

# Bachelorarbeit

Jonathan Wischhusen

**Gerätesteuerung durch ein holografisches Userinterface**

*Fakultät Technik und Informatik  
Studiendepartment Informatik*

*Faculty of Engineering and Computer  
Science  
Department of Computer Science*

Jonathan Wischhusen

**Gerätesteuerung durch ein holografisches Userinterface**

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Angewandte Informatik  
am Department Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Birgit Wendholt  
Zweitgutachter: Prof. Dr. Philipp Jenke

Eingereicht am: 23. Januar 2016

**Jonathan Wischhusen**

**Thema der Arbeit**

Gerätesteuerung durch ein holografisches Userinterface

**Stichworte**

mobile Mid-Air-Interaktion, virtuelle Mid-Air-Bedienelemente, Augmented Reality Interface, virtuelle Hand, adaptive Gerätesteuerung

**Kurzzusammenfassung**

Gegenstand dieser Bachelorarbeit ist die Konzeption und Realisierung einer adaptiven Gerätesteuerung, die sich in der Luft durch eine virtuelle Hand bedienen lässt. Das virtuelle Handmodell imitiert die Bewegungen der realen Hand des Benutzers und ermöglicht so die Interaktion mit einer im Raum dargestellten Benutzeroberfläche. Das Bedienkonzept wird anhand des Prototyps evaluiert.

**Jonathan Wischhusen**

**Title of the paper**

Device control through a holographic user interface

**Keywords**

mobile mid-air interaction, virtual mid-air controls, augmented reality interface, virtual hand, adaptive device control

**Abstract**

Subject of this thesis is the design and implementation of an adaptive device control that can be operated in mid-air by a virtual hand. The virtual hand model imitates the motion of the real hand of the user and enables interaction with a user interface shown in mid-air. The interaction concept is evaluated on the basis of the prototype.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Zielsetzung . . . . .	4
1.3	Gliederung . . . . .	4
<b>2</b>	<b>Vergleichbare Arbeiten</b>	<b>5</b>
2.1	Gliederung grundlegender Verfahren . . . . .	5
2.1.1	Freihändige Interaktion . . . . .	6
2.1.2	Immersive Augmented Reality . . . . .	6
2.2	Wissenschaftliche Projekte . . . . .	8
2.2.1	Invisibilia . . . . .	8
2.2.2	OmniTouch . . . . .	10
2.2.3	Ubii . . . . .	11
2.2.4	Remote-Touch . . . . .	13
2.2.5	RoomAlive . . . . .	14
2.2.6	Weitere interaktive Entwicklungen . . . . .	15
2.2.7	Zusammenfassung . . . . .	16
2.3	Kommerzielle Entwicklungen . . . . .	17
2.3.1	Moverio . . . . .	17
2.3.2	HoloLens . . . . .	18
2.3.3	Leap Motion . . . . .	18
2.3.4	Project Soli . . . . .	19
2.3.5	Weitere Entwicklungen . . . . .	19
2.3.6	Zusammenfassung . . . . .	20
2.4	Fazit . . . . .	21
<b>3</b>	<b>Systemkonzeption</b>	<b>22</b>
3.1	Anforderungsanalyse . . . . .	22
3.2	Wahl der Hardware . . . . .	24
3.3	Systemkomponenten . . . . .	25
3.3.1	Virtual Device . . . . .	25
3.3.2	Hand Tracking . . . . .	29
3.3.3	Virtual Hand . . . . .	32
3.3.4	User Interface . . . . .	34
3.3.5	Virtual Scene . . . . .	36
3.4	Implementierung . . . . .	37



3.5	Zusammenfassung . . . . .	38
<b>4</b>	<b>Evaluation</b>	<b>39</b>
4.1	Laufzeit Messung . . . . .	39
4.1.1	Latenz der virtuellen Hand . . . . .	40
4.1.2	Interaktionsbereitschaft . . . . .	40
4.1.3	Geräteaktion . . . . .	41
4.2	Benutzerstudie . . . . .	41
4.2.1	Teil I . . . . .	41
4.2.2	Teil II . . . . .	42
4.3	Fazit . . . . .	46
<b>5</b>	<b>Zusammenfassung</b>	<b>47</b>
5.1	Fazit . . . . .	48
5.2	Probleme . . . . .	49
5.3	Fortführung . . . . .	49

# Abbildungsverzeichnis

1.1	Reality Virtuality Continuum . . . . .	2
2.1	Invisibilia - Aufbau und Interaktion . . . . .	9
2.2	OmniTouch - Anwendungsbeispiele . . . . .	10
2.3	Interaktion mit Ubii . . . . .	11
2.4	Remote-Touch - Anwendungsfall . . . . .	13
2.5	RoomAlive - Anwendungsbeispiel und Aufbau . . . . .	14
2.6	Moverio PRO BT-2000 . . . . .	17
2.7	Microsoft HoloLens . . . . .	18
2.8	Leap Motion . . . . .	18
2.9	Project Soli Anwendungsfall . . . . .	19
3.1	Display-Sensor-Halterung . . . . .	24
3.2	Black-Box-Sicht der Systemkomponenten . . . . .	25
3.3	Virtual Device Komponente . . . . .	26
3.4	Aktivitätsdiagramm der DevieHandler-Komponente . . . . .	26
3.5	Klassendiagramm der ControllableDevice-Komponente . . . . .	29
3.6	Verteilungdiagramm der Systemkomponenten . . . . .	30
3.7	Aktivitätsdiagramm der Hand Tracking-Komponente . . . . .	30
3.8	Virutal Hand-Komponente . . . . .	33
3.9	Finger- und Kugel-Handmodell . . . . .	33
3.10	User Interface-Komponente . . . . .	35
3.11	Zustandsdiagramm der GuiButton-Komponente . . . . .	36
3.12	Aktivitätsdiagramm der Rendschleife . . . . .	36
3.13	Aktivieren der Benutzeroberfläche . . . . .	38
4.1	Übersicht der Reaktionszeitmessungen . . . . .	40
4.2	Auswertung der ersten Likert-Skala . . . . .	42
4.3	Auslösen einer Aktion mit dem Finger-Handmodell . . . . .	43
4.4	Übersicht der beobachteten Fehler . . . . .	44
4.5	Übersicht der beobachteten Zeit . . . . .	44
4.6	Auswertung der zweiten Likert-Skala . . . . .	45

# 1 Einleitung

Science-Fiction-Autoren haben uns schon oft durch ihre Visionen gezeigt, wie Menschen in Zukunft mit Computern und ihrer Umgebung interagieren.

In dem Film „Iron Man“ ist es dem Protagonisten unter anderem möglich, einzelne Teile seines Anzuges als Hologramm am eigenen Körper Probe zu tragen oder sich auf natürliche Art und Weise mit der künstlichen Intelligenz J.A.R.V.I.S.<sup>1</sup> zu unterhalten[Fav08].

Die Serie „Black Mirror“ zeigt andererseits, wie Menschen mit ihrer Umgebung fast ausschließlich durch Gesten inmitten der Luft interagieren. So werden nicht nur Türen durch einfache horizontale Handbewegung geöffnet, intelligente Badarmaturen erkennen ebenfalls die gewünschte Menge an Zahnpasta anhand des Ein bzw. Ausrollens des Zeigefingers[Bro11].

Der Film „Minority Report“ hingegen zeigt, wie der Protagonist mit speziellen Handschuhen vor einem futuristischen Bildschirm Videoaufnahmen analysieren kann, indem er komplexe Handgesten in der Luft ausführt[Spi02].

Solche Visionen sind oft nicht frei erfunden, sondern entstammen vielmehr aktuellen Trends in verschiedenen Forschungsgebieten. Im Mittelpunkt stehen dabei Ubiquitous Computing[Wei95] und Entwicklungen im Bereich Human Computer Interaction, wie multimodale Interaktion[JS07] und Augmented Reality[ZDB08].

## 1.1 Motivation

Auch wenn die Technologien der Science-Fiction-Autoren teilweise dem Status quo der Forschung vorraus sind, beschränkt sich die Interaktion zwischen Mensch und Computer

---

<sup>1</sup>Just A Rather Very Intelligent System

lange nicht mehr auf traditionelles Desktop Computing mit Maus und Tastatur.

Natural User Interfaces erlauben uns zunehmend, bekanntes Handlungswissen aus der realen Welt auf virtuelle Umgebungen durch Gesten, Mimik und Sprache zu übertragen. So ermöglicht der Kinect-Sensor<sup>2</sup> Spielern, ihren eigenen Körper als Controller zu benutzen, um beispielsweise ein virtuelles Lenkrad zu steuern, als hielten sie es zwischen ihren Fingern[Mica]. Natural User Interfaces (NUIs) erfreuen sich aufgrund der Möglichkeit, multimodal zu interagieren, in verschiedenen Gebieten zunehmend an Beliebtheit. Dabei setzen besonders künstlerische Projekte auf die neue Benutzererfahrung[TP15; Wik15; Vee13]. In der Medizin hingegen ermöglichen sie durch freihändige Gesten sterile Benutzerschnittstellen[Jal+13; Ebe+12]. Letzendlich haben NUIs mit der Verbreitung von Smartphones längst ihren Weg in unser alltägliches Leben gefunden und werden uns in Zukunft den Umgang mit elektronischen Geräten vereinfachen.

Obwohl uns Natural User Interfaces ein natürliches Gefühl für die Bedienung geben, haben sie alleine keine Auswirkung auf unser unmittelbares Umfeld. Mixed Reality-Umgebungen allerdings lassen die Grenzen zwischen realer und virtueller Welt ineinander verschwimmen[MK94], wodurch unser Umfeld manipulierbar wird.

Dabei genießt besonders Augmented Reality große Aufmerksamkeit in der Wissenschaft und in der Industrie. Wie Abbildung 1.1 zeigt, beschreibt Augmented Reality eine Teilmenge der Mixed Reality, die sehr realitätsnah ist. Die physische Welt wird mit virtuellen Informationen überlagert. Dies wird unterschiedlich erreicht, etwa durch Smartphones[Ull+12], Head-Mounted Displays[Micb; Ocu] oder Projektion[Vol15]. Dem Anwendungsbereich von Augmented Reality sind keine Grenzen gesetzt, da in nahezu jedem Bereich virtuelle Inhalte unterstützend eingesetzt werden können, wie bei Operationen[Nav+12] oder in der eigenen Küche[BLS05]. Mit Google Glass<sup>4</sup> wurde bereits eine alltagstaugliche Augmented Reality-Brille konzipiert.

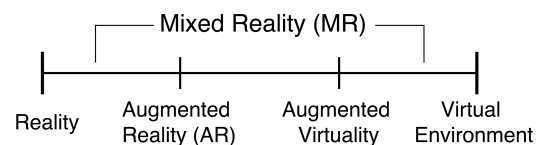


Abbildung 1.1: Reality Virtuality Continuum<sup>3</sup>

---

<sup>2</sup><http://www.xbox.com/de-DE/xbox-one/accessories/kinect-for-xbox-one>

<sup>3</sup>Bild wurde nachgestellt. Originalbild: Milgram und Kishino [MK94, S. 3]

<sup>4</sup><https://www.google.com/glass/start/>

Wissenschaftliche Arbeiten vereinen zunehmend immersive Bildschirme mit multi-modaler Interaktion im Hinblick auf ein natürlicheres Benutzererlebnis[Jon+14; Lea]. Ebenso verknüpft das i2e-Projekt<sup>5</sup> diese Entwicklungen und forscht an kollaborativer Konstruktion in einer verteilten Mixed Reality-Umgebung. Dabei sollen Benutzer durch physikbasierte Gesten, wie Drehen oder Anheben, virtuelle Objekte manipulieren und zusammensetzen können[I2E].

Durch das Internet of Things beschränken sich instrumentalisierte Umgebungen, die durch Sensoren oder eingebettete Systeme Informationen zur Verfügung stellen, nicht mehr auf einzelne wissenschaftliche Labore. Internet of Things (IoT) entspringt dem Gedanken hinter Ubiquitous Computing und beschreibt die Ausbreitung vernetzter Systeme in unserer Umgebung, die uns unmerklich das alltägliche Leben erleichtern sollen. So zeigen uns digitale Spiegel im Einzelhandel, wie uns Kleidungsstücke in anderen Farben stehen[Mem], während Smart Home-Applikationen unseren Energieverbrauch senken[HL10] oder Kühlschränke uns Kochrezepte anhand unserer Essgewohnheiten und im Kühlschrank befindlicher Lebensmittel vorschlagen[Xie+13]. Viele dieser Systeme haben aufgrund ihrer Größe, Lage oder Funktion oftmals keine direkte und für unterschiedliche Anwender individuelle Benutzerschnittstelle. In der Regel werden diese Systeme durch ein zentrales Terminal oder eine Fernbedienung bedient. Dadurch wird der Benutzer an einen Ort oder Gerät gebunden. Dies ist je nach System wenig praktikabel.

Augmented Reality und Natural User Interfaces eignen sich, um der zunehmenden Verwendung elektronischer Geräte in unserem Alltag eine geeignete und einheitliche Benutzerschnittstelle zu schaffen. Anstelle von Fernbedienungen und Steuerungseinheiten für einzelne Systeme können kontextabhängige Inhalte dort angezeigt werden, wo sie erwartet werden. Geräte würden kein physikalisches Interface mit Displays und Knöpfen mehr benötigen. Ein virtuelles Pendant könnte die gleiche Funktionalität bieten und zum Beispiel zusätzliche Informationen darstellen. Das Benutzerinterface könnte flexibel gestaltet werden und wäre für unterschiedliche Benutzer personalisierbar. Betrachtet man die derzeitigen Entwicklungen und die Möglichkeiten von Augmented Reality und Natural User Interfaces, liegt es nicht fern, dass diese grundlegend für neue Systemarchitekturen sein werden.

---

<sup>5</sup>Interactive Immersive Environments

## 1.2 Zielsetzung

Vor dem Hintergrund der in Abschnitt 1.1 geschilderten Entwicklungen setzt sich die Arbeit zum Ziel, prototypisch eine universelle Benutzerschnittstelle für Geräte zu entwickeln. Dabei wird die Vorstellung verfolgt, dass der Benutzer in seinem Alltag die Möglichkeit bekommt, kontextabhängige Informationen und Bedienelemente von Geräten vor sich im Raum zu sehen und diese mit dem Finger bedienen zu können. Das zu entwickelnde System wird im Folgenden als *HoloUI* bezeichnet.

*HoloUI* soll Bedienelemente und Informationen ähnlich einem Hologramm im Raum darstellen, beschränkt sich dabei jedoch auf aktuelle Augmented Reality-Technik. Das System soll als tragbares Gerät entwickelt werden und durch Tip-Gesten in der Luft interaktiv sein. Damit der Benutzer nicht in seinem Alltag eingeschränkt wird, soll *HoloUI* als mobile Lösung entworfen werden. Zudem soll es universell nutzbar sein und sich nicht auf ein bedienbares Gerät beschränken. Dazu müssen Objekte eindeutig identifiziert werden können und es muss eine universelle Schnittstelle bereitgestellt werden, über die verschiedene Geräte kontrollierbar werden.

Der Fokus der Arbeit liegt auf freihändiger Interaktion und weniger auf einer universellen Gerätesteuerung, weshalb sich die bedienbaren Geräte im Rahmen dieser Arbeit auf den VLC-Player beschränken, der durch seine Web API im Prinzip ein beliebiges vernetztes Gerät simuliert. Im Zuge dieser Arbeit wird nach Hardware- sowie Softwarelösungen für Visualisierung, Interaktion und Gerätesteuerung gesucht.

## 1.3 Gliederung

Die Arbeit setzt sich aus fünf Kapiteln zusammen. Kapitel 1 zeigt grundlegende Forschungstrends im Bereich Human Computer Interaction. Mit der Zielsetzung wird die Ausrichtung der Arbeit beschrieben. Relevante wissenschaftliche Entwicklungen und kommerzielle Produkte werden in Kapitel 2 vorgestellt. Die verwendeten Methoden und Geräte werden auf die Eignung für diese Arbeit hin untersucht. Kapitel 3 setzt sich mit der Realisierung des Zielsystems auseinander. Das Benutzererlebnis sowie die ausschlaggebenden Faktoren für dieses werden in Kapitel 4 beschrieben. Kapitel 5 fasst die Arbeit zusammen.

## 2 Vergleichbare Arbeiten

Card, Newell und Moran haben sich bereits in den frühen 80er Jahren der Human Computer Interaction angenommen [CNM83]. Moderne Technologien und leistungsfähigere Hardware wirken seither wie ein Katalysator auf den Forschungsbereich, weshalb es nicht verwundert, dass dieser bereits viele wissenschaftliche Arbeiten und kommerzielle Produkte umschließt. [JS07; ZDB08; Wex14]

Dieses Kapitel setzt sich mit den relevanten Entwicklungen aus Forschung und Industrie, deren Schwerpunkt immersive und freihändige Interaktion ist, auseinander. Die Entwicklungen sind in ihrem Zweck und ihrer Umsetzung zu vielseitig, als dass eine eindeutige Kategorisierung vorgenommen werden kann. Deshalb wird für eine bessere Vergleichbarkeit in Abschnitt 2.1 eine Unterteilung der grundlegenden Verfahren vorgenommen. In Abschnitt 2.2 werden daraufhin einige wissenschaftliche Projekte unter den Gesichtspunkten, die für das Ziel dieser Arbeit interessant sind, vorgestellt: *Interaktion*, *Visualisierung* sowie *Objekterkennung/-bedienung*. Um *HoloUI* realisieren zu können, wird in Abschnitt 2.3 nach geeigneter Hardware gesucht, die zum Aufbau des System verwendet werden kann. Dafür werden kommerzielle Entwicklungen im Hinblick auf die in Abschnitt 1.2 beschriebenen Anforderungen untersucht und bewertet. Abschnitt 2.4 schließt das Kapitel mit einem Fazit der Untersuchung ab.

### 2.1 Gliederung grundlegender Verfahren

Die Unterteilung bezieht sich auf die wesentlichen Bereiche, die sich aus der Zielsetzung in Abschnitt 1.2 ergeben. Dies sind zum einen Augmented Reality als visuelles Ausgabemedium und freihändige Interaktion in der Luft als Eingabemedium. Die Methoden, die in Abschnitt 2.1.1 und 2.1.2 beschrieben werden, sind grundlegende Realisierungsverfahren, auf denen sämtliche aktuellen Projekte aufbauen.

### **2.1.1 Freihändige Interaktion**

Es gibt zwei Herangehensweisen, um freihändige Aktionen und Gesten eines Benutzers in der Luft zu erfassen, ohne dass dieser ein physikalisches Objekt halten oder berühren muss. Entweder wird die Umgebung instrumentalisiert oder der Benutzer. Beide Herangehensweisen haben ihre Vor- und Nachteile und sind je nach angestrebtem Anwendungsbereich auszuwählen.

#### **Instrumentalisierte Umgebung**

Die Umgebung wird mit einem oder mehreren Sensoren ausgestattet. Diese messen und interpretieren in der Regel Tiefeninformationen der Szene. Je nach verwendeter Technik lassen sich so der gesamte Körper oder einzelne Gliedmaßen erfassen. Der Interaktionsbereich des Benutzers beschränkt sich dabei auf die präparierte Umgebung. Dieser Aufbau eignet sich besonders, wenn verschiedenen Benutzern die Interaktion ermöglicht werden soll. Ein beliebter Sensor für solche Umgebungen ist die Kinect.

#### **Instrumentalisierter Benutzer**

Wird der Benutzer hingegen instrumentalisiert, ist er in seiner Mobilität nicht an einen bestimmten Ort gebunden. Der Nachteil dieses Ansatzes ist, dass die Interaktion nur von einem Anwender ausgeht, der entsprechende Sensoren am Körper trägt. Zur Erfassung können neben Tiefensensoren auch Inertialsensoren oder jene, die elektrische Signale von Muskeln messen, eingesetzt werden, um an zusätzliche Informationen über die Bewegung des Benutzers zu gelangen.

### **2.1.2 Immersive Augmented Reality**

Die einfachste Realisierung einer Augmented Reality-Umgebung ist eine Kamera in Verbindung mit einem Bildschirm, auf dem der Videostream der Kamera samt kontextabhängiger virtueller Elemente dargestellt wird. Smartphones vereinen beide Elemente und sind dadurch oft das Mittel der Wahl. Diesem Aufbau fehlt allerdings eine entscheidene Eigenschaft: die Immersion. Hier helfen räumliche Projektion und Head-Mounted Displays weiter. Werden diese um Stereoskopie erweitert, erhält der Benutzer dank zusätzlicher Tiefeninformationen eine dreidimensionale Sicht auf die virtuellen Elemente. Die virtuelle Szene passt sich dadurch besser an die physische Welt an und die Immersion wird positiv beeinflusst.



### **Räumliche Projektion**

„Spatial Projection“ bezeichnet die Projektion von virtuellen Inhalten durch einen oder mehrere unterschiedlich ausgerichtete Beamer auf reale Objekte und Umgebungen. Durch die Projektion können mehrere Benutzer ohne weiteres die Augmented Reality-Umgebung wahrnehmen. Die Projektion reicht dabei von kleinen unscheinbaren Textinformationen bis hin zu einer kompletten Überlagerung der realen Umgebung durch eine virtuelle Umgebung[You]. Ein bekanntes Beispiel für eine großflächige Überlagerung durch Projektion ist RoomAlive[Jon+14]. Durch Ausrichtung der Beamer an die Blickrichtung zweier Benutzer erhalten beide eine verschiedene Sicht auf die virtuelle Szene. Die Immersion des Benutzers kann durch eine stereoskopische Sicht verbessert werden. Um dies innerhalb einer Spatial Augmented Reality zu ermöglichen, muss der Benutzer eine zusätzliche Brille tragen oder sich an einem bestimmten Punkt im Raum befinden.

### **Head Mounted Display**

Will man eine Augmented Reality-Umgebung durch ein Head-Mounted Display darstellen, gibt es zwei Optionen. Ein transparentes Display, ähnlich einer Brille, erlaubt dem Benutzer weiterhin, die reale Welt mit seinen Augen zu sehen. Das transparente Display stellt lediglich die zusätzlichen virtuellen Elemente dar. Ein lichtundurchlässiges Display schirmt hingegen die reale Umgebung vollständig ab. Der Benutzer sieht nur einen Videostream der Kamera von der realen Welt, in dem virtuelle Elemente eingefügt werden. Beide Methoden haben den Vorteil, dass jedem Auge ein separates Bild angezeigt werden kann, wodurch Stereoskopie bereits durch den Aufbau ermöglicht wird. Aktuelle transparente Head-Mounted Displays leiden an einem schmalen Field of View und der Tatsache, dass das menschliche Auge entweder die reale Welt oder das wenige Zentimeter entfernte reflektierende Glas des Displays fokussiert, wodurch das jeweils andere verschwommen wahrgenommen wird. Das lichtundurchlässige Pendant kämpft mit zu niedriger Auflösung und Unwohlsein bei zu langer Benutzung. Obwohl die Auflösung hoch ist, wird sie durch den geringen Abstand zwischen Display und Auge sehr viel geringer wahrgenommen. Head-Mounted Displays erlauben Mobilität und der Benutzer sieht, da das Display kontinuierlich vor seinem Auge ist, die virtuellen Elemente ungeachtet seiner Blickrichtung.

## Stereoskopische Umgebung

In stereoskopischen Umgebungen wird dem Benutzer durch individuelle Bilder für das linke und rechte Auge eine bessere Tiefenwahrnehmung geboten. Dabei erfolgt die Sicht auf die virtuelle Szene aus zwei verschiedenen virtuellen Blickpunkten, deren optische Achsen sich in einer Konvergenzebene schneiden. Virtuelle Inhalte, die vor der Ebene liegen, vermitteln dem Benutzer den Anschein, als würden sie sich räumlich näher befinden (negative Parallaxe). Analog dazu erscheinen Elemente, die sich hinter der Ebenen befinden, weiter entfernt (positive Parallaxe). Stereoskopie imitiert das Verfahren, mit dem wir die reale Welt dreidimensional sehen, und liefert uns aus der virtuellen Umgebung zusätzliche Tiefeninformationen, die unserem Gehirn helfen, die Inhalte räumlicher wahrnehmen und lokalisieren zu können.

## 2.2 Wissenschaftliche Projekte

Wie Abschnitt 1.1 zeigt, sind viele wissenschaftliche Arbeiten dem Bereich Human Computer Interaction zuzuordnen. Im Folgenden werden wissenschaftliche Projekte vorgestellt, die die relevante Aspekte behandeln: *Interaktion*, *Visualisierung* sowie *Objekterkennung/-bedienung*.

### 2.2.1 Invisibilia

Invisibilia wurde von Benavides, Amores und Maes [BAM15] mit dem Ziel entwickelt, unsichtbare Informationen durch eine Augmented Reality-Umgebung sichtbar und interaktiv werden zu lassen. Ziel der Arbeit war es, das Verständnis physikalischer Phänomene durch Beobachten leichter zugänglich zu machen.

**Interaktion** Ein am Display angebrachter Infrarotsensor erfasst „Pinching“- und „Stretching“-Gesten. Die Benutzereingabe beschreibt den Zustand zwischen einem minimalen und maximalen Wert, der durch horizontales bzw. vertikales Zusammen- und Auseinanderführen des Zeigefingers und Daumens abgebildet wird (Abbildung 2.1, links). Das System erkennt die Start- und Endhaltung der Finger und kann so den gewünschten Wert errechnen. Die Gesten werden interpretiert, sobald das System das bedienbare Gerät erkannt hat.



Abbildung 2.1: Interaktion durch Invisibilia(links), Aufbau(rechts)<sup>2</sup>

**Visualisierung** Durch ein lichtundurchlässiges Head-Mounted Display<sup>1</sup> erhält der Benutzer Einblick in eine Augmented Reality-Umgebung. Die angezeigten virtuellen Elemente beschränken sich auf eine Visualisierung des Audiosignals, das vor das bedienbare Gerät gelegt ist. Da nur der Videostream der eingebauten Kamera verwendet wird, sieht der Benutzer im Grunde das selbe Bild auf jedem Auge, jedoch leicht verschoben.

**Objekterkennung/-bedienung** Das Radio wird über ein markantes Kreismuster erkannt. Durch die Verzerrung des Musters wird die Größe und relative Ausrichtung des Radios zum Betrachter errechnet. Das Gerät kommuniziert über WLAN und nimmt so Befehle entgegen. Die genaue Kommunikation ist nicht beschrieben.

Im Gegensatz zu dieser Arbeit erhält der Benutzer nur die Möglichkeit, ein einzelnes Gerät zu bedienen. Ebenso wurde die Objekterkennung mit einer Mustervorlage umgesetzt, die nicht für die Erkennung einer Vielzahl von verschiedenen Geräten geeignet ist. Durch das geschlossene Display wird der Eindruck, sich in der realen Welt zu befinden, gemindert.

Der Aufbau von Invisibilia, der das Erfassen der Finger im Sichtbereich des Benutzers erlaubt, ist für die Arbeit interessant. Ebenso birgt die Erfassung durch Tiefeninformationen viele Möglichkeiten, in einer virtuellen Umgebung zu interagieren.

---

<sup>1</sup>Smartphone und Adapter

<sup>2</sup>Quelle: Benavides, Amores und Maes [BAM15, S. 2]

### 2.2.2 OmniTouch

Mit OmniTouch haben Harrison, Benko und Wilson [HBW11] ein System entwickelt, das beliebige Oberflächen interaktiv werden lässt. Der Benutzer trägt dazu eine Kombination aus Pico-Projektor und PrimeSense-Sensor<sup>3</sup> auf der Schulter (Abbildung 2.2, links).

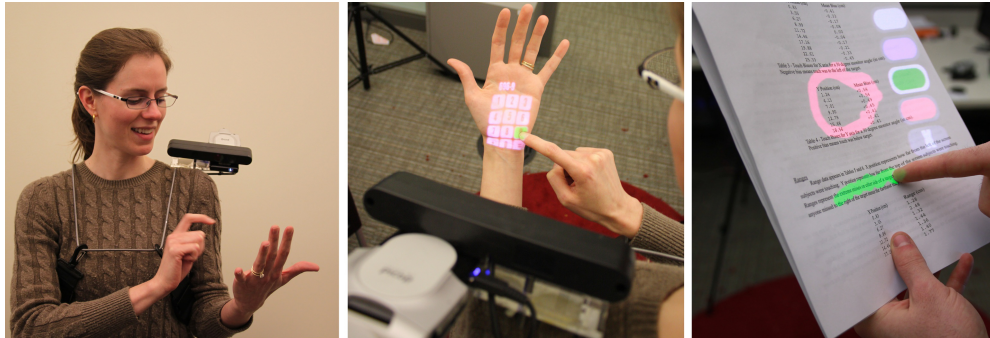


Abbildung 2.2: Anwendungsbeispiele von OmniTouch<sup>4</sup>

**Visualisierung** Der Projektor ermöglicht eine mobile Spatial Augmented Reality-Umgebung. Im Sichtfeld des Benutzers werden zweidimensionale virtuelle Elemente auf beliebige Oberflächen, die sich unmittelbar vor dem Benutzer befinden, projiziert. Ist die Oberfläche beweglich, wie die in Abbildung 2.2 mittig dargestellte Hand, können die virtuellen Elemente an „Lock Points“ der Oberfläche gepinnt werden, so dass projizierte Elemente weiterhin auf der Handfläche dargestellt werden. Liegt eine Oberfläche nicht orthogonal zur Projektionsrichtung, wird die Ebene, in der die projizierten Elemente liegen, passend in der virtuellen Umgebung gedreht, so dass die Elemente ohne Verzerrung auf dem physischen Objekt dargestellt werden. Die anzeigbaren virtuellen Elemente umfassen sämtliche 2D-Interfaces wie Menüs, Uhren, farbige Markierungen und Bilder, sind jedoch fester Bestandteil verschiedener Demos und werden nicht kontextabhängig angezeigt.

**Interaktion** Jede Oberfläche, die sich im Arbeitsbereich der PrimeSense-Kamera befindet, kann als Eingabemedium genutzt werden. OmniTouch interpretiert dazu die Tiefeninformationen und sucht nach Fingern und weiter hinten gelegenen Oberflächen. Nähert sich ein Finger einer Oberfläche, werden „Hovering“- und „Click“-Gesten erkannt.

---

<sup>3</sup><https://en.wikipedia.org/wiki/PrimeSense#Products>

<sup>4</sup>Quelle: Harrison, Benko und Wilson [HBW11, S. 1]

Ebenso wurde eine Demo umgesetzt, die typische Touch-Gesten wie „Drag“ und „Zoom“ verwendet.

**Objekterkennung/-bedienung** OmniTouch ist nicht in der Lage, Objekttypen zu klassifizieren. Oberflächen und Finger werden anhand einfacher mathematischer Annahmen erkannt.

OmniTouch stellt ein Proof-of-Concept für mobile Interaktion dar. Die Userinterfaces setzen sich mangels Objekterkennung nicht aus kontextabhängigen Bedienelementen zusammen. Demos zeigen durch verschiedenen Interfaces lediglich die Interaktionsmöglichkeiten in verschiedenen gestellten Szenarien.

Obwohl OmniTouch ein interessantes Konzept für die mobile Interaktion präsentiert, korreliert die Tatsache, dass zur Darstellung von virtuellen Inhalten eine Oberfläche in unmittelbarer Nähe notwendig ist, nicht mit der Idee der freien Projektion im Raum, auf die sich diese Arbeit stützt. Desweiteren wird auch keine kontextabhängige Visualisierung und Interaktion unterstützt.

### 2.2.3 Ubii

Huang, Li und Hui [HLH15] zeigen mit Ubii<sup>5</sup>, wie die Wahrnehmung und Interaktion mit digitalen Inhalten auf die physische Welt erweitert werden kann. Ein vernetztes System aus Druckern, Bildschirmen und Projektoren bildet ein Szenario, in dem der Benutzer Dateien zwischen Geräten bewegen und drucken kann.



Abbildung 2.3: Interaktion mit Ubii(links), Aufbau(rechts)<sup>6</sup>

---

<sup>5</sup>Ubiquitous interface and interaction

<sup>6</sup>Quelle: Huang, Li und Hui [HLH15, S. 2,4]

**Visualisierung** Mit Google Glass wurde eine klassische Augmented Reality-Umgebung umgesetzt. Der Benutzer sieht einen überlagerten Videostream der eingebauten Kamera auf dem Displayprisma vor dem rechten Auge. Virtuelle Inhalte, wie Dateisymbole und Bedienelemente einzelner Geräte, werden nahe den Referenzmarkierungen im Video in Form eines Ringmenüs eingeblendet und sind an die Ausrichtung des Gerätes angepasst. Der Benutzer sieht die virtuellen Inhalte nur für die Geräte, die sich im Sichtfeld der Kamera befinden. Das Ringmenü ist in aktive und inaktive Elemente unterteilt und wächst mit zunehmenden Dateien.

**Interaktion** Es werden fünf natürliche Fingergesten in der Luft interpretiert. Als Grundlage dienen zweidimensionale Farbbilder der Kamera, anhand welcher die Gesten bestimmt werden. Der verwendete Algorithmus sucht dabei nach einer initialen „Pinch“-Geste, auf der alle weiteren Gesten aufbauen. Will der Benutzer eine Aktion auslösen, muss die initiale Geste in der Nähe des virtuellen Elements ausgeführt werden. Um eine Datei zu drucken, muss diese auf das Druckgerät gezogen werden. Die Interaktion beschränkt sich auf die Menünavigation und das Bewegen von Dateien zwischen Systemen.

**Objekterkennung/-bedienung** Durch QR-Codes werden Objekte identifiziert, die im Blickfeld des Benutzers liegen. Der Code enthält die IP-Adresse und den Objekttyp des physikalischen Gerätes. Diese Informationen werden genutzt, um Befehle des Benutzers an das entsprechende Gerät zu senden. Die möglichen Befehle beschränken sich dabei auf das Kopieren, Anzeigen und Drucken einer Datei.

Ubii mangelt es an Tiefeninformationen. Dadurch beschränkt sich die intuitive freihändige Interaktion auf das einfache Verschieben von Dateien zwischen Geräten durch „Pinch“-Gesten. Die Selektion von virtuellen Elementen durch „einfaches“ Antippen durch den Finger ist nicht möglich, wie es diese Arbeit anstrebt. Ebenso wird die Bedienbarkeit virtueller Menüs, deren Alignment dem physikalischen Gerät angepasst ist, erschwert, sobald der Benutzer in einem spitzeren Winkel auf das Gerät schaut. Diese Tatsache unterstützt im Gegensatz zu dieser Arbeit nicht das Konzept einer einfachen Bedienung.

Auf der anderen Seite bietet Ubii ein Geräteinterface, über das verschiedene Geräte angesprochen werden können. Die Konfigurierbarkeit beschränkt sich dabei allerdings

auf IP Adresse und Gerätetyp. Das Konzept, Daten aus Makern auszulesen, eignet sich für diese Arbeit.

### 2.2.4 Remote-Touch

Ullah u. a. [Ull+12] stellen mit Remote-Touch ein System vor, das älteren Menschen und jenen mit physischen Einschränkungen helfen soll, tägliche Aufgaben zu bewältigen, wie Lichtschalter zu bedienen oder Fenster zu öffnen. Dabei wird dem Benutzer die Möglichkeit gegeben, innerhalb einer intelligenten Umgebung reale Objekte aus der Entfernung über ein Smartphone zu bedienen.



Abbildung 2.4: An-/Ausschalten einer entfernten Lampe mit Remote-Touch<sup>7</sup>

**Visualisierung** Durch ein Smartphone wird eine klassische Augmented Reality-Umgebung realisiert. Sobald der Benutzer dieses auf ein bedienbares Objekt mit einem QR-Code richtet, werden entsprechende Bedienelemente über den Videostream gelegt, siehe Abbildung 2.4. Diese sind allerdings nicht nach dem Objekt ausgerichtet, sondern an dem Bildschirm des Smartphones. Dies erlaubt die einwandfreie Bedienung immer noch dann, wenn das Objekt gerade eben am Rande des Videostreams erkannt wird.

**Interaktion** Die Interaktion beschränkt sich auf das einfache Auslösen einer Tip-Geste mit einem Finger auf dem Touchscreen. Das ausgewählte Bedienelement wird aktiviert und löst die entsprechende Aktion aus.

**Objekterkennung/-bedienung** Das System erkennt interaktive Gegenstände anhand von QR-Codes. Durch X10-Module können Objekte gesteuert werden, die selbst über

---

<sup>7</sup>Quelle: Ullah u. a. [Ull+12, S. 1]

keine Netzwerkschnittstelle verfügen. Ein Web-Service verwaltet alle Gerätezustände und Optionen, die einem Benutzer zur Verfügung stehen. Wird ein QR-Code erkannt, werden entsprechende Optionen geladen und dem Benutzer durch visuelle Bedienelemente angezeigt.

Remote-Touch setzt im Gegensatz zu dieser Arbeit auf ein wenig immersives Augmented Reality-Erlebnis. Der Fokus liegt auf einer einfachen Bedienbarkeit, indem bereits vorhandene Komponenten optimal genutzt werden sollen.

Die visuelle Objekterkennung durch QR-Codes bleibt jedoch erwähnenswert für diese Arbeit. Ebenfalls ist die Steuerung von nicht vernetzten Geräten für eine mögliche Fortführung der Arbeit sinnvoll. Am interessantesten ist jedoch die Ausrichtung der Bedienelemente am Bildschirm, um eine einfache Bedienung zu ermöglichen. Dies fügt sich gut in das Konzept der Arbeit ein.

### 2.2.5 RoomAlive

RoomAlive ist ein Forschungsprojekt von Microsoft, in dem Jones u. a. [Jon+14] eine virtuelle Umgebung über einen physikalischen Raum legen. Das Projekt wird mit Hinblick auf ein neuartiges Spielerlebnis entwickelt, in dem Spieler sich im wahrsten Sinne des Wortes „inmitten“ des Spiels wiederfinden sollen.

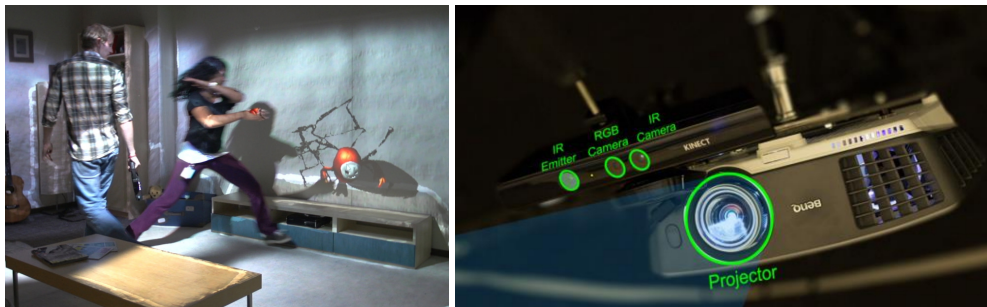


Abbildung 2.5: Interaktion mit RoomAlive(links), Aufbau(rechts)<sup>8</sup>

**Visualisierung** Das Projekt konstruiert durch Projections Mapping eine Spatial Augmented Reality. Mehrere Beamer leuchten dazu einen Raum aus und sind dadurch in der Lage, eine komplette virtuelle Szene über die physikalische Umgebung zu legen, siehe

---

<sup>8</sup>Quelle: Jones u. a. [Jon+14, S. 1-2]



Abbildung 2.5 rechts. Nachdem RoomAlive sich selbst kalibriert hat, werden Tiefeninformationen der Umgebung gesammelt und zusammengesetzt. Dadurch wird die virtuelle Umgebung an die physikalischen Gegebenheiten des Raums angepasst. Virtuelle Figuren können so beispielsweise einen Schrank erklimmen oder über die Wohnzimmergarnitur laufen.

**Interaktion** Mehrere an Beamern montierte Kinect-Sensoren erfassen den Körper eines Spielers. Dadurch können Berührungen und Körperbewegungen relativ zur Umgebung erkannt werden. Der Spieler wird durch „Proxy Partikel“ in der virtuellen Umgebung dargestellt, wodurch physikalisch realistische Rückschlüsse zwischen den ausgeführten Gesten und virtuellen Figuren gezogen werden können.

**Objekterkennung/-bedienung** RoomAlive realisiert keine explizite Objekterkennung und -bedienung. Die Interaktionen wirken sich lediglich auf die virtuelle Szene aus.

Räumliche Projektion, wie sie in RoomAlive verwendet wird, ist nicht für *HoloUI* geeignet. Durch die fest installierten Beamer und Sensoren geht die in dieser Arbeit angestrebte Mobilität gänzlich verloren. Außerdem setzt die Methode grundlegend eine Oberfläche voraus, die nicht immer gegeben ist.

Obwohl Microsoft mit RoomAlive eine andere Richtung als diese Arbeit einschlägt, kann die Bewegungserfassung durch Kinect-Sensoren durchaus für den Anwendungsfall der Arbeit in Betracht gezogen werden.

### 2.2.6 Weitere interaktive Entwicklungen

Die Bewegungserfassung der Benutzer beschränkt sich nicht ausschließlich auf optische Verfahren, wie bisher gezeigt. Pu u. a. [Pu+13] erkennen mit WiSee Benutzergesten über elektromagnetische Wellen von WLAN-Netzen. Dazu wird die Umgebung mit Empfängern ausgestattet. Cohn u. a. [Coh+11] verfolgen einen ähnlichen Ansatz, bei dem allerdings der Benutzer instrumentalisiert wird und bereits das Rauschen in der Nähe befindlicher elektrischer Geräte und Steckdosen ausreicht.

Ein weiterer Interaktionsansatz in Augmented Reality-Umgebungen ist die Interpretation des Auges. Lee u. a. [Lee+14] haben ein System entwickelt, das die Pupille verfolgt

und durch Zwinkern und Blickrichtung dem Benutzer Interaktion ermöglicht.

Die beschriebenen weiteren Entwicklungen zeigen zwar alternative Interaktionsformen, sind für diese Arbeit allerdings uninteressant, da sie in ihren Anwendungsbereichen zu speziell sind. Gestenerkennung durch elektromagnetische Wellen erlaubt nur grobe Aussagen über die tatsächliche Geste und es können nur Ganzkörpergesten erfasst werden. Andererseits bietet das Konzept, nur mithilfe des Auges zu interagieren, zu wenig Spielraum für die angestrebten Funktionalitäten der Arbeit.

### 2.2.7 Zusammenfassung

Durch diese Arbeit soll mit *HoloUI* ein mobiles System entstehen, das dem Benutzer bei Bedarf Geräte bedienen lässt, indem ein virtuelles Menü in der Luft eingeblendet wird und sich mit den Fingern bedienen lässt.

In Abschnitt 2.2 wurden dazu wissenschaftliche Arbeiten vorgestellt, die verschiedene relevante Konzepte zeigen. Während in Abschnitt 2.2.1 und 2.2.3 ein überlagerter Videostream der realen Welt auf einem Head-Mounted Display angezeigt wurde, wird in Abschnitt 2.2.2 ausschließlich der virtuelle Inhalt projiziert. Um den Benutzer weder an Oberflächen zu binden, noch ihm einen überlagerten Videostream anzuzeigen, der die Wahrnehmung der Realität verändert, sollten beide Konzepte kombiniert eingesetzt werden, sodass ausschließlich virtuelle Inhalte auf einem transparenten Head-Mounted Display dargestellt werden.

Um Gesten und Finger zu erfassen, wurde in Abschnitt 2.2.1 und 2.2.2 ein Tiefensensor verwendet. Dieser erlaubt einen größeren Gestaltungsfreiraum für das Benutzerinterface, da so Restriktionen wie eine notwendige Initialgeste in Abschnitt 2.2.3 vermieden werden. Generell ist die räumliche Instrumentalisierung, wie sie in Abschnitt 2.2.5 zur Gestenerfassung genutzt wird, zu vermeiden, um Mobilität zu gewährleisten.

Mit Abschnitt 2.2.4 wurde ein geeigneter Ansatz zur Ausrichtung des Menüs in der virtuellen Umgebung gezeigt. Durch orthogonales Ausrichten der Bedienelemente zur Sicht des Benutzers am Bildschirm bleiben die Bedienelemente gut erreichbar.

Zusammenfassend lässt sich ebenfalls sagen, dass Objektidentifikation durch eine Referenzmarkierung, wie in Abschnitt 2.2.3 gezeigt, für diese Arbeit eine interessante

Methode darstellt, die das Potential hat, gerätespezifische Daten wie IP-Adresse oder Gerätetyp bereitzustellen.

### 2.3 Kommerzielle Entwicklungen

Bisher wurde keine allgemeine Lösung gefunden, die den Anforderungen in Kapitel 1 genügt. Partielle Lösungen für einzelne Aspekte der Interaktion und Visualisierung wurden im vorherigen Abschnitt erläutert. Dieser Abschnitt befasst sich mit möglichen Geräten, die als Ausgangsbasis für *HoloUI* in Frage kommen. Dazu werden aktuelle Entwicklungen im kommerziellen Bereich im Hinblick auf die Interaktion in der Luft und der Darstellung von Augmented Reality-Umgebungen vorgestellt.

#### 2.3.1 Moverio

Epson hat mit der Moverio-Reihe eine leistungsstarke Augmented Reality-Brille entwickelt. Die aktuelle BT-200<sup>10</sup> bietet neben WLAN, GPS und Beschleunigungssensoren Stereoskopie und eine vorderseitige Kamera. Ein Android-Device bietet über ein Touchpad Eingabemöglichkeiten, an welches die Brille als Bildschirm angeschlossen ist. Dadurch ist die Brille für den mobilen Gebrauch geeignet. Das Blickfeld des Displays ist mit 23 Grad bemessen.



Abbildung 2.6: Moverio PRO BT-2000<sup>9</sup>

Durch Android bietet Epson eine flexible Entwicklungsumgebung, in der das standardmäßige Android-SDK in der Version 4.0.4 verwenden werden kann. Die für den industriellen Sektor entwickelte Moverio PRO BT-2000<sup>11</sup> verfügt zusätzlich über einen Tiefensensor, durch den die räumliche Positionierung von Objekten relativ zueinander erfasst werden kann.

Die BT-2000 wäre für diese Arbeit durchaus empfehlenswert, da sie die allgemeinen Anforderungen aus Abschnitt 1.2 erfüllen würde. Sie war zu Beginn dieser Arbeit aber

---

<sup>9</sup>Quelle: [https://macnn.mnmcnd.com/article\\_images/1435062525-md-150623\\_h.jpg](https://macnn.mnmcnd.com/article_images/1435062525-md-150623_h.jpg)

<sup>10</sup><http://www.epson.de/de/de/viewcon/corporatesite/products/mainunits/overview/12411>

<sup>11</sup><https://www.epson.de/de/de/viewcon/corporatesite/products/mainunits/overview/16149>

nicht in Europa erhältlich. Allerdings bleibt die verfügbare BT-200 ein interessanter Kandidat für die Visualisierung der Augmented Reality Umgebung.

### 2.3.2 HoloLens

Ähnlich der Moverio entwickelt Microsoft mit der HoloLens<sup>13</sup> ebenfalls eine transparente Augmented Reality-Brille. HoloLens soll mit umfangreichen Sensoren ausgestattet sein. Ebenfalls soll sie in der Lage sein, Finger zu erkennen und Gesten zu interpretieren. Microsoft bewirbt die freihändige Interaktion in der Luft, so wie es diese Arbeit es zum Ziel hat.



Abbildung 2.7: Microsoft HoloLens<sup>12</sup>

HoloLens wäre ein perfektes Gerät, um *HoloUI* darauf aufzubauen, da die Kriterien der Visualisierung und der Interaktion erfüllt sind. Da sich die Brille jedoch in der Entwicklungsphase befindet, kann sie für diese Arbeit nicht genutzt werden.

### 2.3.3 Leap Motion

Mit Leap Motion<sup>15</sup> hat ein robuster Infrarot-Sensor seinen Weg in den Verbrauchermarkt gefunden. Leap Motion erkennt und interpretiert Hände und Finger im Nahbereich. Es existieren viele SDKs in verschiedenen Programmiersprachen, die sämtliche Informationen bereitstellen. Der Sensor selber stellt eine API über einen Websocket zu Verfügung, über die direkt Daten des Sensors bezogen werden können. Die API bietet bereits eine Segmentierung der einzelnen Fingerknochen und Gelenke sowie Position und Dimensionen der Hand



Abbildung 2.8: Leap Motion<sup>14</sup>

---

<sup>12</sup>Quelle: <https://compass-ssl.surface.com/assets/13/b5/13b57736-9875-4332-af9f-dca0accbc331.jpg>

<sup>13</sup><https://www.microsoft.com/microsoft-hololens/en-us>

<sup>14</sup>Quelle: <http://116.255.173.144/uploads/jpg/2014/08/13/9/3KAb9bGbaY.jpg>

<sup>15</sup><https://www.leapmotion.com/>

und der Finger. Die Informationen werden als JSON-Objekte durch die API bereitgestellt.

Der Sensor ist für diese Arbeit sehr geeignet, da sein Arbeitsbereich und die verfügbaren SDKs alle Anforderungen an die Interaktion ermöglichen.

### 2.3.4 Project Soli

Google stellt mit Project Soli<sup>17</sup> einen auf Radar basierenden Chip vor, der kleinste Fingerbewegungen erfassen kann. Das Projekt befindet sich inmitten der Forschung, ist aber dennoch wegen seiner Möglichkeiten und Funktionsweise sehr interessant. Der Sensor soll vor allem in Wearables Anwendung finden und die Hand selbst als Eingabeinterface definieren.



Abbildung 2.9: Project Soli Anwendungsfall<sup>16</sup>

Durch Streichen über den Zeigefinger können beispielsweise virtuelle Regler eingestellt werden oder durch Reiben von Zeigefinger und Daumen durch Listen gescrollt werden, siehe Abbildung 2.9. Durch die kleine Bauart ist der Chip ideal für unscheinbare eingebettete Systeme und wäre auch für *HoloUI* denkbar. Zum Zeitpunkt dieser Arbeit gibt es keine näheren Informationen über die konkrete Fingererfassung, weshalb eine abschließende Bewertung nicht möglich ist.

### 2.3.5 Weitere Entwicklungen

Wie Project Soli und Leap Motion beschränken sich viele kommerzielle Produkte zur Zeit entweder auf die Eingabe oder auf die Ausgabe.

Der Markt bietet geschlossene Head-Mounted Displays, wie die Oculus Rift<sup>18</sup>, die sich jedoch wegen der notwendigen Darstellung eines Videostreams nicht eignet. Ebenso kann Google Glass nicht verwendet werden, da das kleine Display die Augmented Realität

---

<sup>16</sup>Quelle: <http://i.huffpost.com/gen/3019428/images/o-SOLI-facebook.jpg>

<sup>17</sup><https://www.google.com/atap/project-soli/>

<sup>18</sup><https://www.oculus.com/en-us/>

ty-Umgebung nicht ausreichend darstellen kann.

Microsoft wollte mit dem Kinect-Sensor den Spielemarkt revolutionieren. Der Sensor ist wegen seiner günstigen Anschaffung in wissenschaftlichen Projekten sehr beliebt. Das SDK, das von Microsoft bereitgestellt wird, beschränkt sich auf die Interpretation der Körperbewegungen und kann dadurch nicht für diese Arbeit eingesetzt werden. Es gibt bereits Anstrengungen, Finger über die Kinect zu erfassen[Sha+15], allerdings wurde noch keine Software veröffentlicht.

Myo<sup>19</sup> und Fin<sup>20</sup> sind zwei weitere Produkte, die dem Benutzer freihändige Interaktionsmöglichkeiten bieten. Myo setzt dabei auf Registrierung einzelner Muskelbewegungen im Arm, während Fin als Ring am Finger getragen werden kann. Beide Sensoren beschränken sich auf ein kleines Gesten-Repertoire und bieten keine Erfassung der Hand oder der Finger, wodurch sie ungeeignet sind.

Mit Magic Leap<sup>21</sup> befindet sich ein weiteres kommerzielles Produkt in der Entwicklung. Das Gerät soll durch Projektion direkt auf die Retina des Auges ein natürliches Field of View bieten und nicht den bisherigen technischen Beschränkungen unterliegen. Zudem soll das Gerät über einen Tiefensensor verfügen, um Hände und Finger zu erfassen, ähnlich der HoloLens. Über das Projekt wurden bisher nur wenige Informationen veröffentlicht.

### 2.3.6 Zusammenfassung

In Abschnitt 2.3 wurden kommerzielle Entwicklungen im Hinblick auf die Verwendung für *HoloUI* vorgestellt. Dabei stellte sich heraus, dass Geräte wie die HoloLens in Abschnitt 2.3.2 oder die Magic Leap sich fast gänzlich mit der Zielsetzung dieser Arbeit decken. Da von diesen Geräten bisher nur eine Ankündigung bekannt ist, können sie nicht für *HoloUI* herangezogen werden.

Hingegen eignen sich die Geräte aus Abschnitt 2.3.1 und 2.3.3: Moverio BT-200 sowie Leap Motion. Jedes ermöglicht für sich die geforderten Funktionalitäten. Durch entwicklerfreundliche Umgebungen wie Android und zahlreiche SDKs sind diese als Hardware-Grundlage geeignet.

---

<sup>19</sup><https://www.myo.com/>

<sup>20</sup><https://www.indiegogo.com/projects/fin-wearable-ring/>

<sup>21</sup><http://www.magicleap.com/#/home>

Die Sensoren, die in Abschnitt 2.3.5 gezeigt wurden, können aufgrund ihrer speziellen Anwendungsbereiche unbeachtet bleiben.

### 2.4 Fazit

In diesem Kapitel wurden wissenschaftliche sowie kommerzielle Projekte in Bezug auf Human Computer Interaction vorgestellt und mit Blick auf die Ziele dieser Arbeit untersucht. Es hat sich herausgestellt, dass es bereits viele Anstrengungen in der Industrie gibt, freihändige Interaktion in der Luft zu ermöglichen. Es sind ebenfalls bereits Geräte angekündigt worden, die eine mobile und interaktive Augmented Reality-Umgebung realisieren. Um *HoloUI* umsetzen zu können, werden vorläufig, wie in Abschnitt 2.3.6, die Moverio BT-200 und der Leap Motion Sensor erwogen.

Die wissenschaftlichen Arbeiten in diesem Bereich sind breit gefächert. Aus Abschnitt 2.2 geht hervor, dass Objektidentifikation durch optische Erfassung von Referenzmarkierungen empfehlenswert ist. Ebenso ist eine Ausrichtung des Userinterfaces am Display für die Zielsetzung erstrebenswerter als ein objektzentriertes Alignment, damit der Benutzer nicht das Gerät fokussieren muss, um Bedienelemente auszuwählen.

## 3 Systemkonzeption

Dieses Kapitel setzt sich mit der Realisierung des Zielsystems auseinander. In Abschnitt 3.1 werden die Anforderungen an *HoloUI* konkretisiert. Dabei werden funktionale und nicht-funktionale Anforderungen beschrieben. Darauf beziehend befasst sich Abschnitt 3.2 mit der Wahl der Hardware, die in Kapitel 2 erwogen wurde und als Grundlage dieser Arbeit dienen soll. Es folgt die Umsetzung von *HoloUI* in Abschnitt 3.3 gemäß den Anforderungen. Dabei werden einzelne Komponenten nacheinander vorgestellt. Rahmendetails der Implementation werden in Abschnitt 3.4 erläutert.

### 3.1 Anforderungsanalyse

Im Folgenden werden die Anforderungen der einzelnen Teilbereiche des Prototypen beschrieben. Als Grundlage dienen dabei die in Abschnitt 1.2 erwähnten Funktionen. Die Anforderungen richten sich an das zu entwickelnde System und sind nicht explizit als Hard- oder Softwareanforderungen deklariert.

#### **Userinterface**

Der Prototyp soll unscheinbar den Benutzer in seinem Alltag unterstützen. Um dies zu gewährleisten, muss eine Überschwemmung mit virtuellen Inhalten vermieden werden. Virtuelle Elemente sollen deshalb durch eine Handgeste aktiviert werden können, sobald sich der Benutzer in Reichweite eines bedienbaren Objektes befindet. Das Userinterface soll dem Benutzer Geräteinformationen als Text und Bedienelemente als Buttons anzeigen können. Diese sollen der Hand des Benutzers während der Aktivierungsgeste in einem geeigneten Abstand folgen. Sobald die Hand stillgehalten wird, sollen die virtuellen Elemente in der Luft „einrasten“. Danach kann der Benutzer die Bedienelemente durch Berühren mit der Fingerspitze aktivieren. Dabei sollen die Buttons reale Schalter imitieren und nur ein Mal auslösen, wenn man sie gedrückt hält. Gedrückte Buttons sollen dem Benutzer kenntlich gemacht werden, ebenso müssen Buttons unterschiedliche Zustände annehmen können.



#### **Fingererfassung**

Die Interaktion basiert gänzlich auf der Position und Bewegung der Finger. Dazu ist es notwendig, die gesamte Hand zu erfassen und zu verfolgen. Es gilt, die genaue Position einzelner Finger und des Handballens zu ermitteln. Die Finger sollen in einem geeigneten Bereich vor dem Gesicht des Benutzers erkannt werden. Es ist für diese Arbeit wichtig, dass die Verfolgung und Berechnung wenig Zeit beansprucht. Zudem muss sie robust sein und soll in unterschiedlichen Lichtverhältnissen zuverlässig funktionieren.

#### **Geräteidentifikation**

Der Prototyp muss Geräte eindeutig und in angemessener Zeit identifizieren können. Die Anzahl unterschiedlicher Geräte, die erkannt werden können, ist für die Arbeit nicht sonderlich relevant. Dennoch sollte prinzipiell die Möglichkeit bestehen, alle im Alltag befindlichen Geräte und Applikationen identifizieren zu können. Die Identifikation soll auf der Blickrichtung des Benutzers basieren. Es soll ein Gerät zur Zeit erkannt werden. Befinden sich mehrere Geräte in der Blickrichtung, soll durch leichte Korrektur der Blickrichtung das gewünschte Gerät erfasst werden. Sobald ein Gerät identifiziert wurde, soll es für eine geeignete Zeit lang gemerkt werden, sofern in dieser Zeit kein anderes Gerät erkannt wird. Ist dies jedoch der Fall, soll das neue Gerät das bisherige ersetzen.

#### **Gerätesteuerung**

Um Geräte bedienen zu können, muss der Prototyp eine Schnittstelle bereit stellen, mit der Gerätekonfigurationen geladen werden können. Aus dieser Konfiguration ergeben sich die verfügbaren Informationen und Funktionen eines Gerätes. Die Gerätekonfiguration muss ein gängiges Format haben, das auf vielen Plattformen, insbesondere mobilen, verfügbar ist. Die Schnittstelle muss so gestaltet sein, dass unterschiedliche Geräte und Gerätetypen beschrieben werden können. Dabei soll sich die Schnittstelle nicht auf vernetzte Geräte beschränken, sondern beliebige Geräte beschreiben können. Zudem muss es einen bidirektionalen Kanal geben, um Geräten Befehle senden und Informationen des Gerätes empfangen zu können. Gerätezustände sollen in geeigneten Zeitintervallen auf dem Benutzerinterface aktualisiert werden.

## 3.2 Wahl der Hardware

Nachdem in Abschnitt 2.4 festgestellt wurde, dass der Markt zum Zeitpunkt dieser Arbeit keine geeignete Hardware hergibt, die sowohl der Anforderung an ein transparentes Head-Mounted Display nachkommt, als auch die Forderung an das Erfassen der Hände durch einen Tiefensensor erfüllt, wird eine Kombination folgender Geräte als Hardwaregrundlage für diese Arbeit verwendet:

**Display/Platform** Moverio BT-200

**Sensor** Leap Motion

In Anbetracht der in Abschnitt 3.1 gestellten Anforderungen an das Zielsystem bieten beide Geräte zur Zeit die jeweils beste Grundlage, um die Anforderungen an die Interaktion und die Visualisierung umzusetzen. Beiden Geräte wurden bereits in Abschnitt 2.3.1 und 2.3.3 vorgestellt.

### Sensor Adapter

Um die Geräte physikalisch zu verbinden, wurde im Rahmen dieser Arbeit ein Adapter entwickelt, der in Abbildung 3.1 dargestellt ist. Die Halterung besteht aus drei U-Profilen, die wiederum aus Aluminium bestehen. Zwei dienen als Befestigung an der Brille und umspannen den linken und rechten Bügel. Ein dritter klemmt den Sensor mittig über den Displays der Moverio ein. Jede Innenseite der Profile ist mit Moosgummi verkleidet, um eine material-schonende Befestigung zu ermöglichen, die schnell auf- und abgesteckt werden kann. Ein 3mm starker Bogen aus Acrylglas hält die Profile mit Schrauben zusammen. In einer Klemmvorrichtung an der linken Seite kann das Kabel des Sensors befestigt werden, damit der Sensor nicht aus der Halterung gerissen werden kann. Die Halterung wurde so konstruiert, dass der Benutzer weder den Sensor sieht, noch die Halterung selbst.

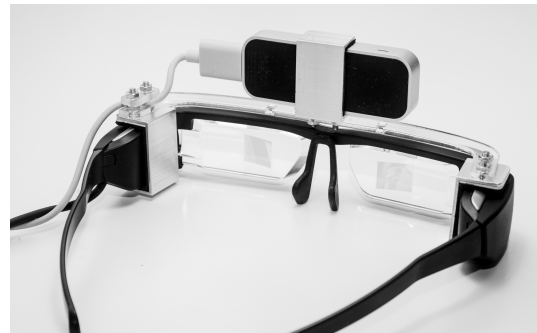


Abbildung 3.1: Display-Sensor-Halterung

### 3.3 Systemkomponenten

*HoloUI* setzt sich aus fünf grundlegenden Komponenten zusammen. Abbildung 3.2 zeigt die Abhängigkeiten und Kommunikationswege der Systemkomponenten sowie das externe Gerät, welches kontrolliert wird.

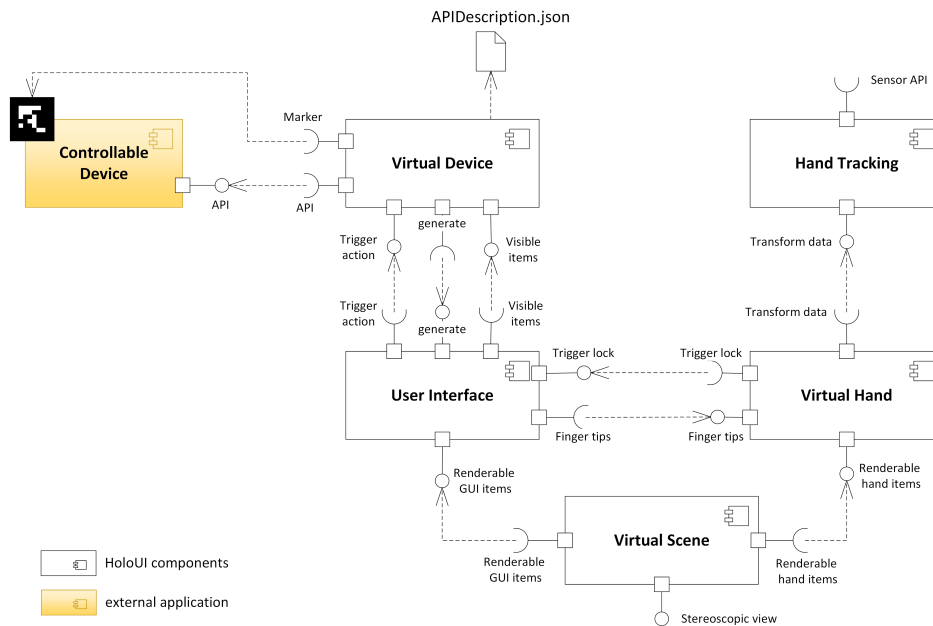


Abbildung 3.2: Black-Box-Sicht der Systemomponenten

Die *Hand Tracking*- und *Virtual Scene*-Komponenten bilden jeweils das Eingabe- bzw. Ausgabemedium für den Benutzer. Erstere kommuniziert dafür mit Leap Motion, während durch letztere die stereoskopische Sicht auf die virtuelle Szene auf der Moverio dargestellt wird. Die fünf Komponenten werden in den folgenden Abschnitten 3.3.1 bis 3.3.5 vorgestellt.

#### 3.3.1 Virtual Device

Es gilt, ein externes Gerät bedienen zu können. Um das zu ermöglichen, bedarf es einer virtuellen Repräsentation des Gerätes samt seiner Schnittstellen. Die *Virtual Device*-Komponente generiert anhand einer Gerätebeschreibung ein virtuelles Geräteabbild, das Aktionen auf dem Gerät auslösen kann und sich mit dessen Zustand synchronisiert. Wie Abbildung 3.5 zu entnehmen ist, setzt sie sich aus zwei Komponenten zusammen, die im Folgenden beschrieben werden.

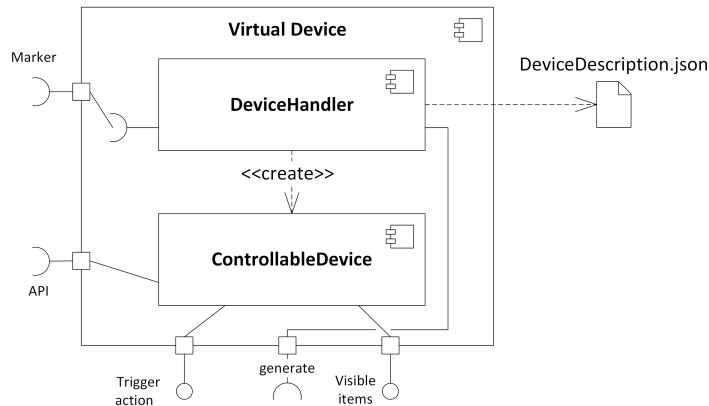


Abbildung 3.3: *Virtual Device*-Komponente

### DeviceHandler

Die *DeviceHandler*-Komponente identifiziert das reale Gerät und verwaltet das virtuelle Abbild. Wie in Abbildung 3.4 gezeigt ist, werden zunächst alle bekannten Gerätebeschreibungen geladen und mit einem eindeutigen Marker, der in der Beschreibung referenziert wird, assoziiert. In einem Intervall wird daraufhin das Kamerabild kontinuierlich abgegriffen, um nach einem Marker zu suchen, der zu einer der Gerätebeschreibungen passt. Sobald eine Übereinstimmung auftritt, wird anhand der assoziierten Beschreibung der Gerätetype bestimmt und ein entsprechendes *ControllableDevice* generiert.

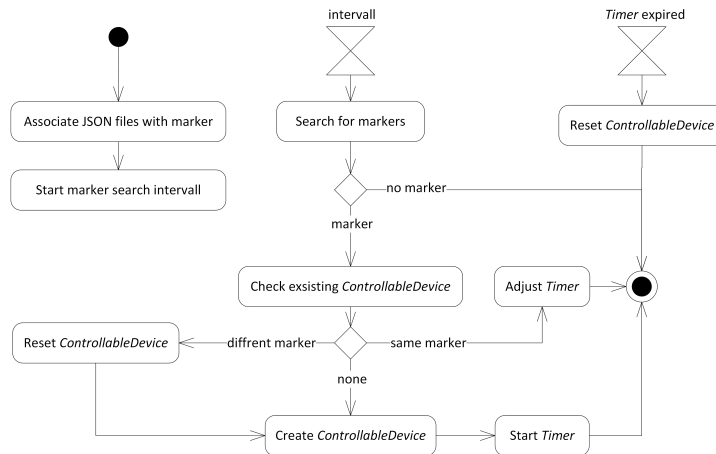


Abbildung 3.4: Aktivitätsdiagramm der *DevieHandler*-Komponente

Nach dem Generieren veranlasst der *DeviceHandler* über eine Schnittstelle der *User Interface*-Komponente das Erzeugen einer entsprechenden *GUI* und startet einen **Timer**, der die Lebenszeit des *ControllableDevice* darstellt. Dies erfüllt die Anforderung an die Bedienbarkeit, die eine bestimmte Zeit erhalten bleiben soll, wenn sich das Gerät nicht mehr im Sichtbereich befindet. Sobald der **Timer** abläuft, wird das *ControllableDevice* zurückgesetzt. Wird der gleiche Marker öfters erkannt, beginnt der **Timer** von vorne.

#### **Beschreibung der Geräteschnittstelle**

Die Grundlage für die Generierung bildet die Beschreibung der Schnittstelle des externen Gerätes. Sie entspricht einem einheitlichen JSON-Schema<sup>1</sup>. Wie in Abschnitt 1.2 erwähnt wurde, wird ein Schema passend für Geräte und Applikationen modelliert, die über eine Web API verfügen und Daten im XML-Format zurückgeben.

Ein Schema beschreibt immer einen Gerätetyp. Wird eine Gerätebeschreibung nach spezifischem Schema erstellt, muss der Dateiname entsprechend mit dem durch das Schema beschriebenen Gerätetypen beginnen. So kann dieser später bestimmt werden, ohne die Struktur des Schemas kennen zu müssen.

Ein abstraktes Basisschema bildet die Grundlage jedes spezifischen Schemas. Dies beschreibt die notwendigen Felder für das *ControllableDevice*, die jeder Gerätetyp mindestens unterstützen muss. Sollen verschiedene Gerätetypen kontrollierbar sein, muss ein spezifisches Schema samt entsprechender Implementation der inneren *ControllableDevice*-Komponente (Abbildung 3.5) vorhanden sein. Dadurch wird der Anforderung genüge getan, verschiedene Gerätetypen verwenden zu können. Im Folgenden wird das Schema für den Gerätetyp `HttpXmlDevice` beschrieben.

#### **HttpXmlDevice Schema**

Das Schema für `HttpXmlDevices` erweitert ebenfalls Basisschema. Die Felder des Basisschemas sind durch einen Stern `*` gekennzeichnet.

*Name\** Der Name beschreibt den Namen des Gerätes der dem Benutzer angezeigt wird.

*Marker\** Jedem Gerät muss ein eindeutiger Marker zugewiesen werden. Dieser wird im System mit der entsprechenden Gerätebeschreibung assoziiert.

---

<sup>1</sup><http://json-schema.org/documentation.html>

*Http* Das Http-Feld besteht aus IP Adresse, Port und Methode, die jeweils für die Kommunikation notwendig sind. Es können zusätzlich Benutzername und Passwort übergeben werden, sollte das Gerät über eine Basic Authentication<sup>2</sup> verfügen.

*Fields\** Dieses Feld beinhaltet eine Auflistung der Gerätezustände, die dem Benutzer angezeigt werden sollen oder für den Zustand eines Bedienelementes verantwortlich sind.

*Display\** In diesem Feld wird ein Template angegeben, welches das Ausgabeformat der in Fields angegebenen Zustände beschreibt. Jedem Zustandsfeld kann zudem eine Funktion zugeordnet werden, welche den Zustand angenehmer zu lesen macht.

*Buttons\** Hier werden die Bedienelemente des Gerätes aufgelistet. Jeder Button besitzt einen Namen\*, ein Icon\*, einen optionalen aktiven Zustand, Parameter und eine entsprechende Requestinformation, über die mithilfe der Parameter die Aktion auf dem Gerät ausgelöst werden kann.

*Requests* In den Requests sind die einzelnen Abfragen, die an das Gerät gesendet werden können, beschrieben. Jeder Request besitzt neben dem Pfad eine Beschreibung, wo sich ein bestimmter Zustand in der XML Antwort befindet.

*Initialize\** Hier können Requests referenziert und Parameter angegeben werden, die gesendet werden, sobald das virtuelle Abbild eines Gerätes generiert wird.

*Update\** Analog zu Initialize enthält Update die Requests und Parameter, über die sich das virtuelle Gerät mit dem realen Gerät synchronisiert und die Zustände abfragt.

#### **ControllableDevice**

Das *ControllableDevice* bildet die Grundlage, von der sich alle virtuellen Geräteabbildungen ableiten lassen. Die für das *HttpXmlDevice*-Schema erforderlichen Klassen sind in Abbildung 3.5 dargestellt.

Sobald ein *ControllableDevice* anhand einer Gerätebeschreibung generiert wurde, werden die Methoden `initDevice()` und `startUpdate()`, die durch die *IController* Schnittstelle bereitgestellt werden, aufgerufen. Letzte Methode gewährleistet die Synchronität der beschriebenen Zustände durch eine Intervallabfrage. Der *HttpController* ruft auf dem externen Gerät die entsprechende URL auf und verwertet die Antwort

---

<sup>2</sup>[https://de.wikipedia.org/wiki/HTTP-Authentifizierung#Basic\\_Authentication](https://de.wikipedia.org/wiki/HTTP-Authentifizierung#Basic_Authentication)

mithilfe eines XML-Parsers, der die Zustände aus der Gerätebeschreibung findet und im *HttpXmlDevice* aktualisiert.

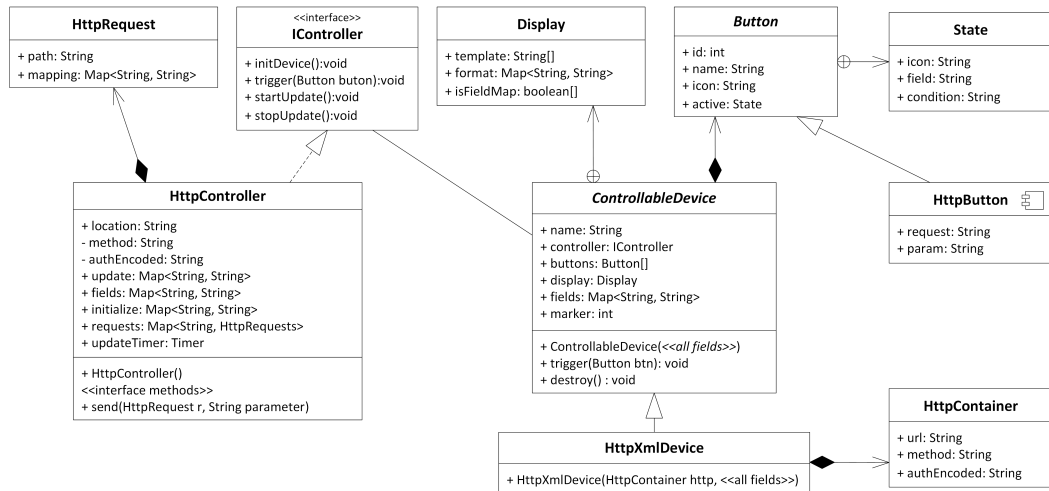


Abbildung 3.5: Klassendiagramm der *ControllableDevice*-Komponente

Über das *IController*-Interface wird ebenfalls das Auslösen von Geräteaktionen bereitgestellt. Die Informationen, die für die *User Interface*-Komponente relevant sind, werden durch die Attribute *display*, *fields* und *buttons* des *ControllableDevices* geliefert.

### 3.3.2 Hand Tracking

*HoloUI* soll eine handbasierte Interaktion ermöglichen. Die Voraussetzung dafür ist das Erfassen der realen Hände relativ zu der virtuellen Umgebung. Dies wird durch die *Hand Tracking*-Komponente umgesetzt, die über eine Schnittstelle aufbereitete Daten über die Transformation und Bewegung der Hände bereitstellt.

Zu Beginn dieser Arbeit gab es kein Android SDK für Leap Motion. Der Sensor kann dadurch nicht direkt von der Moverio aus angesteuert werden. Um trotzdem eine Kommunikation zu gewährleisten, wird der Sensor an ein Notebook angeschlossen, siehe Abbildung 3.6. Die *Hand Tracking*-Komponente wird als externe Applikation konzipiert. Damit Daten auf der Moverio verarbeitet werden können, werden sie im JSON-Format über einen WebSocket bereitgestellt.

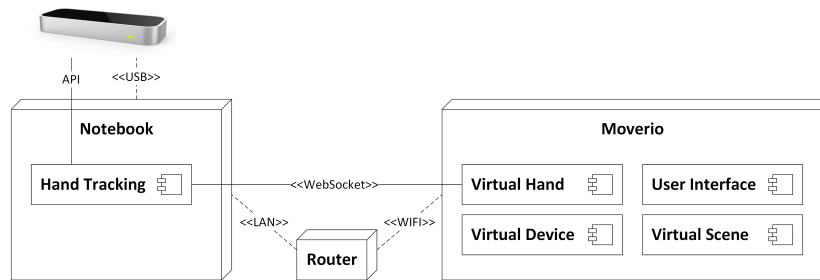


Abbildung 3.6: Verteilungsdiagramm der Systemkomponenten

Leap Motion liefert im Durchschnitt 115 Frames pro Sekunde. Eine so hohe Frequenz ergibt für *HoloUI* wenig Sinn, da einerseits das Display der Moverio mit 60 Hz aktualisiert wird und andererseits die Systemauslastung durch das Deserialisieren der Nachrichten zu hoch wäre. Um dem entgegen zu wirken, muss die *Hand Tracking*-Komponente eine konfigurierbare Frequenz besitzen, die das Senden der Handinformationen beschränkt. In Abbildung 3.7 ist der Ablauf der Komponente dargestellt.

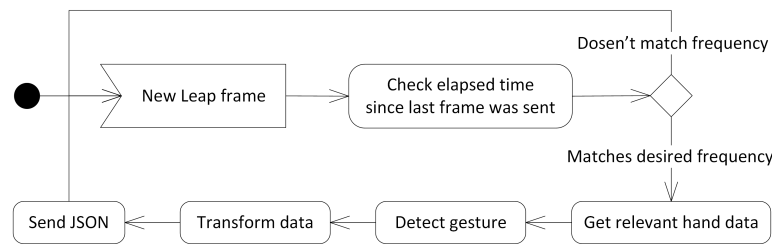


Abbildung 3.7: Aktivitätsdiagramm der *Hand Tracking*-Komponente

Sobald ein neuer Leap Frame zur Verfügung steht, muss entschieden werden, ob dieser verwertet werden darf. Ist das der Fall, werden relevante Informationen über die Translation, Skalierung und Rotation aller fünf Finger sowie die Position der Hand erfasst. Dieses dient der Erkennung einer „Stillhalte“-Geste, deren Dauer ermittelt wird. Dadurch wird die Anforderung erfüllt, die virtuellen Bedienelemente im Raum nach einer Zeit „einrasten“ zu lassen.



### Gestenerkennung

Um eine stillgehaltene Hand zu bestimmen, wird die relative Translation  $T_R$  des Handballens der linken bzw rechten Hand anhand seiner Position  $P_n$  zwischen mehreren Frames nach der Formel 3.1 berechnet.

$$T_R = \sqrt{(p_{n-i,x} - p_{n,x})^2 + (p_{n-i,y} - p_{n,y})^2 + (p_{n-i,z} - p_{n,z})^2} \quad (3.1)$$

Valide Frames, die für die Berechnung genutzt werden, sind jene, in denen die jeweilige Hand kontinuierlich erkannt wurde. Es werden die Position des Handballens und die Zeit der aufeinander folgenden Frames gespeichert. Anschließend wird zwischen dem letzten und jedem vorherigen validen Frame die relative Änderung der Position berechnet und mit einer maximalen Abweichung, damit der Benutzer seine Hand nicht wie versteinert stillhalten muss, verglichen. Befindet sich die relative Translation darunter, wird die Zeit der Frames summiert. Daraus ergibt sich die Zeit, in der die jeweilige Hand stillgehalten wurde.

### Transformation des Koordinatensystems

Leap Motion nimmt an, dass der Sensor auf einem Tisch steht und nach oben zeigt. Die Y-Achse des Koordinatensystems zeigt in dieser Ausrichtung nach oben. *HoloUI* verwendet den Sensor allerdings in einer nach vorne ausgerichteten Position, siehe Abbildung 3.1. Dadurch zeigt die Y-Achse des Sensors nach vorne und die Z-Achse nach oben. Damit die Transformationsdaten der Hände im gleichen Koordinatensystem wie die virtuelle Umgebung liegen, ist der in Abbildung 3.7 gezeigte Transformationsschritt notwendig. Nach der Formel aus He [He] wird zunächst die Translation korrigiert:

$$M_{Leap/world} = \begin{bmatrix} -1 & 0 & 0 & t_x \\ 0 & 0 & -1 & t_y \\ 0 & -1 & 0 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

In einem zweiten Schritt wird die Maßeinheit von Millimeter in Zentimeter umgerechnet. Dies ist notwendig, da die virtuelle Umgebung von *HoloUI* Zentimeter als einheitliches Maß verwendet.

$$\begin{bmatrix} P_{world} \\ 1 \end{bmatrix} = M_{Leap/world} \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_{Leap} \\ 1 \end{bmatrix} \quad (3.3)$$

Die  $t$  Variablen bezeichnen die Ausrichtung des Sensors gegenüber der Moverio.  $t_x$  und  $t_z$  sind 0, da der Sensor mittig über dem Display sitzt.  $t_y$  entspricht dem Höhenunterschied zwischen Display- und Sensormitte. Dies kann allerdings einfach durch die virtuelle Kameraposition ausgeglichen werden.  $P_{world}$  beschreibt die korrekte Position im Raum der virtuellen Umgebung.

$$Q_{World} = (x, z, y, w), Q_{Leap} = (x, y, z, w) \quad (3.4)$$

Die Rotation wird korrigiert, indem das entsprechende Quaternion angepasst wird, sodass der  $z$  mit dem  $y$  Wert vertauscht ist.

#### 3.3.3 Virtual Hand

Um der virtuellen Umgebung von *HoloUI* Interaktivität zu verleihen, muss ein Element existieren, das auf virtuelle Inhalte reagieren kann. Die *Virtual Hand*-Komponente generiert ein Abbild der physikalischen Hand des Benutzers, durch das virtuelle Bedienelemente selektiert werden können.

Die Komponente erstellt zu Beginn durch den *HandCreator* jeweils eine virtuelle linke und rechte Hand. Die Transformationsdaten werden über die Schnittstelle der *Hand Tracking*-Komponente empfangen, wodurch die Bewegung der realen Hand imitiert wird. Die Komponenten, aus denen sich die *Virtual Hand* zusammensetzt, sind in Abbildung 3.8 dargestellt.

#### Hand

Die *Hand*-Komponente repräsentiert die reale Hand durch Kugeln für Gelenke und Zylinder für Knochen, siehe Abbildung 3.9 links. Würde man dieses Handmodell transparent darstellen und über die reale Hand legen, könnte man mit dem eigenen Finger

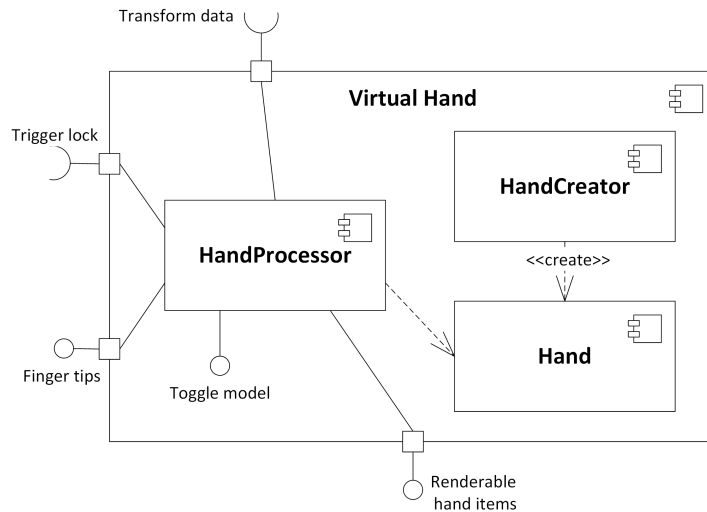


Abbildung 3.8: *Virtual Hand*-Komponente

die virtuellen Elemente „berühren“ und aktivieren. Head-Mounted Displays wie die Moverio bewerben mit dieser Interaktion die Displays. Allerdings wird bereits durch die Helligkeit des Displays und die angezeigten Elemente der reale Finger soweit verdeckt, dass eine Zuordnung zwischen dem virtuellen Element und dem Finger unmöglich wird. Desweiteren kann das menschliche Auge nur eine Ebene fokussieren, wodurch entweder das Display oder der Finger verschwommen und doppelt wahrgenommen wird[BSS13]. Diese Tatsache macht den Schritt unausweichlich, ein sichtbares Handmodell zu verwenden.

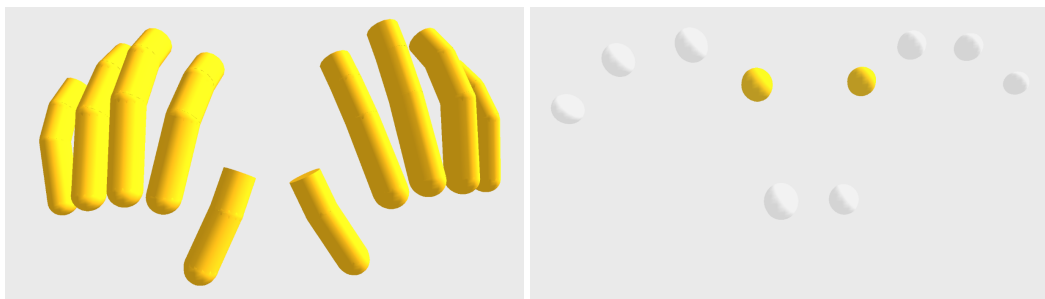


Abbildung 3.9: Finger- und Kugel-Handmodell

Aufgrund des Fat-Finger-Problems[Wig+07], bei dem Bedienelemente, die man auswählen will, durch seinen Finger verdeckt werden, kann die virtuelle Hand durch ein

alternatives Handmodell dargestellt werden. Das Kugel-Handmodell beschränkt sich auf transparente Fingerspitzen, siehe Abbildung 3.9.

Die Interaktion ist ausschließlich durch die Fingerspitze des Zeigefingers vorgesehen, um ungewollte Interaktion durch die anderen Finger zu vermeiden. Beide Modelle heben die Fingerspitze hervor, je näher sie sich einem interaktiven Element befinden, siehe Abbildung 4.3, um die Tiefenwahrnehmung des Benutzers zu verbessern.

#### **HandProcessor**

Der *HandProcessor* nutzt die Schnittstelle der *Hand Tracking*-Komponente und überträgt die Transformationen der einzelnen Fingerglieder auf ihr virtuelles Gegenstück. Dadurch verformt sich die virtuelle Hand, sodass sie proportional der realen entspricht. Der *HandProcessor* nutzt zudem die Schnittstelle der *User Interface*-Komponente, um diese „einrasten“ zu lassen, sobald eine stillgehaltene Hand durch die *Hand Tracking* Komponente erkannt wurde. Der *HandProcessor* stellt mehrere Schnittstellen zur Verfügung, die einerseits Daten über die Position und die Größe der Fingerspitzen und andererseits die darzustellende Kugeln und Zylinder der Finger bereitstellen. Ebenso kann zwischen den beiden Handmodellen über eine Schnittstelle umgeschaltet werden, die dem Benutzer durch ein Bedienelement der Moverio zugänglich ist.

#### **3.3.4 User Interface**

Die *User Interface*-Komponente stellt dem Benutzer ein interaktives generisches Benutzerinterface zur Verfügung. Sie setzt sich aus vier Komponenten zusammen (Abbildung 3.10). Über drei Schnittstellen werden die darzustellenden Elemente sowie Funktionen zum Generieren und „Einrasten“ der Benutzeroberfläche bereitgestellt.

Sobald ein bedienbares Gerät erkannt wird, erhält die *GuiCreator*-Komponente über ihre Schnittstelle `Button`-Elemente des *ControllableDevices* und generiert eine *GUI*-Komponente, die ein horizontales Menü (Abbildung 4.3, links) darstellt.

#### **GUI**

Die Anforderung an die Benutzeroberfläche erst sichtbar zu sein, wenn der Benutzer mit dem Gerät interagieren will, wird realisiert, indem die *GUI*-Komponente die *Virtual Hand*-Element nach aktiven Fingerspitzen fragt. Wenn die *VirtualHand*-Komponente

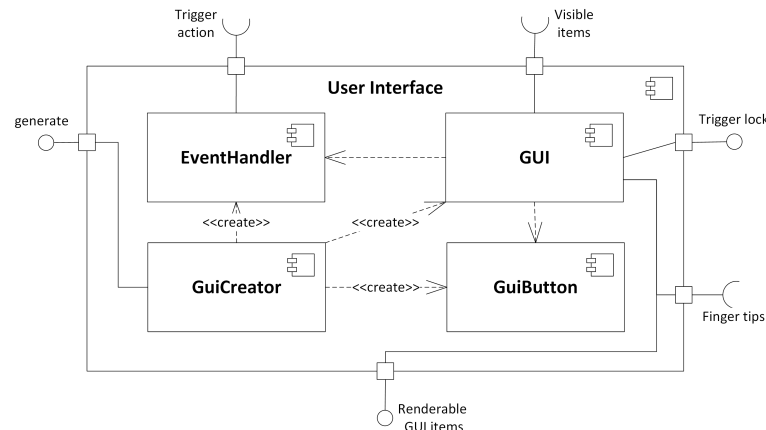


Abbildung 3.10: *User Interface*-Komponente

eine Fingerspitzen bereitstellt, fängt die Oberfläche an diese zu folgen. Dadurch wird dem Benutzer unmerklich ein Tiefenhinweis über die relative Entfernung zwischen seiner Hand und den Bedienelementen geboten, um die Selektion eines Buttons zu vereinfachen. Das Menü ändert seine Transparenz und stoppt die Verfolgung, sobald die *Virtual Hand* die entsprechende Schnittstelle zum „Einrastet“ des Menüs anspricht.

Eine *EventHandler*-Komponente verwaltet die Interaktion mit der Oberfläche. Durch Schnittpunkttests[Arv90] zwischen der Fingerspitze mit der *GuiButton*-Komponente, werden entsprechende Events ausgelöst. Der *EventHandler* leitet die Aktion an die Schnittstelle der *ControllableDevice*-Komponente.

### GuiButton

Die Bedienelemente der *GUI* sind durch Icons dargestellt, die in der Gerätebeschreibung referenziert wurden. Die *GuiButton*-Komponente stellt ein Icon-Repertoire bereit, aus dem die Icons geladen werden. Ein **Button**-Element kann einen aktiven Zustand besitzen (Abbildung 3.5, Button Klasse), wodurch er sein Icon gegen ein anderes austauscht. Durch diese Funktionalität kann zum Beispiel ein Play-Button zu einem Pause-Button werden.

Die *GuiButton*-Komponente erlaubt einen „pressed“ Zustand, der dem Benutzer ein visuelles Feedback vermittelt, sobald ein Buttonevent ausgelöst wurde. Abbildung 3.11 fasst die Zustände zusammen.

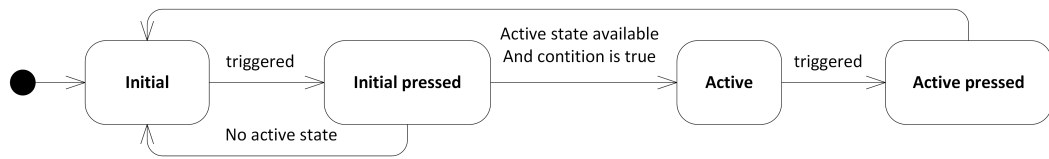


Abbildung 3.11: Zustandsdiagramm der *GuiButton*-Komponente

### 3.3.5 Virtual Scene

Die *Virtual Scene*-Komponente nutzt Schnittstellen der *Virtual Hand* sowie der *User Interface*-Komponente um Elemente zu rendern die aktiv. Das stereoskopische Bild wird auf dem Display der Moverio ausgegeben.

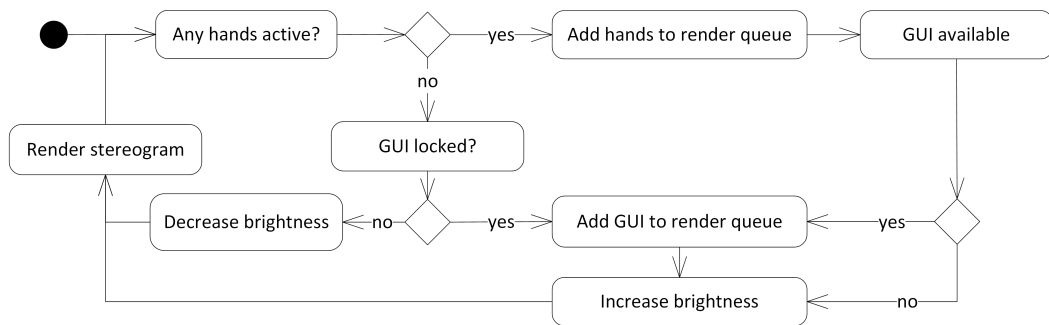


Abbildung 3.12: Aktivitätsdiagramm der Rendschleife

### Stereoskopisches Rendern

In der virtuellen Szene sind zwei Kameras positioniert, die durch einen leicht versetzten Blickwinkel die Szene zeigen und jeweils für das linke und rechte Auge die Bilder berechnen. Die Kameras sind dabei so positioniert, das optimalerweise die virtuellen Fingermodelle die realen Finger überlagern. Dazu muss der Abstand zwischen den internen Kameras des Tiefensensors und der Abstand zwischen den virtuellen Kameras in der Szene dem Abstand der „biologischen Kameras“, den Augen des Benutzers, entsprechen[Col], so dass genaue Tiefeninformationen berechnet werden. Da dies nicht möglich ist, wird für die Position der virtuellen Kameras ein Kompromiss zwischen Sensor und Auge gefunden. Ein weiteres Problem ist der schmale Sichtbereich der Moverio, wodurch die Positionierung der Kameras weiter eingeschränkt wird, um den stereoskopischen Effekt aufrecht zu erhalten.

#### **Anpassung der Helligkeit**

Je nachdem, ob der Benutzer *HoloUI* aktiv nutzt oder nicht, wird die Helligkeit des Bildschirms verringert oder erhöht, um bei Inaktivität dem Benutzer gegenüber unscheinbar zu sein. Abbildung 3.12 zeigt den zugrunde liegenden Entscheidungsweg.

## **3.4 Implementierung**

#### **Android-Applikation**

Für die Moverio wurde eine Android-Applikation für Android v4.0.3 umgesetzt. Als Programmiersprache wurde Java verwendet und als Entwicklungsumgebung kam Android Studio v1.3.2 unter Windows 10 zum Einsatz. Es wurden folgende externe Bibliotheken für die Implementierung benutzt:

*libGDX* ist ein Open-Source-Framework für plattformunabhängige Spieleentwicklung.

*Chilitags* ist eine Bibliothek für die Detektion und Identifizierung von 2D-Markern.

*Java WebSocket* stellt eine nicht-blockierende eventbasierte WebSocket-Implementation in Java bereit.

*Jackson* ist eine Java-Bibliothek zur Verarbeitung des JSON-Datenformats.

*VTD-XML* ist eine Multiplattform-Bibliothek zur Verarbeitung des XML-Datenformats.

#### **Externe Applikation**

Die externe Applikation, durch die das *Hand Tracking* abgebildet ist, wurde in JavaScript für die Node.js-Umgebung geschrieben. Es wurden folgende externe Bibliotheken und Module für die Implementierung benutzt:

*leapjs* stellt das JavaScript-SDK für Leap Motion dar.

*ws* ist eine WebSocket-Implementation für NodeJS.

*glMatrix* ist eine performante Javascript-Bibliothek für Matrizen und Vektoren.

*keypress* ist ein Modul zum Abfangen von Tasteneingaben.

### 3.5 Zusammenfassung

In diesem Kapitel wurde die Konzeption und die Umsetzung von *HoloUI* beschrieben. Die Anforderungen an das Zielsystem aus Abschnitt 3.1 wurden bis auf eine erfüllt.

Ein Kompromiss wurde im Hinblick auf die Mobilität in Kauf genommen, da der Leap Motion-Sensor nicht direkt durch die Moverio betrieben werden konnte. Das Auslagern der *Hand Tracking*-Komponente abstrahiert diesen Schritt allerdings innerhalb der Android-Anwendung, wodurch funktional keine Einschränkungen auftreten. Einzig bleibt die Mobilität durch das Notebook beschränkt.

Der Prototyp erfüllt den in Abschnitt 1.2 beschriebenen Anwendungsfall. Über einen Marker wird eine VLC Player Instanz erkannt, woraufhin dem Benutzer eine entsprechende Benutzeroberfläche zur Verfügung gestellt wird, die anhand der Gerätebeschreibung des VLC Players generiert wurde. Der Benutzer kann sie bei Bedarf durch seine Hand aktivieren (Abbildung 3.13) und mit dem Zeigefinger der linken und rechten Hand Aktionen auf dem VLC Player auslösen. Die Oberfläche übernimmt Gerätezustände, die in der Gerätebeschreibung angegeben sind und aktualisiert sie in einem Intervall. Sobald der Benutzer sich von dem Gerät abwendet, bleibt die Benutzeroberfläche eine Zeit lang erhalten und kann weiter benutzt werden.

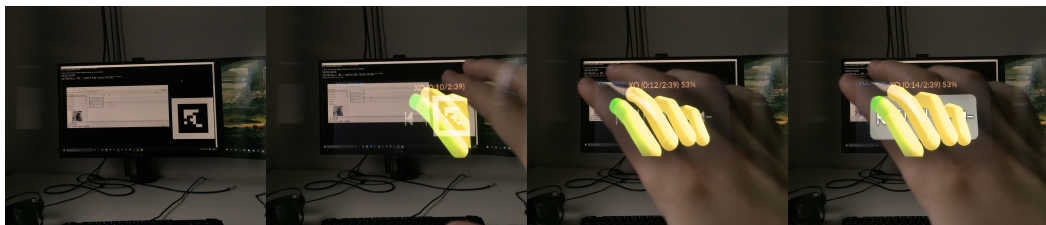


Abbildung 3.13: Aktivieren der Benutzeroberfläche



## 4 Evaluation

Dieses Kapitel evaluiert die Benutzererfahrung von *HoloUI*. Dazu werden in Abschnitt 4.1 zuerst Reaktionszeiten erfasst, die grundlegend für das Benutzererlebnis sind. In einem zweiten Schritt wird anhand einer Benutzerstudie in Abschnitt 4.2 die Benutzung des Zielsystems im Ganzen untersucht. Das Kapitel schließt mit dem Ergebnis in Abschnitt 4.3 ab.

### 4.1 Laufzeit Messung

In Abschnitt 3.1 wurden nicht-funktionale Anforderungen an die Handfassung und die Gerätesteuerung gestellt. Im Folgenden wird die Performance durch eine Zeitmessung dargestellt. Dabei wird in Abschnitt 4.1.1 zuerst die Verzögerung der virtuellen Hand gegenüber der realen betrachtet. In Abschnitt 4.1.2 wird die Betriebsbereitschaft von *HoloUI* gemessen und zuletzt wird die Ausführung einer Aktion des VLC Players in Abschnitt 4.1.3 betrachtet.

#### Messumgebung

Die Messungen wurden innerhalb der Android-Applikation auf der Moverio (Android 4.0.4) durchgeführt. Dazu wurde `System.nanoTime()` verwendet. Das externe Programm der *Fingererfassungs*-Komponente aus Abschnitt 3.3 läuft auf Windows 10. Es wurden die NodeJS-Methoden `console.time()` und `console.timeEnd()` genutzt. Der VLC Player (Version 2.2.1 Terry Pratchett) läuft ebenfalls unter Windows 10.

Die Moverio ist über ein 802.11b/g/n WLAN einer Fritzbox 7560 SL verbunden und zeigt eine volle Signalstärke an. Der Windows 10 PC verfügt über eine Intel E3-1230v2 CPU, eine Samsung 840 SSD Pro und ist über LAN verbunden.

### 4.1.1 Latenz der virtuellen Hand

Die Latenz beschreibt die Zeit, die zwischen einer Bewegung der realen Hand und der Imitation der virtuellen vergeht. Da die Handerfassung mithilfe eines externen Programms realisiert wurde, lässt sich die genaue Zeit nur erschwert über zwei Systeme hinweg messen. Es wird deshalb die benötigte Zeit separat in beiden Applikationen festgehalten. Dadurch bleibt die Übertragungszeit durch das WLAN unbeachtet.

Die Latenz ist in Abbildung 4.1 durch —●— *Hand Imitation* dargestellt. Sie setzt sich aus der Rechenzeit des externen Programmes ( $\leq 1ms$ ), das die Sensordaten abgreift und aufbereitet sowie der Rechenzeit, die die Applikation auf der Moverio benötigt, um die Daten zu deserialisieren, auf die dreidimensionalen Objekte anzuwenden und das Bild zu rendern. Die Zeit dieses Prozesses liegt im unteren Zehntelsekunden-Bereich.

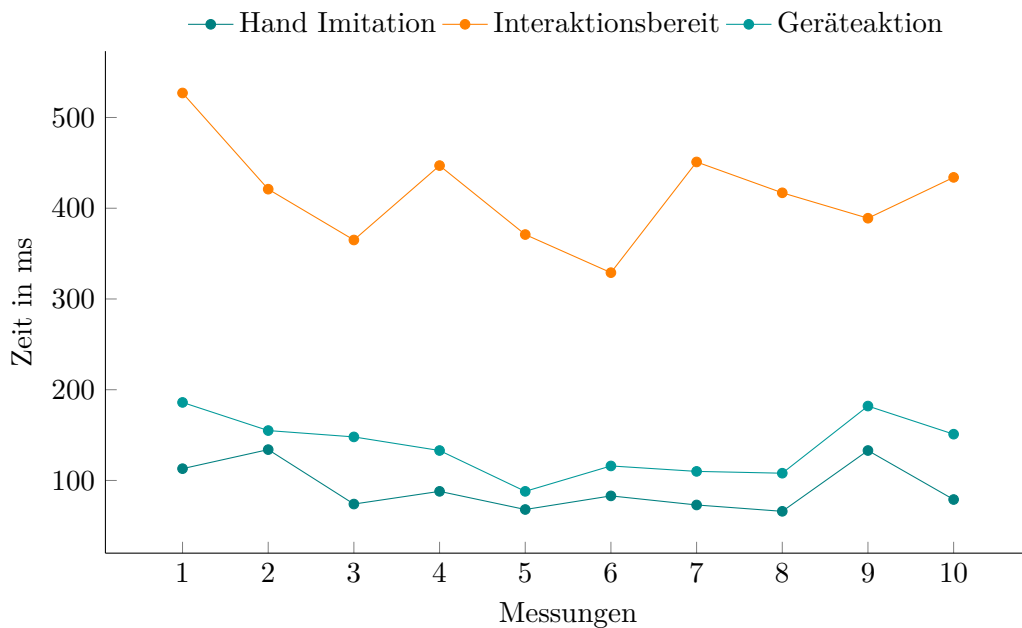


Abbildung 4.1: Übersicht der Reaktionszeitmessungen

### 4.1.2 Interaktionsbereitschaft

Sobald ein bedienbares Gerät angeschaut wird, ermöglicht es *HoloUI*, mit diesem zu interagieren. Die Messung zeigt die Zeit, bis der Benutzer Aktionen auf dem Gerät

auslösen kann. Sie umfasst die Erkennung des Gerätes durch einen Marker, das Laden der Konfigurationsdatei und die Generierung der Benutzeroberfläche. Die benötigte Zeit ist in Abbildung 4.1 durch  $\text{---}\bullet\text{---}$  *Interaktionsbereit* gezeigt. Nachdem ein Gerät im Sichtfeld des Benutzers ist, dauert es etwa eine halbe Sekunde, bis die Interaktion beginnen kann.

### 4.1.3 Geräteaktion

Die Messung der Ausführung einer Aktion auf dem bedienbaren Gerät setzt sich aus der Übertragung der Daten, dem Parsen der Antwort und der Interaktion innerhalb der virtuellen Umgebung zusammen. Dazu zählt das Rendern sowie das Aktualisieren der Finger und Überschneidungstests ( $\leq 1ms$ ) der Buttons und der Finger. Die Zeit ist in Abbildung 4.1 durch die Linie  $\text{---}\bullet\text{---}$  *Geräteaktion* dargestellt. Eine durchschnittliche Aktion benötigt  $\cong 150ms$ .

## 4.2 Benutzerstudie

Es wurde eine Benutzerstudie im Rahmen dieser Arbeit durchgeführt, anhand welcher das Benutzererlebnis untersucht und bewertet wird. Dabei zielte die Studie ebenso in Richtung der Akzeptanz der Technologie. Die Studie setzt sich aus zwei Teilen zusammen. Im ersten Teil wurde den Probanden *HoloUI* zum ersten Mal gezeigt, während im zweiten Teil Probanden zum wiederholten Male das System testen konnten.

### 4.2.1 Teil I

Teil I befasst sich mit dem Herantasten an die neue Interaktionsform. Es wurden 9 Personen im Alter von 11 bis 23 Jahren gebeten, nach 1-3-minütigem Ausprobieren von *HoloUI* fünf Aussagen zu bewerten. Eine Likert-Skala<sup>1</sup> die von *stimme nicht zu(1)*, *stimme eher nicht zu(2)*, *teils-teils(3)*, *stimme eher zu(4)* bis *stimme zu(5)* reicht, diente dabei als Grundlage. Alle Personen haben außerdem angegeben, sich für Technik zu interessieren und hatten zu einem Großteil vorher schon Kontakt mit Interaktionsformen, wie der Playstation Move, Wii Remote und Xbox Kinect. Folgende Aussagen wurden den Probanden gezeigt:

A1: Ich konnte zuverlässig Buttons auswählen.

A2: Ich habe niemals Buttons ungewollt ausgewählt.

---

<sup>1</sup><https://de.wikipedia.org/wiki/Likert-Skala>

A3: Ich konnte die Entfernung zwischen Menü und virtuellem Finger gut einschätzen.

A4: Das Einfärben der Fingerspitze half der Tiefenwahrnehmung.

A5: Die Interaktionsform ist interessant und ich würde sie in Zukunft nutzen wollen.

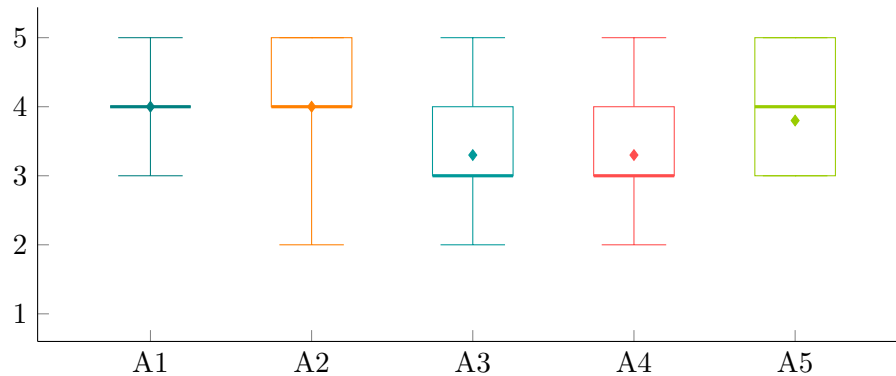


Abbildung 4.2: Auswertung der ersten Likert-Skala

Das Ergebnis der Auswertung der Likert-Skala ist in Abbildung 4.2 dargestellt. Grundsätzlich haben die Aussagen Zuspruch erhalten. Der Median der Aussagen A1, A2, und A5 liegt bei 4, während er generell nicht unter 3 rutscht.

Die unteren Antennen reichen bei A2 und A4 bis auf 2 hinunter. Dies mag der kurzen Zeit geschuldet sein, die zum Testen verfügbar war. Ebenfalls reicht die untere Antenne von A3 bis auf 2 hinunter, wodurch das relative Färben der Fingerspitze im Hinblick auf die Entfernung zum Menü nicht signifikant die Tiefenwahrnehmung unterstützt.

Durch A5 wird das vorhandene Interesse der Probanden an der Technologie deutlich. A1 sowie A2 zeigen zudem, dass die Lernkurve der Interaktionsform relativ kurz ist. Es war den Probanden sofort möglich, *HoloUI* zu bedienen.

#### 4.2.2 Teil II

In Teil II hatten 6 Personen im Alter von 26 bis 33 Jahren die Möglichkeit, *HoloUI* ausgiebiger zu testen. Die Probanden forschten zum Teil selber an interaktiven Umgebungen und sind mit der Technologie vertraut. Die Personengruppen aus Teil I und Teil II überschneiden sich nicht.

Dieser Teil untersucht die Fehleranfälligkeit und Geschwindigkeit der Bedienung. Die Probanden absolvierten dazu eine vorgegebene Interaktionsabfolge. Zusätzlich haben die Probanden eine Likert-Skala bewertet, die auf den Aspekten einer Weiterentwicklung des von Davis, Bagozzi und Warshaw [DBW89] vorgestellten Technology Acceptance Models basiert. Die zu absolvierende Abfolge der Aktionen sieht wie folgt aus:

*Play → Volume + → Next → Next → Volume - → Pause → Previous → Pause*

Die Abfolge wird jeweils viermal mit dem Finger- und dem Kugel-Handmodell (Abbildung 3.9) durchgeführt, um zu sehen, ob die Ergebnisse abweichen. Abbildung 4.3 zeigt das Aktivieren eines Elementes, wie es von den Probanden getan werden muss.

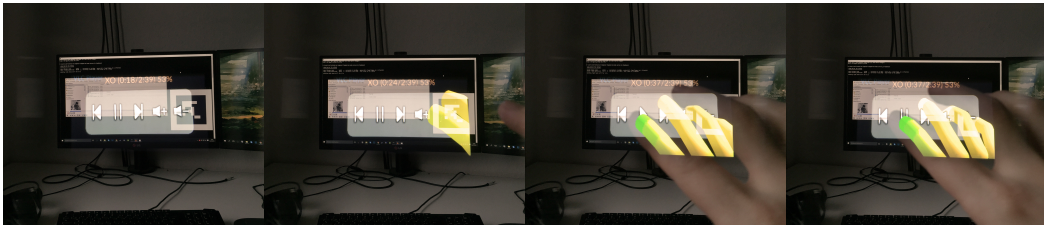


Abbildung 4.3: Auslösen einer Aktion mit dem Finger-Handmodell

### **Fehleranfälligkeit der Bedienung**

Es wurden die Fehler jedes Probanden gezählt, die während der Interaktionsabfolge aufgetreten sind. Als Fehler gilt das unbeabsichtigte Auslösen einer falschen Aktion in der Abfolge. Abbildung 4.4 zeigt pro Durchlauf die Verteilung der Fehler aller Benutzer, die mit dem jeweiligen Handmodell aufgetreten sind.

Die beobachtete Fehlerrate ist sehr gering. Zwei Ausreißer im ersten Durchlauf zeigen, dass 3 oder mehr Fehler vorgekommen sind. Es wurden im Durchschnitt ein Fehler oder weniger gemacht, da der Median des ersten zweiten und vierten Durchlaufes jeweils 1 oder weniger ist. Die Fehlerrate ist bei dem Finger-Handmodell geringfügig größer ausgefallen, was durch die generell höher reichende obere Kante der Verteilung gezeigt wird.

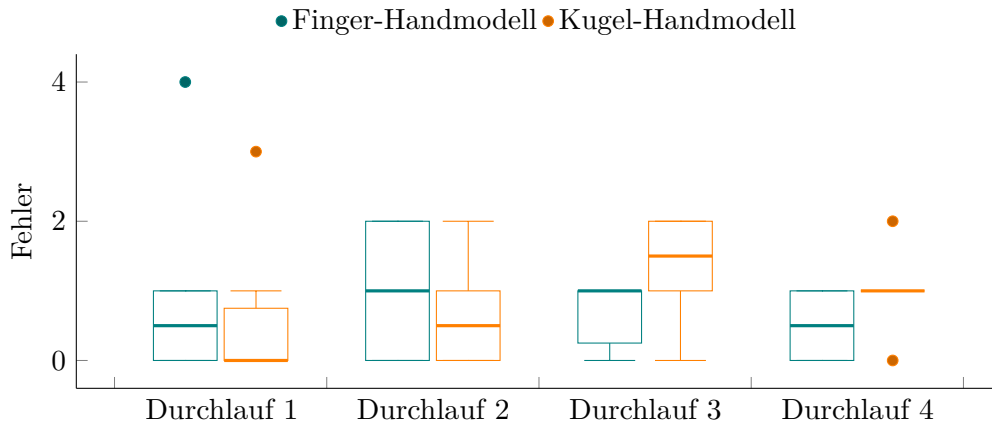


Abbildung 4.4: Übersicht der beobachteten Fehler

### Bediengeschwindigkeit

Die Bediengeschwindigkeit wurde parallel zu den Fehlern eines Benutzers gemessen. Abbildung 4.5 zeigt die durchschnittliche gemessene Zeit, die jeder Proband zum ungezwungenen Auswählen einer Aktion pro Durchlauf benötigt hat.

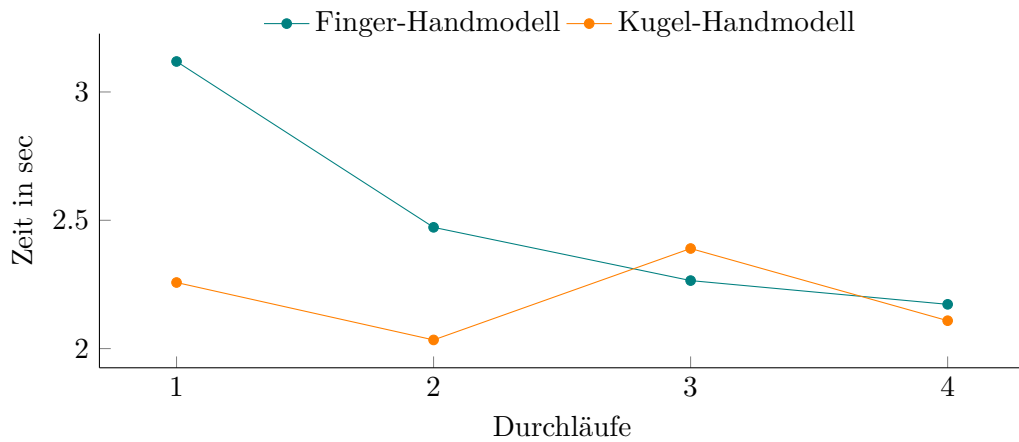



Abbildung 4.5: Übersicht der beobachteten Zeit

Die Linie des Finger-Handmodells zeigt anfängliche Schwierigkeiten der Benutzer, sich mit dem Finger-Handmodell zu orientieren. Allerdings fällt die benötigte Zeit mit jedem weiteren Durchlauf, wodurch gezeigt wird, dass sich die Benutzer schnell an die Bedienung und das Handmodell gewöhnen. Im Vergleich dazu konnten die Benutzer mit

dem -Kugel-Handmodell sofort zügig die Abfolge absolvieren. Die durchschnittliche Zeit, um eine Aktion auszulösen, liegt zu Beginn der Messung bei ~2,7 Sekunden und betrug nach dem vierten Durchgang nur noch ~2,2 Sekunden.

### Technologieakzeptanz

Die Likert-Skala wurde nach dem Absolvieren der Abfolge durch die Probanden bewertet. Dabei wurden Aussagen über vier Kriterien des Technology Acceptance Model und eine bezüglich des Kugel-Handmodells getroffen. Die Bewertung erfolgte dabei wie in Abschnitt 4.2.1. Im folgenden sind die Aussagen aufgelistet:

PE: perceived enjoyment

PU: perceived usefulness

PEOU: perceived ease of use

IOU: intention to use

A6: Das Kugel-Handmodell Handmodell ist hilfreicher als

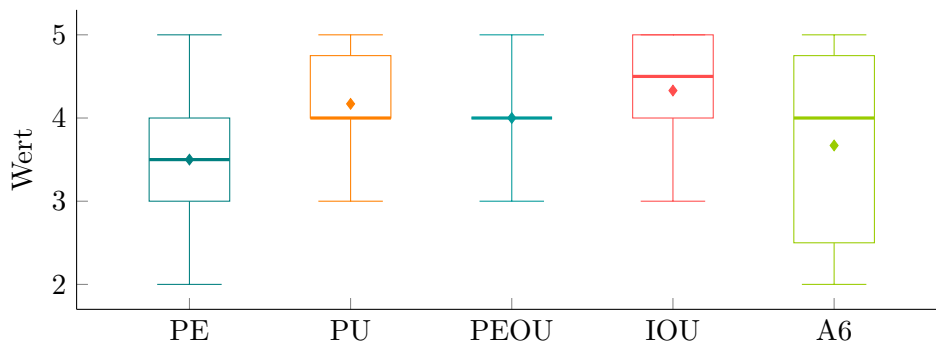


Abbildung 4.6: Auswertung der zweiten Likert-Skala

Abbildung 4.6 zeigt die Auswertung der Skala. Die gemessenen Fehler in der Interaktionsabfolge (Abbildung 4.4) spiegeln sich in dem wahrgenommenem Vergnügen(PE) wider. Dies ist hauptsächlich dem Leap Motion Sensor zuzuordnen, der Schwierigkeiten hat, eine geballte Hand zu erkennen, bei der nur der Zeigefinger ausgestreckt ist. Viele Probanden haben diese Haltung der Hand verwendet.

Der Median sowie der Durchschnitt liegen jeweils bei der wahrgenommenen Nützlichkeit(PU) und der wahrgenommenen Anwenderfreundlichkeit(PEOU) bei 4. Der Absicht, das Interaktionskonzept selbst nutzen zu wollen(IOU), wurde ebenfalls zugestimmt.

Im Vergleich der beiden Handmodelle durch die Aussage A6 ergibt sich eine Tendenz zu dem Kugel-Handmodell, das die virtuelle Oberfläche weniger verdeckt. Allerdings geht bei dem Kugel-Handmodell das Gefühl für die eigene Handausrichtung etwas verloren, die das Finger-Handmodell dem Nutzer unmerklich vermittelt. Dadurch liegt die Fehlerrate des Kugel-Handmodells etwas höher (Abbildung 4.4). Die Wahl der Handrepräsentation hängt von dem Bedienkomfort ab, den der Benutzer erfährt. Ob das Kugel-Handmodell grundsätzlich eine einfachere Bedienung ermöglicht, kann nicht geschlussfolgert werden.

### 4.3 Fazit

In diesem Kapitel wurde untersucht, wie sich das Zielsystem in der Praxis verhält. Anhand der Zeitmessungen wurde gezeigt, dass *HoloUI* schnell genug arbeitet, um eine flüssige Bedienung zu ermöglichen. Dabei wurde festgestellt, dass der Benutzer nach ungefähr einer halben Sekunde das Benutzerinterface zu einem bedienbaren Gerät aktivieren kann.

Durch die Benutzerstudie wurde zudem gezeigt, dass die Technologie und das System auf Interesse bei den Probanden gestoßen sind. Weiter zeigte sich, dass der Umgang mit *HoloUI* intuitiv genug ist, um in kürzester Zeit die Bedienung nutzen zu können. Dies wurde durch Teil II der Studie unterstützt. Außerdem wurde eine geringe Fehlerrate bei der Bedienung festgestellt und die Probanden konnten in kurzer Zeit präzise gewünschte Aktionen auslösen.



## 5 Zusammenfassung

In dieser Arbeit wurde ein System entwickelt, das selbstständig bedienbare Geräte erkennt und dem Benutzer ermöglicht, diese über ein Benutzerinterface in der Luft zu steuern. *HoloUI* stellt Bedienelemente und Informationen von Geräten ähnlich eines Hologramms vor dem Benutzer im Raum dar. Bedienelemente lassen sich durch Antippen mit dem Zeigefinger aktivieren. Eine virtuelle Handrepräsentation imitiert dazu die reale Hand des Benutzers und überträgt so die Geste in die virtuelle Umgebung.

In Kapitel 1 wurde über Visionen von Sci-Fi-Autoren in das Thema Human Computer Interaction eingeleitet. Abschnitt 1.1 befasste sich anschließend mit aktuellen Trends aus der Forschung und zeigte Visionen, wie der Mensch in Zukunft mit Maschinen und seiner Umgebung interagieren wird. Diese Entwicklungen wurden als Grundlage für die Formulierung der Ausrichtung und des Ziels dieser Arbeit in Abschnitt 1.2 herangezogen. Die Struktur der Arbeit wurde in Abschnitt 1.3 festgelegt.

Kapitel 2 befasste sich mit der Einordnung der Arbeit in den wissenschaftlichen Kontext und stellt kommerzielle Kernentwicklungen im Bereich Human Computer Interaction vor. Da eine eindeutige Kategorisierung der vielseitigen Entwicklungen nicht möglich gewesen ist, ging der Vorstellung der wissenschaftlichen Arbeiten in Abschnitt 2.2 eine Gliederung der grundlegenden Verfahren freihändiger Interaktion und Augmented Reality in Abschnitt 2.1 voraus. Die wissenschaftlichen Arbeiten wurden auf drei relevante Kriterien hin untersucht: *Interaktion*, *Visualisierung* und *Objekterkennung/-bedienung*. Abschnitt 2.3 zeigte anschließend kommerzielle Produkte, die für diese Arbeit interessant sind. Der Markt bot zu Beginn dieser Arbeit kein Produkt an, das die nötige Hardware für diese Arbeit vereint, weshalb ein Kompromiss gefunden werden musste. Das Kapitel endet in Abschnitt 2.4 mit der Schlussfolgerung, eine Kombination aus Leap Motion und Moverio BT-200 sowie Geräteidentifikation durch Referenzmarkierungen als Grundlage für diese Arbeit zu verwenden.

In Kapitel 3 wurden zunächst die Anforderungen an das Zielsystem in Abschnitt 3.1 konkretisiert. Abschnitt 3.2 hat sich mit der Zusammenlegung der Hardware befasst, so dass *HoloUI* als einheitliches System genutzt werden kann. Die Konzeption und die Realisierung der einzelnen Softwarekomponenten wurden in Abschnitt 3.3 beschrieben. Die Tatsache, dass sich Leap Motion zum Zeitpunkt dieser Arbeit nicht direkt durch die Moverio ansteuern ließ, war ausschlaggebend für den Kompromiss die *Hand Tracking*-Komponente auf ein externes Notebook auszulagern. Details zur Implementation der Applikationen wurden in Abschnitt 3.4 erwähnt. Das Kapitel endet in Abschnitt 3.5 mit einer Zusammenfassung der realisierten Funktionalitäten, die durch das Zielsystem bereitgestellt werden.

Der Prototyp wurde in Kapitel 4 evaluiert. Dazu wurde in Abschnitt 4.1 eine Reaktionszeit von  $\sim 0.5$  Sekunden für die Betriebsbereitschaft,  $\sim 0.2$  Sekunden für das Auslösen einer Geräteaktion und  $\sim 0.15$  Sekunden für die Handimitation gemessen. Abschnitt 4.2 hat sich mit der Akzeptanz des Interaktionskonzepts befasst, und gezeigt das die Technologie begrüßt wird. Eine Interaktionsabfolge die Probanden ausführen mussten zeigte, dass die Bedienung leicht fällt und schnell zur Gewohnheit wird.

### 5.1 Fazit

Mit der Arbeit wurde gezeigt, dass sich die Interaktion von Mensch und Computer definitiv in Richtung freihändiger und kontaktloser Interaktion über Natural User Interfaces entwickeln wird. Auch wenn die heutigen Techniken noch einige Restriktionen haben, wie in Abschnitt 5.2 erwähnt wird, gibt es durchaus ein Bestreben in der Forschung und gleichermaßen in der Industrie, mit Produkten wie der HoloLens oder MagicLeap (siehe Abschnitt 2.3) dem Menschen neue Interaktionskonzepte für eine intelligente Umgebung zu bieten. Dabei überschreiten die in Werbungen gezeigten Möglichkeiten bei Weitem den aktuellen Stand der Technik. In Kapitel 4 wurde gezeigt, dass Benutzer sich schnell an das neue Konzept der Bedienung gewöhnen können und diesem offen gegenüberstehen. Nicht zuletzt werden die zunehmenden Geräte in unserem Umfeld eine neue Modalität fordern, um die Übersicht und Kontrolle zu behalten. Augmented Reality und Natural User Interfaces sind die Technologien, die dies ermöglichen.

## 5.2 Probleme

Die in dieser Arbeit aufgetretenen Probleme lassen sich auf die verwendete Technik zurückführen bzw. sind durch die Technologie an sich bedingt.

### Tiefenwahrnehmung

Die Tiefenwahrnehmung in Augmented Reality-Umgebungen ist besonders wichtig. Es fällt dem Benutzer allerdings schwer, ein Gefühl für diese zu bekommen, da wesentliche Sinneseindrücke, die man in der physischen Welt hat, fehlen. Besonders das Wissen über Relationen von Größen und Abständen hilft in der echten Welt, Tiefe abzuschätzen. *HoloUI* stellt Elemente frei von jeglichem Bezug auf die reale Umwelt mitten im Raum dar. Die einzige Relation, die der Benutzer unbewusst erfährt, ist die Größe seiner Hand. Diese wird millimetergenau auf die virtuelle Hand übertragen. Allerdings sinkt der Informationswert dadurch, dass die virtuelle Hand nicht genau die reale verdeckt, was mitunter dem Field of View der Moverio geschuldet ist.

### Schmales Field of View

Durch das schmale Sichtfeld der Moverio BT-200 von 23° konnten die für die Stereoskopie relevanten Positionen und das Sichtfeld der virtuellen Kameras nicht optimal umgesetzt werden. Dies hat zur Folge, dass sich die virtuelle Umgebung für den Betrachter nicht exakt über die reale legt. Zum Zeitpunkt dieser Arbeit bieten andere Head-Mounted Displays allerdings keinen größeren Sichtbereich.

## 5.3 Fortführung

Diese Arbeit hat sich auf die Umsetzung eines funktionalen Prototyps konzentriert, der eine einfache Bedienung von Geräten über ein Benutzerinterface in der Luft ermöglicht, weshalb nicht alle Konzepte in ausgearbeiteter Form ihre Anwendung gefunden haben. Mögliche Erweiterungen werden im Folgenden beschrieben.

### Feedback

Sämtlichen Interaktionen in Augmented Reality-Umgebungen mangelt es an taktilem Feedback. Der Benutzer wird meist durch visuelle Mittel auf eine Aktion hingewiesen. Um dem Benutzer beispielsweise die Entfernung zwischen seinem Finger und einem Bedienelement zu verdeutlichen, könnte ein vibrierender Ring an dem Finger helfen.

Innerhalb instrumentalisierter Umgebungen haben Sodhi u. a. [Sod+13] einen auf Luftverwirbelungen basierenden Ansatz gezeigt, während Kruijff, Schmalstieg und Beckhaus [KSB06] auf elektrische Impulse zurückgreifen.

### **Dynamische Gerätekonfiguration**

Zur Zeit müssen Konfigurationsdateien innerhalb der *HoloUI*-Applikation bekannt sein. Eine Idee ist es, die Konfiguration über eine mögliche Datenmatrix am Gerät zu laden und die Referenzmakierung nicht zur alleinigen Identifikation zu nutzen. Huang, Li und Hui [HLH15] zeigen, wie IP-Adresse und Gerätetyp aus einem QR-Code geladen werden. Ein anderer Ansatz wäre, die Konfigurationsdateien zentral durch eine Webapplikation zu verwalten.

### **Virtuelle Elemente**

Das virtuelle Benutzerinterface beschränkt sich auf wenige Icons und Informationen, die dargestellt werden können. Um der Bedienungsmöglichkeit unterschiedlicher Gerätetypen gerecht zu werden und in der Darstellung nicht an ein einheitliches Design gebunden zu sein, kann *HoloUI* um ein individuelles Benutzerinterface erweitert werden. Dies geht allerdings mit einer Erweiterung des Konfigurationsschemas, siehe Abschnitt 3.3.1, einher, das die virtuellen Elemente des Gerätes beschreibt.

### **Nicht-vernetzte Geräte**

*HoloUI* beschränkt sich bisher auf die Bedienung von vernetzten Geräten und Applikationen, die auf XML Responses basieren und über eine HTTP API verfügen. Eine Herausforderung stellen Geräte dar, die über keine direkte Netzwerkanbindung verfügen. Um diese zu bedienen, haben Ullah u. a. [Ull+12] mit X10-Modulen und Han und Lim [HL10] mit ZigBee-Funkmodulen bereits mögliche Ansätze vorgestellt.

## Literatur

- [Arv90] ARVO, James: “Graphics Gems”. In: Hrsg. von GLASSNER, Andrew S. San Diego, CA, USA: Academic Press Professional, Inc., 1990. Kap. A Simple Method for Box-sphere Intersection Testing, S. 335–339. URL: <http://dl.acm.org/citation.cfm?id=90767.90844>.
- [BAM15] BENAVIDES, Xavier ; AMORES, Judith ; MAES, Pattie: “Invisibilia: Revealing Invisible Data Using Augmented Reality and Internet Connected Devices”. In: *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*. UbiComp/ISWC’15 Adjunct. Osaka, Japan: ACM, 2015, S. 341–344. ISBN: 978-1-4503-3575-1. DOI: 10.1145/2800835.2800882. URL: <http://doi.acm.org/10.1145/2800835.2800882>.
- [BLS05] BONANNI, Leonardo ; LEE, Chia-Hsun ; SELKER, Ted: “CounterIntelligence: Augmented reality kitchen”. In: *Proc. CHI*. Bd. 2239. Citeseer. 2005, S. 44.
- [Bro11] BROOKER, Charlie: *Black Mirror, Episode: Das Leben als Spiel*. UK: Channel 4. UK, 2011.
- [BSS13] BRUDER, G. ; STEINICKE, F. ; STURZLINGER, W.: “Effects of visual conflicts on 3D selection task performance in stereoscopic display environments”. In: *3D User Interfaces (3DUI), 2013 IEEE Symposium on*. März 2013, S. 115–118. DOI: 10.1109/3DUI.2013.6550207.
- [CNM83] CARD, Stuart K. ; NEWELL, Allen ; MORAN, Thomas P.: *The Psychology of Human-Computer Interaction*. Hillsdale, NJ, USA: L. Erlbaum Associates Inc., 1983. ISBN: 0898592437.
- [Coh+11] COHN, Gabe u. a.: “Your Noise is My Command: Sensing Gestures Using the Body As an Antenna”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’11. Vancouver, BC, Canada: ACM, 2011,

- S. 791–800. ISBN: 978-1-4503-0228-9. DOI: 10.1145/1978942.1979058. URL: <http://doi.acm.org/10.1145/1978942.1979058>.
- [Col] COLGAN, Alex: *The Alignment Problem: Building Augmented Reality*. <http://blog.leapmotion.com/alignment-problem-position-cameras-augmented-reality/>. (Visited on 01/12/2016).
- [DBW89] DAVIS, Fred D. ; BAGOZZI, Richard P. ; WARSHAW, Paul R.: “User Acceptance of Computer Technology: A Comparison of Two Theoretical Models”. In: *Management Science* 35.8 (1989), S. 982–1003. DOI: 10.1287/mnsc.35.8.982. eprint: <http://dx.doi.org/10.1287/mnsc.35.8.982>. URL: <http://dx.doi.org/10.1287/mnsc.35.8.982>.
- [Ebe+12] EBERT, Lars C. u. a.: “You Can’t Touch This: Touch-free Navigation Through Radiological Images”. In: *Surgical Innovation* 19.3 (2012), S. 301–307. DOI: 10.1177/1553350611425508. eprint: <http://sri.sagepub.com/content/19/3/301.full.pdf+html>. URL: <http://sri.sagepub.com/content/19/3/301.abstract>.
- [Fav08] FAVREAU, Jon: *Ironman*. Deutschland: Concorde. USA, 2008.
- [HBW11] HARRISON, Chris ; BENKO, Hrvoje ; WILSON, Andrew D.: “OmniTouch: Wearable Multitouch Interaction Everywhere”. In: *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*. UIST ’11. Santa Barbara, California, USA: ACM, 2011, S. 441–450. ISBN: 978-1-4503-0716-1. DOI: 10.1145/2047196.2047255. URL: <http://doi.acm.org/10.1145/2047196.2047255>.
- [He] HE, Jimmy: *VR Essentials: What You Need to Build from Scratch*. [Stand 01. Januar 2016]. URL: <http://blog.leapmotion.com/vr-essentials-need-build-scratch>.
- [HL10] HAN, Dae-Man ; LIM, Jae-Hyun: “Smart home energy management system using IEEE 802.15.4 and zigbee”. In: *Consumer Electronics, IEEE Transactions on* 56.3 (Aug. 2010), S. 1403–1410. ISSN: 0098-3063. DOI: 10.1109/TCE.2010.5606276.
- [HLH15] HUANG, Zhanpeng ; LI, Weikai ; HUI, Pan: “Ubii: Towards Seamless Interaction Between Digital and Physical Worlds”. In: *Proceedings of the 23rd ACM International Conference on Multimedia*. MM ’15. Brisbane, Australia: ACM, 2015, S. 341–350. ISBN: 978-1-4503-3459-4. DOI: 10.1145/2733373.2806266. URL: <http://doi.acm.org/10.1145/2733373.2806266>.

- [I2E] I2E: *I2E - Immersive Interactive Environments*. [Stand 23. November 2015]. URL: <http://i2e.informatik.haw-hamburg.de/>.
- [Jal+13] JALALINIYA, Shahram u. a.: “Touch-less interaction with medical images using hand & foot gestures”. In: *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication*. UbiComp '13 Adjunct. Zurich, Switzerland: ACM, 2013, S. 1265–1274. ISBN: 978-1-4503-2215-7. DOI: 10.1145/2494091.2497332. URL: <http://doi.acm.org/10.1145/2494091.2497332>.
- [Jon+14] JONES, Brett u. a.: “RoomAlive: Magical Experiences Enabled by Scalable, Adaptive Projector-camera Units”. In: *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*. UIST '14. Honolulu, Hawaii, USA: ACM, 2014, S. 637–644. ISBN: 978-1-4503-3069-5. DOI: 10.1145/2642918.2647383. URL: <http://doi.acm.org/10.1145/2642918.2647383>.
- [JS07] JAIMES, Alejandro ; SEBE, Nicu: “Multimodal human–computer interaction: A survey”. In: *Computer vision and image understanding* 108.1 (2007), S. 116–134.
- [KSB06] KRUIJFF, Ernst ; SCHMALSTIEG, Dieter ; BECKHAUS, Steffi: “Using Neuro-muscular Electrical Stimulation for Pseudo-haptic Feedback”. In: *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*. VRST '06. Limassol, Cyprus: ACM, 2006, S. 316–319. ISBN: 1-59593-321-2. DOI: 10.1145/1180495.1180558. URL: <http://doi.acm.org/10.1145/1180495.1180558>.
- [Lea] LEAP, Motion: *Leap Motion - 3D Motion and Gesture Control for Virtual Reality*. [Stand 5. Dezember 2015]. URL: <https://www.leapmotion.com/product/vr>.
- [Lee+14] LEE, Jae-Young u. a.: “Design and implementation of an augmented reality system using gaze interaction”. English. In: *Multimedia Tools and Applications* 68.2 (2014), S. 265–280. ISSN: 1380-7501. DOI: 10.1007/s11042-011-0944-5. URL: <http://dx.doi.org/10.1007/s11042-011-0944-5>.
- [Mem] MEMORYMIRROR: *Memomi - MemoryMirror*. [Stand 12. Dezember 2015]. URL: <http://memorymirror.com/>.

- [Mica] MICROSOFT: *Kinect Joy Ride / Xbox Kinect Joy Ride Game / Xbox 360*. [Stand 23. November 2015]. URL: <http://support.xbox.com/en-US/xbox-360/games/kinect-joy-ride-info>.
- [Micb] MICROSOFT: *Microsoft HoloLens*. [Stand 23. November 2015].
- [MK94] MILGRAM, Paul ; KISHINO, Fumio: "A taxonomy of mixed reality visual displays". In: *IEICE TRANSACTIONS on Information and Systems* 77.12 (1994), S. 1321–1329.
- [Nav+12] NAVAB, Nassir u. a.: "First Deployments of Augmented Reality in Operating Rooms". In: *Computer* 45.7 (2012), S. 48–55. ISSN: 0018-9162. DOI: <http://doi.ieeecomputersociety.org/10.1109/MC.2012.75>.
- [Ocu] OCULUS: *Oculus Rift*. [Stand 23. November 2015]. URL: <https://www.oculus.com/en-us/rift/>.
- [Pu+13] PU, Qifan u. a.: "Whole-home Gesture Recognition Using Wireless Signals". In: *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking*. MobiCom '13. Miami, Florida, USA: ACM, 2013, S. 27–38. ISBN: 978-1-4503-1999-7. DOI: 10.1145/2500423.2500436. URL: <http://doi.acm.org/10.1145/2500423.2500436>.
- [Sha+15] SHARP, Toby u. a.: "Accurate, Robust, and Flexible Real-time Hand Tracking". In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. CHI '15. Seoul, Republic of Korea: ACM, 2015, S. 3633–3642. ISBN: 978-1-4503-3145-6. DOI: 10.1145/2702123.2702179. URL: <http://doi.acm.org/10.1145/2702123.2702179>.
- [Sod+13] SODHI, Rajinder u. a.: "AIREAL: Interactive Tactile Experiences in Free Air". In: *ACM Trans. Graph.* 32.4 (Juli 2013), 134:1–134:10. ISSN: 0730-0301. DOI: 10.1145/2461912.2462007. URL: <http://doi.acm.org/10.1145/2461912.2462007>.
- [Spi02] SPIELBERG, Steven: *Minority Report*. Deutschland: 20th Century Fox Home Entertainment. USA, 2002.
- [TP15] TAKEUCHI, Maria ; PHILLIPS, Frederico: *as.phyx.i.a.* 2015. URL: <http://www.asphyxia-project.com/>.



- [Ull+12] ULLAH, A.M. u. a.: “Remote-touch: Augmented reality based marker tracking for smart home control”. In: *Computer and Information Technology (ICCIT), 2012 15th International Conference on*. Dez. 2012, S. 473–477. DOI: 10.1109/ICCITechn.2012.6509774.
- [Vee13] VEENHOF, Sander: *INSTANT SCULPTURE GARDEN*. <http://www.sndrv.nl/instantsculpturegarden/>. 2013.
- [Vol15] VOLKSWAGENAG: *Volkswagen Konzern Einsatz von Spatial Augmented Reality in der Automobilindustrie*. [Stand 23. November 2015]. 2015. URL: [http://www.volkswagenag.com/content/vwcorp/content/de/innovation/Virtual\\_technologies/Use\\_of\\_spatial\\_augmented\\_reality\\_in\\_the\\_automotive\\_industry.html](http://www.volkswagenag.com/content/vwcorp/content/de/innovation/Virtual_technologies/Use_of_spatial_augmented_reality_in_the_automotive_industry.html).
- [Wei95] WEISER, Mark: “Human-computer Interaction”. In: Hrsg. von BAECKER, Ronald M. u. a. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995. Kap. The Computer for the 21st Century, S. 933–940. ISBN: 1-55860-246-1. URL: <http://dl.acm.org/citation.cfm?id=212925.213017>.
- [Wex14] WEXELBLAT, Alan: *Virtual reality: applications and explorations*. Academic Press, 2014.
- [Wig+07] WIGDOR, Daniel u. a.: “Lucid Touch: A See-through Mobile Device”. In: *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*. UIST '07. Newport, Rhode Island, USA: ACM, 2007, S. 269–278. ISBN: 978-1-59593-679-0. DOI: 10.1145/1294211.1294259. URL: <http://doi.acm.org/10.1145/1294211.1294259>.
- [Wik15] WIKIPEDIA: *Laser harp* — *Wikipedia, The Free Encyclopedia*. [Stand 23. November 2015]. 2015. URL: [https://en.wikipedia.org/w/index.php?title=Laser\\_harp&oldid=697451128](https://en.wikipedia.org/w/index.php?title=Laser_harp&oldid=697451128).
- [Xie+13] XIE, Lei u. a.: “iFridge: An Intelligent Fridge for Food Management Based on RFID Technology”. In: *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication*. UbiComp '13 Adjunct. Zurich, Switzerland: ACM, 2013, S. 291–294. ISBN: 978-1-4503-2215-7. DOI: 10.1145/2494091.2494181. URL: <http://doi.acm.org/10.1145/2494091.2494181>.
- [You] YOUTUBE: *Most Insane Immersive Movie Experience EVER, Part 1*. (Stand 13. Dezember 2015). URL: <https://www.youtube.com/watch?v=VrgWH1KUDt4>.

- [ZDB08] ZHOU, Feng ; DUH, H.B.-L. ; BILLINGHURST, M.: “Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR”. In: *Mixed and Augmented Reality, 2008. ISMAR 2008. 7th IEEE/ACM International Symposium on*. Sep. 2008, S. 193–202. DOI: 10.1109/ISMAR.2008.4637362.

*Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.*

Hamburg, 23. Januar 2016

---

Jonathan Wischhusen