem Vorteil. Problematisch sind bei der Verteilung offensichtlich zwei Aspekte. Die einzelnen Prozesse des Systems müssen 1) so unabhängig wie möglich definiert werden, weil 2) zu einem gewissen Zeitpunkt alle Prozesse miteinander kommunizieren müssen, um die Daten zusammenzuführen. Die Kommunikationsstruktur muss so simpel wie möglich gehalten werden, weil komplexe Kommunikationsstrukturen Einbußen in anderen Bereichen des Systems mit sich bringen könnten (Wartbarkeit, Erweiterbarkeit). Dies würde wiederum konträr zum eigentlichen Sinn der Modularität stehen. Dabei spielt auch die Synchronisation des Systems eine entscheidende Rolle, damit beispielsweise eine Parallelisierung möglich gemacht werden kann.

Eine **Serviceorientierung** einer Architektur wird von Perrey und Lycett (2003) vor allem als eine gute Ausdefinierung der Funktionalität in einzelne, unabhängige Dienste (Services) beschrieben. Diese Dienste kommunizieren miteinander, um beispielsweise Daten zu übermitteln oder Aktivitäten anderer Services zu koordinieren. Auch hier ist die Kommunikationsstruktur eine wichtige Komponente. Diese wird jedoch durch die Unabhängigkeit der Service-Module vereinfacht, da sichergestellt ist, dass jeder Service einen kleinen Teil der Funktionalität des Systems autonom bereitstellt. Es müssen lediglich einfache Service-Anfragen über wohl-definierte Schnittstellen gesendet werden. Damit will man auch eine Unabhängigkeit bezüglich eingesetzter Technologien – wie Hardware – schaffen. Ein essenzieller und aufwändiger Teil der Planung und Implementierung einer serviceorientierten Architektur ist folglich die vorherige Definition der Funktionalitäten und Schnittstellen.

Bei einer **Client-Server-Architektur** werden zwar ebenfalls bestimmte Aufgaben ausgelagert, jedoch nicht im Sinne von kleineren abgeschlossenen Diensten, sondern von ganzen Anwendungen. Diese umfangreicheren Anwendungen laufen auf spezialisierten Servern, die angepasst und oft auf spezielle Technologien angewiesen sind. Bei beiden

vorgestellten Architekturen ist es wichtig, sie unabhängig von physischen Computern zu sehen, da auch eine Verteilung des Systems auf dem gleichen Computer denkbar ist.

Auch **modulares Programmieren** nach Parnas (1972) kann als Mittel genutzt werden, um eine modulare Systemarchitektur zu erzeugen. Es ist ein abstrakter Überbegriff für alle Techniken, die die Funktionalitäten eines Programms in möglichst unabhängige und austauschbare Module aufteilen. Darunter fallen auch die bereits genannten Techniken. Da dieses Konzept nicht auf ein spezifisches System oder ein Anwendungsgebiet abzielt, ist auch der Einsatz in VR-Systemen möglich. Als technisches Mittel um die theoretische Denkweise von modularem Programmieren zu realisieren, bietet sich die **systemische Modellierung** einer Architektur (Wendt, 2013) (Tabeling, 2006) nach Fundamental Modeling Concepts (FMC) an. Diese Konzepte wurden von Knöpfel, Grön & Tabeling (2005) vorgestellt, und bieten ein Framework für die ganzheitliche systemische Betrachtung eines Systems mit dem Fokus der Modularisierung nach Funktionalität und nicht nach Technik.

Insgesamt kann also die Aussage getroffen werden, dass Modularisierung im Bereich VR ähnliche Vorteile bietet wie in anderen Bereichen auch, nämlich die Erhöhung der Wiederverwendbarkeit von Systemkomponenten und die Reduzierung der Komplexität eines Systems. Zu letzterem fällt nach Zusammenfassung dieses Kapitels auf Mid- und Low-Level Modularisierung durchaus auch die Reduzierung der Rechenlast.

Ein konkretes Beispiel, welches diese Vorteile sehr deutlich zeigt, ist das ARVIDA-Projekt²¹. In diesem Projekt wird von Behr et al. (2017) eine modulare Referenzarchitektur beschrieben, die gezielt für virtuelle Techniken bestimmt ist. Hoffmann, Schreiber, Zimmermann und Kürl (2017) stellen anhand dieser konkreten Referenzarchitektur besonders die Nachhaltigkeit noch einmal heraus.

2.7 Evaluation von virtueller Realität

Die Anzahl der Faktoren die ein erfolgreiches VR-System auszeichnen ist enorm hoch, mannigfaltig und stark abhängig von der Intention des Systems und der Betrachtungsweise der Evaluation. Im Folgenden wird sich auf drei menschzentrierte Aspekte – insbesondere bezüglich MR – fokussiert:

1. Immersion
2. Präsenz
3. Akzeptanz von MR-Technologie

2.7.1 Messung von Immersion

Aus Slaters Publikation (2003) geht hervor, dass die Terme Immersion und Präsenz im VR-Kontext fälschlicherweise oft synonym gebraucht werden. Er grenzt diese beiden Begriffe jedoch voneinander ab und definiert Immersion als das objektive Niveau der sensorischen Umsetzung eines VE. Damit ist Immersion ein eher technisches Konzept.

Die Qualität der Immersion einer VR ist nach Bowman und McMahan (2001) daher stark abhängig von der Rendering-Software und der Display-Technologie. Die visuelle Immersion ist dabei ein Teilespekt von Immersion und nach Bowman und McMahan messbar anhand von Komponenten wie:

- Field of View (FOV),
- Field of Regard (FOR),
- Displaygröße und -auflösung,
- Stereoskopie,

²¹ https://www.dfki.de/web/aktuelles/dfki-auf-der-cebit-2015/index_html/arvida (8/03/2018)

- Realismus des simulierten Lichts,
- und Frame Rate.

Andere Komponenten technischer Art sind ebenfalls denkbar. Explizit gesehen ist die Evaluation der Immersion daher die Messung dieser Komponenten. Implizit ist es eine Messung die zeigt, dass mit steigender Immersion immer mehr echte Sinneseindrücke substituiert werden durch virtuelle Pendants (Mestre, Fuchs, Berthoz & Vercher, 2006).

2.7.2 Messung von Präsenz

Während der Begriff der Immersion durchaus technischer Natur ist, ist Präsenz nach Autoren wie Mestre, Fuchs, Berthoz und Vercher (2006), Biocca und Delaney (1995) und Schubert, Fiedmann und Regenbrecht (2001) ein subjektiv empfundenes Konstrukt. Im Vergleich zur Immersion – welche durchaus auch nutzerzentriert gemessen werden kann (Jennet et al., 2008) – ist die Präsenz ausschließlich mithilfe der subjektiven Wahrnehmung von Nutzern messbar.

Nach Biocca und Delaney (1995) ist die Präsenz zudem ein Konstrukt mit verschiedenen Ebenen und Dimensionen. Schubert et al. (2001) versuchen daher die verschiedenen Faktoren die für Präsenz verantwortlich sind zu kategorisieren und schlagen folgende drei Unterkategorien für Präsenz vor:

- 1) *Spatial Presence* – Der Eindruck im virtuellen Raum zu sein, anstatt etwas von außen zu steuern (nach Slater, Usoh und Steed (1994)).
- 2) *Involvement* – Die Aufmerksamkeit auf das RE oder VE, wobei ein hohes Involvement bedeutet, dass man dem RE nur wenig Aufmerksamkeit schenkt (nach Witmer und Singer (1998)).
- 3) *Realness* – Das Empfinden von Echtheit für das VE (nach Witmer und Singer (1998)).

Für diese Unterkategorien werden in der Veröffentlichung insgesamt dreizehn Komponenten vorgestellt, womit diese Faktoren messbar gemacht werden. Eine weitere, eher generelle Unterkategorie die durch nur eine einzige Frage behandelt wird ist die selbsterklärende *General Presence*.

Es existieren bereits einige Arbeiten, die sich schon früh speziell auf das Messen und die Evaluation von Präsenz spezialisiert haben, wie von Witmer und Singer (1998), Schuber et al. (2001) und Schaik, Turnbull, Wersch und Drummond (2004), wobei letztere sogar direkt auf den MR-Bereich ausgerichtet ist. Es ist an dieser Stelle anzumerken, dass die Meinungen dabei weit auseinander gehen, sodass Slater (1999a) sogar so weit geht zu schreiben, dass er selbst das etablierte Presence Questionnaire nach Witmer nie benutzen würde, obwohl es eventuell auch zu nützlichen Einsichten führen könnte. Es spieltgelt seiner Ansicht nach nicht die eigentliche Präsenz wieder.

2.7.3 Messung der Akzeptanz von MR-Technologie

Eine weiterer zu evaluierender Aspekt eines MR-Systems, der nach Rasimah, Ahman und Zaman (2011) weitgehend unabhängig von Immersion und Präsenz ist, ist die generelle Akzeptanz eines Nutzers von spezifischer MR-Technologie. In der vorgestellten Methode und einer Beispielstudie wird die Akzeptanz von MR-Technologie anhand folgender Konstrukte evaluiert und gemessen:

- *Personal Innovativeness* – Die Bereitschaft neue Informationstechnologie auszuprobieren (nach Agarwal und Prasad (1998)). Diese wird genutzt, um individuelle Eigenschaften des Nutzers bezüglich der MR-Technologie zu messen.

STAND DER WISSENSCHAFT UND TECHNIK

- *Perceived Enjoyment* – Das Vergnügen während der VE-Erfahrung (nach Lin und Parker (2007)).
- *Perceived Usefulness* – Das Ausmaß der Verbesserung eines Ablaufs (beispielsweise im Beruf), welches der Nutzer dem MR-System zuspricht (nach Davis (1989)). Hier wird die wahrgenommene gewünschte Funktionalität der Nutzer herausgestellt.
- *Perceived Ease of Use* – Der Grad der Anwenderfreundlichkeit des MR-Systems (nach Davis (1989)). Diese trägt einen hohen Teil zur Bewertung der gesamten Zufriedenheit der Nutzer des MR-Systems bei.
- *Intention of Use* – Die Ausdrucksstärke in der die Nutzer einen möglichen, zukünftigen Einsatz des MR-Systems bekundet haben (nach Davis (1989)).

Abschließend ist zu sagen, dass Immersion, Präsenz und Akzeptanz von MR-Technologie zwar prinzipiell unabhängig betrachtet werden können, dass aber auch in den zugehörigen Quellen bereits Abhängigkeiten geschildert werden. Beispielsweise wird ein System mit höherer Immersion nach Witmer und Singer (1998) auch gleichzeitig einen höheren Grad an Präsenz schaffen und umgekehrt.

3. DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

In diesem Hauptkapitel der Thesis wird ein Konzept und ein Design einer Kommunikationsmethode vorgestellt, das die visuelle Kommunikation zwischen virtueller und realer Umgebung verbessern soll. Diese Methode wird in einem Prototyp umgesetzt, sodass dieser ebenfalls konzipiert und designt wird. Auf Grund des hohen technischen Anspruchs der Integration von Funktionalität im VR-Bereich ist der umfangreichste Abschnitt dieses Kapitels der technischen Dokumentation des Prototyps gewidmet. Im Anschluss wird ein konkreter Anwendungsfall dargestellt. Abschließend wird der Prototyp zusammen mit dem Szenario des Anwendungsfalls genutzt, um die Kommunikationsmethode zu evaluieren. Die finale Umsetzung kann zudem in einem kurzen, ergänzenden Video in Anhang A oder online²² betrachtet werden.

Abschnittübergreifend wird in diesem Hauptkorpus der Arbeit auch näher auf die in Kapitel 1.4 genannten zwei wesentlichen Iterationen des Thesisprozesses eingegangen. Der iterative, scrum-ähnliche Ansatz der in der Umsetzung der Thesis verfolgt wurde spiegelt sich wie dort beschrieben zwar auch auf tieferen Ebenen wieder, jedoch lassen sich alle grundlegenden Entscheidungen in genau eine der beiden in Anhang B gezeigten Iterationen einordnen. Im Folgenden wird sich daher auf die Unterteilung in die Hauptiterationen beschränkt.

3.1 Konzeption und Design des Kommunikationskonzepts und eines Visualisierungssystems

Ein maßgeblicher Bestandteil der Konzeption und des Designs der Methode zur visuellen Kommunikation zwischen den Umgebungen fällt auf den Bereich der Anforderungsanalyse zurück. In anfänglichen Gesprächen mit den Projektverantwortlichen des DFKI ging besonders die hohe Relevanz der visuellen Darstellung eines Probanden für den Experimentator hervor. Der Proband, welcher sich in dem immersiven VE befindet, sollte durch einen Experimentator beobachtet werden. Diese Relevanz konnte auch durch eine anschließende Literaturrecherche begründet werden. Die **erste Iteration** wurde daher genutzt, um vorerst Anforderungen an ein System für eine einseitige Beobachtung zu erarbeiten. Diese wurden aus mehr als 20 Veröffentlichungen (Anhang A) grob und vereinfacht herausgestellt. Die ausgewählte Literatur ist vorwiegend interdisziplinären Charakters, da das zu erstellende Konzept nicht nur auf einen spezifischen Bereich anwendbar, sondern so generell wie möglich sein sollte.

Mit der Beschränkung auf die reine Beobachtung von Probanden und deren VE wurden zusammenfassend nach der Literaturrecherche insbesondere folgende Aspekte als äußerst wichtig eingestuft:

1. *Detaillierte und realistische Darstellung der Körperpose* – Dies ging besonders aus Veröffentlichungen hervor, die sich mit qualitativer Ergonomie- und Sicherheitsevaluationen beschäftigen.
2. *Mikro-Expressionen* – Feinste körpersprachliche Merkmale sind von großem Wert für die Kommunikation zwischen Menschen. Auch in einseitiger Beobachtung sagen diese unterbewusst viel über den Probanden und dessen Befinden in der VR aus.
3. *Kollaborationsmöglichkeit* – Die Möglichkeit mehrere Beobachter zuzulassen wurde als hilfreich empfunden.

²² <https://vimeo.com/257680437> (8/03/2018)

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

4. *Datenvizualisierung* – Einige Systeme zur visuellen Evaluation von VE nutzen die Visualisierung von weiteren szenario- und sensorspezifischen Daten. Diese wirken sich positiv auf die Entscheidungsstützung des Systems aus.
5. *Nachträgliche Evaluation* – Die Möglichkeit eine Session nachträglich anzuschauen oder durchzugehen ist besonders förderlich, wenn mehrere Sessions aufeinanderfolgen und erst abschließend eine Gesamtevaluation stattfindet.
6. *Nachhaltigkeit* – Damit das System nicht nur auf genau ein Szenario anwendbar ist, wird oft der Aspekt der Nachhaltigkeit gefordert. Dies ist nicht nur im Sinne von Generalität, sondern ebenfalls im Sinne von Erweiterbarkeit zu sehen.
7. *Visuelle räumliche Interaktion* – In mehreren Publikationen wurde die Notwendigkeit beschrieben, dass in Experimenten mit VR einfache und direkte Zeigegesten aus dem RE in das VE wünschenswert sind.
8. *Menschlicher Faktor* – Realistische menschliche Faktoren müssen nach Autoren aus Anhang A explizit miteinbezogen werden, wenn Menschen in VE beobachtet werden. Mögliche Faktoren sind Größe, Gewicht, Geschlecht und viele mehr.
9. *Individuelle und unterschiedliche Avatare* – Im Einklang mit den menschlichen Faktoren muss ebenfalls eine gewisse Individualisierung der Avatare möglich sein. Menschen mit demselben Gewicht, derselben Größe und demselben Geschlecht werden sich möglicherweise zwar ähnlich sehen, aber sich nie exakt gleichen. Der Ausdruck von individuellem und persönlichem Hintergrund sowie eigener Kultur ist eines der wichtigsten Faktoren bei der Beobachtung von individuellen Probanden.
10. *Kollisionserkennung* – In einigen eher technischen Anwendungsbereichen, wie der Ergonomie und Sicherheit, wird die Möglichkeit zur Erkennung von Kollisionen zwischen echtem Probanden und virtuellem Objekt als maßgeblich beschrieben.
11. *Kommunikation ohne die Immersion zu stören* – In diversen Experimenten psychologischer Natur ist die Immersion von VR von größter Wichtigkeit, um die Ergebnisse nicht zu verfälschen. Die visuelle Kommunikation zwischen Proband und Experimentator darf diese also nicht beeinträchtigen.

Die genannten Punkte stellen insbesondere die realitätsnahe visuelle Darstellungsform des Probanden als höchst bedeutsam für die Beobachtung heraus. Die Art der Darstellung beeinflusst die meisten dieser Aspekte des Systems maßgeblich, sodass 1, 2, 7, 8, 9 und 11 sogar fast ausschließlich durch die Darstellungsform ausgedrückt werden.

Um die Darstellung in höchstem Maße realistisch zu gestalten, und damit in direktem Maße die genannten Aspekte zu erfüllen, ist vor allem die echte Textur des menschlichen Körpers des jeweiligen Probanden geeignet. Sowohl prägnante körperliche Eigenschaften, wie Körperpose, Geschlecht, Gewicht und Größe, aber auch diffizile Eigenschaften, wie Mikro-Expressionen, persönlicher Hintergrund (beispielsweise Kleidung und Haare) und Kultur (beispielsweise Hautfarbe) werden so realistisch wie nur möglich abgebildet.

Um diese Idee weiter zu analysieren, wurde bereits vor der Implementierung der ersten Iteration ein visueller Prototyp zur Kommunikation des „Look and Feel“ entwickelt (Abbildung 3.1-1). In der weiteren Anforderungsanalyse der ersten Iteration wurde anhand dieses Bildes die Wichtigkeit des passenden Skalierungsverhältnisses sowohl von virtueller Szene als auch Proband-Darstellung deutlich. Die Skalierung ist bei herkömmlichen animierten Avataren gut regulierbar, da diese bereits digital erstellt werden. Die technische Problematik der Überblendung von echter Textur in korrekter Korrelation zur Szene wird näher in Abschnitt 3.3 erläutert.

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP



Abbildung 3.1-1: Die geplante Optik der gegenseitigen Darstellung zur Kommunikation zwischen realen und virtuellen Umgebungen ist von hohem Realismus geprägt. Durch die Echt-Textur-Repräsentation werden visuelle Aspekte, wie die Körpersprache, besonders realistisch abgebildet.

Die **zweite Iteration** wurde deutlich nutzer-zentrierter gestaltet. Da nach der ersten Iteration bereits ein funktionaler Medium-Fidelity-Prototyp implementiert wurde, konnte dieser genutzt werden, um mögliche Nutzer und fachliche Experten in direktem Maße hinzuzuziehen. Insbesondere in Experteninterviews mit Domänen-Experten (in psychologischen Experimenten und Ergonomieevaluationen) und VR-Expertern (VR-Ingenieure und VR-Anwender) wurde der Prototyp genutzt, um ein unmittelbares Feedback aus der Industrie zu bekommen. Dafür ist ein Grundaufbau eines Interviews genutzt worden (Anhang C), welcher je nach Personengruppe leicht angepasst wurde. Eine Zeitspanne von 45 Minuten pro Interview wurde als passend angesehen. Dabei ist unterschieden worden zwischen elementar wichtigen Fragen (hervorgehoben) und Fragen, die bei der Anforderungsanalyse unterstützend wirken können.

Die wichtigsten Ergebnisse der Interviews können wie folgt in zwei Bereichen zusammengefasst werden:

1. Die Darstellung echter menschlicher Texturen in virtuellen Umgebungen scheint nicht nur derzeit en vogue, sondern auch einstimmig in generellem Maße nützlich zu sein. Um in den jeweiligen spezifischen Gebieten wirklich einsetzbar sein zu können, muss die Darstellung jedoch jeweils angepasst werden. So muss beispielsweise in Ergonomieevaluationen durchaus eine technische Komponente vorhanden sein, die unverkennbar darstellt, wann bestimmte unergonomische Bewegungen gemacht werden (Vergleich mit „Jack“ aus Phillips und Badler (1988)).
2. Besonders durch die Experteninterviews wurde klar, dass nicht nur die einseitige Beobachtung des Probanden, sondern dass auch die gegenseitige Beobachtung von mindestens gleich hohem Interesse ist. Durch diesen Rückkanal der Darstellung entsteht eine gegenseitige visuelle Wahrnehmung der beiden Akteure, welche die

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

visuelle Kommunikation in großem Maße unterstützt. Bei den Befragungen zeichnete sich jedoch auch die Einschränkung ab, dass bei der Integration einer Experimentatoren-Darstellung für den Probanden in VR keinesfalls die Immersion und die Präsenz gestört werden darf. Der mögliche Nutzen einer Echt-Textur-Darstellung wurde daher noch einmal bekräftigt, ähnlich wie auch in Kapitel 2.4 beschrieben. So können wichtige Rückschlüsse über bedeutsame Faktoren wie die Körpersprache gezogen werden. Gleichzeitig wird erwartet, dass eine korrekte Platzierung und Skalierung der Experimentatoredarstellung im virtuellen Raum die Immersion nur wenig beeinflusst.

Zusammenfassend wurde sich bei dem Ablauf der Erhebung der Anforderungen an die Beobachtungsmethode stark an dem von Kotonya und Sommerville (1998) geschilderten Ablauf orientiert (Abbildung 3.1-2). Die Analyse in der ersten Iteration wurde genutzt, um schnell zu einem lauffähigen Prototyp zu gelangen. Dieser wurde in der zweiten Iteration dann genutzt, um 1) weitere Anforderungen abzuleiten, 2) bestehende Anforderungen zu verifizieren, aber auch um 3) die anfängliche wissenschaftliche und theoretische Idee der Kommunikationsmethode zu validieren. Durch den Prototyp war es einfacher für Nicht-Experten mögliche visuelle Anforderungen zu formulieren.

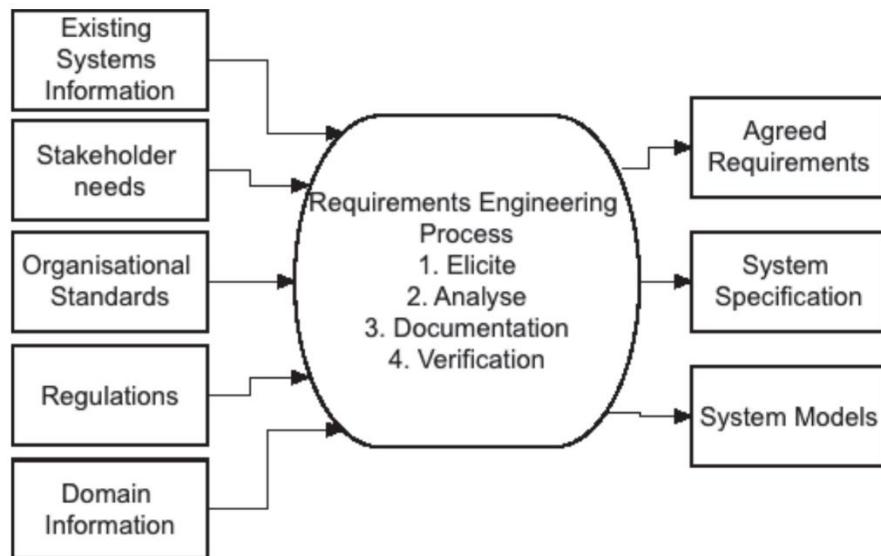


Abbildung 3.1-2: Prozess der Anforderungserhebung nach Kotonya und Sommerville (1998).

Nach der Neu-Justierung des Fokus der Arbeit nach etwa eineinhalb Monaten nach Start, wurden folgende Haupt-Aspekte zur Definition der Kommunikationsmethode festgelegt:

1. *Visuelle Darstellung* – Die echte menschliche Textur wird zusammen mit dem VE visualisiert. Dies gilt sowohl für den Probanden in der VR, als auch für 1-N Beobachter, die sich im selben Raum befinden. Dadurch kann das Uncanny Valley durchquert werden und selbst Mikro-Expressionen können vermittelt werden. Die digitale Individualisierung des Avatars wird damit ausgeschlossen, da dieses Problem wie gewöhnlich in der Realität stattfindet. Der Proband wird mit den selben Merkmalen (Kleidung, Haare etc.) wie in der Realität abgebildet.
2. *Immersion und Präsenz* – Die Immersion und die Präsenz des Probanden wird nicht gestört, da der Beobachter in korrekter Positionierung, Skalierung und zeitlichem Ablauf im virtuellen Raum dargestellt wird. Der Beobachter wird somit nicht mehr

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

implizit, sondern explizit Teil der virtuellen Welt. So wird erwartet, dass weder die Immersion noch die Präsenz beeinträchtigt wird.

3.2 Konzeption und Design des Prototyps

Nachdem nunmehr die Kommunikationsmethode konzipiert und begründet ist, wird ein Konzept eines Prototyps dargestellt, um die Methode in Anwendung zu zeigen. Die Konzeption des Prototyps hat in jedem Fall auch großen Einfluss auf die spätere Kommunikation selbst, da hier technische Grenzen auftreten könnten, welche wiederum in der Methodenkonzeption begründet werden müssen. So hat sich in den Iterationen des Projekts ebenfalls stetig das Konzept des Prototyps verändert. Die praktische Umsetzung und die dabei aufgetretenen Probleme werden im nächsten Kapitel beschrieben.

Die theoretische Konzeption für die **erste Iteration** beschränkte sich besonders auf mögliche grundlegende Techniken, die genutzt werden sollen, um 1) die Kommunikationsmethode aus Abschnitt 3.1 zu realisieren und 2) die Einbettung dieser Methode in ein übergreifendes Visualisierungssystem zu zeigen. Die gänzliche Implementierung des übergreifenden Visualisierungssystems wurde nicht angestrebt. Das Ziel der ersten Iteration war es einen Prototyp zu entwickeln der auf einem Stand ist, auf dem man weiter Schlüsse über die eigentliche Methode ziehen kann. Außerdem wird durch diese technische Konzeption bereits eine erste Machbarkeit und Umsetzbarkeit aufgezeigt.

An dieser Stelle werden dennoch in kürze die elementaren Komponenten eines solchen ganzheitlichen Systems dargestellt, um den Kontext aufzuzeigen in dem sich das Konzept der Kommunikation und der Prototyp befinden. Durch die abstrakte Modellierung (Abbildung 3.2-1), sollten so eventuelle Schwierigkeiten bei der Implementierung früh erkannt werden. Außerdem wird durch die abstrakte Modellierung bereits ein gewisser Grad an Modularität des Systems sichergestellt, sodass das System in der zweiten Iteration einfach erweitert werden konnte.

Folgende Komponenten wurden für ein Visualisierungssystem und die einseitige Beobachtungsmethode der ersten Iteration herausgestellt:

- *Proband-Bild* – Um eine Echt-Körper-Repräsentation von Probanden zu realisieren, ist es unumgänglich die echte Textur mithilfe von Sensoren aufzunehmen. Eine Variante wäre hier eine handelsübliche Webcam zu nutzen, da diese auch den low-budget Charakter der Arbeit gut trifft. Eine andere Möglichkeit stellen RGBD-Kameras, wie die Kinect-Kamera dar.
- *VE-Bild* – Da das VE ebenfalls dargestellt werden soll, muss dieses entsprechend in einer Komponente erstellt werden. Diese Komponente beinhaltet auch die weitere Technik, die zur korrekten Darstellung von VR-Inhalten gebraucht wird, wie beispielsweise Tracking und Rendering.
- *Optionale Daten* – Zusätzliche Daten können für den Experimentator während eines Experiments ebenfalls wichtig sein. Diese können aus verschiedenen Quellen stammen, sodass zusätzliche Sensoren, aber auch semantisches Vorwissen zum Ablauf der virtuellen Szene visualisiert werden.
- *Proband-Compositor* – In dieser Komponente werden die verschiedenen Datenströme miteinander zu einem einfachen zweidimensionalen Bild verschmolzen. Dieses Bild wird dann mithilfe von Streaming an Clients versendet, um letztendlich alles für den Beobachter auf einem Bildschirm darzustellen.

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

- *Datenbank* – Hier werden alle vom Compositor verarbeiteten Daten für eine nachträgliche Evaluation (beispielsweise Abspielen eines Screen Capture Videos des Experiments) gespeichert.
- *Client* – Der Client bietet sowohl die Funktionalität das Live-Experiment mitzuverfolgen und den Probanden in Echtzeit zu betrachten, als auch die Möglichkeit nach dem Experiment Daten mithilfe des Evaluation-Moduls abzurufen.
- *Evaluation* – Dieses Modul bietet die Funktionalität gespeicherte Daten aus der Datenbank zu laden und auszuwerten.

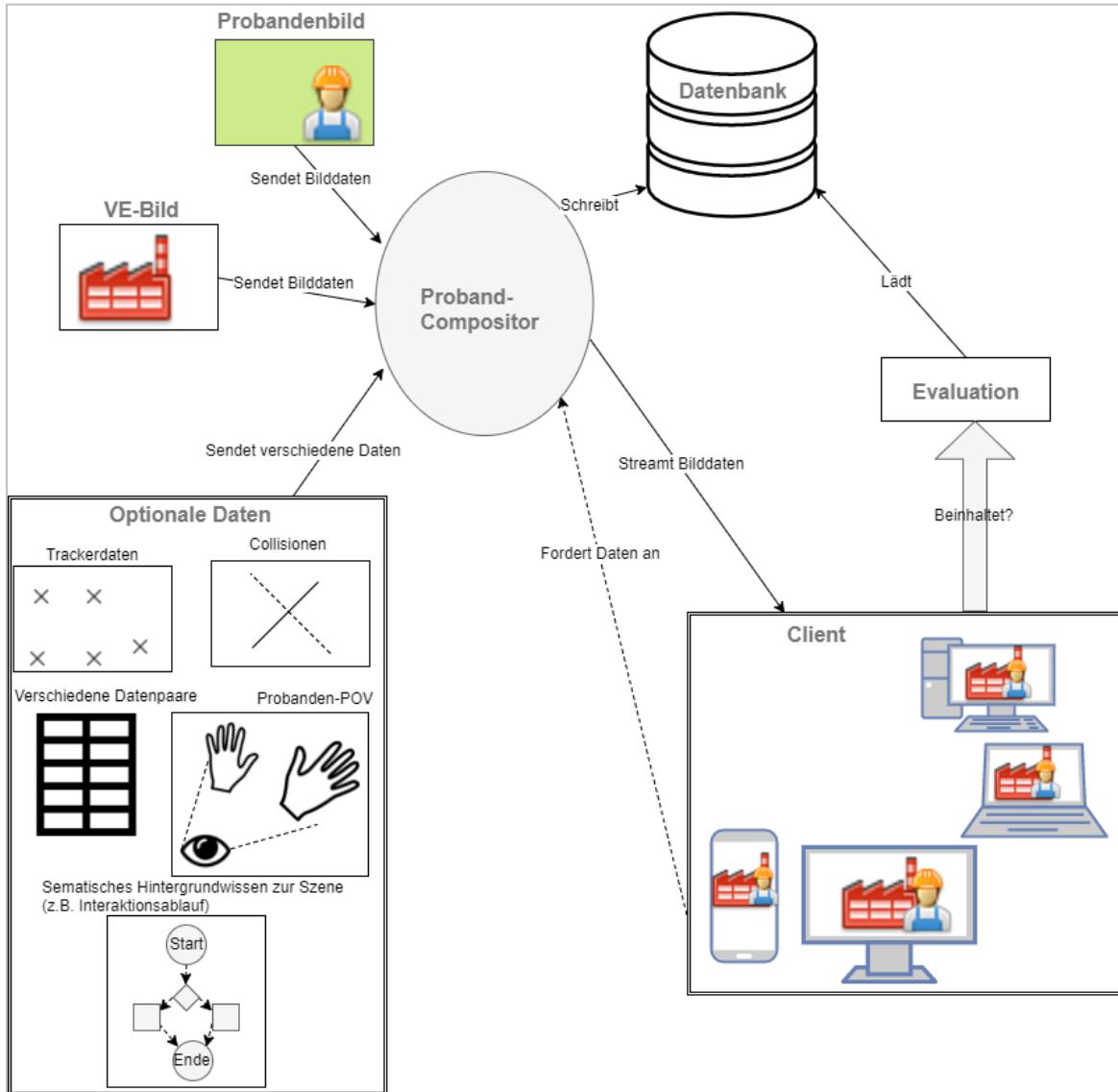


Abbildung 3.2-1: Abstrakter Aufbau des Systems mit dem Fokus auf modulare Kapselung der Funktionalität. Dieses Sub-System bildet die Darstellung des Probanden und des VE für den Experimentator (Iteration 1) ab.

Wichtig bei der Konzeption und dem Design des Prototyps ist auch die Art des Compositings der Daten. Diese Funktionalität ist die wohl elementarste Funktion für das gesamte System in Bezug auf die Kommunikation zwischen den Benutzern. Da Arbeiten technischer Art nicht immer wissenschaftlich dokumentiert und veröffentlicht werden, wurden auch unveröffentlichte, beziehungsweise teilveröffentlichte Projekte den aus Bereichen VR,

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

Computerspiele und Film-Post-Production als Basis für die Recherche herangezogen. Die Einflüsse dieser Projekte spiegeln sich besonders in der technischen Dokumentation in Kapitel 3.3 wieder. Maßgebend für die Konzeption des Compositings sind die Projekte von Owlchemy²³, Alex Anpiogov²⁴, Kert Gartner²⁵, Goran Adrinek²⁶ und Northway Games²⁷. Dabei wird nach zwei Vorgehensarten unterschieden. Die menschliche Textur wird entweder durch Greenscreen-Technik oder durch Tiefen-Compositing extrahiert. Weitere Unterscheidungen werden ebenfalls in Abschnitt 3.3 erläutert.

Nachdem der Grundaufbau des Systems aus der ersten Iteration hervorgeht, setzt die **zweite Iteration** den Fokus auf die Anwendung in Mehrbenutzerumgebungen. Dieser Mehrnutzer-Gedanke wird im späteren Verlauf näher beschrieben und beschränkt sich vorerst auf die Erweiterung des virtuellen Systems um den Experimentator als aktiven Nutzer. Da die Blickrichtung von Proband zu Experimentator eingebunden werden muss, kommen folgende Komponenten zum System hinzu (Abbildung 3.4):

- *Beobachter-Bild* – Das Bild des Beobachters muss nun ebenfalls aufgenommen werden, da es dem Probanden zur Verfügung gestellt wird.
- *Proband-VR* – Das Probanden-VR-Modul beinhaltet vor allem das Rendering des VE. Hier muss das Bild des Beobachters eingebunden werden.
- *Beobachter-Compositor* – Analog zum Proband-Compositor verarbeitet diese Komponente die Daten die nötig sind, um den Beobachter für den Probanden zu visualisieren.
- *VE-Daten* – Um den Beobachter in korrekter Positionierung und Größe innerhalb des VE darzustellen, benötigt der Compositor bestimmte Meta-Daten über die virtuelle Szene.

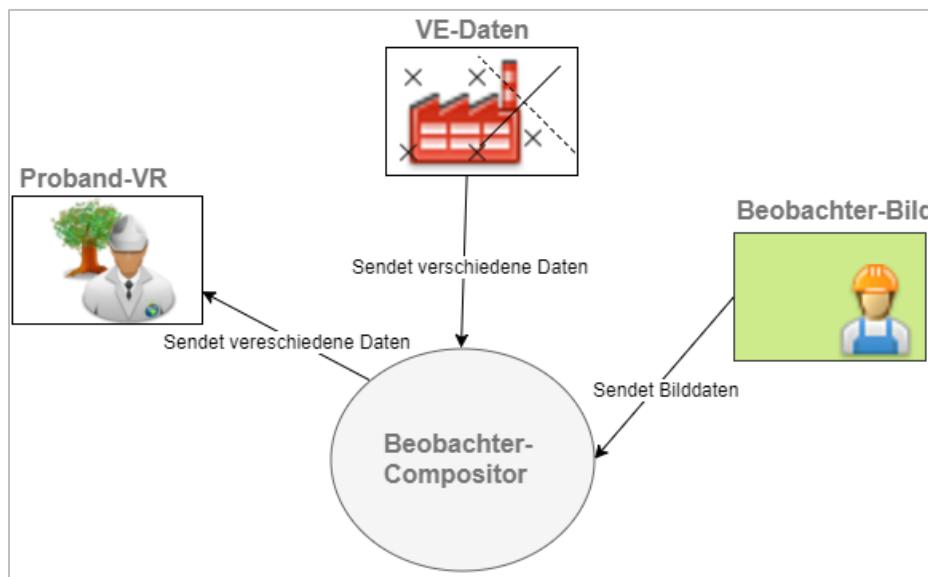


Abbildung 3.2-2: Sub-System, das die Darstellung des Experimentators für den Probanden innerhalb des VE (Iteration 2) bereitstellt.

²³ <https://owlchemylabs.com/owlchemyvr-mixed-reality-tech/> (8/03/2018)

²⁴ <https://alexeverything.net/mixed-reality-in-real-time-no-green-screen/> (8/03/2018)

²⁵ <http://www.kertgartner.com/making-mixed-reality-vr-trailers-and-videos/> (8/03/2018)

²⁶ <http://www.croteam.com/mixed-reality-serious-sam-vr-last-hope/> (8/03/2018)

²⁷ <http://northwaygames.com/how-to-mixed-reality/> (8/03/2018)

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

Obwohl diese Sub-Systeme aus Abbildung 3.2-1 und Abbildung 3.2-2 praktisch gesehen sehr nahegelegen sind, sind diese in Bezug auf die eigentliche Funktionalität völlig separiert. Dies spiegelt auch die Abbildung in Anhang D wieder, die das konzipierte Gesamtsystem zeigt, welches ohne jegliche Kommunikation zwischen den Sub-Systemen auskommt. Explizite Schnittstellen werden allerdings erst im folgenden Abschnitt 3.3 modelliert

Mit dem hier vorgestellten Konzept eines Prototyps kann das in Abschnitt 3.1 beschriebene Kommunikationskonzept praktisch angewandt werden. Die gegenseitige Echt-Körper-Repräsentation zur Verbesserung der visuellen Kommunikation zwischen virtueller und realer Umgebung ist möglich. Die technische Implementierung des Prototyps und damit der Kernfunktion – der Erweiterung der gegenseitigen visuellen Kommunikation zwischen realer und virtueller Umgebung – ist jedoch nicht trivial und wird im Folgenden nun technisch beleuchtet.

3.3 Technische Dokumentation des Prototyps

Um die Umsetzbarkeit und Anwendbarkeit des theoretischen Ansatzes zu zeigen wurde ein Prototyp implementiert. Dieses Kapitel beschreibt zum einen die zugrundeliegende Architektur des Systems, zum anderen die Implementierung selbst. Zu den angewendeten Techniken werden außerdem alternative Möglichkeiten aufgezeigt. Parallel wird dadurch begründet, warum sich für die Implementierung für eine bestimmte Technik und nicht für die Alternativen entschieden wurde. Argumentiert wird vor allem mit den in Kapitel 3.1 und 3.2 beschriebenen Anforderungen an die Kommunikationsmethode und den Prototyp, aber auch mit den in Kapitel 1.2 festgelegten Zielen für die vorliegende Master-Thesis.

Wichtig für das Verständnis des Kapitels der technischen Dokumentation des Prototyps ist, dass sich zugegen nun auf die prototypische Umsetzung der Kommunikationsmethode beschränkt wird. Der entwickelte Prototyp, oder nach Ries (2009) auch das minimal überlebensfähige Produkt (Minimum Viable Product, MVP), bildet daher nicht das in den Abschnitten 3.1 und 3.2 beschriebenen Gesamtsystems ab. Es werden nur die nötigsten Teile des Gesamtsystems implementiert, da nicht alle Module (wie beispielsweise Datenbank, Evaluationsmodul und Teile des Clients) für die wissenschaftliche Relevanz der Thesis notwendig sind.

Unterteilt ist dieses Haupt-Kapitel in acht Abschnitte, beginnend mit der Aufbaubeschreibung des Prototyps und dessen Unterteilung der Komponenten in möglichst unabhängige und wiederverwendbare Module. Aus diesem Abschnitt wird die Wahl der weiteren Abschnitte verständlich. Es folgt ein Überblick über die VR-Implementierung unter Zuhilfenahme eines Szenarios. Danach wird die Bilddatengewinnung aus der Unity-Szene beschrieben. Diese Daten bilden zusammen mit den danach beschriebenen Kinect-Daten die Basis, um in Abschnitt 3.3.4 den Ablauf des Compositings zu erläutern. In den Abschnitten 3.3.2.1 bis 3.3.4 liegt der Fokus besonders auf der Implementierung der Beobachtungsrichtung von Experimentator zu Proband gelegt (Abbildung 3.3-1), welche in der **ersten Iteration** fokussiert wurde.

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

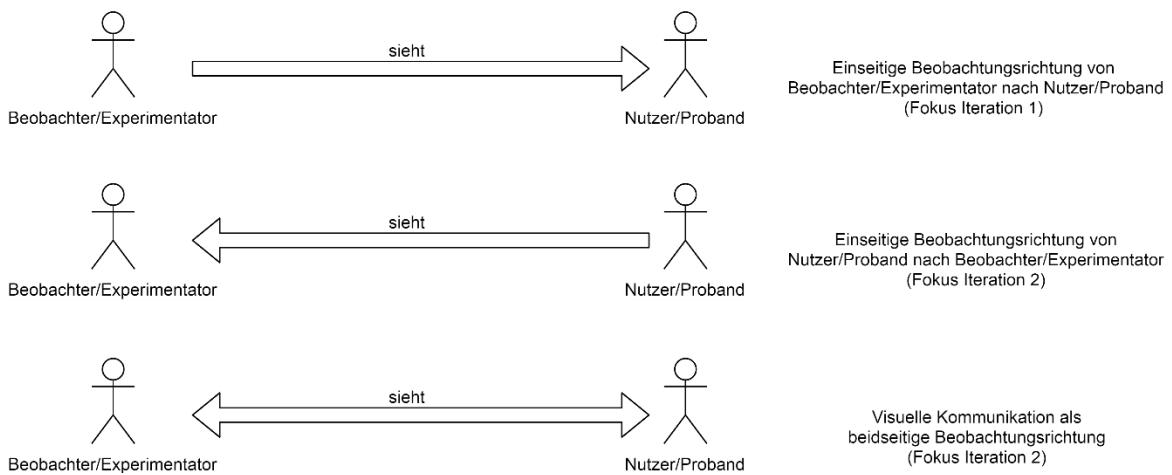


Abbildung 3.3-1: Definition der Beobachtungsrichtungen und dem Verhältnis von Beobachter/Experimentator zu Nutzer/Proband in einem immersiven VE.

Abschnitt 3.3.5 legt den Schwerpunkt auf die Erweiterung des Systems in der **zweiten Iteration**. Hier ist die Modularität und die daraus resultierende Nachhaltigkeit des Systems bereits bemerkbar. Codestücke von verschiedenen Komponenten aus der ersten Iteration können mit nur leichten Abänderungen wiederverwendet werden, um die Funktionalität zu erweitern. Dadurch wird die beidseitige visuelle Kommunikation zwischen Beobachter und Nutzer hergestellt. Da nach diesem Abschnitt nun die gesamte Funktionalität der besagten Kommunikationsmethode gewährleistet ist, wird der finale Datenaustausch der Komponenten des Systems darauffolgend beschrieben. So werden die technischen Schnittstellen klar herausgestellt. Diese gehen abstrakt zwar schon aus Abschnitt 3.3.1 hervor, sind jedoch für eine wirkliche Wiederverwendung von Codefragmenten dann auch anwendungsnahe definiert. Die technische Dokumentation endet mit einem kurzen Abschnitt bezüglich weiterer Mehrbenutzerumgebungen.

3.3.1 Modulare Architektur und Modulkommunikation

Die hohe Relevanz des Aufbaus des Prototyps kann besonders im Zusammenhang mit den in Kapitel 2.6 genannten Aspekten nachvollzogen werden. Die modulare Modellierung steht danach folglich auch in direkter Korrelation mit Ziel Nummer Vier dieser Thesis (Tabelle 1.2-1) und der daraus resultierenden Anforderung an die Nachhaltigkeit (Kapitel 3.1).

Bevor nun die modulare Architektur des entworfenen Prototyps mithilfe der FMC dargestellt wird, wird der hypothetische Einfluss des alternativen Aufbaus – der monolithischen Architektur – beschrieben. Trivialerweise ist die vorherige Gliederung in Module ein nicht zu unterschätzender Mehraufwand in der Planung. Die Frage die daher schnell aufkommt ist: Lohnt sich dieser Aufwand? Es handelt sich in der Tat nicht um die Implementierung einer versandfertigen Software- oder Hardware-Lösung, sondern „nur“ um einen Prototyp. Jedoch sind gerade bei modernen Vorgehensmodellen in der Softwaretechnik iterative und agile Abläufe und Produktzyklen maßgebend. Darunter fällt auch der iterative Ansatz, der in der vorliegenden Thesis verfolgt wird. Wird der Prototyp also nicht als „Wegwerf-Prototyp“ gesehen, sondern als auszubauender Machbarkeitsnachweis und MVP, dann lohnt der anfängliche Mehraufwand. Eine spätere Umstellung des Prototyps von monolithisch auf modular könnte im Vergleich zu den anfänglich geringeren Investitionen einen exponentiellen Anstieg der Kosten verursachen. Die Möglichkeit Runde um Runde weitere Funktionalität hinzuzufügen wird sich auch im weiteren Verlauf der Thesis als besonders hilfreich herausstellen. In der zweiten Iteration wird das Baukastenprinzip des Systems

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

bereits aufgegriffen, sodass Code aus der ersten Iteration direkt genutzt werden kann, um neue Funktionalität in der zweiten hinzuzufügen.

Die Modularität wurde in Kapitel 2.6 in drei verschiedene Ebenen aufgeteilt – Inhalt, Middleware und Systemarchitektur. Für den erarbeiteten Prototyp als Proof of Concept des Kommunikationskonzepts ist die darunterliegende Struktur und Wiederverwendbarkeit der Implementierung nur von geringer Bedeutung. Die inhaltliche Ebene wird aus diesem Grund vernachlässigt. Es muss nicht gewährleistet sein, dass der Inhalt eines möglichen Anwendungsfalls von modularem Charakter ist, da die Nachhaltigkeit des Inhalts keinerlei Bewandtnis hat. Es ist außerdem nicht das Ziel dieser Arbeit ein Middleware-Produkt zu schaffen, das die Nutzer dazu befähigt individualisierte VEs zu erstellen. Infolgedessen ergibt sich die Modularität des Prototyps auf der Architekturebene. Dieser Aspekt des modularen Aufbaus steht in direkter Verbindung mit Ziel Nummer vier aus Tabelle 1.2-1 und ist gleichzeitig auch alleiniger Indikator zur Erfüllung des Ziels.

In Kapitel 2.6.3 wird die Architekturebene nochmals in Soft- und Hardware-Architektur und -Kapselung unterteilt. Während eine Hardware-Modularität zwar äußerst wünschenswert für das genannte Teilziel der Arbeit wäre, muss sich auf Grund der zeitlichen und monetären Begrenzung auf die Software-Modularität beschränkt werden. Zum einen wäre der Aufwand der Implementierung um ein Vielfaches höher, zum anderen müssten zur Evaluation auch diverse alternative Hardware-Komponenten angeschafft werden. Ein serviceorientiertes System wäre von Nöten, um optimal auf verschiedene Anforderungen und Spezifikationen der unterschiedlichen Hardware einzugehen. Hingegen reicht es für den Zweck der Evaluation der Kommunikationsmethode zwischen realer und virtueller Umgebung, aus sich auf ein Hardware-Set-up zu beschränken. Die Wiederverwendbarkeit wird demnach als Kompromiss über die Softwarekomponenten erreicht.

Als Konzept zur Verteilung der Komponenten als einzelne Systeme wird vorwiegend das Client-Server-Konzept genutzt. Dadurch wird im Vergleich zu Shared Memory²⁸ oder anderen intramaskinellen Konzepten eine theoretische Verteilung innerhalb eines Netzwerkes ermöglicht. Implizit wird dadurch auch wieder die vorteilhafte vertikale Skalierung des Gesamtsystems begünstigt, obwohl diese in der ersten Iteration noch nicht genutzt wird (Abbildung 3.3-2), da alles auf einem Rechner lauffähig ist.

3.3.1.1 Beobachtungsrichtung von Experimentator zu Proband – Sub-System erster Iteration

In Abbildung 3.3-2 wird die statische, kompositionelle Struktur des Sub-Systems nach der **ersten Iteration** dargestellt. Hierbei wurde sich nach der FMC-Notation eines Blockdiagramms²⁹ gerichtet. Die Grafik wurde abschließend zur Thesis erstellt und bildet die finale Struktur ab. Es wurden jedoch bereits während der Implementierung und Erarbeitung regelmäßige abstrakte systemische Modelle erstellt, ähnlich wie in den Abschnitten 3.1 und 3.2. Diese regelmäßigen Modellierungen wurden erstellt, um die Modularität des Systems nicht zu vernachlässigen. Konkrete Modelle nach FMC wurden jedoch erst am Ende erstellt, um gleichzeitig keinen großen Overhead darzustellen zu den eigentlichen Implementierungen. So passte es in den iterativen, scrum-ähnlichen Ansatz den diese Thesis verfolgt. Für die konkreten Aussagen in der Evaluation werden diese abschließenden FMC-Diagramme genutzt.

²⁸ <http://users.cs.cf.ac.uk/Dave.Marshall/C/node27.html> (8/03/2018)

²⁹ http://www.fmc-modeling.org/download/notation_reference/Reference_Sheet-Block_Diagram.pdf (8/03/2018)

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

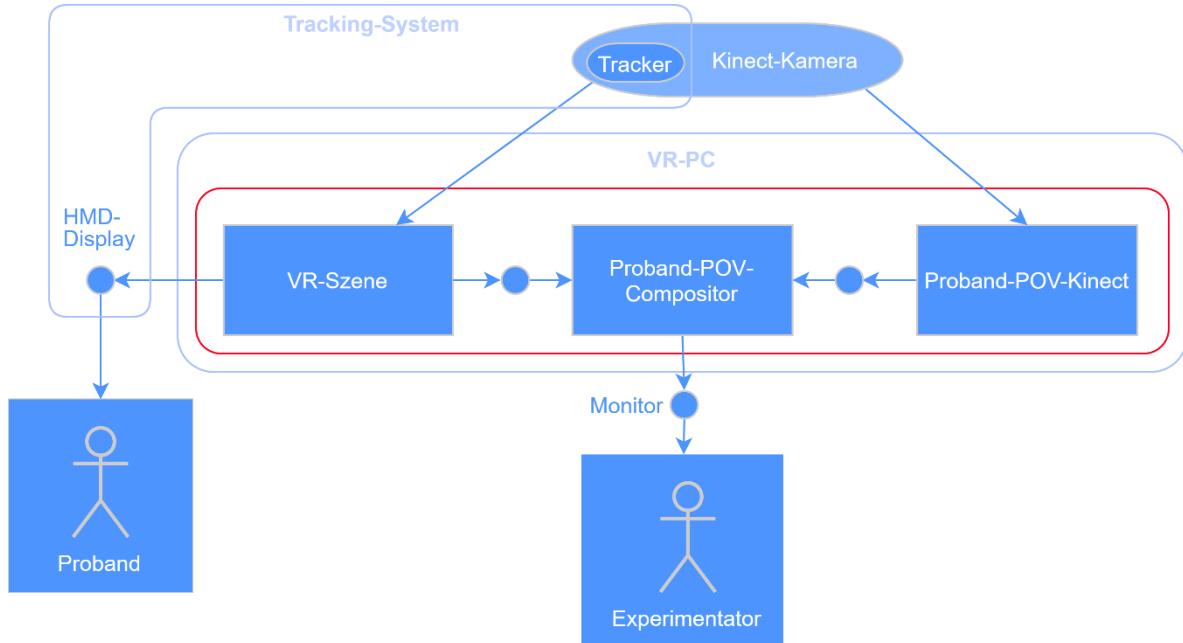


Abbildung 3.3-2: Die Modellierung der kompositionellen Struktur (nach FMC) der Beobachtungsrichtung von Experimentator zu Proband als Blockdiagramm. Das Sub-System spiegelt das Ergebnis der ersten Iteration wieder. Die rote Form ist die Systemgrenze. Alles innerhalb dieser Grenze beinhaltet die fokussierte Funktionalität und ist umfassend modelliert. Diagrammkomponenten außerhalb dieser Grenze beinhalten zu viel unbeeinflussbare, externe Faktoren, als das diese umfassend modelliert werden könnten.

Man sieht in dem Blockdiagramm besonders die Aufteilung des Codes in drei Module, welche über einfache Kanäle kommunizieren. Das **VR-Szene**-Modul beinhaltet jegliche Funktionalität, die sich mit VR auseinandersetzt, wie das Rendering, das Tracking und die Interaktion mit virtuellen Objekten. Hier werden sowohl die Szeneninformationen für den Probanden, als auch die Szenendaten die das **Szenen-Übersicht-Compositor**-Modul benötigt berechnet, um später dem Experimentator die visuellen Informationen zu Szene und Proband bereit zu stellen. Das **Szenen-Übersicht-Kinect**-Modul stellt die Informationen, die die Proband-Textur betreffen für den Compositor bereit. Aus der Modellierung in Abbildung 3.3-2 geht besonders hervor, dass der Datenfluss innerhalb des Sub-Systems nur in eine Richtung stattfindet. Die Tatsache, dass ausschließlich unidirektionale Lese- und Schreibzugriffe genutzt werden ist bereits ein erster Indikator dafür, dass die Modularität sich positiv auf die Einfachheit der Architektur auswirkt.

Um jedoch die Schnittstellen dieser Architektur systemisch und nach FMC klar zu definieren, müssen zur kompositionellen ebenfalls die dynamische Struktur im Sinne eines Ablaufes, und die Wertebereichsstruktur konkretisiert werden. Hierzu werden respektive in Abbildung

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

3.3-3 eine Art Petri-Netz³⁰³¹ und in Abbildung 3.3-4 eine Art Entity-Relationship-Diagramm³² (ER-Diagramm) verwendet.

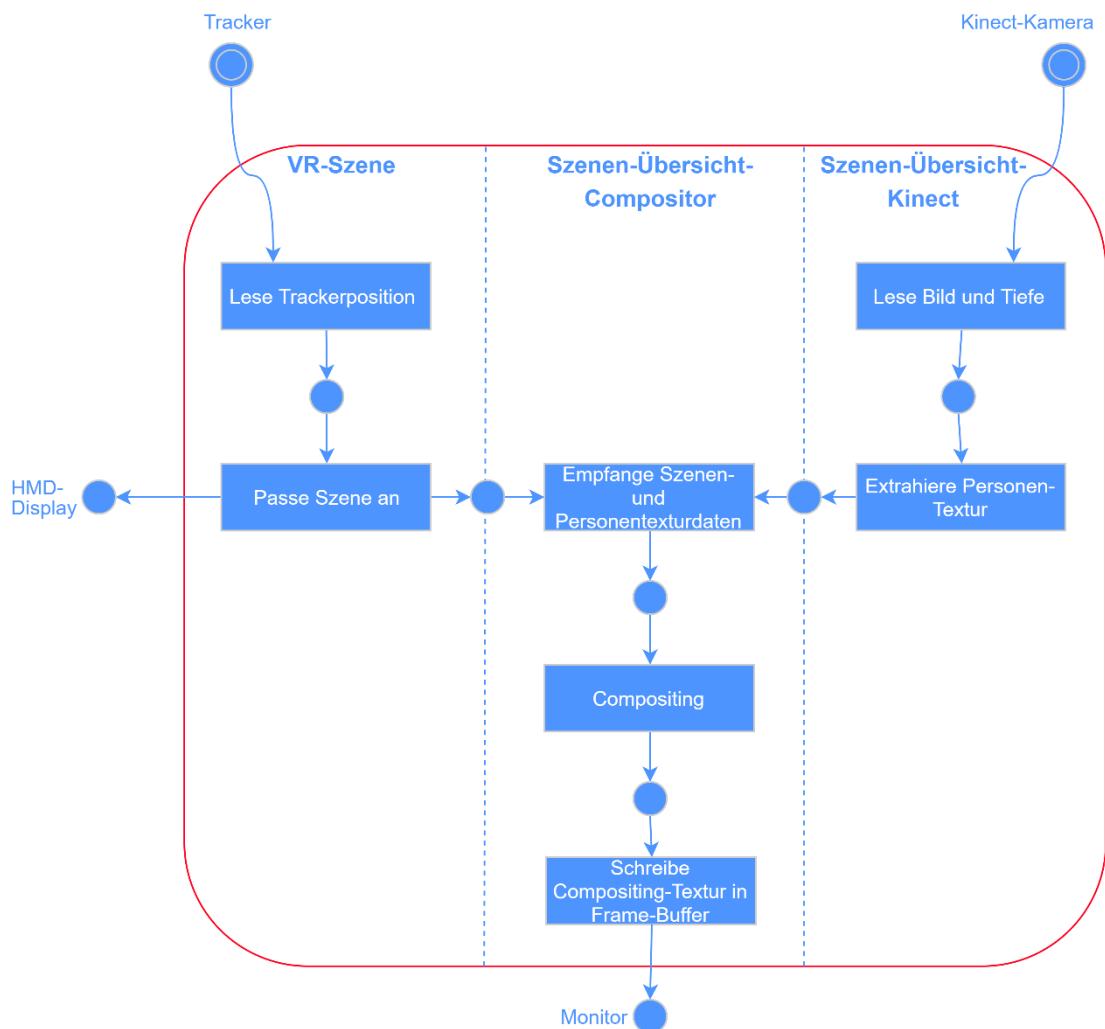


Abbildung 3.3-3: Das Sub-System der ersten Iteration als Petri-Netz spiegelt den Ablauf wider.

Das Petri-Netz in Abbildung 3.3-3 zeigt das zeitliche Verhalten des Sub-Systems und drückt dieses mithilfe von Aktionen aus. Während das Blockdiagramm aufzeigt, *wo* die Kommunikation zwischen den drei Modulen des Sub-Systems stattfindet, kann man hier sehen *wie* sie abläuft. Hier erkennt man besonders den klaren Ablauf der Kommunikation innerhalb, aber auch zwischen den Modulen. Die Funktionalität ist auf einfache, voneinander getrennte Unterfunktionen aufgeteilt. Diese müssen an nur zwei Stellen miteinander kommunizieren.

³⁰ [http://www.fmc-modeling.org/download/notation_reference/Reference_Sheet-Petri_Net_\(1\).pdf](http://www.fmc-modeling.org/download/notation_reference/Reference_Sheet-Petri_Net_(1).pdf) (8/03/2018)

³¹ [http://www.fmc-modeling.org/download/notation_reference/Reference_Sheet-Petri_Net_\(2\).pdf](http://www.fmc-modeling.org/download/notation_reference/Reference_Sheet-Petri_Net_(2).pdf) (8/03/2018)

³² http://www.fmc-modeling.org/download/notation_reference/Reference_Sheet-ER_Diagram.pdf (8/03/2018)

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

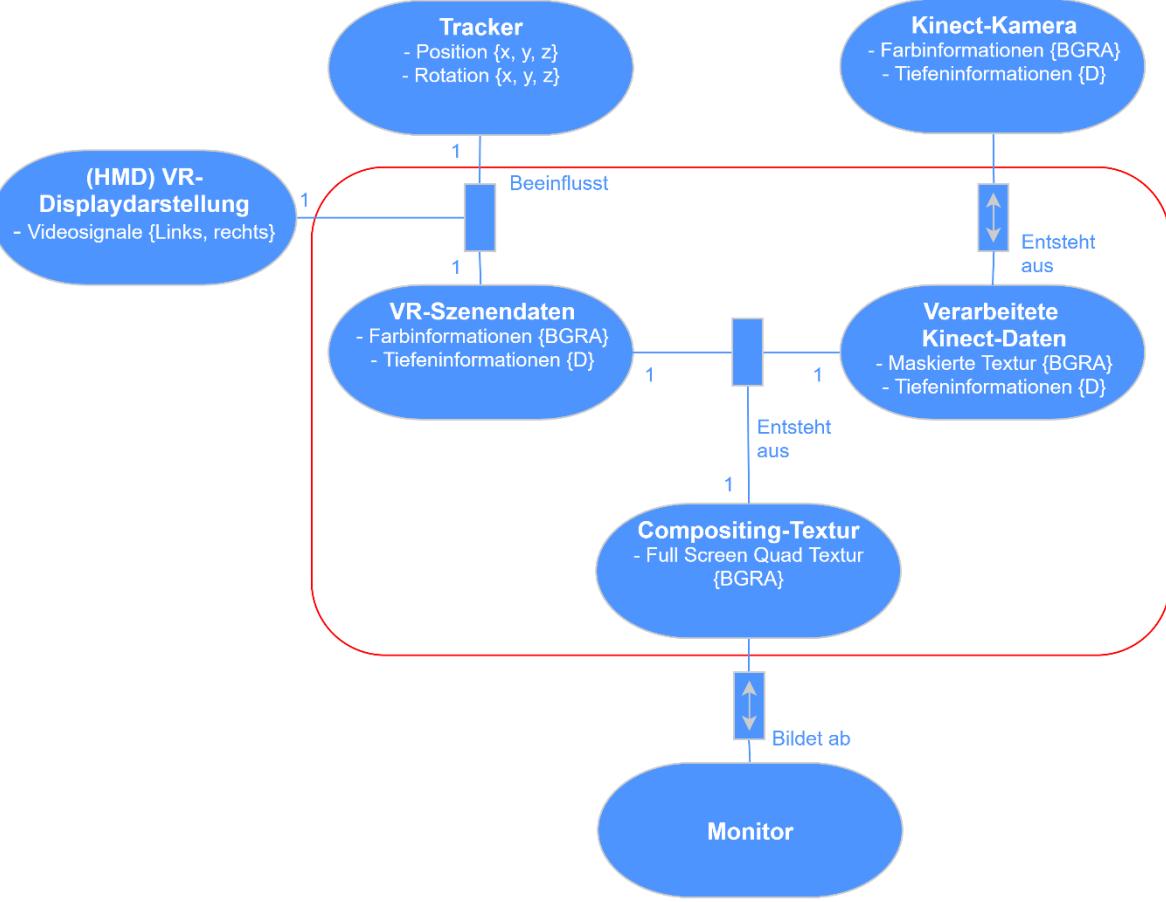


Abbildung 3.3-4: Das Entity-Relationship-Diagramm, das den Funktionsumfang für die Darstellung des Probanden und des VE für den Experimentator beschreibt.

Das ER-Diagramm beschreibt die Wertestruktur des Systems mithilfe von Entitäten. Diese Entitäten gleichen auf abstrakter Ebene den Kanälen und Speichern in einem Blockdiagramm. Jeder Kanal des Blockdiagramms wird hier als Entität dargestellt. Je nach Komplexität können weitere hinzukommen. Der Einblick in die Wertestruktur offenbart an dieser Stelle bereits Spezifikationen an die Daten, die an den vorher definierten Schnittstellen fließen. Man sieht welche Art von Daten benötigt wird, um aus den verschiedenen Quellen die finale Textur auf dem Bildschirm anzuzeigen. Das Zustandekommen des visuellen Mixed-Reality-Signals für den Experimentator ist eine elementare Funktion und daher von hoher Wichtigkeit, um die Module auch in anderen Projekten einzusetzen. Besonders bei der Modellierung des gesamten Systems, am Ende dieses Kapitels, wird dieser Nutzen bereits in der Praxis gezeigt. Insgesamt treten nach der ersten Iteration noch keine Probleme auf, die mithilfe von Architekturmodellierung gelöst werden müssten.

3.3.1.2 Beobachtungsrichtung von Proband zu Experimentator – Sub-System zweiter Iteration

Das Blockdiagramm in Abbildung 3.3-5 zeigt den strukturellen Aufbau der Funktionalität, die in der **zweiten Iteration** fokussiert wurde. Dieses Sub-System ermöglicht es dem Probanden in immersiver VR den Experimentator innerhalb des VE zu sehen, sodass visuelle Kommunikation ermöglicht wird. Es ist anzumerken, dass diese Funktionalität vorerst getrennt von der Funktionalität aus der ersten Iteration betrachtet wird, um auch die modulübergreifende Funktion des Sub-Systems als eigenes Modul zu behandeln. Es wird in

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP
der zweiten Iteration also nur die Beobachtungsrichtung von Proband zu Experimentator beschrieben.

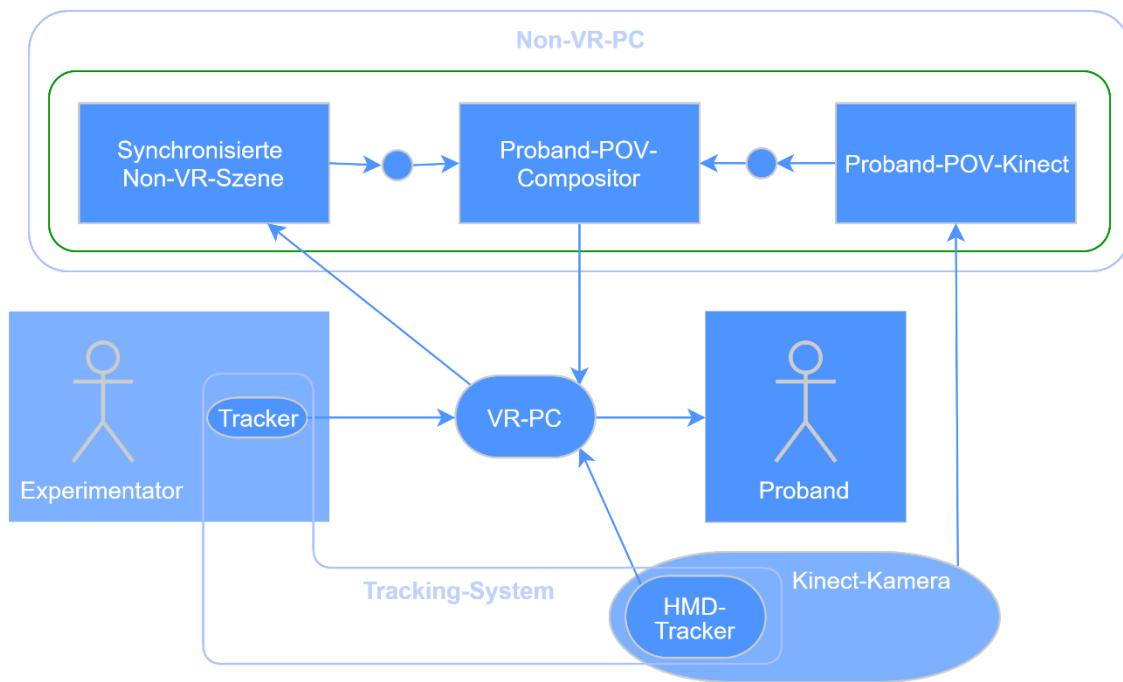


Abbildung 3.3-5: Das Strukturdiagramm der zweiten Iteration. Die Funktionalität der Beobachtung von Proband zu Experimentator ist weitgehend getrennt von der umgekehrten Beobachtungsrichtung aus der ersten Iteration. Die Systemgrenze des Sub-Systems ist in Grün dargestellt.

Die genannte Kapselung ist insbesondere anhand der Platzierung der VR-PC-Komponente außerhalb der Sub-System-Grenze zu sehen. Wie in Abbildung 3.3-2 dargestellt wird, umschließt das VR-PC-System die gesamte vorherige Funktionalität. Das Non-VR-PC-System hingegen umschließt die gesamte Funktionalität, die in der zweiten Iteration behandelt wird. Eine strukturelle Abhängigkeit an außenstehende Systeme besteht nur an eine Kinect-Kamera und den VR-PC. Innerhalb des Sub-Systems sind die Module **Synchronisierte Non-VR-Szene**, **Proband-POV-Compositor** und **Proband-POV-Kinect** wie im vorherigen Modell auch nur mit unidirektionalen Verbindungen verkettet.

Die Funktionen dieser Module sind ebenfalls eng an den Modulen des vorherigen Modells orientiert. Die synchronisierte Non-VR-Szene ist eine exakte inhaltliche Kopie der eigentlichen VR-Szene und wird mit dieser synchronisiert, sodass diese ohne die Proband-Interaktion die gleiche zeitlich-räumliche Information abbildet. Theoretisch wäre es möglich, dass diese vom Proband-POV-Compositor benötigten Informationen auch aus der originalen VR-Szene kommen, jedoch gibt es hier praktische Grenzen. Sowohl die VR-Performance bei den Tests, als auch die Beschränkung von genau einer Kinect-Kamera pro Rechner waren ausschlaggebend für die Verteilung des Systems auf mehrere Rechner. Das Proband-POV-Kamera-Modul stellt für den Compositor das benötigte Bild aus der Sicht des Probanden (Point of View, POV) zur Verfügung, sodass der Compositor dann die nötigen Daten von Szene und POV kombiniert und an den VR-PC zurückgibt.

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

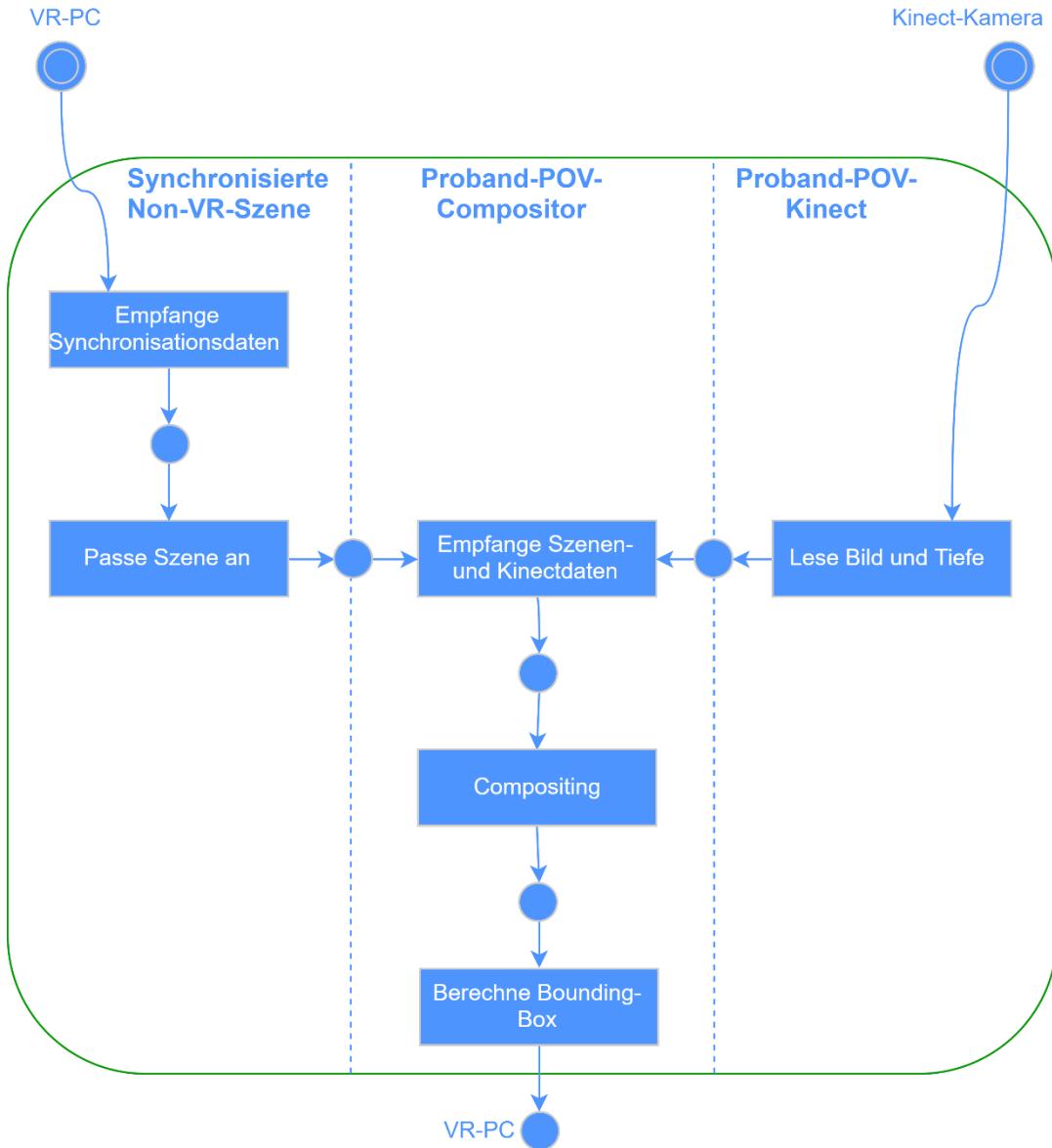


Abbildung 3.3-6: Das Petri-Netz für die Funktionalität die bei der zweiten Iteration hinzugekommen ist verdeutlicht die Ähnlichkeit zum Sub-System der ersten Iteration.

Die dynamische Struktur in Abbildung 3.3-6 ist sehr ähnlich zum vorrangingen Petri-Netz in Abbildung 3.3-3. Unterschiede sind besonders auf die Nutzung der synchronisierten Non-VR-Szene zurückzuführen. Statt des Trackers wird der Prozess vom außenstehenden VR-PC gestartet. Ein weiterer Unterschied ist die fehlende Übergangskomponente im Proband-POV-Modul, die die Personentextur aus den Kinect-Daten extrahiert. Für die hier geschilderte Funktionalität konnte auf Grund vieler Einschränkungen der Kinect-Personenerkennung nicht auf diese zurückgegriffen werden. Die Erkennung arbeitet auf Grund der genannten Kamera-Spezifikationen und des Erkennungsalgoritmus (Deep Randomized Decision Trees) (Kapitel 2.5.6) nur, wenn die Kamera still steht. Ein POV eines Menschen wird jedoch nie still sein, sodass auf eine neuartige Erkennungs- und Segmentierungsmethode zurückgegriffen wird, die in Abschnitt 3.3.5 genauer beschrieben ist. Außerdem wird das Compositing-Ergebnis nicht ohne Weiteres auf einem Bildschirm abgebildet, sondern als Bounding-Box (BB) an den VR-PC weitergegeben..

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

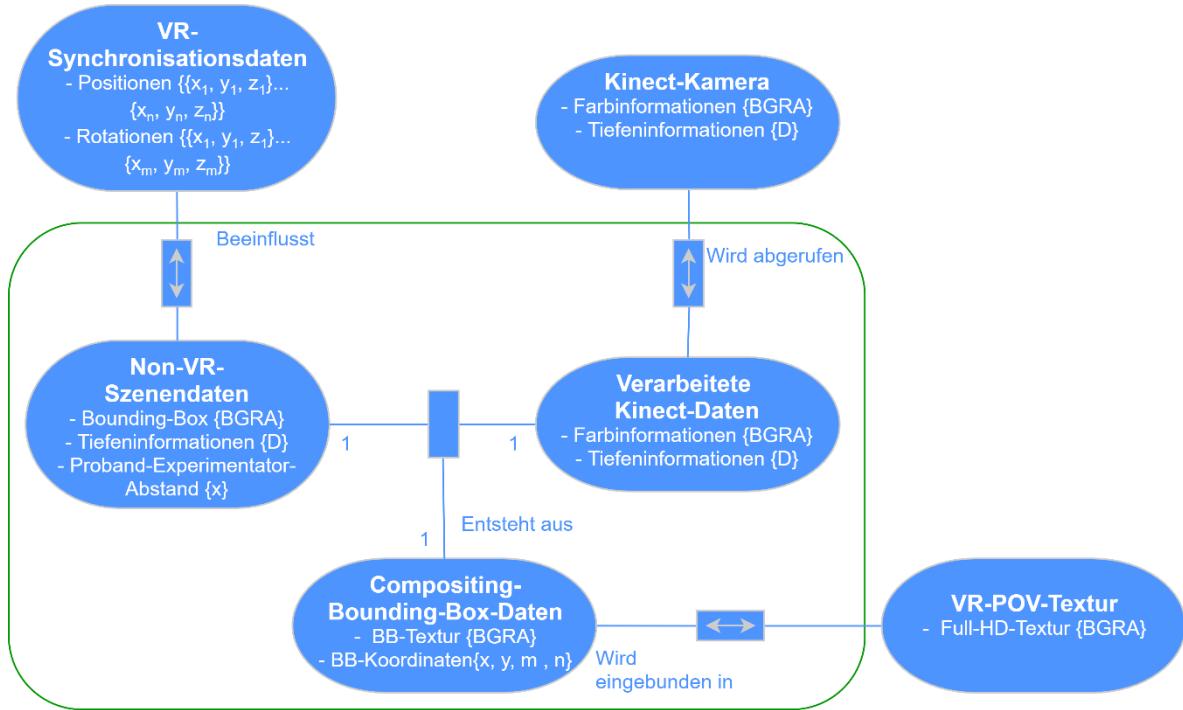


Abbildung 3.3-7: Das Entity-Relationship-Diagramm für die Beobachtungsrichtung von Proband zu Experimentator zeigt die Wertebereichsstruktur an.

Der Unterschied bezüglich der BB, die während der Prozess-Kommunikation genutzt wird, ist in erster Linie in den Attributen der Compositing-BB-Daten-Entität zu sehen (Abbildung 3.3-7). Zusätzlich zur erstellten Textur werden hier auch Koordinaten übermittelt, mit denen es möglich ist, die BB später wieder auf eine Textur im Originalformat zu übertragen. Dies geschieht, wie man hier ebenfalls sieht, außerhalb der Systemgrenze des Sub-Systems. Auch die Funktion des Kinect-Moduls wird klarer, da die Verarbeitete-Kinect-Daten-Entität nur noch die Farb- und Tiefeninformationen von der Kamera abruft und unverändert weiterleitet. Die eigentliche Personenerkennung, -segmentierung und Textur-Extraktion geschieht entsprechend im Compositor-Modul anstelle des Kinect-Moduls. Um weitere funktionale Zusammenhänge zwischen den beiden Sub-Systemen aufzuzeigen müssen die separierten Module als Gesamtsystem modelliert werden.

3.3.1.3 Beidseitige Beobachtungsrichtung – Gesamtsystem

Die strukturelle Architektur, die in Abbildung 3.3-8 dargestellt ist, zeigt die beiden vorangegangenen Funktionalitäten. Die Beobachtungsrichtung von Experimentator nach Proband ist wieder roter und die entgegengesetzte Richtung in grüner Systemgrenze abgebildet. Das gesamte System, in dem auch die Struktur zwischen diesen Sub-Systemen modelliert ist, ist mit der gelben Systemgrenze umschlossen.

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

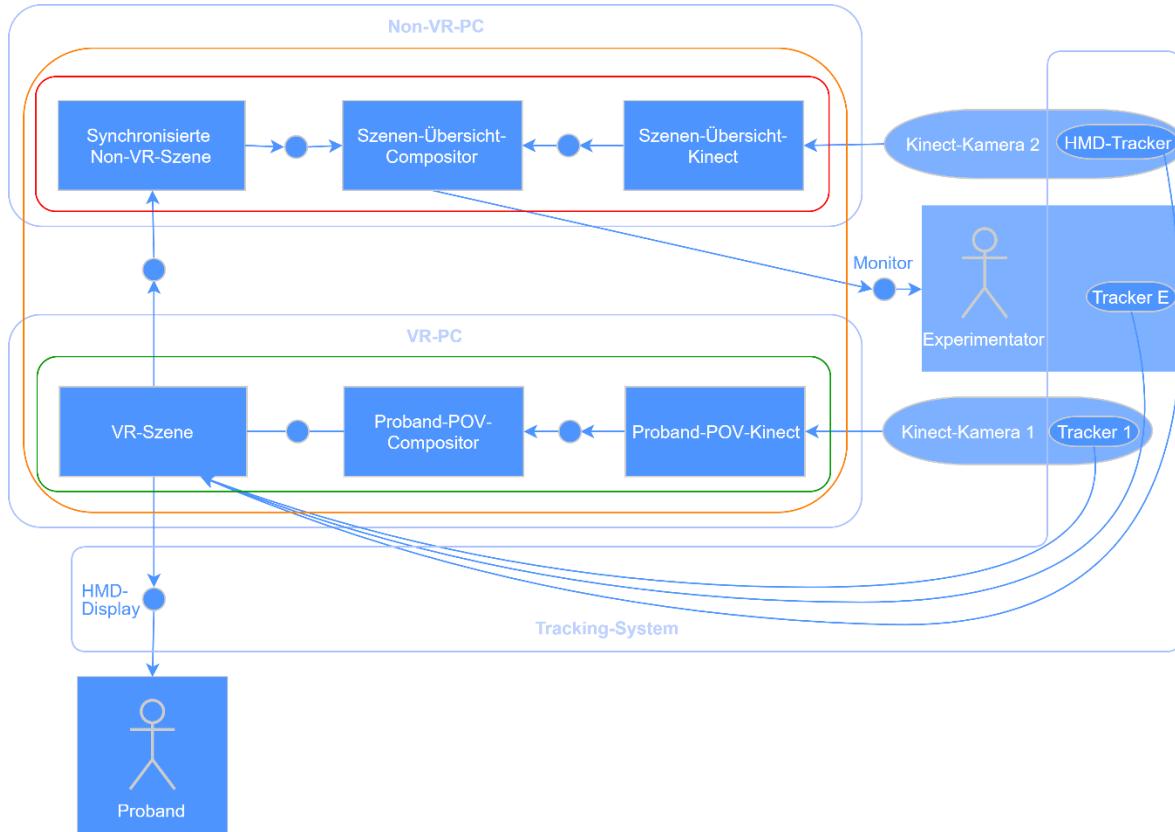


Abbildung 3.3-8: Das Blockdiagramm des Gesamtsystems umfasst sowohl die Funktionalität aus der ersten, als auch aus der zweiten Iteration. Hier wird besonders die Kommunikation zwischen diesen hervorgehoben.

Auf Grund der vorangegangenen systemischen Darstellung und modularen Kapselung wurde bei der Zusammensetzung der Sub-Systeme bereits die Modularität des Systems ausgenutzt. Um die kompositionelle Struktur so einfach wie möglich zu halten, wurden die Funktionalitäten der Sub-Systeme ausgetauscht. Der POV für den Probanden wird damit anhand der Daten aus der VR-Szene berechnet und die Berechnung der Szenen-Übersicht für den Experimentator wurde auf das Non-VR-PC System verlegt. Dieser Austausch ist auf Grund des modulmäßigen Aufbaus relativ einfach zu veranschaulichen und umzusetzen, wie Abbildung 3.3-9 zeigt. Durch die Anwendung einer bidirektionalen Verbindung innerhalb des VR-PC-Systems (Abbildung 3.3-9 untere Markierungen) kann die tendenziell anfällige und unperformante systemübergreifende Verbindung über ein Netzwerk (Abbildung 3.3-9 obere Markierungen) ersetzt werden. Zusätzlich kann der Nutzen aus der vertikalen Skalierbarkeit verwendet werden, um beispielsweise die paarweise verschiedenen Module auch auf unterschiedliche Prozessoren und/oder Grafikkarten zu verteilen.

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

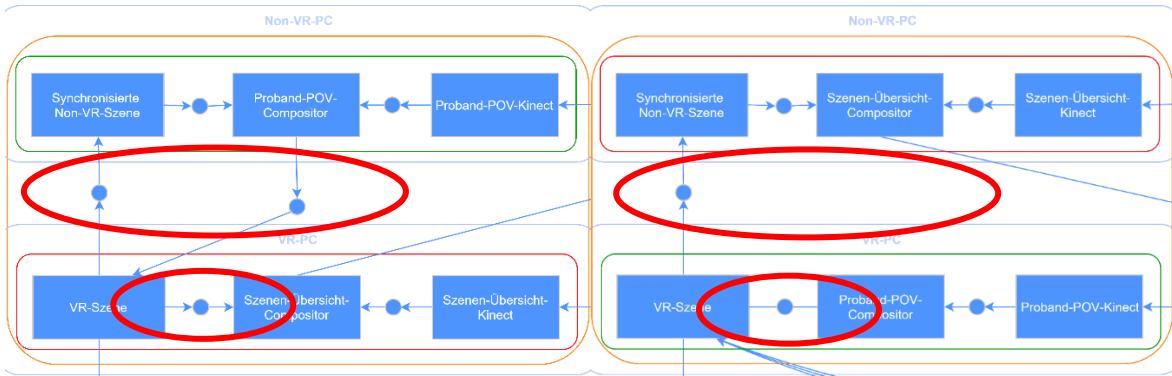


Abbildung 3.3-9: Links: Die erste, naive Struktur des Systems. Rechts: Die durch systemische Modellierung begründete neue Architektur.

Obwohl das umgangene Problem auf den ersten Blick recht trivial erscheint, ist die Verbindung der getrennten Systeme ein durchaus valider Störfaktor, wenn man bedenkt, dass VR-nahe Daten, wie beispielsweise ganze HD-Texturen mit vier Kanälen (RGBA) ausgetauscht werden müssen. Bei einer Implementierung der naiven Architektur (Abbildung 3.3-9 links) und ohne Reduzierung der Textur unter Nutzung einer BB, würden folgende Daten in 30 FPS je einmal pro Frame über das lokale Netzwerk fließen:

- Synchronisationsdaten
- POV-Textur

Um die Übertragung der beiden Datenarten besser nachvollziehen zu können, folgen die Berechnungen dieser Daten und eine Einordnung anhand eines regulären 100 Mbit/s und eines Gigabit Netzwerkes.

$$sync = (n + m) * 3 * 4$$

Formel 3.3-1: Synchronisationsdaten in Byte.

Die Synchronisationsdaten bestehen aus einfachen Positions- und Rotationswerten (Formel 3.3-1). Während n die Anzahl der zu synchronisierenden Positionen darstellt, stellt m die Anzahl der zu synchronisierenden Rotationen dar (Abbildung 3.3-7). Jede Position und Rotation besteht aus drei Float. Ein Float besteht wiederum aus 4 Byte. Insgesamt besteht daher nur eine lineare Abhängigkeit an die Anzahl der Synchronisationsobjekte der Szene. Die Kosten für die Synchronisationsdaten sind damit vorerst vernachlässigbar, da keine hohe zu synchronisierende Objektanzahl (von mehr als 20) erwartet wird.

$$POV = w * h * 4$$

Formel 3.3-2: POV-Textur in Byte.

Die Textur, die für den POV benötigt wird, müsste ebenfalls über das Netzwerk versendet werden. Diese Textur ist bezüglich der Ausmaße w und h (Breite und Höhe, Formel 3.3-2) an die Vorgaben der Kinect-Kamera gebunden. Diese Kamera wird als simulierte „Augen“ des Probanden verwendet. Es werden respektive also $1920 * 1080$ Pixel pro Bild versendet. Pro Pixel sind wiederum vier Kanäle (RGBA) nötig. Alleine durch die Übertragung dieser Textur werden also pro Sekunde $30 * 1920 * 1080 * 4 = 248832000$ Byte übertragen, was in etwa 248 Mb/s entspricht, und damit sowohl die ~12,5 Mb/s eines 100 Mbit-, als auch die ~125 Mb/s eines Gigabit-Netzwerks bei Weitem überschreitet. Das zeigt, dass sowohl die Nutzung der BB – die näher in Kapitel 3.3.6 erläutert wird – als auch die generelle Vermeidung von Netzwerk-Verbindungen in der Architektur des Systems von großer Wichtigkeit sind.

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

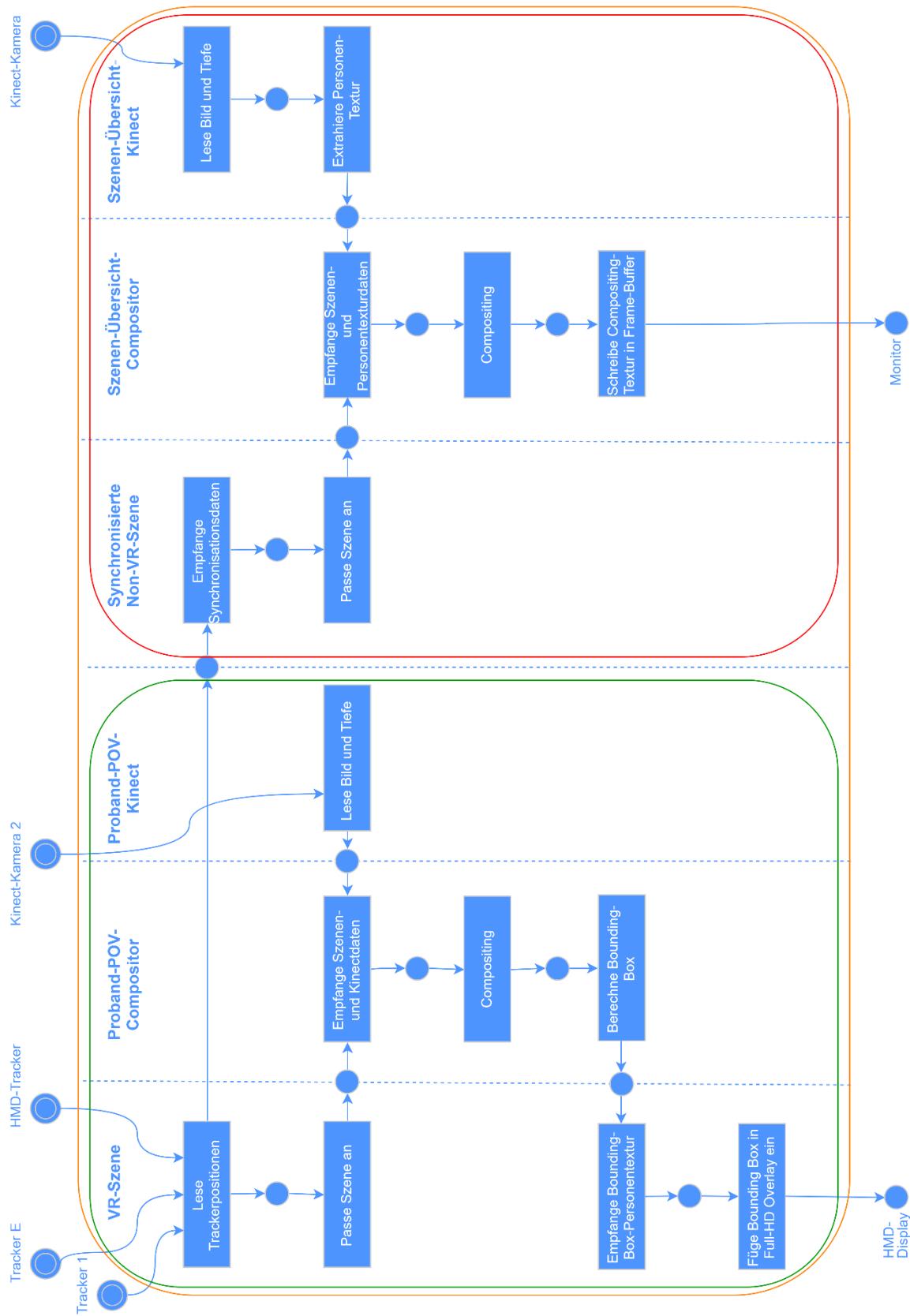


Abbildung 3.3-10: Ablaufdiagramm des Gesamtsystems, visualisiert als FMC-Petri-Netz.

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

Es ist an dieser Stelle auch wichtig zu nennen, dass es durchaus alternative Lösungen gibt, um eine solche Textur zu übertragen, wie beispielsweise modernes Echtzeit-Streaming. Dazu zählt unter anderem das Real-Time Streaming Protocol nach Schulzrinne (1998). Vergleichbare Vorgehen setzen jedoch immer auch eine Kompression der Bilddateien voraus. Somit sind diese Techniken aus zwei Gründen ungeeignet für den Einsatz innerhalb des VR-Prototyps, da 1) die Bildqualität verringert wird, aber viel schwerwiegender, weil 2) die Kompression Rechenzeit in Anspruch nimmt. Diese zusätzliche Rechenzeit müsste zu einem Zeitpunkt in Anspruch genommen werden, der mitten im Ablauf des Systems liegt, wie man in Abbildung 3.3-10 erkennen kann.

Bei der Problematik die Echtzeit-Bilddaten in ein VR-System einzupflegen, ist man jedoch stark abhängig von den Faktoren Performanz und Geschwindigkeit – das gilt auch noch für die verbesserte Architektur der Szene. Mit Geschwindigkeit ist hier gemeint, dass das Ausgangsmaterial für die POV-Textur zu einem Zeitpunkt t von dem Unity- und dem Kinect-Modul erstellt wird. Die finale berechnete Textur muss aber theoretisch zum Zeitpunkt t auch schon für den Probanden zur Verfügung gestellt werden, da die Daten zum Zeitpunkt $t + 1$ schon nicht mehr in Gänze valide sind. Es könnten sich vom Zeitpunkt t zum Zeitpunkt $t + 1$ bereits maßgebliche Veränderungen der Szene zugetragen haben. Diese Problematik lässt sich durchaus als eine Art Producer-Consumer-Problem³³ der VR-Domäne darstellen und kann mit derzeitigiger Technik nicht vollkommen gelöst werden. Eventuell könnte man mit Doppeltberechnung oder Schätzung der Kopfbewegungen und aufwendiger Technik dem Problem zumindest etwas entgegentreten, jedoch schließt dies die Abgrenzung der Thesis unter dem Gesichtspunkt „low-budget“ aus.

Hinzu kommt die Problematik der Synchronisierung der beiden Unity-Szenen. Dem Experimentator wird ebenfalls nie die Szene zu dem exakten Zeitpunkt t aus der VR-Szene wiedergegeben werden können. Eine Diskrepanz zwischen Probanden-Sicht und Experimentatoren-Sicht ist kaum umgänglich. Aus der Petri-Netz-Gesamtmodellierung lässt sich diese naheliegende Schwierigkeit der Synchronisation und des Delays/der Latenz gut visuell erkennen. Nachdem in der VR-Szene die Trackerpositionen gelesen werden, wird direkt darauffolgend die Szene inhaltlich angepasst. In dieser Zeit empfängt die simulierte Szene erst die Daten und passt danach die Szene an, sodass die simulierte Szene immer den Zeitpunkt $t-1$ behandelt, während die VR-Szene bereits den Zeitpunkt t darstellt. Hinzu kommt, dass die Synchronisation der Unity-Szenen, und damit auch die Kommunikation der entsprechenden Module zusätzlich noch in die Event-Pipeline³⁴ von Unity eingebettet werden müssen.

Der Ablauf müsste idealerweise daher wie folgt aussehen:

1. Die Update-Funktion der VR-Szene wird abgehandelt.
2. Vor dem eigentlichen VR-Rendering werden die Synchronisationsdaten an die Non-VR-Szene gesendet.
3. Diese werden in der Update-Funktion der Non-VR-Szene verarbeitet.
4. Nach dem notwendigen Rendering der Non-VR-Szene werden die Bilddaten an den Szenen-Übersicht-Compositor gesendet.
5. Außerdem erfolgt ein Vor-Rendering der VR-Szene, da der POV-Compositor ebenfalls die Szenendaten als Bilder benötigt. Diese Bilder werden dann an den POV-Compositor gesendet.

³³ <https://docs.oracle.com/cd/E19455-01/806-5257/sync-31/index.html> (8/03/2018)

³⁴ <https://docs.unity3d.com/Manual/ExecutionOrder.html> (8/03/2018)

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

6. Die VR-Szene empfängt dann die Daten des POV-Compositors, pflegt diese in die Szene ein und erst dann wird sie für den Probanden gerendert. Zum gleichen Zeitpunkt rendert der Szenen-Übersicht-Compositor dann die Ansicht für den Experimentator.

So würden beide beschriebenen Probleme,

1. Die Latenz der Experimentatoren-Textur
2. und der Delay der Szenenanzeige für den Experimentator,

in Relation zum Zeitpunkt der VR-Szene, gelöst werden. Diese Beschreibung macht aber auch besonders die Aufschiebung des Renderings der VR-Szene deutlich, sodass keine VR-konforme Performanz mehr gewährleistet werden könnte. Der beschriebene Ablauf ist also rein hypothetisch und ideeller Natur.

So bestehen Synchronisations- und Performanz-Einschränkungen bei der angepassten Verteilung der Funktionalität nach Abbildung 3.3-8 nach wie vor. Wie Abbildung 3.3-10 zeigt, wird der Prozess zwar geschickt auf die bedeutend schnellere Interprozess-Kommunikation der einzelnen physischen Rechner verschoben, trotzdem ist es nicht möglich den Kinect-Frame zum Zeitpunkt t auch zum gleichen Zeitpunkt bereits in die Visualisierungen einzubinden, ohne dass der VR-Renderingprozess gestört werden würde. Jedoch umgeht man die meisten Hardware-Beschränkungen, da Daten weder das Netzwerk, noch die Festplatte tangieren. Durch die Nutzung des Loopback-Adapters kann die Non-VR-Textur beispielsweise unkomprimiert an den zugehörigen Compositor gesendet werden, da die Übertragung nur durch die CPU-Geschwindigkeit beschränkt wird. Dadurch wird – je nach System – durchaus Loopback-Traffic von bis zu 40 Gb/s möglich.

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

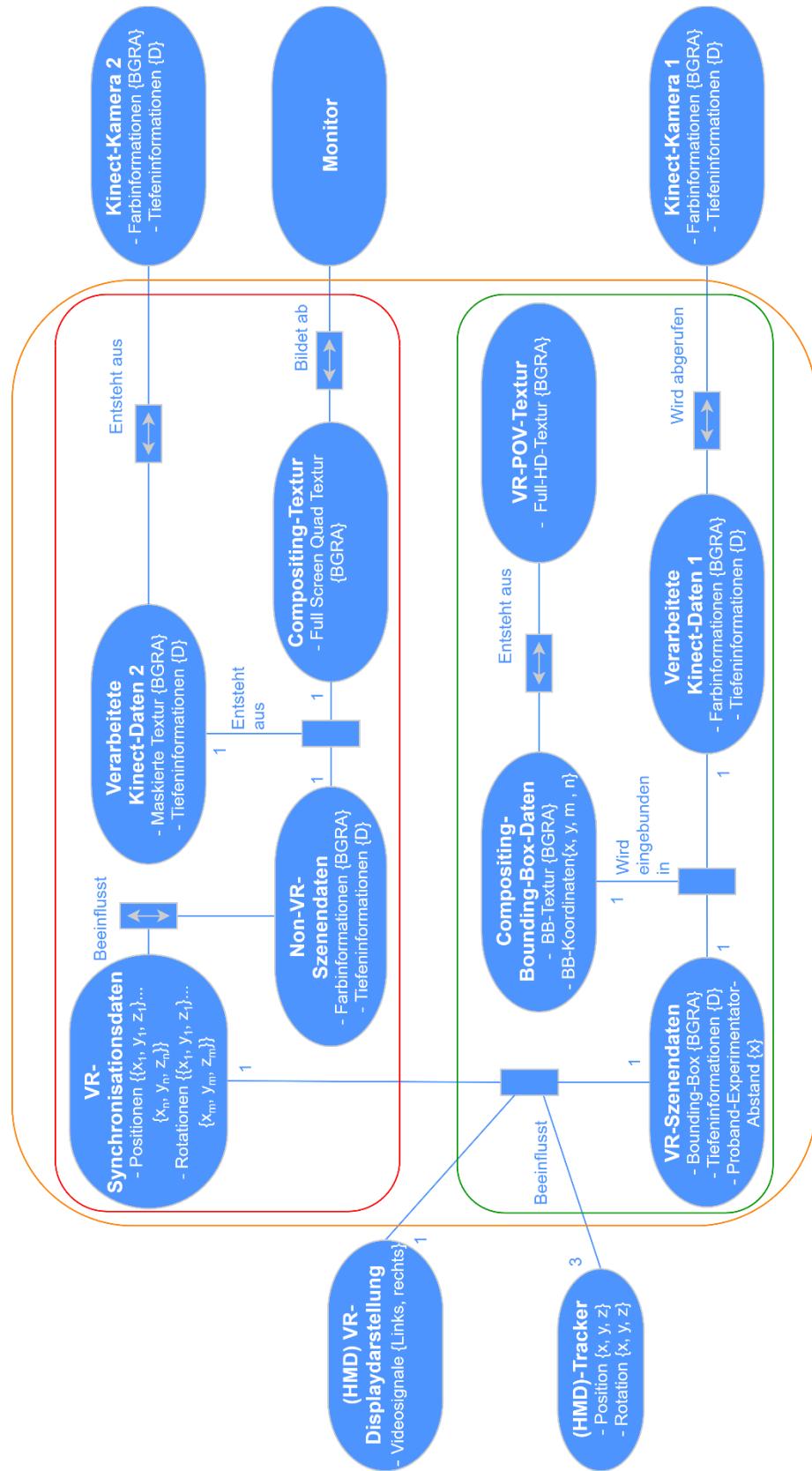


Abbildung 3.3-11: Modellierung der Werte-Struktur des Gesamtsystems.

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

Der Datenfluss in Kombination mit dem Werte-Modell aus Abbildung 3.3-11 zeigt, dass die Netzwerkkommunikation insgesamt also kein Problem mehr darstellt, da nur die Synchronisationsdaten darüber laufen. Diese werden als geringe Anzahl von Float erwartet. Nicht auszuschließen ist jedoch, dass die Summe der Daten, die pro Sekunde über den Loopbackadapter fließen, durchaus höher ist als das besagte Limit. Da die POV-Funktionalität durch die vorherige Netzwerk-Implementierung mit einer BB die Daten reduziert, wird exemplarisch das Limit der Szenen-Übersicht-Funktionalität berechnet, weil dieses mit vollen Auflösungen arbeitet.

Aus der Non-VR-Szene werden zwei Texturen an den Compositor gesendet, die Farb- und die Tiefentextur. Beide werden als vier Kanäle dargestellt, worauf später eingegangen wird. Bei voller Auflösung sind diese Texturen also in BGRA32 insgesamt $1920 * 1080 * 4 = 8294400$ Byte groß. Bei 30 FPS entsteht dadurch ein Throughput von $(1920 * 1080 * 4 + 1920 * 1080 * 1 * 4) * 30 \sim = 497664000$ Byte/s, was 497,664 Mb/s entspricht. Die Kinect-Kamera versendet ebenfalls zwei Bildströme, wobei der Farbkanal auch in BGRA32 vorhanden ist. Der Tiefenkanal benötigt ein UShort pro Pixel, womit der Kinect-Strom auf insgesamt $(1920 * 1080 * 4 + 1920 * 1080 * 1 * 2) * 30 \sim = 373248000$ Byte/s kommt. Der gesamte Throughput des Systems beläuft sich also auf ~ 870912000 Byte/s, beziehungsweise 870,912 Mb/s. Damit stellt die Berechnung der beiden Sub-Systeme in diesem Sinne kein Problem dar, da der Loopback-Adapter mehrere Gb/s unterstützt.

Hinzu kommt, dass die BB den Kinect-Strom sogar noch um den gemessenen Faktorbereich von 0.42404513888 im besten, bis 0.06437885802 im schlechtesten Fall reduziert. Damit wird der Kinect-Datenstrom nochmals von $\sim 248,832$ Mb auf bis zu 16,01952 Mb reduziert. In Kapitel 3.3.5 wird außerdem herausgestellt, dass dadurch auch die Latenz nicht weiter verstärkt wird. Hier muss jedoch angemerkt werden, dass auf eine Verwendung von größeren Protokollen und Format-Header verzichtet wurde, obwohl dies die gängige Praxis ist. So könnte beispielsweise der Zeitstempel eines Bildes verwendet werden, um die unterschiedlichen Datenströme zeitlich korrekt zu synchronisieren. Aus genannten Gründen wird also doktrinisch weitergeführt, dass zugunsten von schnellerer Verarbeitung auf etwaige Ungenauigkeiten in der Synchronität verzichtet wird. So wird also auch Datenumfang gespart, der für gewöhnlich mitgesendet werden müsste. Ansätze für eigene Protokolle sind in der Implementierung skizziert und wurden im Code für mögliche zukünftige Arbeiten beibehalten. Genutzt werden sie in der aktuellen Form des Systems jedoch nicht.

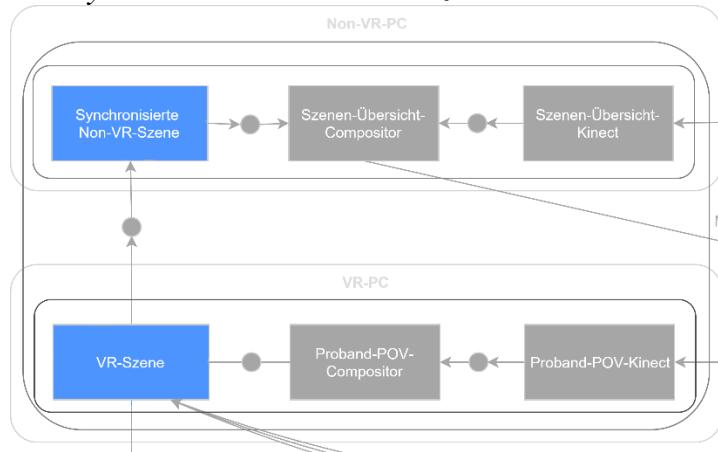
Abschließend wurde in dem Kapitel über die modulare Architektur und die Modulkommunikation der prototypischen Implementierung gezeigt, wie das gesamte System in einfach händelbare Module aufgeteilt wurde. Jedes dieser Module trägt eine bestimmte Funktionalität zur übergreifenden Funktion des Systems bei. Nach außen hin verhält sich das System wie eine einzige Komponente, durch die das Konzept zur gegenseitigen Kommunikation zwischen realer und virtueller Umgebung dargestellt wird. Durch die ausführliche Modellierung des Gesamtsystems wurde der Vorteil einer einfachen Umverteilung der Funktionen erkannt. Dadurch muss kein privates Netzwerk erstellt werden, da die Hauptlast jeweils auf den separierten Rechnern bleibt. Lediglich die Synchronisationsdaten der Szenen müssen über das echte Netzwerk gesendet werden. Dies war besonders einfach möglich, auf Grund der vorherigen, ausführlichen modularen Modellierung der Sub-Systeme.

Herauszustellen ist außerdem, dass im Vergleich zur abstrakten Modellierung in Kapitel 3.2 oder 3.1 nun doch eine konkrete Abhängigkeit zwischen den beiden Sub-Systemen vorhanden ist, die durch die FMC-Modellierung aufgedeckt wurde. Außerdem konnte durch

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

die FMC-Modellierung verstanden werden, dass es effizienter ist, die Funktionalitäten anders aufzuteilen, als es zu Anfang geplant war. Es wurde dadurch nicht nur bereits die Kapselung der Funktionalitäten klar definiert, sondern der Nutzen der Kapselung konnte bei der Umverteilung der Module auch exemplarisch aufgezeigt werden. Der Austausch einzelner Module konnte ohne größeren Aufwand durchgeführt werden, da die Schnittstellen sowohl per kompositioneller und dynamischer Struktur, aber auch bezüglich des Wertebereichs sehr einfach und deutlich beschrieben wurden. Im Folgenden werden nun die besagten einzelnen Funktionalitäten und Module implementierungsnahe beschrieben.

3.3.2 VR-Szene und Synchronisierte Non-VR-Szene



In diesem Abschnitt wird in Kürze auf die VR-Technik eingegangen, die in dem VR-Szene-Modul genutzt wird. Im Zuge dessen wird auch das Non-VR-Szene-Modul dargestellt, da die Funktionalität der beiden Module sich lediglich um den VR-, beziehungsweise den Synchronisierungsaspekt unterscheidet.

Für die Implementierung des VE wurde die Game-Engine Unity3D genutzt. Diese ist nicht nur als eine der wenigen freien Engines gut dokumentiert, sie bietet auch bereits diverse Plug-Ins und Schnittstellen für die gegenwärtige prosumentenorientierte VR-Technologie. Das Plug-In zur zugehörigen VR-Middleware Steam VR³⁵ wird eingesetzt, um ein HTC Vive System einfach und flexibel in die Unity-Umgebung einzubinden. Es wurde sich für ein HTC Vive System entschieden, da dieses zum einen State-of-the-Art HMD- und Room-Scale-VR-Technik bietet und zum anderen auch erweiterbar ist. So können Vive Tracker genutzt werden, um externe Gegenstände, wie Kameras, zu tracken und die Position des RE auf das VR-Koordinatensystem zu übertragen.

Um den Inhalt der Non-VR-Szene mit der VR-Szene zu synchronisieren und die modellierte Verbindung zu implementieren, wird eine UDP-Verbindung (User Datagram Protocol) verwendet. Aus bereits angeführten Gründen wurde diese über eine komplexere Streaming-Verbindung gewählt. UDP wurde auch anstelle einer gängigen TCP-Verbindung (Transmission Control Protocol) verwendet. Im Vergleich zu TCP spart UDP weitere Daten im Header des Protokolls, womit wiederum teurer Netzwerk-Throughput gespart wird. Außerdem ist damit jedes der Module weitgehend unabhängig von der Verbindung, da keinerlei Handshake oder ähnliches stattfinden muss, um die Verbindung zu etablieren. Lediglich der Nachteil von möglichem Paket-Verlust ist anzumerken, da keinerlei Kontrolle der versendeten im Vergleich zu den empfangenen Paketen durchgeführt wird. Ein Paket kann insgesamt jedoch bis zu 65507 Byte groß sein (begrenzt durch die Maximum

³⁵ <http://store.steampowered.com/steamvr?l=german> (8/03/2018)

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

Transmission Unit (MTU) und den UDP-Header). Damit können bis zu 16376 Float-Variablen gleichzeitig in einem einzigen Paket zur Synchronisation versenden werden. Der Verlust eines Paketes ist also vernachlässigbar. Es wird erwartet, dass jedes dieser Pakete den vollständigen, abgeschlossenen Satz an Synchronisationsvariablen für einen Zeitpunkt beinhaltet. Schickt man die Pakete also in sehr kurzen Abständen wird der Verlust eines Paketes nicht bemerkbar sein.

Alternativen zu dieser elementaren Synchronisation wären unityspezifische Lösungen, wie Unitys UNET API³⁶ oder auch der PUN SDK³⁷ von Photon. Beide Alternativen bieten gute und kostenfreie, vor allem aber einfache Zugriffe über High-Level-Funktionen. Der Grund diese nicht zu nutzen ist der, dass diese zur Folge hätten, dass die Module vorerst auf Unity als Render-Back-End beschränkt wären. Damit würde ein Teil der geforderten Nachhaltigkeit und Wiederverwendbarkeit vernachlässigt werden. Daher wurde auch bei den noch folgenden Modulen ausschließlich generische UDP-Kommunikation verwendet.

3.3.2.1 Bilddatengewinnung aus Unity: *Plug-In und Renderloop*

Im Kontext des Prototyps ist die grundlegendste Aufgabe der beiden Szenen-Module die Bereitstellung der Bilddaten (Farb- und Tiefeninformationen) für den jeweiligen Compositor. Bei der Umsetzung dieser Aufgabe sind besonders zwei Aspekte von größter Wichtigkeit. Während 1) das separate Rendering der Szenendaten in die Unity Event-Pipeline eingepflegt werden muss, muss auch 2) das (betriebs-) systemabhängige Render-Back-End mit einbezogen werden. Beides muss wieder unter der Kontradiktion von Performanz und Geschwindigkeit abgewogen werden.

Diese Schwierigkeit kann anhand eines einfachen Beispiels dargestellt werden. Nutzt man die Unity API, um Renderings der Szene zum aktuellen Zeitpunkt t zu generieren, so wird die Unity Event-Pipeline verzögert (gestallt), da die Daten, die erst in diesem Frame berechnet werden, gleichzeitig bereits im selben Frame zum Zeitpunkt t zur Verfügung stehen sollen. Dabei ist auch die CPU-GPU-Kommunikation von maßgeblicher Bedeutung. Die Verarbeitung eines Frames, beziehungsweise die Rückgabe einer Textur im Bereich der Computergrafik, wird nach aktueller Dokumentation von Microsofts Direct3D wie folgt dargestellt:

³⁶ <https://docs.unity3d.com/Manual/UNet.html> (8/03/2018)

³⁷ <https://www.photonengine.com/en/PUN> (8/03/2018)

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

<i>Zeitpunkt des Frames</i>	<i>GPU und CPU Zustand</i>
t	- CPU gibt den Befehl den aktuellen Frame t zu rendern.
$t + 1$	- GPU führt den Befehl der CPU zu Frame t aus. - CPU gibt den Befehl den aktuellen Frame $t + 1$ zu rendern.
$t + 2$	- GPU beendet die Ausführung des Befehls der CPU zu Frame t . - GPU führt den Befehl der CPU zu Frame $t + 1$ aus. - CPU gibt den Befehl den aktuellen Frame $t + 2$ zu rendern.
$t + 3$	- GPU beendet die Ausführung des Befehls der CPU zu Frame $t + 1$. - GPU führt den Befehl der CPU zu Frame $t + 2$ aus. - CPU gibt den Befehl den aktuellen Frame $t + 3$ zu rendern.
$t + 4$...

Tabelle 3.3-1: Zugriff auf Rendering-Daten nach Microsofts Direct3D Dokumentation³⁸.

Bei Betrachtung des Ablaufs dieser Tabelle wird klar, dass ein aktueller Frame zum Zeitpunkt t also frühestens zum Zeitpunkt $t + 2$, durch die Übertragung von GPU zu CPU eher aber zum Zeitpunkt $t + 3$ verfügbar ist, um weiter von der CPU verarbeitet werden zu können. Für die Unity-Pipeline heißt das, dass das Back-End, welches auch die VR-Displays der Vive bespielt, für drei bis vier Frames gestaltet wird. Das Abrufen einer Rendertextur wird für gewöhnlich durch den Aufruf von `Texture2D.Readpixels()`³⁹ durchgeführt. Um die Pipeline jedoch nicht zu stallen, muss das Abrufen der Texturinformationen asynchron implementiert werden, was in Unity selbst nur schwer umzusetzen ist, da Unity nur eine bedingte Parallelisierung durch Threads erlaubt.

Echte Asynchronität in Verbindung mit dem Zugriff auf die Grafikkarte ist in Unity also nur mithilfe von Native Plug-Ins⁴⁰ möglich. Über Dynamic Link Libraries (DLLs) wird so beispielsweise eine Schnittstelle auf die Programmiersprache C++ geschaffen, die Funktionen bereitstellt, die maschinennahen Zugriff auf Grafikfunktionen erlaubt. Zu Beginn der Konzeption der vorliegenden Thesis sollte diese Problematik im Detail behandelt werden. Nach einiger Vorarbeit und tiefgehender Recherche wurde jedoch kurz nach Anmeldung der Thesis bekannt gegeben, dass eben jener asynchrone Zugriff (Readback) auf die GPU in den nächsten Versionen von Unity (2018.1) bereits eine bereitgestellte Funktion der Unity API sein soll⁴¹. Aus diesem Grund wurde darauf verzichtet eine völlig neue Lösung für dieses Problem zu erstellen. Hingegen wurde versucht, dieses Problem lösungsorientiert zu bewältigen, sodass es im Sinne des Prototyping-Prozesses kein Hindernis darstellt. Dies stellt zwar eine Abhängigkeit an Unity dar, womit die Wiederverwendbarkeit eingeschränkt würde, jedoch ist man an dieser Schnittstelle in jedem Fall schon bedingt durch die Wahl der Grafik-Bibliothek – letztendlich auch durch die Wahl der physischen Grafikkarte. Daher erscheint eine zusätzliche Abhängigkeit an dieser Stelle für den Zweck der Thesis von geringerer Bedeutung.

³⁸ [https://msdn.microsoft.com/en-us/library/windows/desktop/bb205132\(v=vs.85\).aspx#Performance_Considerations](https://msdn.microsoft.com/en-us/library/windows/desktop/bb205132(v=vs.85).aspx#Performance_Considerations) (8/03/2018)

³⁹ <https://docs.unity3d.com/ScriptReference/Texture2D.ReadPixels.html> (8/03/2018)

⁴⁰ <https://docs.unity3d.com/Manual/NativePlugins.html> (8/03/2018)

⁴¹ <https://www.youtube.com/watch?v=tGmnZdY5Y-E> (ca. ab Min. 93+) (8/03/2018)

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

Das asynchrone Readback der Textur wird im Prototyp deswegen durch die Adaption eines bestehenden Open Source Plug-Ins⁴² und offizieller Vorschläge⁴³ von Unity-Technologies (Anhang A) durchgeführt. Es basiert auf C++ und unterstützt ausschließlich DirectX 11, und damit auch nur windows-basierende Systeme. In Abbildung 3.3-12 wird der abstrakte Vorgang des Plug-Ins für die Textur genau eines Frames beschrieben.

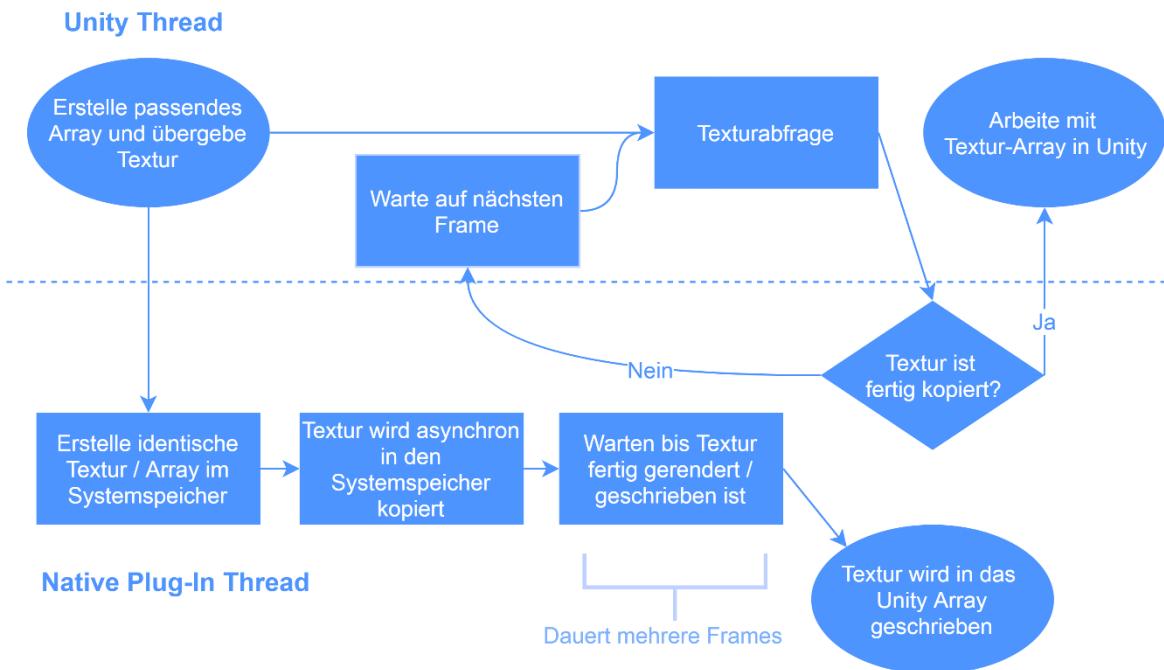


Abbildung 3.3-12: Das Flussdiagramm beschreibt den Ablauf und das Zusammenspiel des Native Plug-In zum Textur Readback mit der Unity Instanz.

Die Problematik die damit ebenfalls umgangen wird ist, dass nur der Unity Hauptthread die Unity-API aufrufen kann, was ein zusätzlicher Thread nicht könnte. Durch die Nutzung der DLL bearbeitet der eigene Render-Thread des Plug-Ins dann unabhängig von Unity das Textur Readback von der Grafikkarte und setzt eine Flag für Unity, wenn das jeweilige Unity Byte-Array erfolgreich die gewünschten Texturdaten beinhaltet. Diese Flag kann von Unity in jedem Frame abgefragt werden. Daraus geht implizit hervor, dass das Plug-In keine Arrays vom Datentyp „Texture“ unterstützt. Dies ist jedoch nötig. Es muss mit mehreren Texturen gearbeitet werden, um nicht nur jedes dritte oder vierte Frame erfolgreich zu bearbeiten. Diese Funktionalität muss also auf der Seite des Unity-Moduls eingebettet werden.

In Unity wird eine Kombination aus Queue und Liste genutzt, um das Array, bestehend aus acht Texturen, zu managen. Um häufige Speicherverschiebung zu verhindern beinhaltet diese Strukturen lediglich die Indizes der Texturen in dem Texturarray. Die Queue beinhaltet zu jedem Zeitpunkt die Indizes der Texturen, die derzeit nicht in Gebrauch sind. Die Liste beinhaltet alle Indizes für die Texturen die momentan belegt sind, da das Plug-In diese Texturen derzeit bearbeitet. Da dieser Vorgang pro Textur mehrere (in etwa drei bis vier) Frames andauert, werden acht bis zwölf Texturen genutzt, um noch etwas Puffer zu haben.

In Unity wird dann in jedem neuen Frame eine Textur als Rendertextur für die virtuelle Kamera festgelegt. Der Index dieser Textur wird von der Queue entfernt und dann in die

⁴² <https://github.com/SlightlyMad/AsyncTextureReader/tree/e37ca6c76214f6903a4b5bb6d32ebe75a13d5430> (8/03/2018)

⁴³ <https://bitbucket.org/Unity-Technologies/graphicsdemos/overview> (8/03/2018)

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

Liste gefügt. An dieser Stelle wird eine Liste genutzt, da es vorkommen kann, dass das Plugin in manche Texturen schneller zurückgibt als andere, womit die First-In-First-Out Queue nicht mehr funktionieren würde. Jedes Mal, wenn eine Textur erfolgreich als Byte-Array zurückgegeben wurde, wird der Index entsprechend an die Queue angestellt und aus der Liste entfernt. Durch diesen Zyklus der Texturen wird gewährleistet, dass jeder Frame erfolgreich bearbeitet werden kann. Das Flussdiagramm in Abbildung 3.3-13 zeigt diesen Zyklus für mehrere Frames.

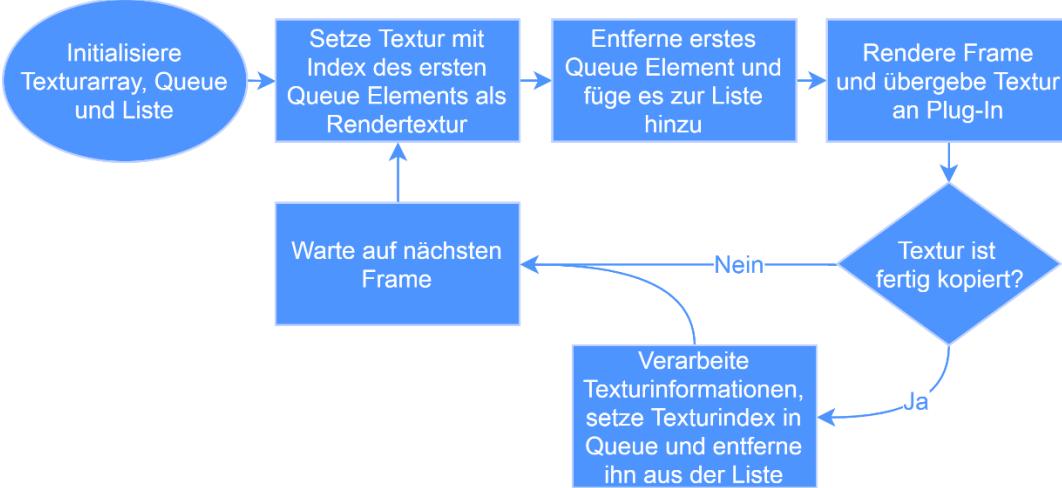


Abbildung 3.3-13: Der unityseitige Ablauf als Flussdiagramm zeigt die Handhabung mehrerer Texturen, mit der Unsicherheit wann die jeweilige Textur fertig für den Zugriff in Unity ist.

Der gesamte Vorgang wird sowohl für die Tiefeninformationen als auch die Farbinformationen durchgeführt. Um an die Tiefeninformationen zu gelangen wird ein Shader genutzt und damit ein zweiter Rendervorgang durchgeführt. In der zugehörigen Rendertextur stehen dann die Tiefeninformationen, die mithilfe des vorher beschriebenen Native Plug-Ins dann als Byte-Array von der Grafikkarte gelesen werden.

Anschließend werden die Arrays dann an das jeweilige Compositor-Modul gesendet. Da die Arrays zu groß sind, um sie als ein einziges Paket per UDP versenden zu können, werden sie vorher manuell fragmentiert. Die Aufteilung in Pakete mit MTU-Größe von 65535 Byte ist performanter implementiert als die in C# gängige `Array.Copy` oder `System.Buffer.Blockcopy` Methode. Die externe, auf C/C++ basierende Methode `memcpy` wird stattdessen innerhalb des Unityzyklus verwendet. Diese wird – ähnlich wie das Native Plug-In – über eine DLL abgerufen. Die DLL muss jedoch nicht in die Unity Szene geladen werden, da die `msvcrt.dll`⁴⁴ bereits standardmäßig integriert ist. Der zugehörige Code des UDP-Clients wird unter dem Stichwort `unsafe` geführt, da die Methode C#-untypische Pointer-Arithmetik benötigt, womit gravierende Systemfehler hervorgerufen werden können. Bei einfachem Ersatz, wie dem Kopiervorgang eines großen Byte-Arrays in kleinere, besteht jedoch keinerlei Gefahr. Das resultierende System wurde daraufhin etwa vier Stunden bei Dauerlast getestet – ohne Auftreten von Fehlern.

⁴⁴ <https://msdn.microsoft.com/en-us/library/hh875057.aspx> (8/03/2018)

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

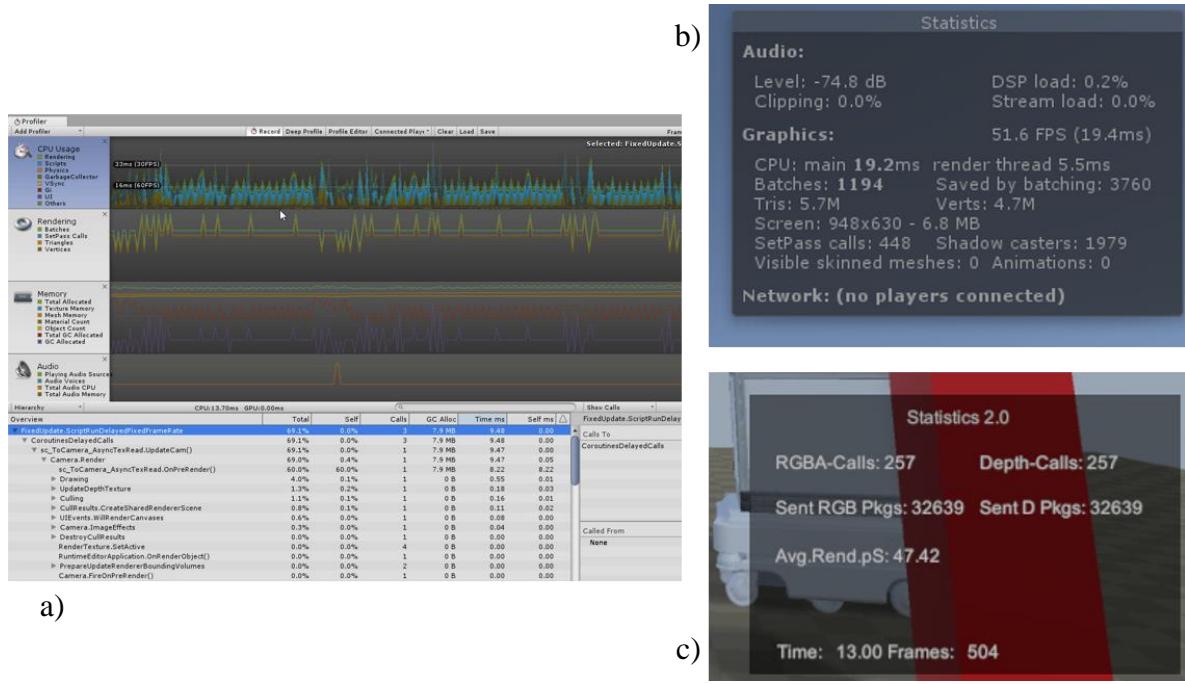


Abbildung 3.3-14: a) UnityProfiler Fenster b) Integriertes Statistics Fenster c) Eigens implementiertes Statistics 2.0 Fenster, welches Informationen zu den alternativen Bildraten und Rendervorgängen gibt.

Der letzte wichtige Punkt, der die Implementierung der Bildgewinnung aus Unity betrifft, ist die Optimierung der Szene. Unity sollte im Idealfall, trotz vergleichbar teurem Mehrfachrendering und Fragmentierung immer noch fähig sein, das VR-Erlebnis des Probanden uneingeschränkt zu vermitteln. Dafür wurde besonders die Funktion des Unityprofiler (Abbildung 3.3-14 a)) genutzt. Auf Grund dieser Aufrufstatistiken konnte festgestellt werden, welche Codeabschnitte besonders oft aufgerufen werden oder besonders viel Rechenzeit in Anspruch nehmen. Da bereits durch die Echtzeitstatistiken aus dem Statistics Fenster (Abbildung 3.3-14 b)) klar wurde, dass bei Weitem keine ausreichende FPS Anzahl gegeben war, wird das Rendering der Extratexturen in einer separaten Bildrate ausgeführt. Dafür wurden verschiedene Funktionen evaluiert. Weil Unity jedoch nur bedingt parallelisierbar ist, greifen unityspezifische Implementierungen⁴⁵, wie *Coroutines* und die *YieldInstructions*-Klasse nicht, da der gängige *WaitForSeconds* Rückgabewert zu ungenau ist. Die gewartete Zeit wird hier noch mit der internen Zeit der Unity-Engine skaliert. Der Rückgabewert *WaitForSecondsRealTime* bietet zwar genau diese gewünschte Wartefunktion, mit der eine stabile Bildrate berechnet werden kann, jedoch ist man immer noch abhängig von der Variable *z*, der eigentlichen Render- und Verarbeitungszeit der auszuführenden Methode. Die Wartezeit lässt sich hier also in FPS ausdrücken als

$$w = 1/x + z$$

Formel 3.3-3: Die Abhängigkeit der Umsetzung einer abweichenden Bildrate *w* an die Render- und Verarbeitungszeit *z* und die gewünschte FPS Anzahl *x* werden anhand dieser Formel mit der *Coroutine* und *WaitForSecondsRealTime* implementiert.

, wobei *x* die gewünschte FPS Anzahl darstellt. Diese wird um den Wert der systemabhängigen Variable *w* verzögert. Diese müsste also für jedes System neu gemessen

⁴⁵ <https://docs.unity3d.com/ScriptReference/> (8/03/2018)

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

werden und dann in die zu wartende Sekundenanzahl miteinbezogen werden. Glücklicherweise bietet Unity jedoch auch die *InvokeRepeat* Funktion. Diese garantiert den regelmäßigen Aufruf einer Methode anhand einer einzigen Zeitvariable:

$$w = 1/x$$

Formel 3.3-4: Die abweichende Bildrate w ist mit der *InvokeRepeat* Implementierung nur von einer einzigen Variable x abhängig.

Diese Funktion wird letztendlich auch in der prototypischen Implementierung genutzt, um die unabhängige Bildrate zu gewährleisten. Ein Nachteil besteht jedoch bei einer Verwendung auf Maschinen mit nur mäßiger Performanz. Hier wird das teure Mehrfachrendering trotz hoher Performanceinbußen stoisch in 30 FPS ausgeführt ohne dabei das Hauptrendering der VR-Szene zu berücksichtigen.

Für weitere Optimierung sollte eine Unity Szene auch inhaltlich angepasst werden. Da der Prototyp vorerst besonders auf einer einzigen 3D-Szene verwendet wird (Kapitel 3.4), die in vorherigen Arbeiten auf Simulation ausgerichtet war, ist diese inhaltlich nicht auf Echtzeitanwendung angepasst. Die 3D-Modelle, welche ebenfalls ursprünglich für offline Simulation und Marketing erstellt wurden, wurden ohne weitere Bearbeitung aus den CAD-Programmen importiert. Somit hatten diese 1) eine enorm hohe Anzahl an Triangles im Mesh und 2) wurden selbst für Kleinstteile gänzlich eigene Texturen verwendet. Ein einziges Objekt hatte durchaus die Anzahl der Triangles eines ganzen umfangreichen Levels eines modernen Computerspiels (hier gemessen ~10.000.000). Es fanden also sowohl deutlich zu viele Drawcalls, durch die einzelnen Texturen, als auch eine lang-andauernde Geometrieberechnung statt, durch die hohe Anzahl der Triangles. Die Anzahl der Drawcalls belief sich auf etwa 4000 und konnte auf 540 reduziert werden. Um sowohl 90 FPS für das VR-HMD, als auch 2*30 FPS für einen gängigen „Videostrom“ des Compositors zu gewährleisten, wurde die Szene dann weiter inhaltlich angepasst – Texturen wurden in Adobe's Photoshop⁴⁶ teilweise gemerged und herunterskaliert und Meshs wurden in 3DS Max auf bis zu 1/30 der Triangles optimiert, ohne zu starke optische Einbußen zu haben. Eine Referenzangabe⁴⁷⁴⁸ für aktuelle VR Konsumententechnologie ist etwa 50.000 bis 100.000 Triangles für die gesamte Szene. Da es sich bei den Quellen um Mobile Technologie handelt, wurde in dieser Thesis ein deutlich höherer Wert als angemessen angesehen. Mit etwa 1.000.000 Triangles werden letztendlich sowohl optisch als auch quantitativ zufriedenstellende Ergebnisse erreicht.

Damit die Informationen für das Hauptrendering auf dem HMD der Vive verfolgt werden konnten, wurde regelmäßig das Statistics Fenster von Unity genutzt. Wichtige Informationen, wie beispielsweise die aktuellen FPS, Auflösung oder Anzahl der zu rendernden Triangles werden dort in Echtzeit angezeigt. Will man nun jedoch ähnliche Statistiken für die besagten nebenläufigen Renderverfahren, die von den regulären Bildraten abweichen, ebenfalls einsehen, so müssen diese manuell gemessen und für Debugging Zwecke in einem eigenem Statistics 2.0 Fenster abgebildet werden (Abbildung 3.3-14 c)). Daher wurde großer Aufwand in das gleichzeitige Debugging investiert, um ein direktes Feedback für die Auswirkung von neuer Funktionalität oder Änderungen zu haben. Dieser

⁴⁶ <https://www.adobe.com/de/products/photoshop.html> (8/03/2018)

⁴⁷ <https://developer.oculus.com/blog/squeezing-performance-out-of-your-unity-gear-vr-game/> (8/03/2018)

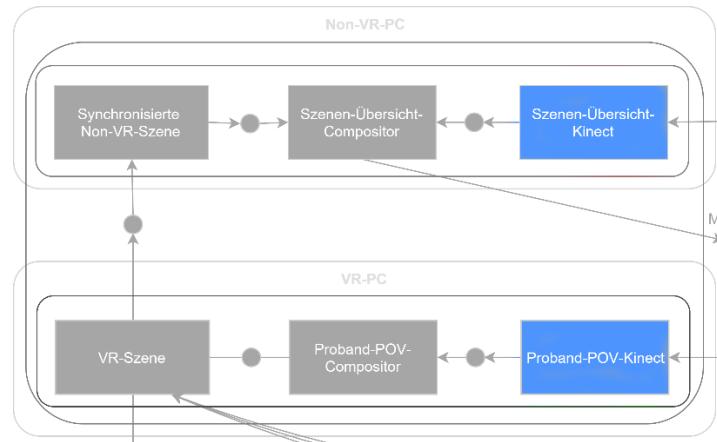
⁴⁸ <https://developer.oculus.com/documentation/unity/1.5/concepts/unity-perf/> (8/03/2018)

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

Aufwand wurde als sinnvoll angesehen, da die Performanz eine der kritischsten Variablen im gesamten VR-System ist, und gleichzeitig die Nutzbarkeit stark einschränken könnte.

Durch die stetige Nutzung der genannten drei Performanz-Monitore (Profiler, Statistics und Statistics 2.0) wurde die schwerwiegendsten Optimierungen durchgeführt. Zusätzlich gibt es auch noch weitere⁴⁹ Möglichkeiten, um eine Szene unityspezifisch anzupassen. Auf Grund der inhaltlichen Gegebenheiten der Szene (vergleichbar wenige Objekte) und dem VR-Rendering war es jedoch nicht nötig diese anzuwenden. Einige empfohlene Optionen, wie das Anti-Aliasing zu verringern oder zugunsten der Performanz gänzlich auszuschalten, wurden gezielt ausgelassen. So wurde das VR-Erlebnis nur wenig beeinträchtigt. Insgesamt konnte auf dem System eine stetige Bildrate von ~90 FPS erzielt werden, ohne zu große Einschnitte in die VR-Visualisierung vorzunehmen. Damit ist die Behandlung der beiden Szenen-Module abgeschlossen, die zusammen mit den Kinect-Modulen das Bildmaterial für das Compositing liefern.

3.3.3 Kinect Module



Die beiden Kinect-Module bieten die Funktionalität der Farb- und Tiefendatengewinnung aus den Kinect-Kameras. Dabei unterscheidet sich lediglich der Faktor der Personenerkennung, beziehungsweise -segmentierung. Dieser wird nur im Szenen-Übersicht-Kinect-Modul verwendet, womit die Extraktion der Personentextur auf die Kinect-API ausgelagert wird. Die Textur-Extraktion ist jedoch aus bereits genannten Gründen nicht möglich, wenn sich die Kamera bewegt. An dieser Stelle ist es allerdings praktikabel, da angenommen wird, dass die Kamera, die für die Szenen-Übersicht des Experimentators zuständig ist, vorwiegend statisch eingesetzt wird. Durch das Tracking der Kamera kann sie trotzdem im Raum verstellt werden. Es werden lediglich wenige Sekunden gebraucht, bis die Personenerkennung danach wieder funktioniert. Die finale Implementierung des **Szenen-Übersicht-Kinect-Moduls** befindet sich in Anhang A. Als Programmiersprache wurde sich nach ersten Tests mit Processing⁵⁰ letztendlich für C# entschieden, da sich andere Module (Unity) ebenfalls für C# eignen und die Kinect-API damit am umfangreichsten ist. Außerdem bietet sie bereits beispielhafte Implementierungen⁵¹ der verschiedenen Kinect-Funktionalitäten, an denen sich bei der Umsetzung orientiert werden konnte. Dadurch ist es bereits ohne weitere Berechnungen möglich, die beiden Bildströme (Farbe- und Tiefe) pixelweise in Verbindung zu setzen, sodass das Matching und die Kalibrierung der separaten

⁴⁹ <https://unity3d.com/de/learn/tutorials/topics/virtual-reality/optimisation-vr-unity> (8/03/2018)

⁵⁰ <https://processing.org/> (8/03/2018)

⁵¹ <https://msdn.microsoft.com/en-us/library/hh855373.aspx> (8/03/2018)

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

Kameras nicht manuell umgesetzt werden musste. Die *CoordinateMapper*-Klasse⁵² der Kinect API stellt Folgendes zur Verfügung:

- 1920x1080 Pixel Farbbild im BGRA32-Format als Byte-Array
- 511x423 Pixel Tiefenbild als Int-Array
- Umrechnungs-Array mit 1920x1080 Int-Tupeln

Jeder der Int-Tupel des Umrechnungs-Arrays ist ein X-Y Paar und gibt die Indizes der korrespondierenden Pixelpositionen im Farbbild an. Wenn also der *erste* Wert des Umrechnungs-Arrays das beispielhafte Datenpaar (3,4) wäre, dann wird der *erste* Wert aus dem Tiefenbild-Array dem Pixel des Farbbildes an Stelle (3,4) zugeschrieben. So findet eine Abbildung von den ungleichgroßen Arrays aufeinander statt. Die Werte des Farbarrays werden klassischerweise als 1D-Array wiedergegeben und können dann mit der Formel

$$1DLOCATION = X + Y * WIDTH$$

Formel 3.3-5: 1D Repräsentation eines 3D Arrays.

unter Angabe der 2D-Position abgerufen werden.

Das Problem der Abbildung von den Farbpixeln auf die Tiefendaten erscheint zwar trivial, jedoch bestehen einige Faktoren, die eine einfache Abbildung verhindern. In der Kinect werden zwei separate Kameras verwendet – eine Infrarotkamera für die Gewinnung der Tiefendaten und eine herkömmliche Farbkamera für die Gewinnung der Farbdaten. Diese Kameras unterscheiden sich nicht nur hinsichtlich der Auflösung des Bildes, sondern auch hinsichtlich des Seitenverhältnisses, der Linsenkrümmung, des FOV und der Brennweite. Ohne die technisch aufwendige Kalibrierung aller intrinsischen und extrinsischen Kameraparameter der beiden Kameras ist eine triviale Abbildung der Pixel nicht möglich. Wie am Schraffurmuster in Abbildung 3.3-15 zu sehen ist, kann nicht allen Pixel ein exakter Tiefenwert zugeordnet werden, weswegen die Nutzung der ausführlichen Kinect-API an dieser Stelle notwendig ist, die eine surjektive Abbildung von Farbpixeln auf Tiefenpixel bereitstellt.

⁵² <https://msdn.microsoft.com/de-de/library/dn785530.aspx> (8/03/2018)

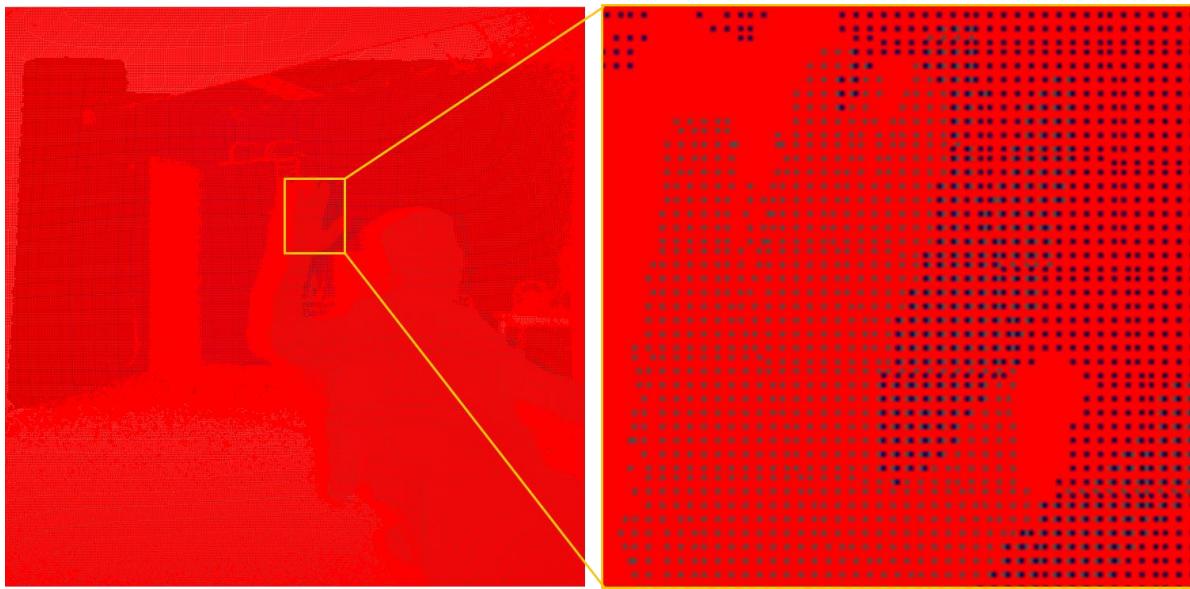


Abbildung 3.3-15: Problematik der Extrapolation von Tiefeninformationen anhand einer trivialen, injektiven Abbildung der Farbpixel auf Tiefenpixel. Farbpixel ohne Tiefenwert sind rot dargestellt. Farbpixel mit Tiefeninformationen sind in Schwarz-Blau-Abstufungen dargestellt. Große rote Bereiche sind zurückzuführen auf nicht sichtbare Bereiche der Tiefenkamera auf Grund mehrerer Zentimeter Versatz der physischen Kameras.

Letztendlich wird aus dem Szenen-Übersicht-Kinect-Modul dann ein maskiertes Bild in 1920x1080 Pixeln (Abbildung 3.3-16 links) und ein Array in gleicher Auflösung mit entsprechenden Tiefenwerten (Abbildung 3.3-16 rechts) an den Szenen-Übersicht-Compositor gesendet.

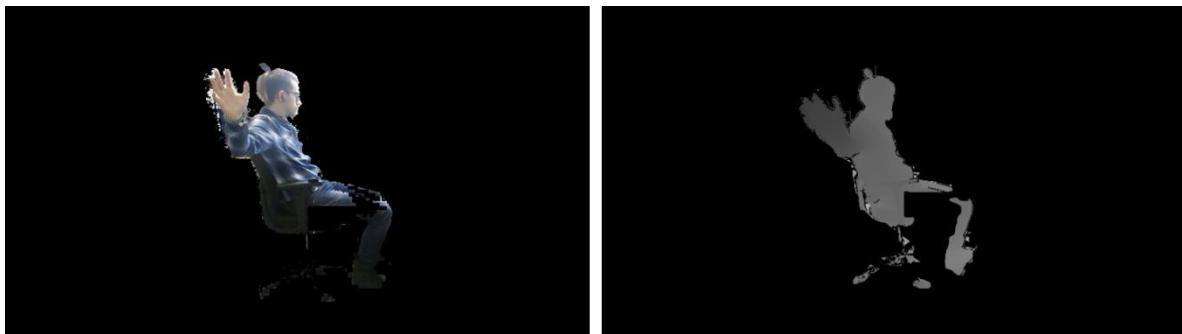
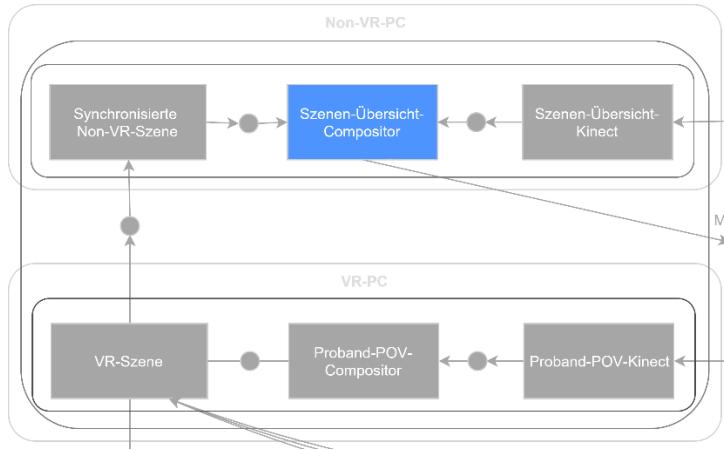


Abbildung 3.3-16: *Links:* Maskiertes Ausgabebild des Szenen-Übersicht-Kinect-Moduls. *Rechts:* Visualisierung vergleichbarer Tiefendaten.

Da zwar eine zweite Kinect-Kamera vorhanden war, die Texturextraktion der Kinect API jedoch nicht für die POV-Simulation geeignet ist, wurden nur die Rohdaten der zweiten Kamera genutzt. Im **Proband-POV-Kinect-Modul** wird also nicht das Computer-Vision-Backend der Kinect API verwendet. Hier wird lediglich die Abbildung der Farbpixel auf die Tiefendaten der Bildströme verwendet, um dann die unbearbeiteten Informationen an den Proband-POV-Compositor zu senden. Die Personenerkennung und Texturextraktion für den Proband-POV wird ausführlich in Kapitel 3.3.5 erläutert. Im Szenen-Übersicht-Compositor hingegen wird die statische Segmentierung wie beschrieben verwendet.

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

3.3.4 Szenen-Übersicht-Compositor / Darstellung von Szene und Proband für den Experimentator



Das *Szenen-Übersicht-Compositor-Modul* ist sowohl für die Berechnung, als auch für die Darstellung der virtuellen Umgebung und der darin befindlichen Personen zuständig. So können Außenstehende, wie ein Experimentator, einen Einblick in die Szene bekommen. Er sieht gleichzeitig den Probanden und sich selbst in Relation zur virtuellen Position im physischen Raum. Das Modul verwendet insgesamt vier Datenströme (RGBA-Szene, Tiefe-Szene, RGBA-Kinect, Tiefe-Kinect) und verarbeitet diese zu einem einzelnen Bildstrom. Am Ende des Compositings stellt das Modul das Ergebnis außerdem mithilfe von Grafikprogrammierung für den Experimentator auf einem Bildschirm zu Verfügung.

Bereits vor dem eigentlichen Compositing ergibt sich hier schnell ein weiteres Problem der Kameraangleichung. Die Auflösung der Kinect-Kamera entspricht in jedem Fall 1920x1080 Pixel. Die Rendertextur in Unity ist ebenfalls bereits auf diese Größe angelegt, sodass hier eine bijektive Abbildung zwischen den Kinect- und den Unity-Pixeln besteht. Auch die Position der Kameras ist bereits angeglichen, unter Nutzung der Vive Tracker. Dadurch alleine ergibt sich jedoch noch kein akzeptables Compositing zwischen zwei unterschiedlichen Kamerabildern. Obwohl für den Zweck des Prototypings bereits viele Parameter vernachlässigt werden können, wie beispielsweise die radiale Verzerrung der optischen Linse, extrinsische sowie intrinsische Parameter der Kinect, so muss doch zumindest der FOV angeglichen werden. Dieser ist maßgeblich dafür verantwortlich, welcher Ausschnitt der Szene auf den 1920x1080 Pixeln dargestellt wird. Der Unterschied des FOV der virtuellen Kamera wird in Abbildung 3.3-17 verdeutlicht.

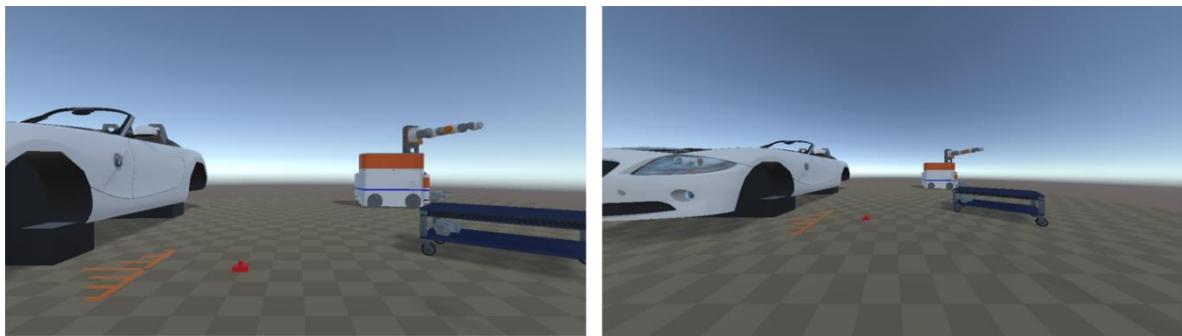


Abbildung 3.3-17: Links: Unity-Kamera mit vertikalem FOV von 60°. Rechts: Selbe Kameraposition mit geändertem vertikalem FOV von 100°.

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

Da die Parameter der physischen Kamera an dieser Stelle nicht verändert werden können, ist die einzige Möglichkeit die Anpassung der virtuellen Kamera in Unity. Außerdem soll die Auflösung des Kinectbildes in keiner Weise verringert werden, sodass beispielsweise nur ein Teilausschnitt genutzt werden könnte, um die Ausschnitte anzulegen. Um Bilder verschiedener Kameras oder Linsen in Korrelation zu setzen, müssen die Kameras für gewöhnlich aufeinander kalibriert werden. Etablierte Verfahren werden von Zhang (2000) und Tsai (1987) vorgestellt. Die umfangreiche Kalibrierung und Angleichung ist jedoch im Umfang dieser Thesis zeitlich nicht realisierbar, wie beschrieben für den Zweck der Thesis aber auch nicht nötig.

Die Kinect hat einen festgelegten FOV von etwa 60° vertikal und 70° horizontal⁵³. Der FOV ist eng verknüpft mit der Brennweite der genutzten Linse, womit ein erster Ansatz die Angleichung der digitalen Linse der Unity-Kamera an die Brennweite der Kinect wäre. In Unity wird die Kamera jedoch nur digital beschrieben. Der Eingabeparameter, der hier direkt verändert werden kann, ist daher der FOV – jedoch nur der vertikale. Der horizontale wird anhand des Viewport-Seitenverhältnisses berechnet und ist nicht unabhängig vom vertikalen FOV zu behandeln. Durch diverse Umrechnungen und die interne Kalibrierung der Farbkamera mit dem Tiefenbild, hat der FOV der Kinect jedoch kein reguläres Verhältnis im Vergleich zur Ausgabe im Format 16:9. Standardmäßig, wie auch in Unity verwendet, wird der horizontale Wert des FOV in Abhängigkeit zum vertikalen FOV und Seitenverhältnis mit folgender Formel berechnet:

$$H = 2\arctan(\tan(V/2) * w/h)$$

Formel 3.3-6: Berechnung des horizontalen Wertes des FOV H anhand des vertikalen Wertes V und des Seitenverhältnisses, bzw. Höhe h und Breite b des Viewports der Kamera nach aktueller Microsoft DirectX Dokumentation⁵⁴. V und H sind in Radian angegeben.

Damit würde sich für den vertikalen Wert der Kinect rechnerisch ein horizontaler Wert von $2\arctan(\tan(1,0472/2) * 1920/1080) = 1,59685421$ Rad ergeben, was etwa 91° entspricht und nicht den für die Kinect zutreffenden 70° . Somit ist es ausgeschlossen auf einfacherem Wege den FOV einer Unity-Kamera mit dem der Kinect anzulegen. Diese Diskrepanz müsste in der finalen Umsetzung auf niedrigster Stufe durch Anpassung der Projektionsmatrix der virtuellen Kamera behandelt werden. Im aktuellen Prototyp wird jedoch das Mittel der Werte genommen. Die gewünschten Parameter, 60° vertikal und 70° horizontal, ergeben bei Anwendung und Umstellung der Formel an einem 1920x1080 Bild $\sim 91^\circ$ horizontalen und $\sim 43^\circ$ vertikalen FOV. Da nur der vertikale Wert angepasst werden kann, wird entsprechend das Mittel zwischen den gewünschten vertikalen Werten von 43° und 60° genommen. Die Wahl des vertikalen FOV-Wertes für die Unity-Kamera fällt damit auf das arithmetische Mittel $(43 + 60)/2 = 51,5$. Die optische Bewertung des Ergebnisses wird als dem Zweck entsprechend sehr gut angesehen (Abbildung 3.3-18 oben).

Nachdem nun alle Bildsignale bereits weitgehend im Voraus angeglichen wurden, können alle weiteren Probleme, wie die verschiedenen Datenformate der Tiefe oder das gespiegelte Bild der Kinect nun durch einfache Indexierung in einer geschachtelten For-Schleife behandelt werden. Für das Compositing kommen aber auch andere Methoden und Programmiersprachen als C# infrage. Erste Implementierungen wurden wieder in der vielversprechenden Processing-Sprache implementiert, die jedoch dadurch, dass sie auf Java

⁵³ <https://smeenk.com/kinect-field-of-view-comparison/> (8/03/2018)

⁵⁴ [https://msdn.microsoft.com/en-us/library/windows/desktop/bb147302\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/bb147302(v=vs.85).aspx) (8/03/2018)

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

basiert, nicht für Echtzeit-Bildverarbeitung solch großer Datenmengen geeignet war. Eine weitere getestete Alternative ist Python, unter Nutzung von OpenCV⁵⁵. OpenCV wurde als besonders geeignet angesehen, da das Umfeld der Funktion sich in Felder wie Bildverarbeitung und Computervision einordnen lässt. Exakt darauf ist OpenCV ausgelegt. Die Einfachheit von Python lässt wiederum agiles Prototyping zu. Diese Python-OpenCV-Kombination wurde jedoch ebenfalls nach ersten Tests ausgeschlossen. Python und OpenCV bieten zwar sehr passende Funktionalität, jedoch sind diese einzelnen Funktionen nicht individualisierbar, sodass jede Transformation einzeln durchgeführt werden muss, obwohl diese auch in einem Durchlauf einer einzigen geschachtelten Schleife vollzogen werden können. Da Python große Performanz-Nachteile hat, wenn man Elemente in Arrays einzeln und nicht gruppenweise indexiert, ist die finale Version letztendlich ebenfalls in C# umgesetzt. Somit basiert das gesamte Projekt auf C# (ausgenommen Native Plug-Ins und Grafikoperationen).

Das Compositing sieht abstrakt wie folgt aus:

```
INIT arrays: KinectDepth, KinectColor, SceneDepth, SceneColor, FinalOutput
Fetch data from different external sources
Wait until every source's output is available
FOR each row of the theoretical 2D images
    FOR each column of the theoretical 2D image
        IF no valid depth value is located in flipped KinectDepth at
            pixel [row,column] THEN
            Write SceneColor at the pixel at [row, column] in FinalOutput
        ELSE IF depth value in flipped KinectDepth at pixel [row,column] is
            lower than value in SceneDepth at pixel [row,column] THEN
            Write KinectColor at the pixel at [row, column] in FinalOutput
        ELSE
            Write SceneColor at the pixel at [row, column] in FinalOutput
        END IF
    END FOR
END FOR
Render FinalOutput to screen
```

Wie man sieht, können Pre-Processing Schritte, die in Python beispielsweise mehrmals die gesamten Arrays verarbeiten müssten, vollständig ausgelassen werden. Dies wird durch direkte Vergleiche (If-Statements) und unterschiedliche Indexierung (flipped-Stichwort) umgesetzt. Außerdem müssen je nach Fall auch keine Werte umgerechnet werden, die bei vorherigen Lösungen bereits für jedes Array durchgeführt werden mussten. Falls der Tiefenwert eines Pixels der Kinect beispielsweise einen ungültigen Wert beinhaltet, so werden alle weiteren Operationen für diesen Pixel auf allen verbleibenden Arrays eingespart. Dies ist besonders häufig der Fall, da die Kinect-Maskierung der Person durch solche Werte dargestellt wird.

Durch die Bearbeitung von 2D-Bild-Daten und die Einbeziehung der Tiefe als dritte Dimension ist zum einen die Berechnung des Compositings kostengünstig und zum anderen die Ausgabe trotzdem realistisch, da die Verdeckungen sehr präzise mitberechnet werden, wie in Abbildung 3.3-18 rechts zu sehen ist.

⁵⁵ <https://opencv.org/> (8/03/2018)

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

Während die Interprozesskommunikation (IPC) wie gehabt über UDP abläuft, müssen innerhalb des Compositor-Moduls mehrere Threads genutzt und auch synchronisiert werden. Die Inter-Thread-Kommunikation wird hier über eine Implementierung generischer *ConcurrentQueues* gehandelt. Diese sind thread-safe und werden für die Synchronisierung nur dann beansprucht, wenn auf allen Queues mindestens ein Element liegt, das aus der Schlange genommen werden kann. Diese Implementierung löst somit auch das Producer-Consumer-Problem, welches durch den Empfang mehrerer unabhängiger Datenströme aufkommt.

Der letzte Schritt des Compositings, nachdem das finale Array vollständig beschrieben ist, stellt diesen Array-Inhalt dann als Textur auf dem Bildschirm dar. Für diese Visualisierung wird die C#-Bibliothek OpenTK genutzt. Diese bindet die OpenGL Grafikbibliothek in C# ein. Es ist anzumerken, dass die Compositing-Operationen selbst ebenfalls sehr geeignet wären, um auf die GPU ausgelagert zu werden, da es pixelweise Operationen sind. Weil das Profiling der Unity-Szene jedoch bereits gezeigt hat, dass die Grafikkarte der Flaschenhals des verwendeten physischen Systems ist, und für kürzere FPS-Einbrüche verantwortlich ist, wurde die Compositing-Berechnung vorerst auf der CPU belassen. Bei einer produktfertigen Implementierung müssten genauere Messungen vorgenommen werden, beziehungsweise bestenfalls eine zweite Grafikkarte für das Compositing miteinbezogen werden, da die pixelweisen Berechnungen unter anderen Umständen ganz klar der Zuständigkeit der Grafikkarte unterliegen sollten.

Letztendlich wird die Compositing-Textur dann auf ein Fullscreen-Quad gezogen, danach wird der Doublebuffer getauscht, wie es in der Computergrafik gängig ist. Das Ergebnis ist ein Bildstrom der in 30 FPS wiedergegeben und optisch in Abbildung 3.3-18 links dargestellt wird. Ein Bewegtbild ist in Anhang A angefügt. Die Avatarform, die hier gewählt wurde, lässt sich auf dem Realismus-Avatarkontinuum wie in Abbildung 3.3-18 unten einordnen.

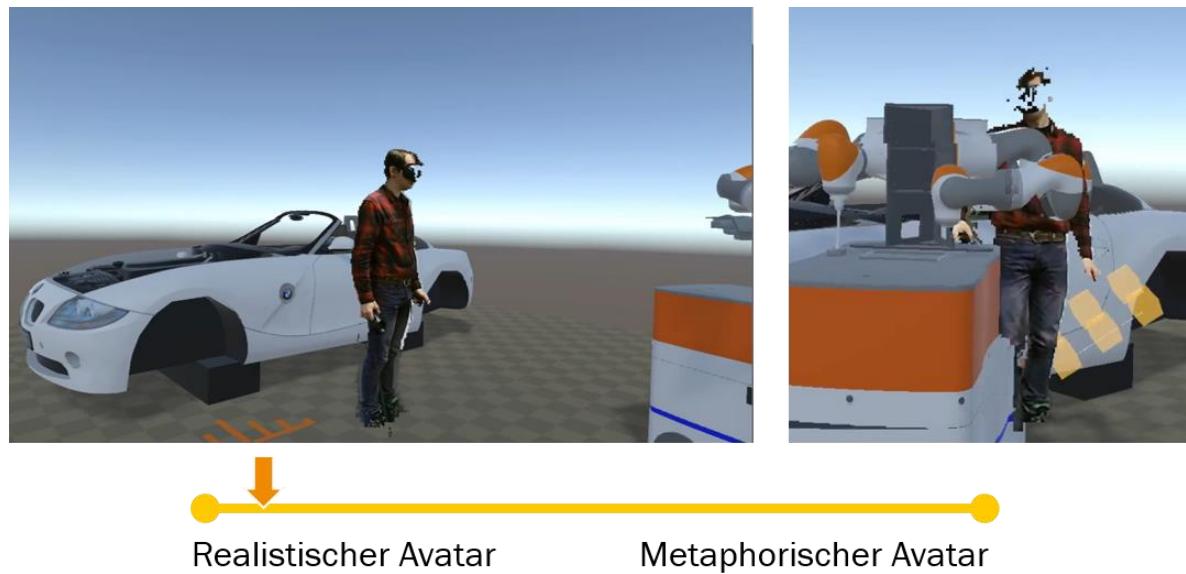
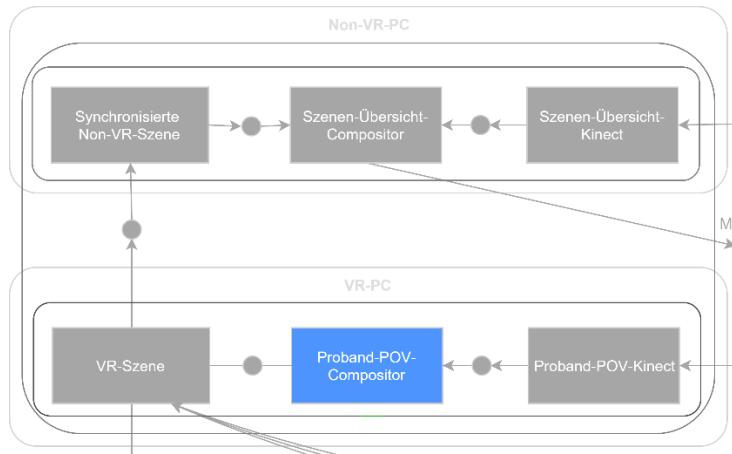


Abbildung 3.3-18: Oben links: Finales Compositing des Szenen-Übersicht-Compositor-Moduls. Oben rechts: Fokus auf präzise Verdeckungsberechnung durch Einbeziehung von 3D-Daten. Unten: Einordnung der Avatarrepräsentation in das Realismus-Metapher-Avatarkontinuum.

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

3.3.5 Proband-POV-Compositor-Modul / Darstellung des Experimentators für den Probanden



Im Zuge der zweiten Iteration des Thesiszyklus wurde das Compositor-Modul entwickelt, welches letztendlich für die Darstellung von Außenstehenden, wie dem Experimentator, innerhalb des VE zuständig ist. Obwohl das Modul auf den ersten Blick eine ähnliche Funktionalität wie das vorher beschriebene Compositor-Modul bietet, so ist das Vorgehen grundverschieden. Der Ansatz dieses Compositors ist völlig unabhängig von der Personensegmentierung, welche die Kinect-API zu Verfügung stellt. Der Ansatz der Segmentierung ist als solcher ebenfalls noch nicht in Publikationen zu finden und somit neuartig. Um die Textur des Experimentators für den Probanden im VE zu visualisieren, wird VR-nahe Technologie genutzt:

- *Kinect-Farb- und Tiefeninformationen*: Die Kinect liefert lediglich das unverarbeitete Farbbild und die dazugehörigen Tiefeninformationen pro Pixel (Abbildung 3.3-19).
- *VR-Szenen-Tiefe*: Des Weiteren benötigt das Modul die rohen Tiefeninformationen der VR-Szene (Abbildung 3.3-19).
- *Experimentator-Position*: Außerdem wird die Position des Experimentators mithilfe eines Trackers in das Koordinatensystem der Szene übertragen.
- *Experimentator-Bounding-Box*: Auf Basis dieser Positionsangabe wird in der Szene eine virtuelle BB mit menschlichen Maßen gerendert (nicht sichtbar für den Probanden) (Abbildung 3.3-19). Dieses BB-Bild wird ebenfalls an den Compositor gesendet.
- *Proband-Experimentator-Distanz*: Die letzte Komponente, die das Modul für die Segmentierung und das Compositing benötigt, ist die absolute Distanz zwischen dem Probanden und dem Experimentator.

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

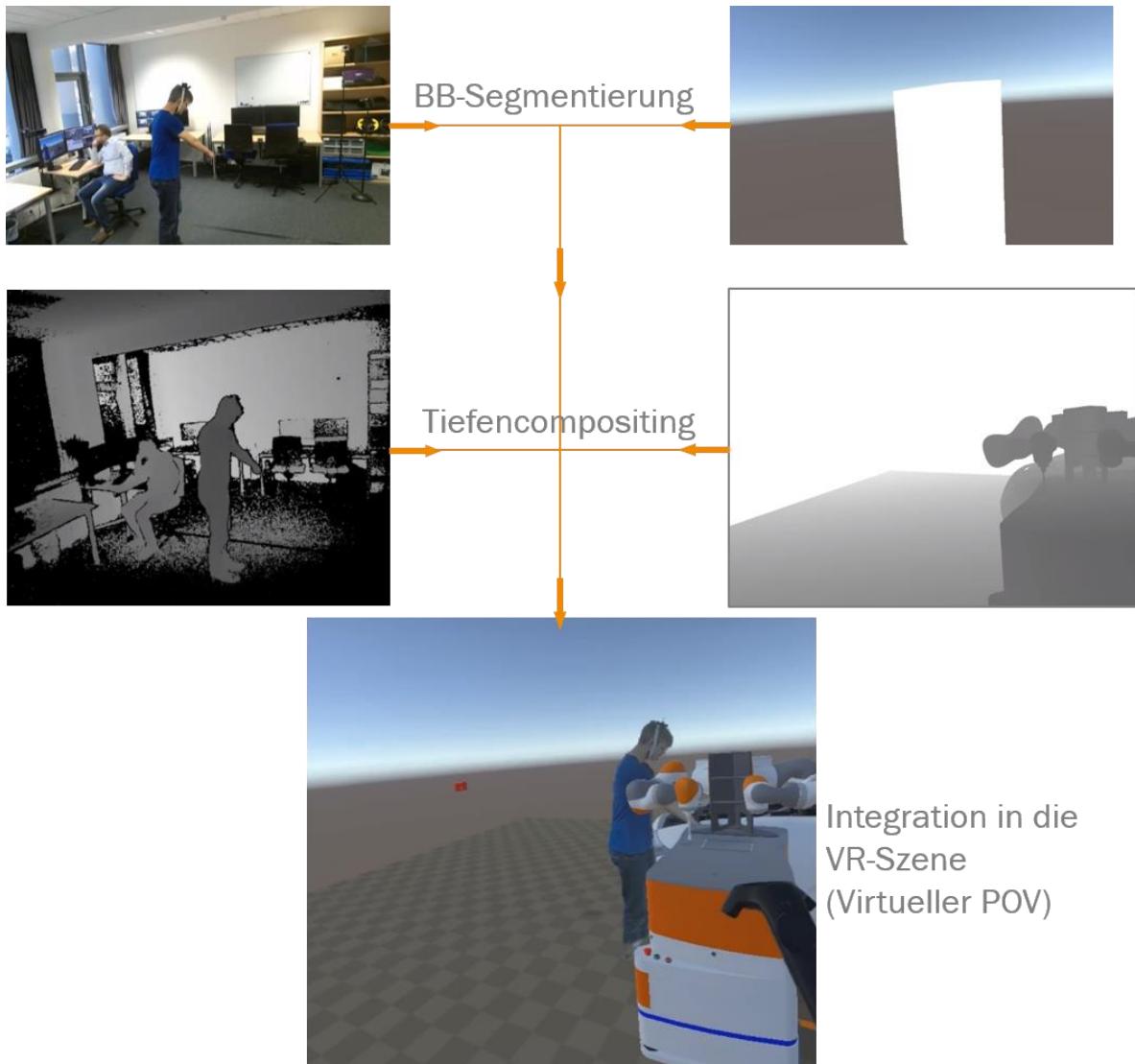


Abbildung 3.3-19: Ablauf der POV-Visualisierung unter Nutzung des Bounding-Box-Ansatzes und des Tiefencompositings.

Die BB kann durch a priori Wissen des physischen Raums genutzt werden. Es wird angenommen, dass sich während des VR-Experiments keine anderen Objekte innerhalb der Fläche befinden, in der das Experiment stattfindet. Auf Grund dieser Annahme ist es möglich, dass durch einfache Tiefenvergleiche die Experimentatorenpixel von den Nicht-Experimentatorenpixeln unterschieden werden können. Innerhalb dieser BB wird eine pixelweise Operation durchgeführt, die jeden zugehörigen Tiefenwert mit der Proband-Experimentator-Distanz vergleicht. Falls ein Pixel nun einen Tiefenwert besitzt, der sich innerhalb des Bereichs $[0, \text{ProbandExperimentatorDistanz}]$ befindet, so gehört dieser Pixel zur Textur des Experimentators und kann separat abgespeichert werden. Letztendlich sendet das Proband-POV-Compositor-Modul dann eine Textur die nur noch die Größe einer neuen BB hat. Mithilfe der Berechnungen werden auch die Indizes der exakten BB für den Experimentator klar – unabhängig von der approximierten BB in Unity.

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

Der folgende Pseudo-Code beschreibt den Algorithmus näher:

```
INIT arrays: KinectDepth, KinectColor, SceneDepth, SceneColor, FinalOutput
INIT int: SubjectExperimenterDistance, RightBBIndex, TopBBIndex, LeftBBIndex,
BottomBBIndex

Fetch data from different external sources
Wait until every source's output is available
FOR each row of the theoretical 2D images
    FOR each column of the theoretical 2D image
        IF no valid depth value is located in flipped KinectDepth at
pixel [row,column] THEN
            Write alpha of value 0 at [row, column] in FinalOutput
        ELSE IF color value in SceneColor at pixel [row,column] is not
pure white THEN
            Write alpha of value 0 at [row, column] in FinalOutput
        ELSE IF depth value in flipped KinectDepth at pixel [row,column] is
not lower than value in SceneDepth at pixel [row,column] THEN
            Write alpha of value 0 at [row, column] in FinalOutput
        ELSE IF the depth value in flipped KinectDepth at pixel [row,column] is in
range between 0 and SubjectExperimenterDistance THEN
            Write KinectColor at the pixel at [row, column] in FinalOutput and
update bounding-box indices
        ELSE
            Write alpha of value 0 at [row, column] in FinalOutput
        END IF
    END FOR
END FOR

IF the bounding-box indices state a minimum-size bounding box THEN
    INIT array: BBArrayToSend
    FOR each row of the theoretical 2D bounding-box
        FOR each column of the theoretical 2D bounding-box
            Write corresponding BGRA values from FinalOutput into BBArrayToSend
    END FOR
END FOR

Send BBArrayToSend, RightBBIndex, TopBBIndex, LeftBBIndex and BottomBBIndex to
Unity VR-Scene-Module
```

Durch diese VR-nahe Segmentierungsmethode ist es möglich eine Personentextur aus einem 2D-Bild zu extrahieren, ohne dabei auf das Computervision-Back-End der Kinect API zurückzugreifen, welches der Grund ist, dass die Segmentierung im Szenen-Übersichts-Compositor nur statisch verwendet werden kann. Die hier vorgeschlagene Methode ermöglicht es nun die Kinect-Kamera dynamisch – also bewegbar – zu verwenden. Somit ist es beispielsweise möglich, die Kamera am Kopf des Probanden zu befestigen und so den POV des Probanden zu simulieren. Dafür wurde ein spezielles Rig „gebaustelt“ (Abbildung 3.3-20), da eine Kamera mit den Maßen und dem Gewicht der Kinect sonst nur schwer am Kopf befestigt werden kann. Eine weitere Schwierigkeit ist es an dieser Stelle, die Kinect nur den Bewegungen des Kopfes anzupassen, und nicht noch eigene Bewegungsfreiheitsgrade zu geben.

Um dieses Problem zu lösen, wurde ein Fahrradhelm als Basis des Rigs mit Do-It-Yourself (DIY)-Charakter verwendet. Ein Teil des Helms, der in etwa den Maßen der Kinect

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

entspricht, wurde aus der Helmfront geschnitten. Damit die Stabilität und die eingeschränkte Freiheit der Kamera gewährleistet werden kann, wurde dann eine Aluminium-Trägerplatte an dieser Stelle befestigt. Diese dient als Sockel für die Kamera. In der Mitte der Trägerplatte wurde mit einem Stufenbohrer in drei Stufen ein Loch mit verschiedenen Ausmaßen gebohrt. Diese kegelförmigen Abstufungen mit harten Kanten gewährleisten einer Schraube und einer gegenziehenden Mutter Halt. Die Kinect-Fassung wurde dann an diese Schraube angebracht. Die Trägerplatte selbst ist mit Kabelbindern wiederum am Helm befestigt.

Insgesamt garantiert diese Technik, dass 1) die Kinect nicht verrutscht, sondern durch die Schraube, die auf der mittleren Abstufung des Lochs sitzt, fest an der Platte gehalten wird, und, dass 2) die Kinect trotzdem wieder lösbar von dem Konstrukt ist, da die unterste Abstufung des Lochs zulässt, die Schraube von unten zu lösen.



Abbildung 3.3-20: Überblick über das DIY-Rig für die Kinect zur Befestigung der Kinect am Kopf des Probanden.

Während zuerst angenommen wurde, dass die Position der POV-Kamera durch die getrackte Position des HMDs bestimmt werden kann, stellte sich nach ersten Tests heraus, dass die Kamera separat getrackt werden muss. Damit annähernd ein menschliches Sichtfeld simuliert werden kann, muss die Kinect einige Grad weit nach unten rotiert werden. Dazu kommt, dass der Abstand zwischen Kinect und HMD tatsächlich stark variiert – je nach Kopfform des Probanden. Mit dem zusätzlichen Tracking der Kamera wurde diese dann vorerst optisch kalibriert. Insgesamt werden bezüglich der Kalibrierung nun Ergebnisse erzielt, die bei verschiedenen Probanden für gut befunden wurden, ohne dass jedes Mal manuell kalibriert werden müsste. Die optische Kalibrierung wurde ebenfalls anhand eines Trackers durchgeführt (Abbildung 3.3-21 links und mitte).

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

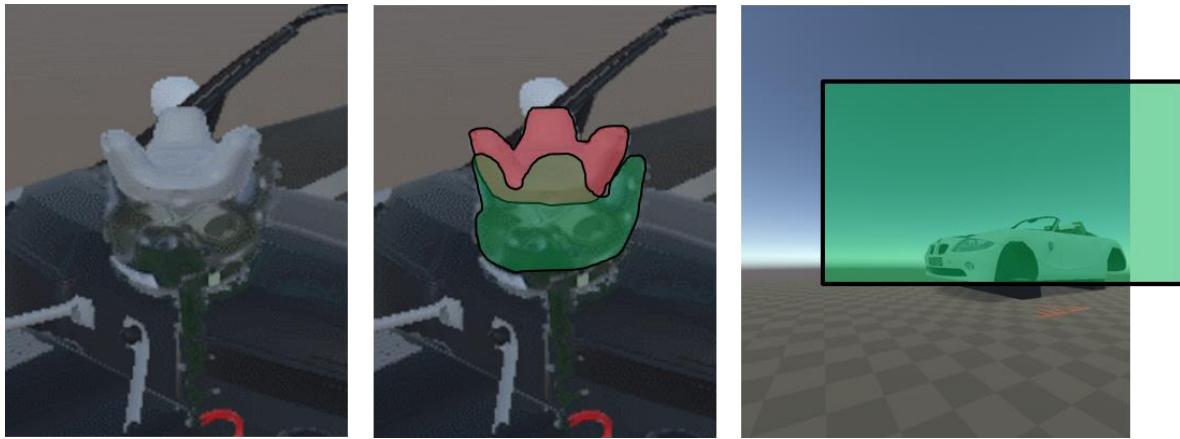


Abbildung 3.3-21: *Links:* Zugrundeliegendes Bild der Kalibrierung. *Mitte:* Hervorgehoben grün: Einbeldung der Textur eines realen Trackers, rot: Darstellung eines digitalen Modells des Trackers in der Szene als Groundtruth. Zu beobachten sind insbesondere Unterschiede, die sich auf die verschiedenen Betrachtungswinkel von Augen zu Kinect zurückführen lassen. *Rechts:* Der POV von einem (!) Auge in Relation zum Sichtfeld, das die Kinect simulieren kann (in grün).

Nachdem nun also die Sicht des Probanden simuliert wird, die Daten im Compositor segmentiert und von diesem wieder an die Unity-Instanz gesendet werden, muss die Textur noch in Unity korrekt abgebildet werden. Dafür wird innerhalb der Szene eine Ebene im Format des ursprünglichen Kinect-Bilds verwendet. Diese wird in Abhängigkeit der Distanz von Proband und Experimentator im virtuellen Raum positioniert und skaliert. Die Kinect bildet jedoch in keiner Weise den gesamten POV eines Menschen ab. Daher wurde der Ausschnitt, den die Kinect abzubilden vermag, ebenfalls optisch kalibriert. Das Verhältnis bezüglich SteamVR und dem Vive HMD wird in Abbildung 3.3-21 rechts illustriert. Es kann also mit der Kinect nur ein Ausschnitt des eigentlichen Sichtfelds simuliert werden.

Die Z-Position der Textur im dreidimensionalen virtuellen Raum wird anhand der Distanz von Proband zu Experimentator berechnet. Die gesamte Textur ist ebenfalls kalibriert, sodass diese ohne Probleme auf der Z-Achse verschoben werden kann. Durch Skalierung wird dann der Inhalt angepasst, sodass die Größe optisch relativ wieder zur Entfernung passt. Damit jedoch die BB, welche die Experimentator-Textur beschreibt, auch an der korrekten Position innerhalb der Full-HD-Textur der Kinect angezeigt wird, werden die berechneten Indizes aus dem Compositor verwendet. So wird die BB an die entsprechende Stelle der Textur geschrieben.

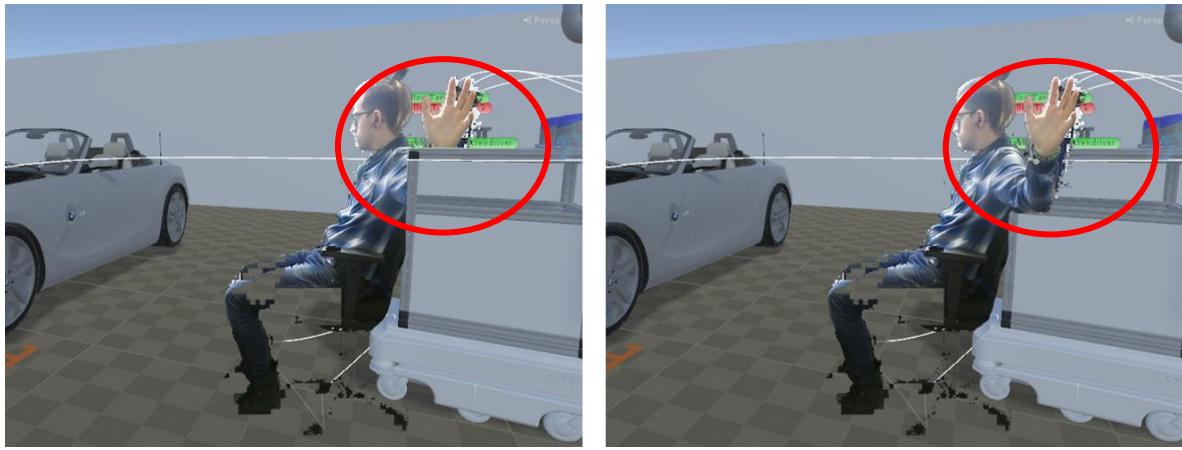


Abbildung 3.3-22: Beispieldfall für falsch-negatives Rendering. Da die texturierte Ebene räumlich hinter dem Objekt platziert ist, können die Pixel des ausgestreckten Arms nicht vor das Objekt gerendert werden. **Links:** Falsch negatives Beispiel. **Rechts:** Richtig positives Beispiel.

Eine weitere Schwierigkeit bei diesem Vorgehen ist die Behandlung der in Kapitel 2.2.1 beschriebenen Verdeckungsproblematik. Die Verdeckungen werden auf Grund der Einbeziehung der Tiefeninformationen von VR-Szene und Kinect bereits präzise berechnet. Jedoch fällt in Abbildung 3.3-22 auf, dass die Integration der 2D-Textur in der virtuellen Szene trotz akkurate Berechnung der Verdeckungen dennoch noch zu Fehlern führen kann. Man stelle sich die 2D-Textur im 3D-Raum vor. Durch die Verdeckungsberechnung wird die Textur an allen Stellen, an denen die Person von virtuellen Objekten verdeckt werden würde, ausgeblendet (Alphawert = 0). So werden alle Fälle, in denen ein Pixel der eigentlich hinter einem virtuellen Objekt liegen würde, aber durch die planare 2D-Darstellung der 3D-Informationen auf einer Ebene vor das Objekt gerendert wird, ausgeschlossen. Betrachtet man Tabelle 3.3-2, so werden alle falsch-positive Fälle ausgeschlossen. Falsch-negative Pixel hingegen können immer noch auftreten, wie es in Abbildung 3.3-22 zu sehen ist. Weil die texturierte Ebene in der Tiefe auf einen einzigen Wert beschränkt ist, kann es passieren, dass die eigentlich dreidimensionalen Abbildungen darauf soweit von diesem Wert abweichen, dass Teile davon das virtuelle Objekt verdecken würden. Dies ist jedoch nicht ohne weiteres abbildbar.

	Pixel ist vor Objekt	Pixel ist hinter Objekt
Pixel wird vor Objekt dargestellt	Richtig positiv (true positive)	Falsch positiv (false positive)
Pixel wird hinter Objekt dargestellt	Falsch negativ (false negative)	Richtig negativ (true negative)

Tabelle 3.3-2: Einordnung des Renderings nach falsch/richtig negativ/positiv.

In der prototypischen Implementierung wird dieses Problem gelöst, indem Wissen um den menschlichen Körper mit einberechnet wird. Die Ebene wird um eine Armlänge in Richtung der POV-Kamera versetzt. Da falsch-positive Pixel bereits durch die Verdeckungsberechnung ausgeschlossen werden, kann die Ebene also ohne größere Probleme vor Objekte versetzt werden. Lediglich sehr große Differenzen würden Probleme, wie eine verfälschte Parallaxe der Textur mit den virtuellen Objekten erzeugen. Durch den Versatz in Richtung Proband wird wiederum falsch-negativen Pixeln weitgehend entgegengewirkt. Aufgrund der angenommenen Heuristik, dass Extremitäten eines Körpers

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

sich nur maximal um eine Armlänge von dessen Mittelpunkt entfernen können, wird die Ebene um diesen Wert vor mögliche verdeckende virtuelle Objekte gesetzt (Abbildung 3.3-22 rechts). Das einzige größere Problem was sich dabei ergibt, ist wie bereits genannt die empfundene Parallaxe – faktisch ist die Textur nun an einer inkorrekt Stelle im virtuellen Raum. Dies wird jedoch in Kauf genommen, da die Verdeckungen als größerer Störfaktor eingestuft werden. Abschließend werden also falsch-positive Fälle durch Verdeckungsberechnung und falsch-negative durch Versatz der Textur ausgeschlossen.

Insgesamt wird die Segmentierung der Textur im POV-Compositor von einem 3D-Problem auf 2D-Berechnungen reduziert, was leistungsmäßig wiederum sehr kostengünstig bezüglich der Rechenzeit ist. Weitere Kosten werden gespart, da die meisten Operationen nur auf einem BB-Bereich des Full-HD-Bildes durchgeführt werden müssen. Die Methode ist zusätzlich von keinerlei bestehenden Computervision-Algorithmen oder anderer Technik abhängig. Es ist mithilfe von regulärer VR-Technologie möglich, die Personentextur zu erkennen und zu segmentieren. Letztendlich zeigen Messungen, dass es auch möglich ist, ganze POV-Verfahren auf einem einzigen Rechner ablaufen zu lassen, ohne dass Performanzseinbrüche bestehen. Die Textur wird ohne Probleme in die Unity-Szene integriert, während die eigentliche Berechnung der VR weiterhin in ~90 FPS abläuft.

Einzuordnen ist die Methode in der Nähe von Phantom Model Rendering (Kapitel 2.2.1, nach Fischer et al. 2004), da das Vorgehen an einen Mix aus Optical See-Through HMDs erinnert. Die Textur wird jedoch nicht auf einem haptischen Glas/Display dargestellt, sondern auf einer virtuellen Ebene. Weil die Realität (die Personentextur) jedoch wie bei Video See-Through Vorgehensweisen gefilmt und dann in die virtuelle Szene integriert wird, ist es naheliegend die Methode als „Virtual Video See-Through“ zu bezeichnen. Das finale Ergebnis des Compositings ist in Abbildung 3.3-23 zu sehen. Zusätzlich ist hier die Einordnung der Avatardarstellung in das Realismus-Avatar-Kontinuum abgebildet. Ein Videobeispiel ist in Anhang A zu finden. Da eine vergleichbare Darstellungsform wie im Szenen-Übersicht-Compositor-Modul genutzt wird, entsprechen sich die Einordnung der beiden Avatardarstellungen bezüglich des Grades an Realismus.

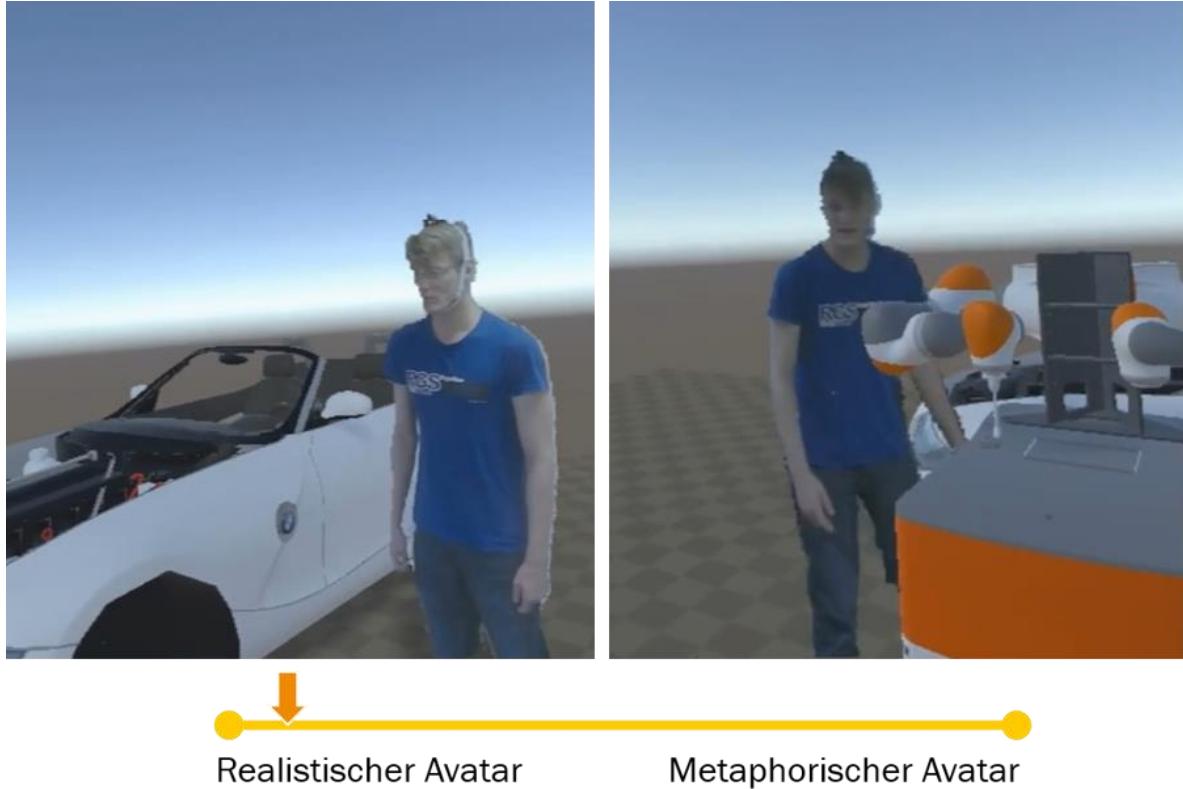


Abbildung 3.3-23: Oben links und rechts: Die finale Visualisierung eines Experimentators innerhalb des VE des Probanden. Unten: Einordnung der Avatardarstellung in das Realismus-Metapher-Avatarkontinuum.

3.3.5.1 Weitere Darstellungs- und Segmentierungsmethoden

Zur Vollständigkeit der Beschreibung des Proband-POV-Compositor-Moduls gehört auch die kurze Betrachtung von alternativen Darstellungs- und Segmentierungsmöglichkeiten. Da viel Energie in die Darstellung des Experimentators als 2D-Objekt investiert wurde, wie beispielsweise in Verdeckungsberechnung und korrekte Kalibrierung und Positionierung, könnten 3D-Darstellungen diese Probleme einfach umgehen. Auch weil die Tiefendaten der Person bereits vorliegen, könnten diese potenziell als Z-Koordinaten der einzelnen Pixel dienen. Aus diesem Grund wurde für die vorliegende Thesis auch dahingehend exploriert. Wie Abbildung 3.3-24 zeigt, sind diese Darstellungsarten mit den selben Mitteln möglich, wie sie derzeit vorgeschlagen werden. Einzig und allein zusätzliche Berechnungen innerhalb der Unity-Umgebung müssten durchgeführt werden. Zwei verschiedene Alternativen, die mit den vorhandenen Daten zu realisieren wären, sind zum einen die Darstellung der Person als Punktwolke (Abbildung 3.3-24 oben links) oder auch als vereintes Mesh (Abbildung 3.3-24 unten rechts).

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

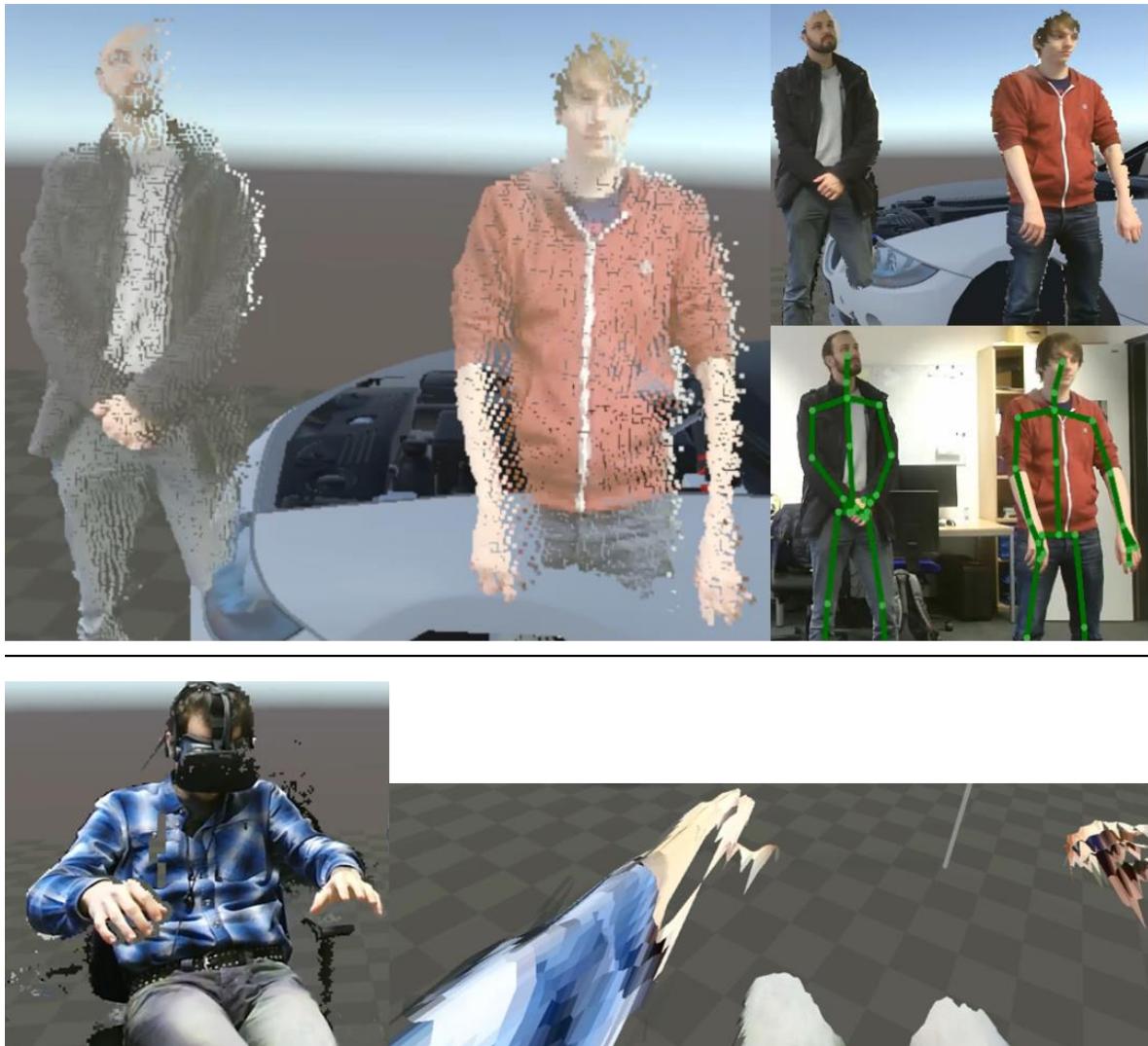


Abbildung 3.3-24: Weitere getestete Darstellungsmöglichkeiten für realistische Menschrepräsentationen in VR. *Oben:* Punktfolke (Billboard-Effekt) (links), Referenzbild (rechts oben) und Skelettschätzung der Kinect API (rechts unten). *Unten:* Referenzbild (links) und Meshrepräsentation eines Selbstavatars aus dem POV eines Nutzers (rechts).

Obwohl beide Darstellungsberechnungen relativ einfach zu implementieren sind, besonders da die Koordinatensysteme von Szene und Kinect-Kamera bereits angeglichen wurde, waren beide Verfahren nicht akzeptabel für die Verwendung in VR. Die Punktwolkendarstellung wurde auf zwei verschiedene Arten getestet. Zum einen wurde versucht um jeden Bildpunkt der Textur eine Würfelform an der Position der zugehöriger Tiefe zu rendern. Dadurch ist jeder Pixel als Würfel dargestellt und räumlich entsprechend der Tiefe versetzt. Diese Berechnungen des Verfahrens sind jedoch deutlich zu rechenintensiv, als dass man es zusammen mit VR betreiben könnte. Bei einem weiteren Test wurden die Bildpunkte lediglich durch planare Quads ersetzt (Abbildung 3.3-24 oben links, Billboard.Effekt). Außerdem wurde die Anzahl der Pixel der Textur durch Downsampling drastisch reduziert. Insgesamt war diese Darstellung immer noch zu unperformant. Die zweite Methode verfolgte das Ziel aus den räumlich zusammenhängenden Punkten ein zusammenhängendes Low-

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

Poly-Mesh zu generieren (Abbildung 3.3-24 unten rechts). Auch hier kam es zu hohen Performanzseinbrüchen. Bewegtbildbeispiele dieser Alterantiven sind in Anhang A angefügt.

Weitere Möglichkeiten die Textur in die VR-Szene zu bringen, wurden nur theoretisch behandelt, wie beispielsweise die Projektion der 2D-Textur mithilfe eines „Projector“-Objekts in Unity. Außerdem wurden erste Versuche mit skelettalen Animationen (Abbildung 3.3-24 oben) durchgeführt. Durch ein exaktes Skelett könnte beispielsweise eine konvexe Hülle erstellt werden, die wiederum genutzt würde, um die menschliche Textur kostengünstig zu extrahieren und diese dann auf ein vereinheitlichtes Menschmodell-Mesh abzubilden. Dafür kämen Anwendungen wie OpenPose infrage, die jedoch eine separate Grafikkarte und CPU allein für die Skelettberechnung benötigen. Die meisten Darstellungsverfahren scheiden damit aus Gründen der Performanzseinbußen als Hauptgrund aus.

Zusätzlich zur Darstellung wäre es auch möglich, die Segmentierung und Extraktion von Personentexturen auszutauschen. Alternativen wären hier die in Kapitel 2.5 bereits vorgestellten Methoden. Prototypisch getestet wurden HOG (mit OpenCV, Anhang A), Haar-Merkmal-basierter Cascade Classifier (mit OpenCV, Anhang A) sowie Part Affinity Fields und Convolutional Pose Machines (mit OpenPose). Diese Techniken waren jedoch entweder ebenfalls zu unperformant, als dass es sich in VR-Technik integrieren ließe (HOG, Part Affinity Fields, Convolutional Pose Machines), oder zu ungenau und müssten je nach Situation manuell programmatisch neu justiert werden (Cascade Classifier). Verfahren wie GrabCut, Deep Randomized Decision Forests und gängige Hintergrundsegmentierungen können auf Grund mangelnder Interaktionsmöglichkeit, Echtzeitabhängigkeit oder Kamerabewegungen nicht verwendet werden.

Die letztendlich verwendete Erkennung, Segmentierung und Darstellung der Experimentatoren-Textur basiert vor allem auf Verfahren wie GrabCut und Adaptive Mixture Background Segmentation, da letztendlich kein statistisches Menschmodell verwendet wird, sondern eine reine Hintergrund-Vordergrund-Segmentierung auf Grund von „a priori“-Wissen stattfindet. Diese Verfahren wurden durch VR-nahe Technik, wie Tracking und Tiefeninformationen erweitert, so dass eine performante Echtzeit-Nutzung in POV-Perspektive innerhalb eines VE möglich ist.

3.3.6 Nutzerverhältnis in Mehrbenutzerumgebungen

Im Zuge der vorangegangenen beschriebenen Implementierung des Prototyps wurde an vielen Stellen der Bezug zu Mehrbenutzerumgebungen deutlich. Dieser Bezug wurde auch im andauernden Konzeptionsprozess immer wieder thematisiert. In Anhang E sind diverse projektbezogene Aspekte dargestellt und zu verschiedenen Aufgabenbereichen zusammengefasst. Diese Grafik zeigt mögliche Permutationen für die Konzeption von Mehrbenutzerumgebungen, die auf Basis bereits vorhandener Technik und Fokus der Thesis in Betracht zu ziehen sind.

Es fällt besonders auf, dass der Begriff des „Nutzers“ in dieser Thesis vor allem für den Probanden genutzt wird. Im bestimmten Kontext trifft er jedoch auch für den Experimentator zu. Dieser ist zwar Teil der realen Umgebung, wird jedoch implizit auch Nutzer/Stakeholder des VE, da er auf dem Bildschirm ebenfalls Informationen zum VE, sowie Positionierung des Probanden und sich selbst bekommt. Obwohl der Titel der Thesis suggeriert, dass der größte Unterschied von Proband und Experimentator in der Trennung von realer Umgebung und virtueller Umgebung besteht, kann der eigentliche Unterschied der beiden Nutzergruppen durchaus bezüglich der Immersion ausgedrückt werden: Während der

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

Proband durch das HMD in immersiver VR eingebunden ist, ist die Immersion des Experimentators zwar deutlich geringer, aber dennoch unweigerlich vorhanden. Sie ist nur entsprechend gering, da dieser nur über einen einfachen, statischen Monitor Einblicke in die Szene bekommt. Das asymmetrische Verhältnis der Immersion kann vereinfacht wie in Abbildung 3.3-25 dargestellt werden.

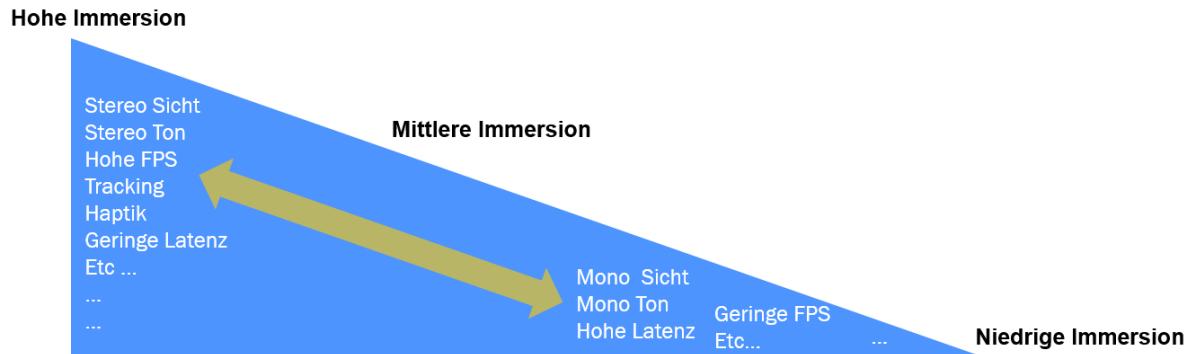


Abbildung 3.3-25: Beispielhafte Skala für die Unterscheidung zwischen verschiedenen Nutzern anhand des Immersionsverhältnisses. Die Grenze zwischen verschiedenen Immersionsgraden ist fließend.

3.4 Anwendungsfall

Der konkrete inhaltliche Anwendungsfall, der in der erarbeiteten Unity-Szene genutzt wird, ist dem Projekt „*Hybr-iT*“ zuzuordnen (*Hybride und intelligente Mensch-Roboter-Kollaboration – Hybride Teams in wandlungsfähigen, cyber-physischen Produktionsumgebungen*⁵⁶). Eine teilweise vorhandene Szene wurde hier von der ursprünglichen Simulation zu einer VR-fähigen Umgebung verarbeitet. Einige Probleme, wie beispielsweise die fehlende Interaktionsschnittstelle oder unoptimierte Modelle, wurden bereits in vorherigen Kapiteln behandelt (Kapitel 3.3.2).

Ziel des Vorhabens ist „der Aufbau und die industrielle Erprobung von hybriden Teams“⁵⁷. Diese hybriden Teams können aus unterschiedlichsten Teilsystemen bestehen, sodass neben Menschen und Robotern unter anderem auch VEs ein Teil des Teams sein können. Ein relevanter Aspekt, der im Anwendungsfall der Thesis genutzt wird, ist die Kommunikation zwischen Menschen, Robotern und Menschen in VE unter Anwendung im Bereich Industrie 4.0. Konkret beschreibt die Umgebung ein Szenario der Automobilindustrie, wobei Mensch und Roboter kollaborativ ein Getriebelager in eine Karosserie einbauen sollen. Die 3D-Szene besteht aus den folgenden Hauptkomponenten (Abbildung 3.4-1):

- *Getriebelager* – Das Getriebelager muss an die passende Stelle in der Karosserie gelegt und festgeschraubt werden. Vor dem eigentlichen Festschrauben müssen die Schrauben angeschraubt werden.
- *Karosserie* – Die Karosserie befindet sich in einem teilzusammengesetzten Zustand.
- *Roboter* – Der Roboter soll dem Werker auf verschiedene Weisen unterstützend zur Seite stehen.

⁵⁶ <http://hybr-it-projekt.de/> (8/03/2018)

⁵⁷ https://www.dfgi.de/web/forschung/asr/projekte/base_view?pid=1008 (8/03/2018)

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

- *Werker* – Während der Werker in der Simulations-Szene durch einen simulierten Avatar dargestellt wurde, wird diese Position in der VR-Szene von einem Menschen übernommen, der mit allen anderen Komponenten interagieren kann.



Abbildung 3.4-1: Links: Getriebelagermodell, Mitte: Karosseriemodell, rechts: Robotermodell.

Um die Kollaboration aufzuzeigen oder vorab zu simulieren, wurde die vorherige Simulationsszene auf den dargestellten Ablauf des Vorgangs in Abbildung 3.4-2 beschränkt. Vergrößerte Versionen der Abläufe sind in Anhang H zu finden. Neben verschiedenen parallelen Abläufen holt der Roboter das Getriebelager, setzt die Schrauben an und setzt es in die Karosserie. Der Werker dreht lediglich die Schrauben an. Danach kann der Roboter die Schrauben selbst auf das Drehmoment anziehen.

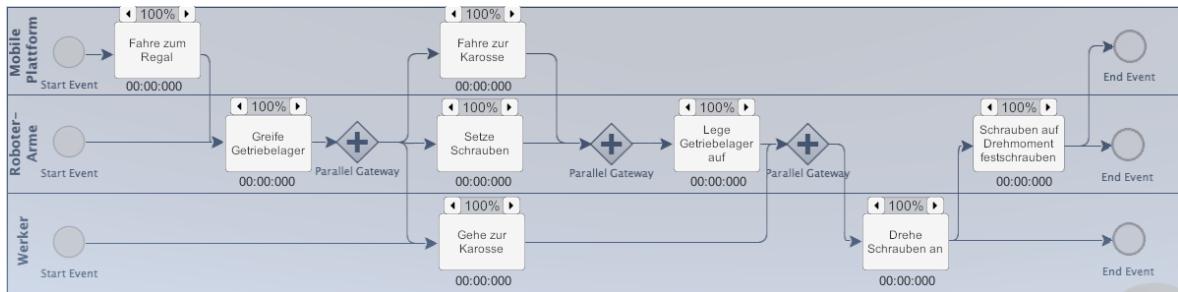


Abbildung 3.4-2: Darstellung des ursprünglichen Ablaufes der Simulationsszene.

Die Interaktion in der überarbeiteten VR-Szene der Thesis ist jedoch äußerst wichtig, da der Werker durch einen Menschen ersetzt wurde. Der Ablauf wurde daher so angepasst, dass der Werker möglichst unabhängig von den anderen Systemen agieren kann (Abbildung 3.4-3, stark vereinfacht). Außerdem wurde der nunmehr freie Ablauf der Kollaboration in Anhang G ausführlich modelliert, da diese durch die größere Autonomie des Werkers recht komplex ist.

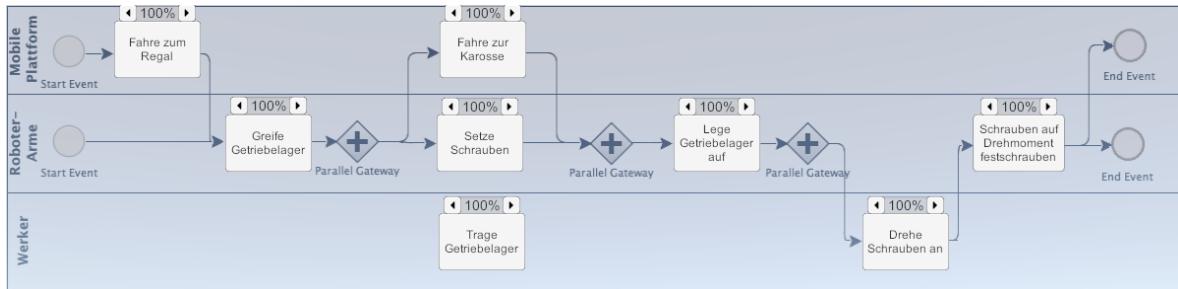


Abbildung 3.4-3: Vereinfachte Darstellung des für VR angepassten Ablaufes.

Die Steuerung des Roboters wird vorerst vom Werker selbst übernommen, sodass auch ein entsprechendes Interface erstellt werden musste. Dieses wurde jedoch nicht ausführlich gestaltet und evaluiert, da zum einen der Roboter in zukünftigen Projekten letztendlich durch

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

künstliche Intelligenz und echte Sensorsimulation autonom agieren soll, und zum anderen der Prototyp durch das Interface des Experimentators in Unity auch als Wizard of Oz (nach Kelley, 1983) Prototyp genutzt werden kann. Die Grundfunktionalität wurde jedoch trotzdem implementiert, sodass dem Werker jegliche Möglichkeiten offen liegen. Der Wizard of Oz Ansatz wurde vor allem in den kleineren Iterationen der scrum-ähnlichen, iterativen Anforderungsanalyse innerhalb des DFKI genutzt.

Auf technischer Seite bleibt die Synchronisation der Hauptkomponenten zu beschreiben. In vorherigen Kapiteln wurde die Annahme verfolgt, dass vorerst nicht mehr als zwanzig Objekte synchronisiert werden müssen, wobei ein Objekt hier als Paar von Rotation und Position definiert ist. Während das Getriebelager somit ein einziges Objekt darstellt, ist der Roboter eine Kombination von insgesamt acht Objekten, da die verschiedenen Achsen der Roboterarme separat rotiert werden müssen, aber Ihre Position immer relativ zum Rumpf des Roboters ist. Des Weiteren müssen dann nur noch die Position des Workers und des Experimentators übertragen werden. Damit ergeben sich die folgenden achtzehn Objekte, von denen jedoch nicht alle Positionen übertragen werden müssen:

- Roboter-Rumpf (Position+Rotation)
- Arm-Achsen, links und rechts jeweils Achsen Nummern: 0,1,2,3,4,5,6 (jeweils nur Rotation)
- Getriebelager (Position+Rotation)
- Trackerposition des Workers (Position+Rotation)
- Trackerposition des Experimentators (Position+Rotation)

$$\Rightarrow 18 * \text{Rotation} + 4 * \text{Position} = 22 * 3 \text{ Floats}(xyz) = 66 \text{ Floats}$$

Im Anwendungsfall dieser Thesis müssen also für einen Synchronisationsabschnitt lediglich 66 Float übertragen werden. Damit ergeben sich etwa $66 \text{ Floats} * 30 \text{ FPS} = 1980 \frac{\text{Floats}}{\text{s}} = 7920 \frac{\text{byte}}{\text{s}}$. Diese können ohne Probleme über das Netzwerk gesendet werden.

3.5 Evaluation des vorgeschlagenen Konzepts mithilfe des Prototyps

Zur Evaluation wurde ein Experiment durchgeführt, welches die folgende Hypothese beurteilt:

Falls eine Kommunikation von einer Person A aus ihrer immersiven virtuellen Umgebung mit einer Person B aus der realen Umgebung stattfindet, wird die empfundene Präsenz von Person A durch die Erweiterung der visuellen Kommunikation, mittels realistischer Einblendung von Person B in der virtuellen Umgebung, nicht negativ beeinträchtigt.

Das Experiment wird im Rahmen der Master-Thesis als kleinere Pilotstudie durchgeführt, um so mögliche Fehler und Schwierigkeiten für eine eventuelle, ausführliche Evaluation in größerem Rahmen herauszufinden. Diese ausführliche Evaluation wird jedoch nicht Teil dieser Thesis sein. Mittels der Pilotstudie kann dennoch eine erste Aussage über das vorgeschlagene Konzept getroffen werden, die einen Trend wiederspiegelt. Insgesamt hätten viele Experimente durchgeführt werden können, um diverse andere Hypothesen und Variablen zu evaluieren. Es wurde sich jedoch aus mehreren Gründen für diese entschieden:

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

1. Aus Zeit- und Kostengründen kann nicht jeder Aspekt der Thesis mittels Experiment bestätigt werden.
2. Es wird nur die Kommunikationsrichtung von Proband zu Experimentator evaluiert, da dieses Konzept ein Novum im VR-Bereich darstellt.
3. Es wird explizit keine statistische Aussage über die eigentliche Kommunikation zwischen den Parteien getroffen. Der Mehrwert der gegenseitigen realistischen Visualisierung zur Erweiterung der visuellen Kommunikation wird – gestützt durch Literaturrecherche – als trivialer Zusammenhang angesehen. Etwa 55% der zwischenmenschlichen Kommunikation wird über visuelle Aspekte wie Körpersprache transportiert. Diese sind bisher nicht in VEs vorhanden.
4. Um zusätzlich auch den praktischen Nutzen zu gewährleisten, muss gezeigt werden, dass die vorgeschlagene Kommunikationsmethode die Präsenz des Probanden nicht beeinträchtigt. Möglicher negativer Einfluss der Visualisierung könnte sonst den positiven Einfluss nicht nur negieren, sondern insgesamt den eigentlichen Nutzen der Anwendung einer VR mindern.

Wenn also durch ein Experiment keine statistische Signifikanz aufgezeigt werden kann, die die Präsenz einer Experimentalgruppe und die einer Kontrollgruppe unterscheidet, so ist sowohl die Verbesserung der Kommunikation, als auch der praktische Nutzen in VEs belegt. Im Idealfall würde die Präsenz sogar verbessert werden. Dies liegt im Bereich des Möglichen, weil zum Zeitpunkt, an dem eine Kommunikation von VE zu RE nötig wird, die Präsenz ohnehin schon beeinträchtigt wird. Diesem negativen Effekt könnte durch die explizite Integration einer Visualisierung außenstehender Nutzer in VR entgegengewirkt werden.

3.5.1 Teilnehmer

Zwölf bezahlte, freiwillige Teilnehmer (5 weiblich, 7 männlich) aus der campusnahen Umgebung der lokalen Universität wurden für das Experiment geworben. Die Anzahl wurde aufgrund von vergleichbaren Pilotstudien bestimmt. Die Altersspanne reichte von 23 bis 35 Jahren mit einem Mittel (arithmetisch) von 26,5 Jahren und einer Standardabweichung (SD) von 3,34 Jahren (gerundet). Zwei Teilnehmer hatten sehr wenig, alle anderen gar keine Vorerfahrung im Bereich VR, womit alle als unerfahrene Nutzer eingestuft wurden. Des Weiteren war keiner der Teilnehmer professionell (Studium/Beruf) mit den Bereichen Informatik oder Psychologie vorbelastet.

3.5.2 Technik

Die genutzte Technik beläuft sich auf die in vorangegangenen Kapiteln bereits genannte. Ein HTC-Vive-System mit drei zusätzlichen Trackern wurde betrieben von einem Windows PC (Intel Core i7 Prozessor mit 3,5 GHz, 32Gb RAM, AMD Radeon R9 Fury Grafikkarte). Auf diesem PC wurden zusätzlich alle Module ausgeführt, die zum VR-PC-System gehören (nach Kapitel 3.3.1). Auf einem zweiten PC (Intel Core i7 Prozessor mit 3,5 GHz, 16Gb RAM, NVIDIA GTX 1070 Grafikkarte) wurden alle Komponenten des Non-VR-Systems betrieben.

Insgesamt wurde eine 12 qm große Fläche kalibriert, sodass diese als „Play Area“ der Vive genutzt werden konnte. Diese Fläche war sowohl der Bereich, in dem sich der Proband innerhalb des VE bewegen konnte, als auch der Bereich in dem der Experimentator getrackt und dargestellt wurde.

Als Bild- und Tiefensensoren wurden, wie bereits beschrieben, Kinect-Kameras genutzt. Die Kamera die für den POV zuständig war, wurde mithilfe des Kinect-Rigs aus Abbildung 3.3-20 an, dem jeweiligen Teilnehmer befestigt.

3.5.3 Ablauf

Nachdem der Teilnehmer durch den (immer gleichen) Experimentator begrüßt wurde, wurde er gebeten das Entry-Questionnaire (Anhang F) an einem dafür vorbereiteten Tisch auszufüllen. Danach wurde er etwa fünf Minuten in die von ihm zu benutzende Technik eingewiesen, wobei keinerlei inhaltliche Informationen zum Anwendungsfall der Szene gegeben wurden. Außerdem wurden Sicherheitshinweise in Kürze geklärt. Danach wurde das Experiment begonnen, bei dem das VE aus Kapitel 3.4 verwendet worden ist. Dabei wurde dem Teilnehmer nur wenig erklärt was genau gemacht werden muss. Ohne weitere inhaltliche Einführung wurde er gebeten zu starten. Mit solchen und weiteren Impulsen wurde die Notwendigkeit von gegenseitiger visueller Kommunikation zwischen virtueller und realer Umgebung hergestellt. Der Teilnehmer wurde so auf implizite Weise zu diversen Rückfragen ermutigt.

Unabhängig vom eigentlichen Stand der Durchführung des Anwendungsfalls wurde der praktische Teil des Experiments nach etwa 12 Minuten beendet. Danach wurde dem Teilnehmer die Technik wieder abgelegt und er wurde gebeten das Post-Study-Questionnaire (Anhang F) an dem dafür vorbereiteten Tisch auszufüllen. Danach wurde dem Teilnehmer für die Teilnahme gedankt und ihm wurde ein Schokoriegel sowie der geringfügige Lohn überreicht. Danach wurde er aus dem Raum geführt und verabschiedet.

3.5.4 Design des Experiments

Die Teilnehmer wurden zufällig in eine Kontroll- und eine Experimentalgruppe eingeteilt. Die Kontrollgruppe absolvierte das Experiment ohne die Nutzung der Einblendung des Experimentators, die Experimentalgruppe nutzt diese Technik und das dazugehörige Rig. Damit wurde ein Between-Subject-Design verwendet, da jede Gruppe nur auf eine einzige Testkondition/Verhalten getestet wurde. Als Umgebung für das Experiment wurde eine kontrollierte Labor-Umgebung verwendet. Diese Umgebung hat bei dem Experiment jedoch keinen negativen Einfluss auf die externe Validität, die man für gewöhnlich in höherem Maße durch eine natürlichere Umgebung/ein Feldexperiment („in the wild“) erhält. In diesem Spezialfall ist die kontrollierte Laborsituation gleichzeitig auch die natürliche Situation, da das Experiment auf die Meta-Ebene – die Kommunikation zwischen realer und virtueller Umgebung mit Anwendung in Experimenten – abzielt.

Das Experiment behandelte genau eine unabhängige Variable, die Visualisierung von dem Experimentator innerhalb eines VE für den Probanden. Gemessen wurde genau eine abhängige Variable: die Präsenz des Teilnehmers. Das Experiment wurde nach zwei Testkonditionen unterteilt: Der Ablauf wurde entweder ohne die Visualisierung oder aber mit Visualisierung des Experimentators durchgeführt. Während das Entry-Questionnaire nur auf demografische Informationen abzielte, wurde das Post-Study-Questionnaire genutzt, um die Präsenz zu messen. Die Präsenz ergibt sich, wie von Schubert, Friedmann, und Regenbrecht (2001) vorgeschlagen, durch die Einheiten:

- Spatial Presence,
- Involvement
- und Experienced Realism.

Das von ihnen entwickelte igroup Presence Questionnaire⁵⁸ (IPQ) zielt genau auf diese Einheiten der Präsenz ab und besteht unter anderem bereits aus etablierten Fragen von Vorgängerfragebögen (Kapitel 2.7.2). Die Fragen aus dem IPQ wurden in das verwendete

⁵⁸ <http://www.igroup.org/pq/ipq/index.php> (8/03/2018)

DIE ERWEITERUNG DER VISUELLEN KOMMUNIKATION – EIN PROTOTYP

Post-Study-Questionnaire eingearbeitet (Anhang F). Alle weiteren Variablen waren entweder Kontrollvariablen und somit für alle Teilnehmer gleich und ohne Einfluss, oder Störfaktoren, wie mögliche Vorerfahrungen von Probanden. Es wurde jedoch bereits beschrieben, dass keine Vorerfahrung der Teilnehmer vorhanden war. Zusätzlich würde die Auswirkung durch die randomisierte Zuweisung in die Gruppen geschwächt werden. Ein fester Faktor war der Inhalt des VE, der genau dem Anwendungsfall der Thesis entsprach.

4. ANALYSE DER ERGEBNISSE

In der vorliegenden Master-Thesis wurde nach ausführlicher Literaturanalyse des Stands der Wissenschaft und Technik ein Konzept vorgeschlagen, das die Kommunikation zwischen virtueller und realer Umgebung erweitert. Als Nebenprodukt dieser Literaturanalyse wurde das Realismus-Metapher-Avatarkontinuum vorgeschlagen, um die behandelten Darstellungsformen mit denen anderer Arbeiten leichter vergleichen zu können.

Unter Anwendung im Kontext von Experimenten ist es dem Experimentator mit dem Konzept möglich, den Probanden und sich selbst räumlich innerhalb der virtuellen Szene zu sehen. Dem Probanden, der in einem immersiven VE ist, ist es ebenfalls möglich, mit dem Experimentator visuell zu kommunizieren, da der Experimentator mit räumlichem Bezug innerhalb der Szene visualisiert wird.

Dieses Konzept wurde unter Verwendung eines modularen und verteilten Systementwurfs prototypisch umgesetzt. Im Zuge der Implementierung wurde eine neuartige Methode zur Erkennung, Segmentierung und Darstellung für außenstehende Personen in immersiven VEs vorgeschlagen. Diese Darstellung wird aus der POV-Perspektive eines Nutzers des immersiven VEs visualisiert. Dabei werden ausschließlich prosumentenorientierte, VR-nahe Techniken und Technologien verwendet, die ohne stärkere Abhängigkeit an bestehende Computervision-Algorithmen auskommen. Während der Modellierung des entsprechenden Systementwurfs wurde außerdem aufgezeigt, welche technischen Schwierigkeiten, Abhängigkeiten und Bedingungen an eine Integration von Echtzeit-Realbild-Daten in ein VR System bestehen.

Anhand des implementierten Prototyps wurde gemessen, dass weiterhin ein VR-Rendering von ~90 FPS stattfindet, während die Experimentatorentextur mit 30 FPS in der Szene aktualisiert wird. Auch weitere Parameter für den technischen Aspekt der Immersion werden nicht beeinflusst, da beispielsweise FOV, FOR, Stereoskopie, Displayauflösung und vieles mehr nicht verändert werden. Dem Experimentator werden die Szene und die Avatardarstellungen in 30 FPS auf einem einfachen Bildschirm angezeigt.

Mithilfe dieses Prototyps wurde eine Evaluation anhand eines Experiments durchgeführt. Die Evaluation fokussierte den Einfluss der Visualisierung des Experimentators in VR auf die Präsenz des Probanden.

4.1 Analyse der Evaluation

Die Analyse der Evaluation zielt auf die Präsenz der Probanden im beschriebenen Experimentaufbau ab. Diese Analyse wurde mit statistischen Methoden – mittels eines Zweistichproben-t-Tests – durchgeführt. Der t-Test wurde auf die Ergebnisse der beiden Populationen (Anhang I) angewandt und bezüglich der Hypothese anhand des Mittelwertunterschieds der Präsenz wie folgt analysiert:

ANALYSE DER ERGEBNISSE

	<i>PRES-Kontrollgruppe</i>	<i>PRES-Experimentalgruppe</i>
Mittelwert	3,4212963	3,5462963
Varianz	0,11586934	0,068107
Beobachtungen	6	6
Pearson Korrelation	-0,53620961	
Hypothetische Differenz der Mittelwerte	0	
Freiheitsgrade (df)	5	
t-Statistik	-0,57942049	
P(T<=t)	0,58743025	
Kritischer t-Wert	2,57058184	

Tabelle 4.1-1: Analyse des Mittelwertunterschieds der beiden Populationen bezüglich der Präsenz.

Die Werte für den t-Test sind in Anhang I zu finden und wurden anhand der dort beschriebenen Formeln errechnet. Diese wurden wiederum aus den Quellen des IPQ entnommen. Des Weiteren zeigt Tabelle 4.1-2, dass die Werte der Experimentalgruppe bei den Kategorien Spatial Presence und Involvement leicht höher als die der Kontrollgruppe sind. Diese sind jedoch statistisch nicht signifikant höher. Da die Werte bezüglich Experienced Realism gleich hoch sind, ist folglich auch die Overall Presence der Experimentalgruppe unsignifikant höher. Sonstige statistische Auffälligkeiten sind nicht festzustellen, was auch die Kastengrafiken in Abbildung 4.1-1 verdeutlichen.

	<i>Experimentalgruppe</i>	<i>Kontrollgruppe</i>
<i>Spatial Presence</i>	4,30555556	4,2222222
<i>Involvement</i>	3,16666667	2,875
<i>Experienced Realism</i>	3,16666667	3,1666667
<i>Overall Presence</i>	3,5462963	3,4212963

Tabelle 4.1-2: Gegenüberstellung der finalen Ergebnisse der zwei Populationen: Experimentalgruppe und Kontrollgruppe.

ANALYSE DER ERGEBNISSE

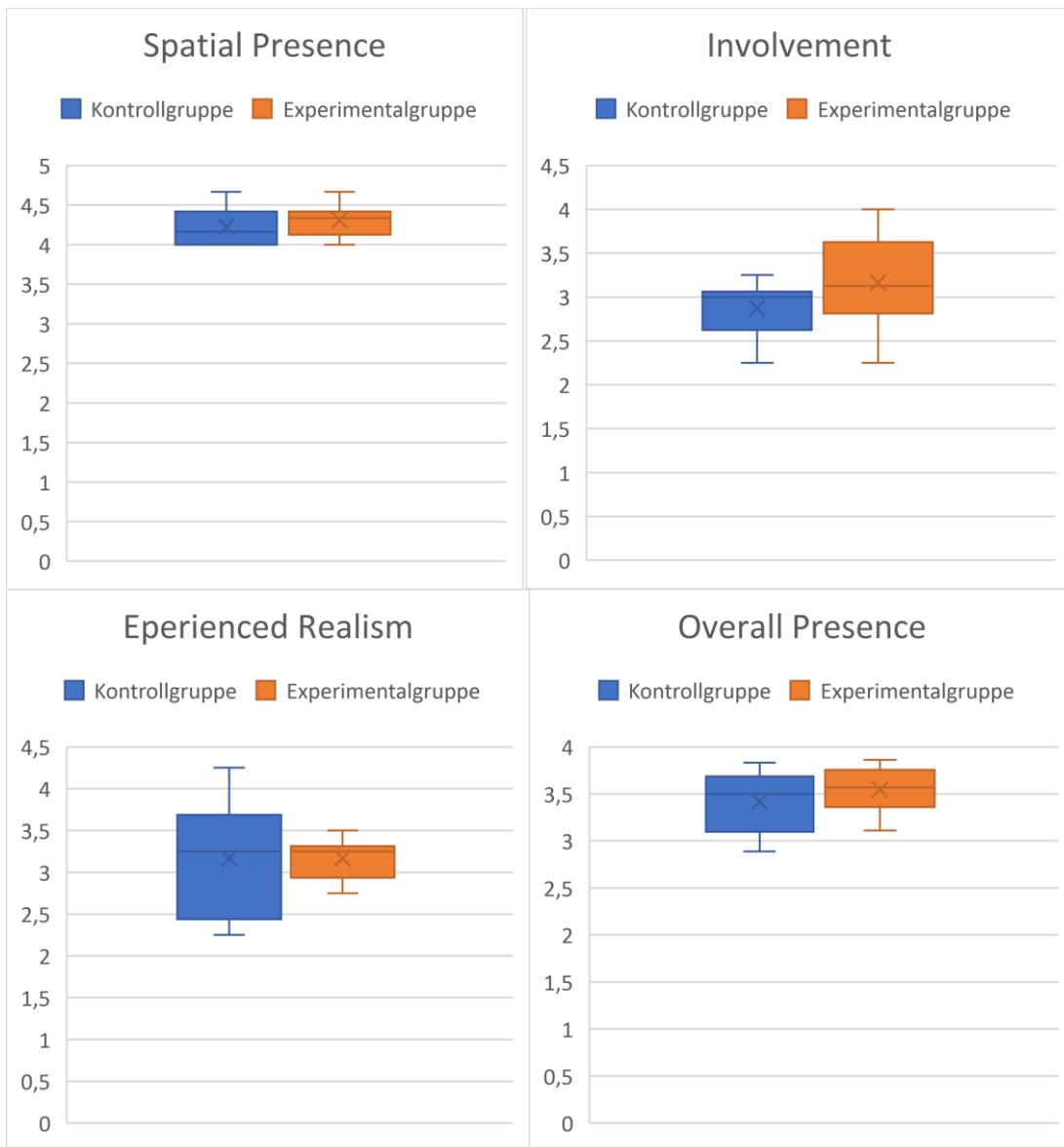


Abbildung 4.1-1: Kastengrafiken der einzelnen Unterkategorien SP, EXP, INV und PRES.

Sowohl die Freitextkommentare des Post-Study-Questionnaires als auch aufgenommene Äußerungen während des Experiments wurden zusätzlich verwendet, um eine Aussage über die Usability des Prototyps treffen zu können. Diese wurden zuerst einzeln gesammelt und dann nach Ähnlichkeit der angesprochenen Thematik gruppiert. Zur analytischen Anordnung wurden, als gängige Praxis im Bereich Usability, Haftzettel genutzt. Die Teilprobleme wurden dann in ein Raster von Teilmengen bezüglich {Art des Problems x Ausmaß des Problems} eingeordnet. Die Art des Problems wurde in Benutzerfehler, Verbesserungsideen, Erfolge und unerwarteten Beobachtungen unterteilt. Das Ausmaß des Problems wurde in Priorität 1, Major und Minor eingeordnet.

5. DISKUSSION DER ERGEBNISSE

Die Ziele, die für die vorliegende Master-Thesis gesetzt wurden, sind in Tabelle 1.2-1 beschrieben. Anhand dieser Ziele werden die Ergebnisse bewertet.

1. Das erste Ziel war die Erstellung eines Systems zur Verbesserung der gegenseitigen Wahrnehmung und Kommunikation zwischen Nutzern von VEs. Während der Erarbeitung des Thesisinhalts wurde auch klar, dass selbst Außenstehende, wie ein Experimentator durchaus Nutzer eines VE sind – mit entsprechend niedrigerer Immersion. Da ein System vorgeschlagen wurde (sowohl konzeptionell als auch in prototypischer Umsetzung), welches die gegenseitige Kommunikation und Wahrnehmung verbessert und dieses hinsichtlich technischer Machbarkeit evaluiert wurde, wird das erste Ziel als erfüllt betrachtet.
2. Im Zuge des zweiten Ziels sollte insbesondere die visuelle Kommunikation zwischen realer und virtueller Umgebung gefördert werden. Auch dies wurde hinsichtlich des Prototyps erfüllt. Gut ersichtlich wird dieser Aspekt im Video aus Anhang A. Zusätzlich wird bei dem vorgestellten System eine gegenseitige Visualisierung umgesetzt, die bezüglich des Realismus-Metapher-Avatarkontinuums mit aktuellsten Arbeiten zu realistischen Avataren durchaus vergleichbar ist.
3. Anlässlich der Implementierung des Prototyps unter ausschließlicher Nutzung von semi-professioneller Technologie, ist auch das dritte Ziel erfüllt. Besonders Aspekte wie das DIY-Rig für die POV-Kinect und die Verwendung von Kinect-Kameras selbst sind klare Anzeichen dafür. Auch die vorgeschlagene Erkennungs-/Segmentierungsmethode beruht ausschließlich auf kostengünstiger prosumentenorientierter Technik, sowohl was die Hardwareanforderungen, als auch die Wahl der Hardware selbst betrifft.
4. Die ausführliche Modellierung aller Systemkomponenten in Kapitel 3.3 macht den modularen Charakter des Systems klar. Die funktionale Kapselung in abstrakte Komponenten, die in der Architekturbeschreibung nach FMC herausgestellt wird, verdeutlicht dies. Außerdem werden Schnittstellen an das System aufgezeigt, sodass es möglich ist einzelne Komponenten durchaus in einfacher Weise auszutauschen. Dadurch wird das System, aber auch einzelne Sub-Systeme, wiederverwendbar in weiteren Projekten. Auch der Datenfluss wird in der umfassenden Modellierung des Gesamtsystems klar dargestellt. Somit ist auch Ziel Nummer vier als erfüllt anzusehen.
5. Das fünfte Ziel wird formal auch bereits als erreicht angesehen. Als Maß dafür wurde angesetzt, dass der Proband die virtuelle Szene nicht verlassen muss während er kommuniziert. Dies ist durch das Video in Anhang A schon belegt. Darüber hinaus wurde eine Pilotstudie mithilfe eines Experiments durchgeführt, um den Aspekt der Präsenz noch genauer zu untersuchen.

5.1 Bewertung der Evaluation

Um besonders das fünfte Ziel der Thesis genauer zu beleuchten, wurde im kleineren Rahmen ein Experiment als eine Pilotstudie durchgeführt, welches auf die Messung der Präsenz abzielt. Die statistische Analyse des Experiments zeigt einen klaren positiven Einfluss des vorgeschlagenen Kommunikationskonzepts.

Entscheidend für die Bewertung der statistischen Analyse ist der P-Wert von 0,58743025 und der t-Wert von -0,57942049. Diese sagen auf die Präsenz bezogen Folgendes aus (besonders, da $P>0,05$ ist):

Es gibt keinen statistisch signifikanten Unterschied der Präsenz zwischen den beiden Populationen, womit der Effekt der Visualisierung des Experimentators innerhalb des VE auf die Präsenz des Probanden ebenfalls nicht statistisch signifikant ist.

Auch der absolute Vergleich der Präsenzmessung mithilfe des IPQ zeigt, dass die Präsenz innerhalb von dem durchgeführten Experiment sogar von der Erweiterung der Kommunikation profitiert hat. Auch die Sub-Kategorien, wie sie in Abbildung 4.1-1 und genauer in Anhang I dargestellt werden, zeigen keine Verschlechterung der gemessenen Ergebnisse. Der größte positive Unterschied ist in der Kategorie Involvement zu sehen. Folglich bedeutet das, dass die Aufmerksamkeit die von den Personen der Experimentalgruppe dem RE geschenkt wurde absolut geringer war als die der Kontrollgruppe. Dies spricht für den positiven Einfluss des Konzepts auf diese Kategorie.

Die zuletzt genannten Ergebnisse sind relativ gesehen jedoch nicht statistisch signifikant, insbesondere da auf Grund der Durchführung als Pilotstudie auch nur eine geringe Anzahl von zwölf Personen miteinbezogen werden konnte. Somit gilt, dass eine signifikante Trendaussage getroffen werden konnte, die aufzeigt, dass es durchaus lohnenswert sein könnte, weiter an dem vorgestellten Konzept und der gegenseitigen Kommunikation zu arbeiten und den Prototyp weiter auszubauen. Letztendlich müsste dann eine weitreichendere Evaluation durchgeführt werden, die weitere Variablen, wie die Darstellung des Probanden in seinem VE für den Experimentator, hinzuzieht.

Des Weiteren bleibt die Konzeption des Experiments kritisch zu bewerten. Im Maße einer Pilotstudie ist sowohl die interne, als auch externe Validität als hoch einzustufen. Die behandelte Fragestellung ist sehr zielgerichtet und die Studienteilnehmer kommen aus verschiedenen Bereichen der Zielgruppe, da Experimentteilnehmer für gewöhnlich als Nicht-Experten angesehen werden. Dies konnte durch die relevanten demographischen Informationen über die Teilnehmer bestätigt werden. Zusätzlich wurden die Teilnehmer randomisiert in die beiden Populationen eingeordnet.

Ein weiterer großer Vorteil, der für hohe interne sowie externe Validität spricht, ist das eigentliche Anwendungsgebiet der Studie. Da das Anwendungsgebiet selbst im Bereich einer Experimentsituation stattfindet, ist die gewählte Laborsituation gleichzeitig auch die natürliche Umgebung. Dadurch wird zum einen gewährleistet, dass nur wenige unkontrollierte externe Variablen miteingeflossen sind, weil die Umgebung natürlich war, und zum anderen, dass trotzdem alle Vorteile eines Laborexperiments genutzt werden konnten (beispielsweise eine kontrollierte Vorbereitung des Experiments). Der klassische Trade-Off zwischen interner und externer Validität, den vergleichbare Experimente für gewöhnlich haben, konnte so gut umgangen werden.

Durch den Vorteil einer Laborsituation konnten außerdem weitere Daten aufgenommen werden, die eine Aussage bezüglich der Usability des Prototyps ermöglichen. Im Bereich der Benutzerfehler konnten keine Aspekte beobachtet werden, die sich auf den Prototyp beziehen. In der Teilmenge der Verbesserungen wurden vor allem Beobachtungen und Kommentare eingeordnet, die sich auf jeweils einen von zwei Aspekten beziehen. Ein kritischer Aspekt, der daher auch in Priorität 1 eingeordnet wurde, war das DIY-Rig in Kombination mit der Kinect-Kamera, die auf dem Kopf befestigt wurden. Damit wurden etwa 2 kg auf dem Kopf des Probanden platziert. Mit der starken Äußerung von fünf der sechs Teilnehmer der Experimentalgruppe ist es der kritischste Aspekt bezüglich der

DISKUSSION DER ERGEBNISSE

Usability – was jedoch bereits vor der eigentlichen Evaluation vermutet wurde. Diese Komponente sollte trotzdem in der nächsten Iteration am stärksten ausgebaut werden. Der zweite Aspekt, der als Verbesserungsvorschlag eingeordnet wurde, hängt ebenfalls mit der Kinect zusammen. Da diese nicht den ganzen Bereich des FOV abdeckt, den der Nutzer durch das Vive HMD bekommt, wurden oft Extremitäten (vorwiegend Füße) in der Experimentatorenvisualisierung abgeschnitten. Da dies sich zum größten Teil nur auf die Füße bezieht, fiel es nicht immer auf und wurde nur von zwei der sechs Teilnehmer thematisiert. Damit wurde es eingeordnet in die Major-Sparte der Verbesserungsvorschläge. Wahrscheinlich wird sich dieses Problem jedoch mit der Ersetzung der Kinect auf Grund des kritischen Fehlers bereits von selbst lösen.

Als positiv empfunden, und damit in den Bereich der Erfolge eingeordnet, wurde besonders die generelle Darstellung des Experimentators im VE. Dieses wurde von fünf von sechs Teilnehmern als äußerst positiv und hilfreich umschrieben. Im Bereich der unerwarteten Beobachtungen wurde lediglich ein Aspekt aufgenommen, der von einem einzigen Teilnehmer geäußert wurde: Die Visualisierung des Experimentators wurde nicht zu jedem Zeitpunkt als Mehrwert angesehen. Zwar hat diese auch nicht gestört, jedoch wurde eine ein- und ausschaltbare Visualisierung als passend beschrieben. Dadurch könne im Falle der Notwendigkeit die Kommunikation dediziert erweitert werden.

Insgesamt konnte mit der Pilotstudie das Beste aus beiden Welten für die Evaluation des Kommunikationskonzepts und des Prototyps vereinigt werden: Die Bestätigung der Hypothese durch die statistische Auswertung des Experiments und das Aufzeigen erster Usability-Probleme mithilfe qualitativer Analyse.

5.2 Bewertung der Methoden

Für den Ablauf der Thesis wurde ein nutzerzentrierter Ansatz gewählt, der von verschiedenen Iterationen geprägt war. Durch die iterative, scrum-ähnliche Anforderungsanalyse und daran anschließende Evaluationen konnte das vorgestellte Konzept bereits durch Experten aber auch Nutzer validiert werden. Neben der Konzeption wurde ein Prototyp entwickelt, der in mehreren Stadien bereits genutzt wurde, um direktes Feedback zu dem vorgestellten Konzept zu erhalten. Erste Verifikationen des Systems wurden durch qualitative Analyse des durchgeföhrten Experiments durchgeführt.

Der Nutzen der gegenseitigen realistischen Darstellung für die Kommunikation zwischen realen und virtuellen Umgebungen wurde als trivial angesehen, da etwa 55% der zwischenmenschlichen Kommunikation durch visuelle Aspekte, wie Körpersprache, transportiert wird. Dieser Aspekt wurde durch Literaturanalyse belegt. Die Nützlichkeit des gesamten Systems wurde von Experten im Themengebiet von VR und Psychologie bestätigt. Das finale Experiment bestätigt, dass dieser Nutzen von Seiten der VR-Nutzer nicht durch negativen Einfluss auf deren Präsenz vermindert wird.

Kritisch zu betrachten ist die finale Evaluation bezüglich des verwendeten Anwendungsfalls. Obwohl dieser eine feste Variable war, ist es möglich, dass er sich tendenziell auf die Aussagen der Probanden ausgewirkt hat. Die Evaluation des Kommunikationskonzeptes hätte also im besten Fall in größerem Maße und mit verschiedenen inhaltlichen Szenarios stattfinden sollen, womit die externe Validität des Experiments noch gesteigert werden könnte. Diese Evaluation des Konzepts in Gänze wurde jedoch bezüglich des zeitlichen und finanziellen Rahmens der Thesis ausgeschlossen.

Insgesamt ist es, wie im Kapitel der Wissenschaft und Technik (2.7) beschrieben, auch kritisch zu hinterfragen, ob der gewählte Ansatz der Immersions- und Präsenzbeurteilung

DISKUSSION DER ERGEBNISSE

aussagekräftig ist. Während der IPQ-Fragebogen durchaus von aktuellem Charakter und in Industrie und Forschung ein oft genutztes Medium ist, so scheint die Beurteilung der Präsenz durchaus angemessen. Als Verknüpfung von Fragen aus mehreren etablierten Fragebögen ist er ein praktisches Mittel. Die Beurteilung der Immersion hingegen bleibt kritischer zu betrachten. Die technischen Aspekte, die von Bowman und McMahan (2001) vorgeschlagen wurden, lassen eine Bewertung der technischen Immersion zwar zu, jedoch wurde im Verlauf der Thesis klar, dass sie nicht vollständig sein kann. Besonders die rasant fortschreitende Entwicklung der prosumentenorientierten VR-Technik bietet viele Aspekte, die weitgehend nicht mit in die Beurteilung der Immersion einfließen Ein Beispiel dafür wäre der Maßstab des virtuellen Raums (Room-Scale, World-Scale etc.). Daher wurde die technische Immersion in der vorliegenden Thesis nur in Kürze analysiert. Weil der Kern der Arbeit jedoch auf dem vorgeschlagenen Kommunikationskonzept und dem dazu passenden Systementwurf für einen Prototyp beruht, ist die Immersion der Präsenz durchaus legitimerweise unterzuordnen. Technische Komponenten werden und müssen sich aus bereits genannten Gründen in weiteren Iterationen ohnehin noch in großem Maße ändern.

6. ZUSAMMENFASSUNG

In der vorliegenden Master-Thesis wurde ein Konzept vorgeschlagen, das die Kommunikation zwischen virtueller und realer Umgebung erweitert. Dieses Konzept wurde mithilfe eines modularen, verteilten Systementwurfs prototypisch umgesetzt und iterativ evaluiert. Dabei wurden Methoden zur Erkennung, Segmentierung und Visualisierung menschlicher Texturen in verschiedenen immersiven VEs verwendet. Zusätzlich zu bereits etablierten, wurde eine neuartige Methode vorgeschlagen, welche Personen zielgerichtet unter ausschließlicher Verwendung von semi-professioneller VR-Technik segmentiert und in immersiven VEs darstellt. Dieses Verfahren lässt sich insgesamt zu Algorithmen wie GrabCut und Adaptive Mixture Background Segmentation zuordnen, da letztendlich kein statistisches Menschmodell verwendet wird, sondern eine reine Hintergrund-Vordergrund-Segmentierung stattfindet. Eine weitere Besonderheit ist dabei die Anwendung für eine POV-Simulation eines Nutzers eines immersiven VEs. Durch die verschiedenen Immersionsabstufungen entstanden zwei Mixed-Reality-Systeme, die zu einem übergreifenden Gesamtsystem verknüpft wurden. Dadurch kann die gegenseitige visuelle Kommunikation in Echtzeit unterstützt werden.

Insgesamt wird die Immersion durch das vorgeschlagene Konzept nicht negativ beeinträchtigt. Außerdem konnte durch ein Experiment gezeigt werden, dass die Präsenz ebenfalls nicht statistisch signifikant negativ beeinflusst wird. Das Konzept erweitert also lediglich das bestehende VE. Durch die modulare Ausrichtung des Systementwurfs ist außerdem gewährleistet, dass man das Konzept in verschiedenste Systeme einbinden kann. Einzelne Module können leicht ausgetauscht werden. Es ist also sowohl inhaltlich als auch technisch relativ losgelöst. Obwohl das Konzept also eher genereller Natur ist, zeigt sich die Spezialisierung bezüglich des Grades an Realismus der Avatardarstellungen. Das Konzept ist stark fokussiert auf sehr realistische Darstellungsformen und lässt daher nur wenig „Übernatürlichkeit“ zu.

Der Grad des Realismus der gegenseitigen Repräsentation kann besonders im angefügten Video (Anhang A) nachvollzogen werden. Um den Vergleich mit anderen Arbeiten in diesem Bereich zu vereinfachen, wurde nebenläufig das Realismus-Metapher-Avatarkontinuum vorgeschlagen. Obwohl keine wirkliche 3D-Darstellung der Avatare gewählt wurde, wurden durch Nutzung der echten Textur der Nutzer und der Einbeziehung von Tiefendaten äußerst realistische Avatare erzeugt, die nicht nur optisch, sondern auch räumlich realistisch eingeordnet werden können.

Abschließend ist zu sagen, dass viele Teile des vorgeschlagenen Konzepts und der Umsetzung als relativ „einfach“ dargestellt werden. Damit genau diese Einfachheit erzielt werden konnte, mussten verschiedenste Techniken aus diversen interdisziplinären Themengebieten detailliert behandelt werden, sodass die vorliegende Master-Thesis ein durchaus breites inhaltliches Spektrum widerspiegelt. Diese sehr aufwändig erhaltene Einfachheit des modellierten Systems ist es insgesamt jedoch auch, die den Mehrwert des Kerns der Thesis reflektiert. Sie bildet die Basis für die Nutzung der Module in weitere Arbeiten, aber auch für folgende zukünftige Forschungsarbeiten, welche die Bereiche Kommunikation und Virtuelle Realität verknüpft.

6.1 Ausblick

Einige potentielle Ansätze für zukünftige Arbeiten auf der Basis der Master-Thesis wurden bereits genannt. Im Folgenden wird abschließend zwischen technischen Verbesserungen der

Umsetzung und weiteren inhaltlichen Ansätzen, die mithilfe des hier vorgestellten Konzepts möglich werden, unterschieden.

6.1.1 Technische Verbesserungen

Zu den technischen Verbesserungen zählt allen voran die Verbesserung der Personen-Maske der Kinect-API. Die Personenerkennung und -segmentierung der Kinect ist von Rauschen um die eigentliche Person herum geprägt. Die finale Maske könnte jedoch durch einfache Computergrafik- und Bildverarbeitungsalgorithmen, wie Erodieren und Dilatieren, deutlich verbessert werden. Eine andere Möglichkeit die Maske optisch zu verbessern wäre es, eine Graph-Cut Segmentierung nach Boyko und Jolly (2001) darauf anzuwenden. Diese Möglichkeiten wurden bisher nicht durchgeführt, da das Computervision-Back-End der Übersichts-Kinect letztendlich ebenfalls durch die deutlich performantere und genauere Segmentierungsmethode, die hier vorgestellt wurde, ersetzt werden könnte und sollte.

Eine weitere technische Begebenheit, die in dieser Thesis bisher gänzlich ausgelassen wurde, ist die Integration von Ton. Da das Szenario des Anwendungsfalls bisher keinen Ton unterstützt, wurde auch bei dem Kommunikationskonzept kein Ton miteingebunden. Die Personen befinden sich letztendlich im selben Raum, sodass direkte auditive Kommunikation kein Problem darstellt. Wird in einer zukünftigen VR-Szene jedoch Ton abgespielt, könnte es sein, dass dieser direkte Kommunikationskanal hinfällig wird. Man müsste auch den Ton des Experimentators digitalisieren und in die VR-Szene einbinden, damit die Immersion hier nicht gestört wird. Auch der umgekehrte Weg wäre denkbar, da außenstehende Personen bisher keine Möglichkeit haben, den Ton einer Szene mitzuerleben.

Eine aus dem Nutzerfeedback abgeleitete Zusatzfunktion ist die optionale Auswahlmöglichkeit, ob der Experimentator in der virtuellen Umgebung visualisiert werden soll oder nicht. Damit ist die Erweiterung der Kommunikation lediglich eine Option und muss nicht dauerhaft genutzt werden. Aus der Evaluation durch die Nutzer geht außerdem hervor, dass der Austausch der Kameratechnik bei folgenden Iterationen eine hohe Priorität hat. Dadurch könnte beispielsweise ein größerer Sichtwinkel simuliert werden, der in etwa das FOV eines Menschen abbilden kann. Die Kinect deckt derzeit nur einen kleinen Bereich des menschlichen FOV ab.

Obwohl, wie beschrieben, einige unityspezifische Änderungen in der 3D-Szene vorgenommen werden mussten, wäre es trotzdem denkbar aus dem Konzept eine einzelne Unitypackage-Datei zu generieren. Der Vorteil wäre, dass man die Unity-Module als einheitliche Plug-Ins behandeln könnte. Man müsste in der nächsten Iteration lediglich auf die vorhandene Modularität aufbauen und die implementierte Funktionalität serviceorientiert gestalten. So würden die meisten Abhängigkeiten, wie das Format der Texturen oder auch die Maßeinheiten der Abstände bereits gelöst werden.

6.1.2 Zukünftige Arbeiten

Durch die Meta-Ebene des Konzepts kann man es in diverse Abläufe einbinden. Hier wurde der Kontext von Experimenten gewählt. Zusätzlich wären jedoch auch andere Anwendungsbereiche, wie Social Media, VR-Demonstrationen oder auch Spiele denkbar. Auf dem aktuellen Stand wird ebenfalls davon ausgegangen, dass sich die kommunizierenden Personen im selben Raum befinden. Da das vorgeschlagene System bereits ein verteiltes ist, wäre es interessant, das Kommunikationskonzept so zu verändern, dass Telekommunikation möglich wird. Die Personen könnten so in verschiedenen Räumen oder über weitere Entfernung miteinander kommunizieren. Verhandlungssituationen, Meetings und andere seriöse Teilbereiche des kooperativen Arbeitens würden somit

ZUSAMMENFASSUNG

erweitert werden. Wenn das System dann an mehreren Standorten gebraucht würde, wäre besonders die Einschränkung auf „low-budget“ Technik von hohem Nutzen.

Ein weiterer wichtiger Punkt ist die Erweiterung des Systems durch einen realistischen Selbstavatar. Bisher werden nur andere Nutzer realistisch im VE dargestellt. Durch Sensorfusion mehrerer Kameras ist es durchaus denkbar auch den eigenen Körper aus dem POV des Nutzers korrekt zu visualisieren. Dadurch wäre es möglich, weitere Schlüsse über die Perzeption und Kognition von realistischen Selbstavataren zu ziehen. In der bisherigen Implementierung bietet die Kinect dafür einen zu geringen FOV.

Als ein weiterer großer Punkt ist eine ausführlichere Evaluation des Konzepts anzumerken. Durch eine vollständige Studie und ein Experiment mit höherer Anzahl an Teilnehmern, könnte man nicht nur einen signifikanten Trend begründen, sondern eine elaborierte, umfassende Aussage treffen. Zusätzlich wäre es nicht nur gut zu wissen, dass die reale Darstellung von Körpersprache die Kommunikation erweitert, sondern auch welcher Zusammenhang zwischen Realismus des Avatars und Förderlichkeit für die zwischenmenschliche Kommunikation besteht. Eine vergleichende Studie wäre hierzu nötig.

Durch die Pilot-Studie wurde zwar nicht die Seite des Experimentators evaluiert, jedoch liegt es nach einigen Tests nahe, dass die Darstellung auf einem fest positionierten Bildschirm nicht das ideale Medium ist, um auch seine Seite der Kommunikation optimal zu unterstützen. Nächste Schritte wären daher, die Szene und den Probanden für ihn ebenfalls aus Egoperspektive zu visualisieren. Dazu wären AR-Techniken geeignet, wie Microsofts HoloLens oder Googles Glasses. So würde der Grad an Immersion, den der Experimentator erfährt, deutlich gesteigert werden, ohne den Bezug zur realen Umgebung zu vernachlässigen. Sobald dann solche professionellere aber auch kostspieligere und rechenintensivere Hard- und Software genutzt werden müsste, sollte vorher in jedem Fall das Laufzeitverhalten des vorgeschlagenen Konzepts evaluiert werden. Unter anderem wäre die Größe der verwendeten Texturen eine denkbare Variable dafür. So könnte man eine Aussage über zukünftiger Trends und Standards treffen, wie etwa 4K-Auflösung oder höhere 60-FPS-Videos.

Abschließend kann man also sagen, dass besonders in der eigentlichen Evaluation des Konzepts viele zukünftige Arbeiten liegen. Dies ist auf die bisher geringe Forschungsaktivität in diesem Bereich zurückzuführen. Insgesamt zeigt die vorliegende Thesis, dass dort durchaus ein hohes Potential für weitere tiefgehende Forschung zu virtueller Realität und menschzentrierter Kommunikation und Kooperation existiert.

DANKSAGUNGEN

Ich bedanke mich bei meinen Betreuern am Deutschen Forschungszentrum für Künstliche Intelligenz, Sebastian Alberternst und Jan Sutter. Außerdem danke ich dem gesamten Team des Forschungsbereichs Agenten und Simulierte Realität, insbesondere der Arbeitsgruppe für Forschungstransfer. Ich danke ebenfalls meinen Prüfern, Prof. Kloos und Prof. Tullius für die Ermöglichung der kooperativen Thesis.

Die vorliegende Thesis wurde durch ein Projekt des Bundesministeriums für Bildung und Forschung gefördert.

This work is supported by the Federal Ministry of Education and Research of Germany in the project Hybr-iT (Förderkennzeichen 01IS16026A).

LITERATURVERZEICHNIS

- Abate, A. F., Narducci, F., & Ricciardi, S. (2014, June). An image based approach to hand occlusions in mixed reality environments. In International Conference on Virtual, Augmented and Mixed Reality (pp. 319-328). Springer, Cham.
- Adolph, S., Cockburn, A., & Bramble, P. (2002). Patterns for effective use cases. Addison-Wesley Longman Publishing Co., Inc..
- Agarwal, R., & Prasad, J. (1998). A conceptual and operational definition of personal innovativeness in the domain of information technology. *Information systems research*, 9(2), 204-215.
- Alexiadis, D. S., Zarpalas, D., & Daras, P. (2013). Real-time, full 3-D reconstruction of moving foreground objects from multiple consumer depth cameras. *IEEE Transactions on Multimedia*, 15(2), 339-358.
- Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11), 1330-1334.
- Tsai, R. (1987). A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal on Robotics and Automation*, 3(4), 323-344.
- Allard, J., Lesage, J. D., & Raffin, B. (2010). Modularity for large virtual reality applications. *Presence: Teleoperators and Virtual Environments*, 19(2), 142-161.
- Alvarado, P., Bockholt, U., Canzler, U., Herbort, S., Heuser, N., Keitler, P., ... & Schwerdt, J. (2017). ARVIDA-Technologien. In *Web-basierte Anwendungen Virtueller Techniken* (pp. 193-217). Springer Berlin Heidelberg.
- Antakli, A., Moya, P. A., Brüderlin, B., Canzler, U., Dammertz, H., Enderlein, V., ... & Keitler, P. (2017). Virtuelle Techniken und Semantic-Web. In *Web-basierte Anwendungen Virtueller Techniken* (pp. 17-116). Springer Berlin Heidelberg.
- Schulzrinne, H. (1998). Real time streaming protocol (RTSP).
- Kelley, J. F. (1983, December). An empirical methodology for writing user-friendly natural language computer applications. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* (pp. 193-196). ACM.
- Anthes, C., García-Hernández, R. J., Wiedemann, M., & Kranzlmüller, D. (2016, March). State of the art of virtual reality technology. In *Aerospace Conference, 2016 IEEE* (pp. 1-19). IEEE.
- Argelaguet, F., Hoyet, L., Trico, M., & Lécuyer, A. (2016, March). The role of interaction in virtual embodiment: Effects of the virtual hand representation. In *Virtual Reality (VR), 2016 IEEE* (pp. 3-10). IEEE.
- Asteriadis, S., Chatzitofis, A., Zarpalas, D., Alexiadis, D. S., & Daras, P. (2013, June). Estimating human motion from multiple kinect sensors. In *Proceedings of the 6th international conference on computer vision/computer graphics collaboration techniques and applications* (p. 3). ACM.
- Schwaber, K., & Beedle, M. (2002). Agile software development with Scrum (Vol. 1). Upper Saddle River: Prentice Hall.
- Fischer, J., Bartz, D., & Straßer, W. (2004, November). Occlusion handling for medical augmented reality using a volumetric phantom model. In *Proceedings of the*

LITERATURVERZEICHNIS

- ACM symposium on Virtual reality software and technology (pp. 174-177). ACM.
- Ries, E. (2009). Minimum viable product: a guide. *Startup Lessons Learned*.
- Badler, N. (1997, June). Virtual humans for animation, ergonomics, and simulation. In Nonrigid and Articulated Motion Workshop, 1997. Proceedings., IEEE (pp. 28-36). IEEE.
- Bailenson, J. N., Yee, N., Merget, D., & Schroeder, R. (2006). 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. *Presence: Teleoperators and Virtual Environments*, 15(4), 359-372.
- Bär, T., Bockholt, U., Hoffmann, H., Jundt, E., Roth, M., Schreiber, W., ... & Zürl, K. (2017). Das Verbundprojekt ARVIDA. In *Web-basierte Anwendungen Virtueller Techniken* (pp. 1-16). Springer Berlin Heidelberg.
- Behr, J., Blach, R., Bockholt, U., Harth, A., Hoffmann, H., Huber, M., ... & Schubotz, R. (2017). ARVIDA-Referenzarchitektur. In *Web-basierte Anwendungen Virtueller Techniken* (pp. 117-191). Springer Berlin Heidelberg.
- Berger, K., Ruhl, K., Schroeder, Y., Bruemmer, C., Scholz, A., & Magnor, M. A. (2011, October). Markerless motion capture using multiple color-depth sensors. In *VMV* (pp. 317-324).
- Bierbaum, A., Just, C., Hartling, P., Meinert, K., Baker, A., & Cruz-Neira, C. (2001, March). VR Juggler: A virtual platform for virtual reality application development. In *Virtual Reality, 2001. Proceedings. IEEE* (pp. 89-96). IEEE.
- Billinghurst, M., & Kato, H. (1999, March). Collaborative mixed reality. In *Proceedings of the First International Symposium on Mixed Reality* (pp. 261-284).
- Billinghurst, M., Kato, H., & Poupyrev, I. (2001). The magicbook-moving seamlessly between reality and virtuality. *IEEE Computer Graphics and applications*, 21(3), 6-8.
- Biocca, F. (1992). Communication within virtual reality: Creating a space for research. *Journal of Communication*, 42(4), 5-22.
- Biocca, F., & Delaney, B. (1995). Immersive virtual reality technology. *Communication in the age of virtual reality*, 57-124.
- Biocca, F., & Levy, M. R. (Eds.). (2013). *Communication in the age of virtual reality*. Routledge.
- Bowman, D. A., & McMahan, R. P. (2007). Virtual reality: how much immersion is enough?. *Computer*, 40(7).
- Boykov, Y. Y., & Jolly, M. P. (2001). Interactive graph cuts for optimal boundary & region segmentation of objects in ND images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on* (Vol. 1, pp. 105-112). IEEE.
- Brey, P. (1999). The ethics of representation and action in virtual reality. *Ethics and Information technology*, 1(1), 5-14.
- Burdea, G. C., & Coiffet, P. (2003). *Virtual reality technology* (Vol. 1). John Wiley & Sons.

LITERATURVERZEICHNIS

- Cao, Z., Simon, T., Wei, S. E., & Sheikh, Y. (2016). Realtime multi-person 2d pose estimation using part affinity fields. arXiv preprint arXiv:1611.08050.
- Caplan, S. E., & Turner, J. S. (2007). Bringing theory to research on computer-mediated comforting communication. *Computers in human behavior*, 23(2), 985-998.
- Charmaz, K., & Belgrave, L. L. (2007). Grounded theory. John Wiley & Sons, Ltd.
- Collins, J. (2016). Occlusion and Interaction in Mixed Reality: A Systematic Inquiry (Doctoral dissertation, University of Otago).
- Collins, J., Regenbrecht, H., & Langlotz, T. (2017). Visual Coherence in Mixed Reality: A Systematic Enquiry. *PRESENCE: Teleoperators and Virtual Environments*, 26(1), 16-41.
- Daft, R. L., & Lengel, R. H. (1986). Organizational information requirements, media richness and structural design. *Management science*, 32(5), 554-571.
- Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* (Vol. 1, pp. 886-893). IEEE.
- Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, 319-340.
- de Groot, A. D. (1969). Methodology: Foundations of inference and research in the behavioral sciences. MTH.
- De Troyer, O., Kleinermann, F., Pellens, B., & Bille, W. (2007, November). Conceptual modeling for virtual reality. In *Tutorials, posters, panels and industrial contributions at the 26th international conference on Conceptual modeling- Volume 83* (pp. 3-18). Australian Computer Society, Inc..
- Dodds, T. J., Mohler, B. J., De La Rosa, S., Streuber, S., Bültlhoff, H. H., AN, A., & Hoven, E. (2011, May). Embodied interaction in immersive virtual environments with real time self-animated avatars. In *Workshop Embodied Interaction: Theory and Practice in HCI (CHI 2011)* (pp. 132-135). ACM Press.
- Dodds, T., Mohler, B. J., & Bültlhoff, H. (2010, June). A communication task in hmd virtual environments: Speaker and listener movement improves communication. In *23rd Annual Conference on Computer Animation and Social Agents (CASA 2010)* (pp. 1-4).
- Doerr, N. (2007). Heavy Rain devs have “conquered”the Uncanny Valley. Joystiq.
- Ducheneaut, N., Wen, M. H., Yee, N., & Wadley, G. (2009, April). Body and mind: a study of avatar personalization in three virtual worlds. In *Proceedings of the SIGCHI conference on human factors in computing systems* (pp. 1151-1160). ACM.
- Fast, J. (1988). Body language. Simon and Schuster.
- Ferretti, G., Filippi, S., Maffezzoni, C., Magnani, G., & Rocco, P. (1999). Modular dynamic virtual-reality modeling of robotic systems. *IEEE Robotics & Automation Magazine*, 6(4), 13-23.
- Findeli, A. (2010). Searching for design research questions: some conceptual clarifications. *Questions, Hypotheses & Conjectures: discussions on projects by early stage and senior design researchers*, 286-303.

LITERATURVERZEICHNIS

- Ford, P. J. (2001). A further analysis of the ethics of representation in virtual reality: multi-user environments. *Ethics and Information Technology*, 3(2), 113-121.
- Gao, C., Lin, Y., & Hua, H. (2012, November). Occlusion capable optical see-through head-mounted display using freeform optics. In *Mixed and Augmented Reality (ISMAR)*, 2012 IEEE International Symposium on (pp. 281-282). IEEE.
- Garau, M., Slater, M., Vinayagamoorthy, V., Brogni, A., Steed, A., & Sasse, M. A. (2003, April). The impact of avatar realism and eye gaze control on perceived quality of communication in a shared immersive virtual environment. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 529-536). ACM.
- Gaver, W. W., Beaver, J., & Benford, S. (2003, April). Ambiguity as a resource for design. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 233-240). ACM.
- Godbehere, A. B., Matsukawa, A., & Goldberg, K. (2012, June). Visual tracking of human visitors under variable-lighting conditions for a responsive audio art installation. In *American Control Conference (ACC)*, 2012 (pp. 4305-4312). IEEE.
- Gouskos, C. (2006). The depths of the uncanny valley. DOI= <http://uk.gamespot.com/features/6153667/index.html>.
- Greene, J. C., Caracelli, V. J., & Graham, W. F. (1989). Toward a conceptual framework for mixed-method evaluation designs. *Educational evaluation and policy analysis*, 11(3), 255-274.
- Gutekunst, M., Geuss, M., Rauhoeft, G., Stefanucci, J., Kloos, U., & Mohler, B. J. (2014, December). Short Paper: A Video Self-avatar Influences the Perception of Heights in an Augmented Reality Oculus Rift. In *ICAT-EGVE* (pp. 9-12).
- Hagen, I. (2017). Perception Virtuality. *INFORMATIK* 2017.
- Hashemipour, M., Manesh, H. F., & Bal, M. (2011). A modular virtual reality system for engineering laboratory education. *Computer Applications in Engineering Education*, 19(2), 305-314.
- Hebborn, A. K., Höhner, N., & Müller, S. (2017, October). Occlusion Matting: Realistic Occlusion Handling for Augmented Reality Applications. In *Mixed and Augmented Reality (ISMAR)*, 2017 IEEE International Symposium on (pp. 62-71). IEEE.
- Ho, C. C., MacDorman, K. F., & Pramono, Z. D. (2008, March). Human emotion and the uncanny valley: a GLM, MDS, and Isomap analysis of robot video ratings. In *Human-Robot Interaction (HRI)*, 2008 3rd ACM/IEEE International Conference on (pp. 169-176). IEEE.
- Hoffmann, H., Schreiber, W., Zimmermann, P., & Zürl, K. (2017). Verwertung und Nachhaltigkeit. In *Web-basierte Anwendungen Virtueller Techniken* (pp. 445-448). Springer Berlin Heidelberg.
- Hoggins, T. (2010). Heavy Rain Video Game Review. *The Telegraph*.
- Jamieson, G. H. (2007). Visual communication: More than meets the eye. Intellect Books.

LITERATURVERZEICHNIS

- Jerald, J. (2015). The VR book: human-centered design for virtual reality. Morgan & Claypool.
- Joinson, A. N. (2004). Self-esteem, interpersonal risk, and preference for e-mail to face-to-face communication. *CyberPsychology & Behavior*, 7(4), 472-478.
- Jones, B., Sodhi, R., Murdock, M., Mehra, R., Benko, H., Wilson, A., ... & Shapira, L. (2014, October). RoomAlive: magical experiences enabled by scalable, adaptive projector-camera units. In Proceedings of the 27th annual ACM symposium on User interface software and technology (pp. 637-644). ACM.
- KaewTraKulPong, P., & Bowden, R. (2002). An improved adaptive background mixture model for real-time tracking with shadow detection. *Video-based surveillance systems*, 1, 135-144.
- Kiyokawa, K., Billinghurst, M., Campbell, B., & Woods, E. (2003, October). An occlusion-capable optical see-through head mount display for supporting co-located collaboration. In Proceedings of the 2nd IEEE/ACM International Symposium on Mixed and Augmented Reality (p. 133). IEEE Computer Society.
- Knöpfel, A., Gröne, B., & Tabeling, P. (2005). Fundamental modeling concepts. Effective Communication of IT Systems, England.
- Kotonya, G., & Sommerville, I. (1998). Requirements engineering: processes and techniques. Wiley Publishing.
- Kowalski, M., Naruniec, J., & Daniluk, M. (2015, October). Live Scan3D: A Fast and Inexpensive 3D Data Acquisition System for Multiple Kinect v2 Sensors. In 3D Vision (3DV), 2015 International Conference on (pp. 318-325). IEEE.
- Kreylos, O. (2009). Vrui VR toolkit.
<http://idav.ucdavis.edu/~okreylos/ResDev/Vrui/index.html> (8/03/2018)
- Krueger, M. (1991). Artificial Reality II. Reading: Addison-Wesley.
- Latoschik, M., Roth, D., Gall, D., Achenbach, J., Waltemate, T., & Botsch, M. (2017). The Effect of Avatar Realism in Immersive Social Virtual Realities. In Proceedings of ACM Symposium on Virtual Reality Software and Technology.
- Lienhart, R., & Maydt, J. (2002). An extended set of haar-like features for rapid object detection. In Image Processing. 2002. Proceedings. 2002 International Conference on (Vol. 1, pp. I-I). IEEE.
- Lin, J. J., & Parker, D. E. (2007, July). User experience modeling and enhancement for virtual environments that employ wide-field displays. In International Conference on Digital Human Modeling (pp. 423-433). Springer, Berlin, Heidelberg.
- Linowes, J. (2015). Unity Virtual Reality Projects. Packt Publishing Ltd.
- Lowe, D. G. (1999). Object recognition from local scale-invariant features. In Computer vision, 1999. The proceedings of the seventh IEEE international conference on (Vol. 2, pp. 1150-1157). Ieee.
- Mania, K., & Chalmers, A. (1998). A Classification for User Embodiment in Collaborative Virtual Environments. Proceedings of VSMM'98-Fourth International Conference on Virtual Systems and Multimedia IOS Press.

LITERATURVERZEICHNIS

- Mehrabian, A., & Ferris, S. R. (1967). Inference of attitudes from nonverbal communication in two channels. *Journal of consulting psychology*, 31(3), 248.
- Mehrabian, A., & Wiener, M. (1967). Decoding of inconsistent communications. *Journal of personality and social psychology*, 6(1), 109.
- Mestre, D., Fuchs, P., Berthoz, A., & Vercher, J. L. (2006). Immersion et présence. Le traité de la réalité virtuelle. Paris: Ecole des Mines de Paris, 309-38.
- Milgram, P., & Kishino, F. (1994). A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12), 1321-1329.
- Milgram, P., Takemura, H., Utsumi, A., & Kishino, F. (1995, December). Augmented reality: A class of displays on the reality-virtuality continuum. In *Telemanipulator and telepresence technologies* (Vol. 2351, pp. 282-293). International Society for Optics and Photonics.
- Mohler, B. J., Creem-Regehr, S. H., Thompson, W. B., & Bülthoff, H. H. (2010). The effect of viewing a self-avatar on distance judgments in an HMD-based virtual environment. *Presence: Teleoperators and Virtual Environments*, 19(3), 230-242.
- Mori, M. (1970). The uncanny valley. *Energy*, 7(4), 33-35.
- Mori, M., MacDorman, K. F., & Kageki, N. (2012). The uncanny valley [from the field]. *IEEE Robotics & Automation Magazine*, 19(2), 98-100.
- Mulder, J. D., & Boscker, B. R. (2004, March). A modular system for collaborative desktop vr/ar with a shared workspace. In *Virtual Reality, 2004. Proceedings*. IEEE (pp. 75-280). IEEE.
- Nigolian, V., Mutlu, M., Hauser, S., Bernardino, A., & Ijspeert, A. (2017). Self-reconfigurable Modular Robot Interface Using Virtual Reality: Arrangement of Furniture Made out of Roombots Modules.
- Nordby, K., Gernez, E., & Børresen, S. (2016). Efficient Use of Virtual and Mixed Reality in Conceptual Design of Maritime Work Places. In *15th International Conference on Computer and IT Applications in the Maritime Industries-COMPIT'16*.
- Nowak, K. L., & Biocca, F. (2003). The effect of the agency and anthropomorphism on users' sense of telepresence, copresence, and social presence in virtual environments. *Presence: Teleoperators and Virtual Environments*, 12(5), 481-494.
- Ohta, Y., & Tamura, H. (2014). Mixed reality: merging real and virtual worlds. Springer Publishing Company, Incorporated.
- Pai, Y. S., Yap, H. J., Dawal, S. Z. M., Ramesh, S., & Phoon, S. Y. (2016). Virtual Planning, Control, and Machining for a Modular-Based Automated Factory Operation in an Augmented Reality Environment. *Scientific reports*, 6, 27380.
- Pandzic, I., Babski, C., Capin, T., Lee, W., Magnenat-Thalmann, N., Musse, S. R., ... & Thalmann, D. (2001). Simulating virtual humans in networked virtual environments. *Presence*, 10(6), 632-646.
- Parisi, T. (2015). Learning virtual reality: developing immersive experiences and applications for desktop, web, and mobile. " O'Reilly Media, Inc.".

LITERATURVERZEICHNIS

- Parnas, D. L. (1972). On the criteria to be used in decomposing systems into modules. *Communications of the ACM*, 15(12), 1053-1058.
- Pavlik, R. A., & Vance, J. M. (2010, January). A modular implementation of Wii remote head tracking for virtual reality. In ASME 2010 World Conference on Innovative Virtual Reality (pp. 351-359). American Society of Mechanical Engineers.
- Peng, G., Wang, G., Liu, W., & Yu, H. (2010). A desktop virtual reality-based interactive modular fixture configuration design system. *Computer-Aided Design*, 42(5), 432-444.
- Perrey, R., & Lycett, M. (2003, January). Service-oriented architecture. In Applications and the Internet Workshops, 2003. Proceedings. 2003 Symposium on (pp. 116-119). IEEE.
- Phillips, C. B., & Badler, N. I. (1988, January). Jack: A toolkit for manipulating articulated figures. In Proceedings of the 1st annual ACM SIGGRAPH symposium on User Interface Software (pp. 221-229). ACM.
- Pomi, A., & Slusallek, P. (2005). Interactive ray tracing for virtual TV studio applications. *Journal of Virtual Reality and Broadcasting*, 2(1), 2.
- Regenbrecht, H., Meng, K., Reepen, A., Beck, S., & Langlotz, T. (2017, October). Mixed Voxel Reality: Presence and Embodiment in Low Fidelity, Visually Coherent, Mixed Reality Environments. In Mixed and Augmented Reality (ISMAR), 2017 IEEE International Symposium on (pp. 90-99). IEEE.
- Roberts, J. C., Headleand, C., & Ritsos, P. D. (2016). Sketching designs using the five design-sheet methodology. *IEEE transactions on visualization and computer graphics*, 22(1), 419-428.
- Roo, J. S., & Hachet, M. (2017, October). One Reality: Augmenting How the Physical World is Experienced by combining Multiple Mixed Reality Modalities. In Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology (pp. 787-795). ACM.
- Roth, D., Lugrin, J. L., Galakhov, D., Hofmann, A., Bente, G., Latoschik, M. E., & Fuhrmann, A. (2016, March). Avatar realism and social interaction quality in virtual reality. In Virtual Reality (VR), 2016 IEEE (pp. 277-278). IEEE.
- Rother, C., Kolmogorov, V., & Blake, A. (2004, August). Grabcut: Interactive foreground extraction using iterated graph cuts. In ACM transactions on graphics (TOG) (Vol. 23, No. 3, pp. 309-314). ACM.
- Rothwell, J. D. (2010). In the company of others: An introduction to communication. Oxford University Press, USA.
- Sagert-Müller, A. (2008). Zur Bedeutung der Körpersprache in der LehrerSchüler-Interaktion. Nordestadt: GRIN Verlag.
- Schaik, P. V., Turnbull, T., Wersch, A. V., & Drummond, S. (2004). Presence within a mixed reality environment. *CyberPsychology & Behavior*, 7(5), 540-552.
- Scheflen, A. E. (1972). Body Language and the Social Order; Communication as Behavioral Control.
- Schoett, O. (1986). Data abstraction and the correctness of modular programming.

LITERATURVERZEICHNIS

- Schubert, T., Friedmann, F., & Regenbrecht, H. (2001). The experience of presence: Factor analytic insights. *Presence: Teleoperators and virtual environments*, 10(3), 266-281.
- Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., ... & Moore, R. (2013). Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1), 116-124.
- Slater, M. (1999a). Measuring presence: A response to the Witmer and Singer presence questionnaire. *Presence: Teleoperators and Virtual Environments*, 8(5), 560-565.
- Slater, M. (1999b). Real people meeting virtually real people—a review of some experiments in shared virtual environments. *BT Technology Journal*, 17(1), 120-127.
- Slater, M. (2003). A note on presence terminology. *Presence connect*, 3(3), 1-5.
- Slater, M., & Wilbur, S. (1997). A framework for immersive virtual environments (FIVE): Speculations on the role of presence in virtual environments. *Presence: Teleoperators and virtual environments*, 6(6), 603-616.
- Slater, M., Perez-Marcos, D., Ehrsson, H. H., & Sanchez-Vives, M. V. (2009). Inducing illusory ownership of a virtual body. *Frontiers in neuroscience*, 3(2), 214.
- Slater, M., Usoh, M., & Steed, A. (1994). Depth of presence in virtual environments. *Presence: Teleoperators & Virtual Environments*, 3(2), 130-144.
- Sless, D. (1981). Learning and visual communication. Taylor & Francis.
- Smith, A. G. (1997). Testing the surf: criteria for evaluating Internet information resources. *Public Access-Computer Systems Review*, 8(3).
- Sung, W. T., & Ou, S. C. (2001). Development of a distribution VR-based complex graphics design system. In *Advanced Intelligent Mechatronics, 2001. Proceedings. 2001 IEEE/ASME International Conference on* (Vol. 2, pp. 793-798). IEEE.
- Szalavári, Z., Schmalstieg, D., Fuhrmann, A., & Gervautz, M. (1998). “Studierstube”: An environment for collaboration in augmented reality. *Virtual Reality*, 3(1), 37-48.
- Tabeling, P. (2006). Softwaresysteme und ihre Modellierung: Grundlagen, Methoden und Techniken. Springer-Verlag.
- Tanenbaum, A. S., & Van Steen, M. (2007). Distributed systems: principles and paradigms. Prentice-Hall.
- Taylor, T. L. (2002). Living digitally: Embodiment in virtual worlds. In *The social life of avatars* (pp. 40-62). Springer, London.
- Tinwell, A., Grimshaw, M., Nabi, D. A., & Williams, A. (2011). Facial expression of emotion and perception of the Uncanny Valley in virtual characters. *Computers in Human Behavior*, 27(2), 741-749.
- Tramberend, H. (1999, March). Avocado: A distributed virtual reality framework. In *Virtual Reality, 1999. Proceedings.*, IEEE (pp. 14-21). IEEE.
- Vapnik, V. N., & Chervonenkis, A. J. (1974). Theory of pattern recognition.
- Ventrella, J. J. (2011). Virtual body language: the history and future of avatars: How nonverbal expression is evolving on the internet. Etc Press.

LITERATURVERZEICHNIS

- Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on (Vol. 1, pp. I-I). IEEE.
- Walton, D. R., & Steed, A. J. S. (2017, November). Accurate Real-Time Occlusion for Mixed Reality. In VRST. ACM.
- Wei, S. E., Ramakrishna, V., Kanade, T., & Sheikh, Y. (2016). Convolutional pose machines. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4724-4732).
- Wendt, S. (2013). Nichtphysikalische Grundlagen der Informationstechnik: Interpretierte Formalismen. Springer-Verlag.
- Whiting, N. and Donaldson, N. 2016. Lessons from Integrating the Oculus Rift into Unreal Engine 4. Oculus Connect Developers Conference, 2014. Letzter Zugriff am 18.12.2017, auf http://static.oculus.com/connect/slides/OculusConnect_Epic_UE4_Integration_and_Demos.pdf.
- Witmer, B. G., & Singer, M. J. (1998). Measuring presence in virtual environments: A presence questionnaire. *Presence: Teleoperators and virtual environments*, 7(3), 225-240.
- Wright, K. (2002). Social support within an on-line cancer community: An assessment of emotional support, perceptions of advantages and disadvantages, and motives for using the community from a communication perspective. *Journal of Applied Communication Research*, 30(3), 195-209.
- Yee, N., & Bailenson, J. (2007). The Proteus effect: The effect of transformed self-representation on behavior. *Human communication research*, 33(3), 271-290.
- Zhu, Y., Zhu, K., Fu, Q., Chen, X., Gong, H., & Yu, J. (2016, December). SAVE: shared augmented virtual environment for real-time mixed reality applications. In Proceedings of the 15th ACM SIGGRAPH Conference on Virtual-Reality Continuum and Its Applications in Industry-Volume 1 (pp. 13-21). ACM.

ABKÜRZUNGSVERZEICHNIS

API = Application Programming Interface

AR = Augmented Reality

ASR = Agenten und Simulierte Realität

AV = Augmented Virtuality

BB = Bounding-Box

CAD = Computer-Aided-Design

CPU = Central Processing Unit

DFKI = Deutsches Forschungszentrum für Künstliche Intelligenz

DIY = Do-It-Yourself

DLL = Dynamic Link Library

FOR = Field of Regard

FOV = Field of View

FPS = Frames per Second

GHz = Gigaherz

GPU = Graphics Processing Unit

HD = High Definition

HMD = Head-Mounted-Device

IPC = Inter-Process-Communication

Mb = Megabyte

Mb/s = Megabyte pro Sekunde

Mbit = Megabit

Mbit/s = Megabit pro Sekunde

MR = Mixed Reality

MTU = Maximum Transmission Unit

MTU = Maximum Transmission Unit

POV = Point of View

RGB = Rot, Grün, Blau

RGBD = Rot, Grün, Blau und Tiefe

RV = Reality-Virtuality

SD = Standardabweichung

SDK = Software Development Kit

TCP = Transmission Control Protocol

UCD = User-Centred Design

UDP = User Datagram Protocol

VE = Virtual Environments

VR = Virtual Reality

VT = Virtuelle Techniken

ANHANG

- A. Medium mit digitalen Artefakten (Programmcode, Videos etc.)
- B. Zeitplan mit potentiellen Milestones
- C. Abstrakter Aufbau des Anforderungsinterviews bei dem der Prototyp nach Iteration 1 genutzt wurde
- D. Modellierung des Gesamtsystemkonzepts
- E. MultiVR-Aspekte mit Bezug zu dem Projekt der Thesis
- F. Evaluationsunterlagen/Fragebögen
- G. Ausführlicher Ablauf des angepassten Szenarios
- H. Vergrößerte Darstellungen der Unity-Ablaufdiagramme
- I. Experimentenanalyse (Tabellen und Umrechnung)

ANHANG B - Zeitplan mit potentiellen Milestones

- Anmeldung 15.10.:
- 15.10. – 29.10. Anforderungs-Recherche für Iteration 1 (2 Wochen)
- 29.10. – 12.11. Digitaler fake Prototyp (Mock-Up) (2 Wochen)
 - o für Kommunikation von Look and Feel
- 12.11. – 10.12. Erster Prototyp mit Grundfunktionalität (ca. 4 Wochen)
 - o (am I building the right thing)
 - o 1 Woche Kinect
 - o 1 Woche IPC
 - o 2 Wochen Compositing
 - o ➔ Ergebnis: Funktionaler Medium-Fidelity-Prototype
- 10.12. – 31.12. Use-Case: Industrie 4.0 Unity Szene in VR & Interaktion (ca. 3 Wochen)
 - o 2 Iterationen:
 - 2 Wochen Implementierung
 - 1 Woche Einbindung von Feedback
- 31.12. – 14.01. Anpassen von Anforderungen und Fokus des Systems nach erster Iteration
 - o Rücksprache mit Verantwortlichen am DFKI + Experten Interviews (2 Wochen)
- 14.01. – 11.02. Zweiter Prototyp (ca. 4 Wochen)
 - o (“am I building the right thing & am I building the thing right”)
 - o Probanden POV
 - o ➔ Ergebnis: High-Fidelity-Prototyp
- 11.02. – 25.02. PUFFER + Exploration bezüglich Multi-Person-VR (ca. 2 Wochen)
- 25.02.- 25.03. Evaluation (ca. 4 Wochen)
 - o 2 Wochen Planung & Terminvorbereitung
 - o 1 Woche aktive Evaluation (Experiment)
 - o 1 Woche Auswertung
- 15.03-15.04 Schreibprozess / Puffer

ANHANG C - Abstrakter Aufbau des Anforderungsinterviews bei dem der Prototyp nach Iteration 1 genutzt wurde

Setting and Structure of the interview (5 min)

1. Structure:
 - 1.1. Start with some questions about the general approach and workflow during VR experiments
 - 1.2. Show prototype and talk about this in more detail
2. Introduce myself and let interviewee introduce his-/herself
3. Setting of my thesis: Thesis placed in general VR experiment setting
 - 3.1. Especially: Observance of participants in VR and communication during experiment
 - 3.1.1. Problem: Visual Representation
 - 3.1.2. First small research / experiments → pointed out a lack of visual representation and observation possibilities, which complicated communication
 - 3.1.2.1. People with HMD always “tried” to look in the right direction while talking, me always staring at the screen trying to point him to the right direction
 - 3.1.3. Mutual visual representation:
 - 3.1.3.1. How could it improve general experimentation workflow and influence outcome?

Questions about the general approach of observance (15 min)**1. Could you explain the process of an experimentation? What are the steps?**

1.1. Questions on introduction and occurrence of problems

 1.1.1. Where does the process start? How do you give the subjects an introduction into the scene? Why?

 1.1.2. What happens if there occur any problems while the participant is in VR?

 1.1.2.1. E.g. when he has a question on what he has to do now?

 1.1.2.1.1. Maybe he does not find what he was looking for.

 1.2. How do you ask them questions during the experiment? How do they ask

 questions during the experiment? How do you / they answer? Why?

 1.2.1. Verbal / Physical interaction?

 1.2.2. Where are you located at this point?

 1.2.2.1. Near participant? Why?

 1.2.2.2. At the computer? Why?

 1.2.2.3. Within the tracking area? Why?

 1.2.2.3.1. What is a typical location for VR experiments? How does the physical setting look like?

1.3. How do you observe participants and the virtual environment?

 1.3.1. In order to help them for example. Why?

 1.3.2. In order to get insights that you are looking for. Why?

 1.3.2.1. Influence on the outcome of the experiment.

2. What does a tool for observing and communicating with participants need to do?

 2.1. Helpful information for you?

 2.2. Visualize the participant for you?

 2.3. Visualize you for the participant?

 2.4. Static / mobile view?

3. How would you describe the necessity or importance of the participant being able to see you, compared to you seeing the participant? Why?

 3.1. Would this solely be useful for the experiment workflow itself? Why?

 3.1.1. Do you analyse and evaluate your experiments afterwards or do you sometimes try to get insights during the experiment?

3.2. Would an abstract visualization or an animated avatar be sufficient for this mutual representation?

 3.2.1. Why do you think an CG avatar could be sufficient?

 3.2.1.1. Why do you think the real-texture avatar representation could be helpful?

3.3. What would you think about being able to see the “real” participant within the VR scene? (real-world textured avatar)

3.3.1. What would you think about the participant seeing the real body presentation of you while giving an introduction of the experiment or the VR scene?

 3.3.1.1.1. What are the scenarios where a real body presentation could be helpful? Why?

 3.3.1.2. Multi-person VR, seeing each other's real body?

 3.3.1.3. When working with participants that are unexperienced with VR, could you imagine that seeing a familiar person (like you at this point) could counteract nervousness? Why? Examples?

 3.3.1.3.1. Would you agree, that this could positively affect the workflow of the experiment? Why?

 3.3.1.4. Would you agree, that this could positively affect the outcome of the experiment?

 3.3.1.4.1. Maybe because the participants feel more comfortable?

 3.3.1.4.1.1. Leading to more natural movements of the participants?

4. What are your expectations regarding a system that enables to observe people in VR?

4.1. Do you think that communication is more essential than pure observation? Why?

5. Which software do you normally use for VR experiments? Why?

Prototype introduction (5 min)

1. Show 1-2 Videos
2. Implemented a prototype for interactive real-texture observance of participants
 - 2.1. In order to (theses):
 - 2.1.1. Improve the abstract workflow of VR experiments
 - 2.1.2. To use it content-related:
 - 2.1.2.1. For example, here in analysing hybrid team interactions

Feedback on the prototype (15 min)

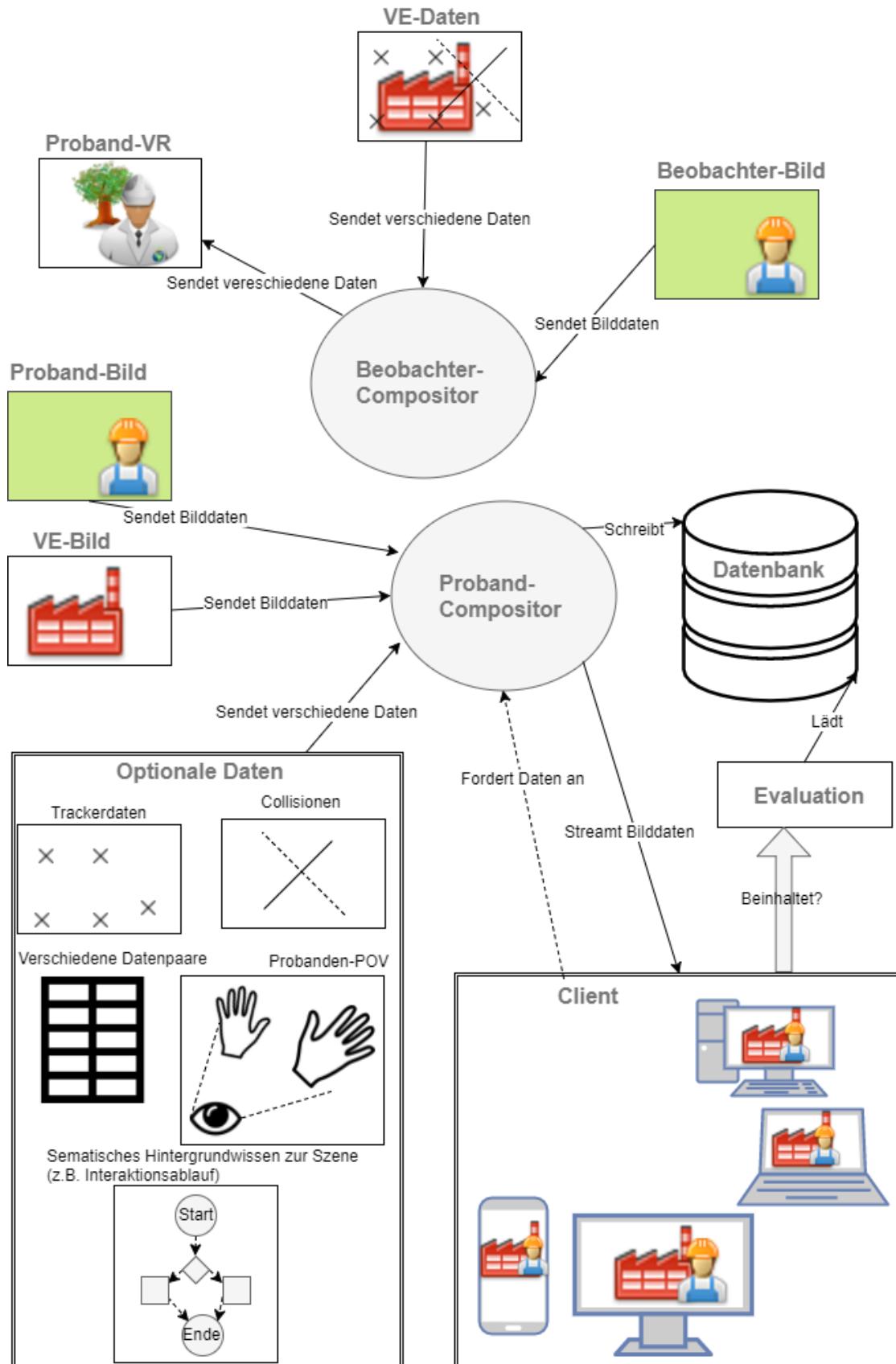
1. What do you think of the general concept of observing the real peoples texture as an experimenter?
 - 1.1. What do you think of the concept as a tool for orientation and communication for participants seeing the experimenter? Why?
 - 1.1.1. (and the experimenter being able to for example physically point onto things that are within the VR scene)
 - 1.2. Would you generally prefer an alternative solution for observation and communication?
 - 1.2.1. CG avatars?
 - 1.2.2. Fully integrated into the scene (so that you do not have to leave the computer)?
 - 1.3. How would you estimate the usage of the tool within specific disciplines rather than general experimentation scenarios? Why?
 - 1.3.1. Planning to apply it in Human-Machine Interaction (evaluating hybrid team interactions, telepresence) / Learning scenarios within training and education
 - 1.3.1.1. Assumption: Real texture representation improves the relationship or familiarity between the persons, by providing more personal information of the other side, like real clothes, height, weight, skin-type, hair, body language etc.
 - 1.3.1.1.1. ...which could be more personal than an CG avatar representation (at least for more low-cost solutions of course)
2. How would this method integrate in your actual experimentation workflow?
 - 2.1. What features would you like to change, so that it fits your needs? Why?
 - 2.1.1. What problems do you see using the tool? Why?
 - 2.1.2. In which way does the tool need to be changed in order to solve these problems?
 3. How important is the reusability of such a tool?
 - 3.1. Eg. independent usage concerning unity3D / unrealengine / vicon etc?

Short summary of the interview, in order to verify that we built a consent result (5 min)

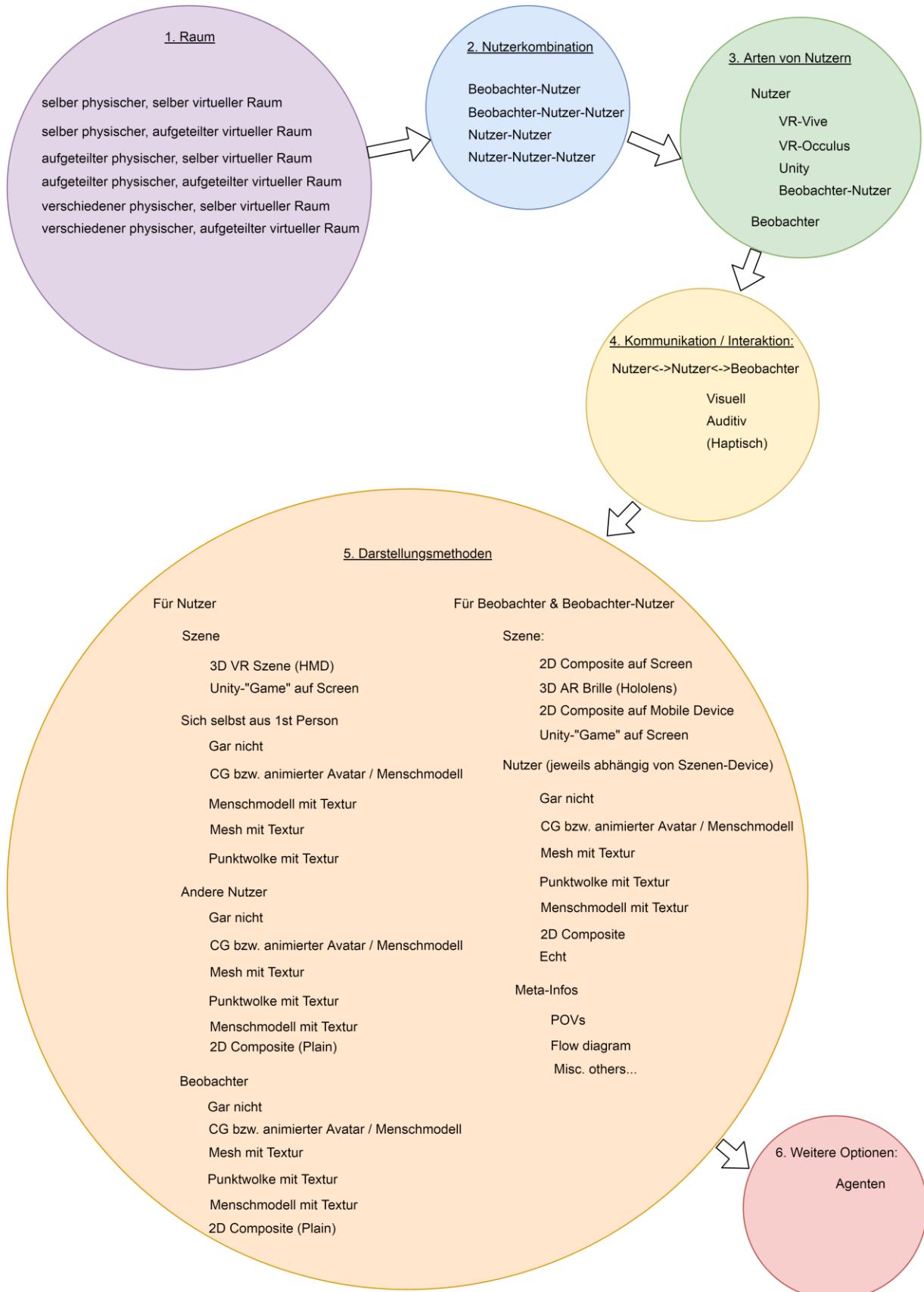
1. State general problem
2. State actual solution / workflow
3. State opinions on improvements on the concept / system
4. Are there any questions from your side?

⇒ ~ 45 min all over

ANHANG D - Modellierung des Gesamtsystemkonzepts



ANHANG E - MultiVR-Aspekte mit Bezug zu dem Projekt der Thesis



ANHANG F - Evaluationsunterlagen / Fragebögen

German
Research Center
for Artificial
Intelligence



Hochschule Reutlingen
Reutlingen University



Software Evaluation Entry Questionnaire

You are invited to participate in a project called Avatar2Avatar by completing the following questionnaire.

The aim of the project is to evaluate a virtual reality system with relation to communication. Because different users have different backgrounds and requirements, it would help us to know a little more about you. You do not have to answer any questions that you do not wish to.

You are:

- Female
- Male

How old were you at your last birthday? Age:

The software has to do with virtual reality (VR).

In a typical month, how often do you use VR technology?

- I have never used VR technology
- Less than once
- 1 to 5 times
- 5 to 10 times
- More than 10 times

If you already did use VR technology, which of the following systems did you use? (Multiple are possible)

- HTC Vive
- Oculus (Development Kit / The Rift)
- Samsung Gear VR
- Google Cardboard
- Other:

If you already did use VR technology, what is the type of setting that you use it in? (Multiple are possible)

- Gaming
- Software development
- Simulation / Data Visualization
- Social media
- Other:

Are you currently enrolled as a student?

- No
- Yes

Are you currently employed?

- No
- Yes

Your profession/studies is/are closely related to:

- Engineering and computer science
- Psychology
- Education
- Law and public policy
- Business
- Arts and Entertainment
- Architecture and design
- None of the above, it is more likely related to the field of:
- The question is not applicable

Ablauf / Aufgaben:

Der Ablauf des Experiments ergibt sich aus verschiedenen Aufgaben. Folgende Aufgaben werden mündlich vom Experimentator an den Proband gestellt:

1. Begrüßung des Probanden durch den Experimentator (im selben Raum, indem das Experiment stattfindet).
2. Der Proband wird gebeten sich kurz am vorbereiteten Platz zu setzen (Tisch am Eingang des Raums) und den Entry-Fragebogen auszufüllen.
3. Tutorial / 5 Min Eingewöhnungszeit:
 - a. Einweisung in Technik. Dem Proband wird in Kürze die Technik angelegt und erklärt. Es wird darauf hingewiesen, dass er weder etwas „falsches“ tun kann, noch etwas kaputt machen kann – alles was er tut ist stets positiv für das Ergebnis des Experiments.
 - b. Sicherheitshinweise werden berücksichtigt (wie z.B. das Symbol des blauen „Käfigs“, welcher bei Näherkommen die Wand darstellt).
4. Experiment beginnt / Kommunikation muss hergestellt werden:
 - a. Der Proband wird gebeten anzufangen (ohne weitere Erklärung)
 - i. Erwartete Gegenfrage: „Mit was soll angefangen werden?“
 - b. Alle weiteren Schritte werden ebenfalls nur einzeln und sehr kurz beschrieben, sodass der Proband gezwungen ist, mit dem Experimentator zu kommunizieren.
 - c. Außerdem wird mindestens ein Gegenstand vom Experimentator benannt, der nur mit Hilfe gefunden werden kann (d.h. es wird notwendig, dass der Experimentator visuell darauf deutet).
5. Dem Proband wird die Technik wieder abgelegt und er wird gebeten, sich an den dafür vorbereiteten Tisch zu setzen und den Post-Study-Fragebogen auszufüllen (anderer Tisch als der für das Entry-Questionnaire).
6. Dem Proband wird für die Teilnahme gedankt und ihm wird ein Schokoriegel seiner Wahl und der geringfügige Lohn überreicht. Danach wird er aus dem Raum geführt und verabschiedet.

Research Information Sheet for the Software Requirements Questionnaire

You are invited to participate as a subject in a project entitled

Avatar2Avatar

The aim of this project is:

To evaluate a virtual reality system with relation to communication and interaction.

The project is funded by the Hybr-iT project (<http://hybr-it-projekt.de/>)

Expected time commitment is ~20-30 min.

Your participation in this project will involve completing two questionnaires.

The results of the project may be published, but you may be assured of the complete confidentiality of data gathered in this investigation. Only the researcher and supervisor will have access to the raw data. To ensure anonymity and confidentiality your name will not be used in any results or publication. We will record your name only so that we can withdraw your results from the research if you ask us to.

The project is being carried out by:

Name of principal researcher: Robin Horst

Contact Details: robinhorst@gmx.de

He will be pleased to discuss any concerns you have about participation in the project.

Consent Form for Requirements and/or Software Evaluation

Avatar2Avatar

I have read and understood the research information sheet for the above-named project. On this basis I agree to participate as a subject in the project, and I consent to publication of the results of the project with the understanding that anonymity will be preserved. I also understand that I may withdraw from the project at any time, including withdrawal of any information I have provided.

I confirm that I am over 16 years of age.

Signed: _____ Date: _____

ID: _____

(This number will be recorded with your data only so we can withdraw it at your request)

Software Evaluation Post Study Questionnaire

Thank you for your interest in our survey. It'll take you about 5 minutes to fill in the questionnaire. You can work through the questionnaire offline, but in order to submit the results at the end you'll have to be online. Please answer all questions only with reference to one single episode of interaction with a virtual environment.

Were there other real persons within the virtual environment besides yourself?

- No
- Yes, but I did not interact with them
- Yes, and I did interact with them

Were there artificial characters (e.g. computer game opponents) within the virtual environment?

- No
- Yes, but I did not interact with them
- Yes, and I did interact with them

Now you'll see some statements about experiences. Please indicate, whether or not each statement applies to your experience. If a question is not relevant to the virtual environment you used, just skip it. You can use the whole range of answers. There are no right or wrong answers, only your opinion counts. You will notice that some questions are very similar to each other. This is necessary for statistical reasons. And please remember: Answer all these questions only referring to this one experience.

How aware were you of the real world surrounding while navigating in the virtual world? (i.e. sounds, room temperature, other people, etc.)?

	-3	-2	-1	0	+1	+2	+3	
<i>Extremely aware</i>								<i>Not aware at all</i>
<i>Moderately aware</i>								

How real did the virtual world seem to you?

	-3	-2	-1	0	+1	+2	+3	
<i>Completely real</i>								<i>Not real at all</i>

I had a sense of acting in the virtual space, rather than operating something from outside.

	-3	-2	-1	0	+1	+2	+3	
<i>Fully disagree</i>								<i>Fully agree</i>

How much did your experience in the virtual environment seem consistent with your real-world experience?

	-3	-2	-1	0	+1	+2	+3	
<i>Not consistent</i>								<i>Very consistent</i>
								<i>Moderately consistent</i>

How real did the virtual world seem to you?

	-3	-2	-1	0	+1	+2	+3	
<i>About as real as an imagined world</i>								<i>Indistinguishable from the real world</i>

I did not feel present in the virtual space.

	-3	-2	-1	0	+1	+2	+3	
<i>Did not feel</i>								<i>Felt present</i>

I was not aware of my real environment.

	-3	-2	-1	0	+1	+2	+3	
<i>Fully disagree</i>								<i>Fully agree</i>

In the computer generated world I had a sense of "being there".

	-3	-2	-1	0	+1	+2	+3	
<i>Not at all</i>								<i>Very much</i>

Somehow I felt that the virtual world surrounded me.

	-3	-2	-1	0	+1	+2	+3	
<i>Fully disagree</i>								<i>Fully agree</i>

I felt present in the virtual space.

	-3	-2	-1	0	+1	+2	+3	
<i>Fully disagree</i>								<i>Fully agree</i>

I still paid attention to the real environment.

	-3	-2	-1	0	+1	+2	+3	
<i>Fully disagree</i>								<i>Fully agree</i>

The virtual world seemed more realistic than the real world.

	-3	-2	-1	0	+1	+2	+3	
<i>Fully disagree</i>								<i>Fully agree</i>

I felt like I was just perceiving pictures.

	-3	-2	-1	0	+1	+2	+3	
<i>Fully disagree</i>								<i>Fully agree</i>

I was completely captivated by the virtual world.

	-3	-2	-1	0	+1	+2	+3	
<i>Fully disagree</i>								<i>Fully agree</i>

List the most **negative** aspects:

- 1.
- 2.
- 3.

List the most **positive** aspects:

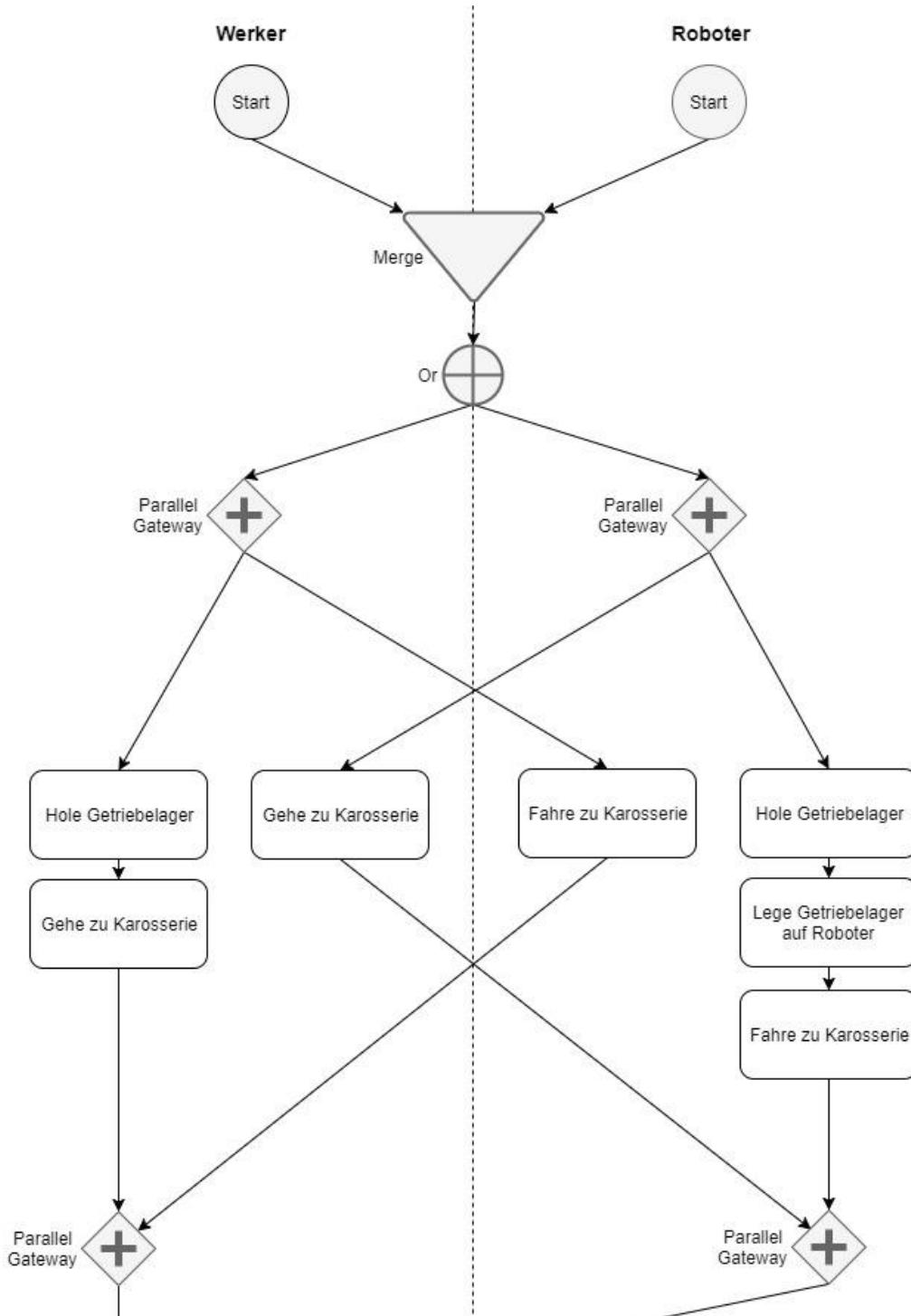
- 1.
- 2.
- 3.

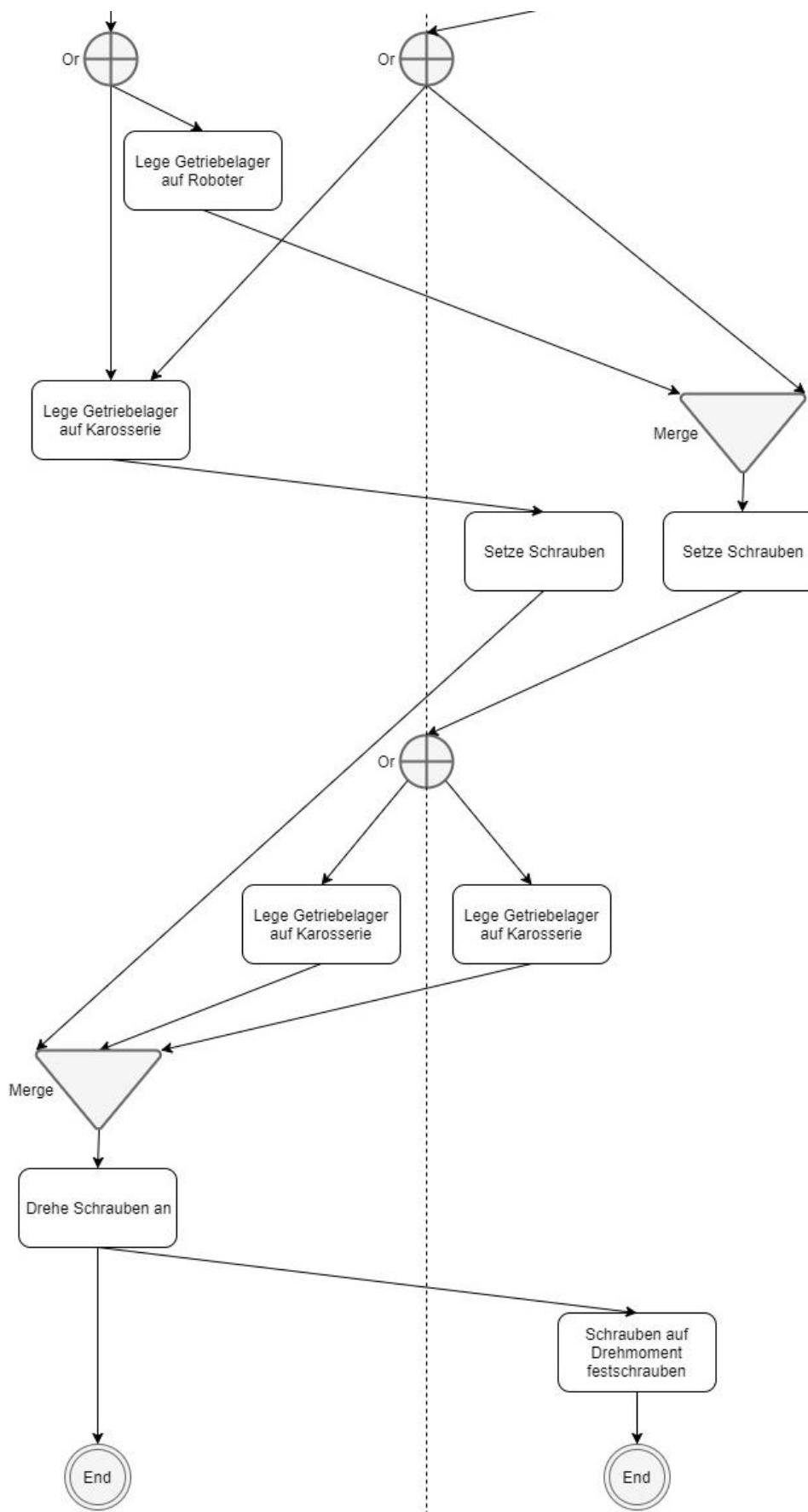
Other things that you would like to tell us about the project:

.....
.....

The team thanks you very much for your participation!

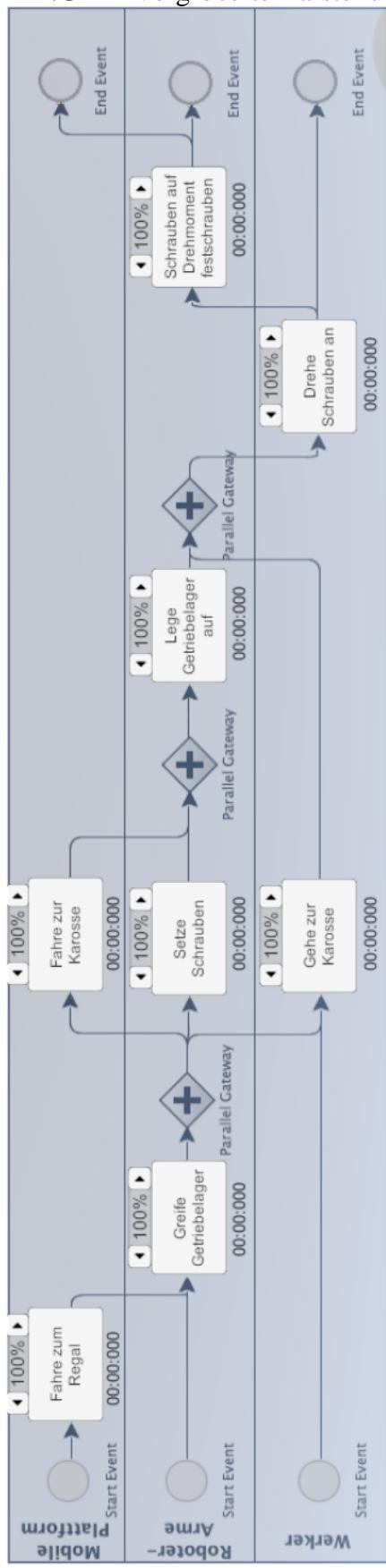
ANHANG G - Ausführlicher Ablauf des angepassten Szenarios



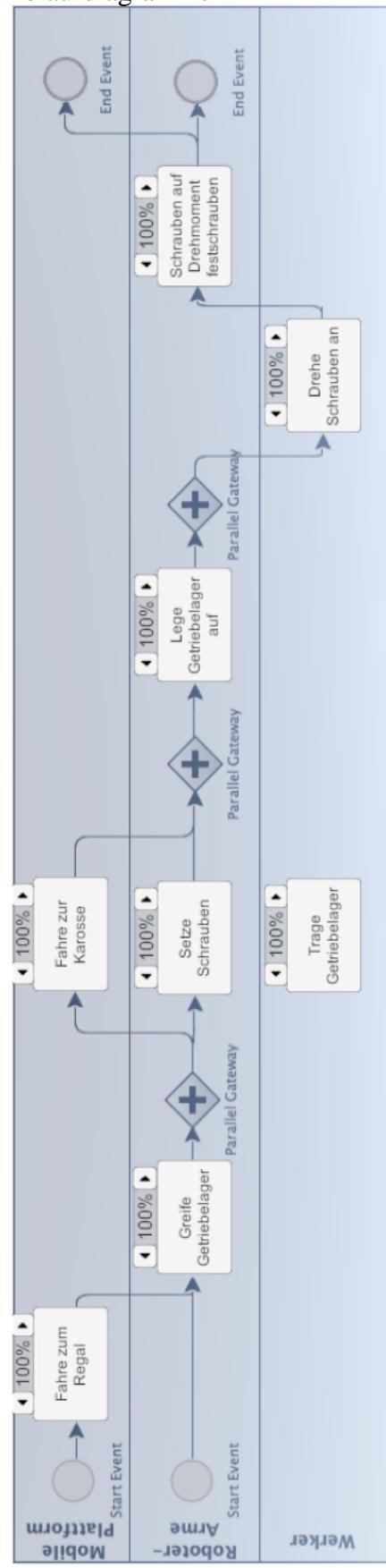


ANHANG H - Vergrößerte Darstellungen der Unity-Ablaufdiagramme

Ursprünglicher Ablauf



Angepasster Ablauf (vereinfacht)



ANHANG I - Experimentenanalyse (Tabellen und Umrechnung)**Zuordnung der Fragen zu den korrespondierten Elementen**

Frage Nr.	Element
1	INV1
2	REAL1
3	SP4
4	REAL2
5	REAL3
6	SP3
7	INV2
8	G1
9	Sp1
10	SP5
11	INV3
12	REAL4
13	SP2
14	INV4

Mapping von Fragebogen- zu Auswertungspunkten

3=6

2=5

1=4

0=3

-1=2

-2=1

-3=0

Zweistichproben t-Test (Paarvergleichstest)

	PRES-Kontrollgruppe	PRES-Experimentalgruppe
Mittelwert	3,4212963	3,5462963
Varianz	0,11586934	0,068107
Beobachtungen	6	6
Pearson Korrelation	-0,53620961	
Hypothetische Differenz der Mittelwerte	0	
Freiheitsgrade (df)	5	
t-Statistik	-0,57942049	
P(T<=t)	0,58743025	
Kritischer t-Wert	2,57058184	

Kontrollgruppe:

Proband Nr.	G1	SP1	SP2	SP3	SP4	SP5	SP	INV1	INV2
1	5	5	1	4	4	4	4,16666667	3	2
2	5	5	1	4	5	4	4,33333333	2	2
5	4	6	1	4	4	4	4,16666667	1	2
8	6	5	0	5	5	5	4,66666667	0	0
9	5	5	0	5	5	2	4	2	5
11	4	4	1	5	6	4	4	1	3
Ø	4,83333333	5	0,66666667	4,5	4,83333333	3,83333333	4,22222222	1,5	2,33333333

INV3	INV4	INV	REAL1	REAL2	REAL3	REAL4	REAL	PRES
3	4	3	2	6	5	0	3,25	3,47222222
4	4	3	2	4	5	2	3,25	3,52777778
6	2	2,25	1	4	4	0	2,25	2,88888889
5	4	2,75	1	5	5	3	3,5	3,63888889
3	1	3,25	5	5	5	2	4,25	3,83333333
5	3	3	2	4	4	0	2,5	3,16666667
4,33333333	3	2,875	2,16666667	4,66666667	4,66666667	1,16666667	3,16666667	3,4212963

SP = Spatial Presence = $((SP1+Sp4+SP5+G1)+(-1*SP2+6)+(-1*SP3+6)) / 6$

INV = Involvement = $((INV4+INV2)+(-1*INV1+6)+(-1*INV3+6))/4$

REAL = Experienced Realism = $(REAL1+REAL2+REAL3+REAL4)/4$

PRES = Overall Presence = $(SP+INV+REAL)/3$

Experimentalgruppe:

Proband Nr.	G1	SP1	SP2	SP3	SP4	SP5	SP	INV1	INV2
3	6	5	4	2	6	5	4,66666667	3	2
4	5	5	5	2	5	5	4,16666667	2	2
6	4	5	4	2	4	5	4	2	2
9	5	5	4	1	5	4	4,33333333	2	2
10	5	4	3	1	5	4	4,33333333	1	1
12	6	6	3	6	5	6	4,33333333	2	4
Ø	5,16666667	5	3,83333333	2,33333333	5	4,83333333	4,30555556	2	2,16666667

INV3	INV4	INV	REAL1	REAL2	REAL3	REAL4	REAL	PRES
4	5	3	1	5	5	3	3,5	3,72222222
4	6	3,5	1	6	4	2	3,25	3,63888889
2	3	3,25	2	4	5	2	3,25	3,5
4	4	3	2	4	4	2	3	3,44444444
5	2	2,25	1	5	5	0	2,75	3,11111111
3	5	4	2	5	5	1	3,25	3,86111111
3,66666667	4,16666667	3,16666667	1,5	4,83333333	4,66666667	1,66666667	3,16666667	3,5462963

SP = Spatial Presence = $((SP1+Sp4+SP5+G1)+(-1*SP2+6)+(-1*SP3+6)) / 6$

INV = Involvement = $((INV4+INV2)+(-1*INV1+6)+(-1*INV3+6))/4$

REAL = Experienced Realism = $(REAL1+REAL2+REAL3+REAL4)/4$

PRES = Overall Presence = $(SP+INV+REAL)/3$