



UNIVERSITÄT
KOBLENZ · LANDAU

Fachbereich 4: Informatik

Interaktion mit ubiquitären User-Interfaces

Masterarbeit

zur Erlangung des Grades eines Master of Science (M.Sc.)
im Studiengang Computervisualistik

vorgelegt von
Andreas Hellwig

Erstgutachter: Prof. Dr.-Ing. Stefan Müller
(Institut für Computervisualistik, AG Computergraphik)

Zweitgutachter: Dipl.-Inform. Dominik Grüntjens
(Institut für Computervisualistik, AG Computergraphik)

Koblenz, im Mai 2013



Aufgabenstellung für die Masterarbeit
Andreas Hellwig
(Mat. Nr. 207200040)

Thema: Interaktion mit ubiquitären User-Interfaces

Mobile Geräte haben sich seit ihrer Erfindung mit rasender Geschwindigkeit weiterentwickelt. Vor allem die Art der Nutzung und der Interaktion mit mobilen Geräten hat sich innerhalb kürzester Zeit grundlegend verändert, und mittelfristig werden neue Technologien noch besser auf uns Menschen zugeschnittene Lösungen ermöglichen.

Das aktuelle Projekt „Glass“ der Firma Google zeigt bspw., dass die längst in Vergessenheit geratene Head-Mounted-Display-Technologie noch lange nicht ausgereizt ist und mobile Geräte in Zukunft wahrscheinlich einen immer wichtigeren Platz in unserem Leben einnehmen werden, wobei die Interaktion mit ihnen dabei gleichzeitig immer intuitiver werden wird.

Allerdings bringen solch ubiquitäre User-Interfaces auch neue Herausforderungen mit sich. Wichtige Fragen sind zum Beispiel: Welche Möglichkeiten der Interaktion mit ubiquitären User-Interfaces gibt es? Welche davon eignen sich besser als andere? Wie schafft man es, diese Interaktion in sicherheitskritischen Situationen nicht zu einem Risiko werden zu lassen? Wie können solche Interaktionskonzepte besser in das alltägliche Leben integriert werden?

Ziel dieser Arbeit ist es, eben diese Fragen zu bearbeiten. Hierzu soll ein Konzept einer neuartigen Interaktion mit User-Interfaces entwickelt und evaluiert werden. Darüber hinaus soll eine prototypische Umsetzung mit Hilfe bestehender Head-Mounted-Display-Lösungen realisiert werden. Es soll Benutzern ermöglicht werden, mit ihrem Smartphone über das Head-Mounted-Display zu interagieren, ohne dass das Smartphone dabei aus der Tasche genommen werden muss. Mit Hilfe dieses vollständigen, prototypischen Systems sollen grundlegende Forschungsergebnisse erzielt werden.

Die inhaltlichen Schwerpunkte der Arbeit sind:

1. Recherche und Analyse verwandter Arbeiten
2. Erstellung des Interaktionskonzepts
3. Prototypische Umsetzung
4. Evaluation
5. Dokumentation und Bewertung der Ergebnisse

Koblenz, den 22.10.2012

Zusammenfassung

Diese Arbeit befasst sich mit der Entwicklung und Evaluierung eines Konzepts zur neuartigen Interaktion mit ubiquitären, also allgegenwärtigen, User-Interfaces. Für die Durchführung der Evaluierung dieses Interaktionskonzeptes wurde darüber hinaus eine prototypische Umsetzung einer Augmented Reality-Anwendung mit Hilfe einer bestehender Head-Mounted-Display-Lösung und einem Android-Smartphone realisiert.

Zudem wurde im Rahmen dieser Arbeit ein konkreter Anwendungsfall für diesen Prototyp – das Navigieren durch einen Stadtteil mittels elektronischer Übersichtskarte – entwickelt und als lauffähige Anwendung implementiert, um die Qualität des Interaktionskonzeptes evaluieren zu können. Auf Grundlage dessen wurden sodann wesentliche Forschungserkenntnisse gewonnen.

Abstract

This thesis deals with the development and evaluation of a concept of novel interaction with ubiquitous user interfaces. To accomplish the evaluation of this interaction concept, a prototype was implemented by using an existing head-mounted display solution and an android smartphone.

Furthermore, in the course of this thesis, a concrete use case for this prototype – the navigation through a city block with the aid of an electronic map – was developed and built as an executable application to help evaluate the quality of the interaction concept. Therefore, fundamental research results were achieved.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Zielsetzung	2
1.3	Aufbau dieser Arbeit	2
2	Grundlagen	3
2.1	Augmented Reality	3
2.2	Displays	6
2.2.1	Optical See-Through Displays	6
2.2.2	Video See-Through Displays	7
2.2.3	Head-Mounted Displays	8
2.3	Tracking	11
2.3.1	Technologien und Prinzipien	12
2.3.2	Optisches Tracking im Detail	14
2.3.3	Herausforderungen im praktischen Einsatz	18
2.4	Eingabe und Interaktion	20
2.4.1	Eingabe	21
2.4.2	Interaktion	25
2.4.3	Anforderungen an User-Interfaces	31
3	Konzept	32
3.1	Grundkonzept	32
3.2	Anwendung	35
3.3	Interaktion	38
4	Aufbau des prototypischen Systems	45
4.1	Komponentendiskussion	45
4.1.1	Webcam und Notebook im Rucksack vs. Smartphone-Kamera und Android-Applikation	45
4.1.2	Optical See-Through vs. Video-See-Through	48
4.2	Konstruktion und Aufbau	50
4.2.1	Helm mit Smartphone-Halterung	51
4.2.2	Bildübertragung Smartphone zu Videobrille	52
4.2.3	Marker-Befestigung am Unterarm	52
4.2.4	Gesamtsystem	54
5	Implementierung	56
5.1	Verwendete Bibliotheken	56
5.1.1	Android OS und OpenGL ES	56
5.1.2	Vuforia SDK	56
5.1.3	Google Maps API	60
5.2	Vorbereitungen	61

5.3	Systemüberblick	62
5.4	Gestenerkennung	65
5.4.1	Vorbereitungen	66
5.4.2	Algorithmus	68
5.4.3	Herausforderungen	70
6	Evaluation	75
6.1	Zielsetzung	75
6.2	Planung und Durchführung	75
6.2.1	Auswahl der Probanden	76
6.2.2	Definition des praktischen Benutzertests	76
6.2.3	Erstellung des persönlichen Fragebogens	81
6.2.4	Durchführung der Evaluation	84
6.3	Auswertung	85
6.3.1	Fragenblock 2: Vorwissen und Kenntnisse	86
6.3.2	Fragenblock 3.1: Bearbeitung der Aufgabe und Benutzbarkeit des Systems – Teil 1	87
6.3.3	Fragenblock 3.2: Bearbeitung der Aufgabe und Benutzbarkeit des Systems – Teil 2	89
6.3.4	Fragenblock 3.3: Position des Markers	90
6.3.5	Fragenblock 4: Tragen des Gesamtsystems	91
6.3.6	Fragenblock 5.1: Potential des evaluierten Systems	93
6.3.7	Fragenblock 5.2: Potential eines optimalen Systems	94
6.3.8	Fragenblock 6: Allgemeine Fragen	95
7	Fazit und Ausblick	98
7.1	Fazit	98
7.2	Ausblick	100
	Literatur	102
A	Fragebogen zur Evaluation	106
B	Checkliste zur Evaluation	111
C	Auswertung der Evaluation – Rohdaten	114

Abbildungsverzeichnis

1.1	Google's Project Glass	1
2.1	Vereinfachte Darstellung des Reality-Virtuality Continuums nach Milgram et al.	4
2.2	Augmented Reality Web-Anwendung <i>Social Shopper</i>	5
2.3	Augmented Reality Game Demo für das Smartphone	5
2.4	i-Glasses High Resolution Head Mounted Display	8
2.5	Verschiebung der fokalen Ebene bei Head-Mounted Displays	9
2.6	AiRScouter, ein Virtual Retinal Display von Brother	10
2.7	Prinzip der Projektion von Bildern auf die Retina mittels VRDs	10
2.8	Funktionsweise des Virtual Retinal Displays AiRScouter	11
2.9	Prinzipien des Trackings	13
2.10	Beispiel eines reflektierenden Baum-Markers	15
2.11	Flachmarker des ARToolKit	16
2.12	Würfel aus Markern des ARToolKit	18
2.13	Die Tischoberfläche wird als virtuelles Objekt ebenfalls gerendert, um den linke Marker möglichst zu verstecken	19
2.14	Visueller Hinweis auf mögliche Geste	22
2.15	Reactable – ein Tangible User-Interface	23
2.16	Der Datenhandschuh CyberGlove II	24
2.17	Das Personal Interaction Panel	29
3.1	Konzeptentwurf: Visuelle Informationen überlagern Handgelenk des Benutzers	35
3.2	Erster Konzeptentwurf: Navigationsanwendung	38
3.3	1. Entwurf eines Interaktionskonzeptes: Buttons zur Manipulation des Kartenausschnitts	39
3.4	2. Entwurf des Interaktionskonzeptes: Manipulation des Kartenausschnitts ohne Buttons	41
3.5	Veranschaulichung einer Pan-Geste	42
3.6	Veranschaulichung einer Rotate-Geste	43
3.7	Veranschaulichung einer Pinch-Close-Geste	44
3.8	Veranschaulichung einer Pinch-Open-Geste	44
4.1	Konstruktion eines AR-Systems mittels Notebook, Rucksack, AR-Brille und Webcam	46
4.2	Konstruktion eines AR-Systems mittels Smartphone und AR-Brille	48
4.3	Die Videobrille Sony Glasstron PLM-A55	50
4.4	Fahrradhelm mit Smartphone-Halterung	51
4.5	Smartphone-Halterung im Detail	52
4.6	Ligawo HDMI zu Composite/S-Video AV-Konverter	53
4.7	Konstruktion zur Befestigung eines Papier-Markers am Unterarm	53
4.8	Prototypisches Gesamtsystem – frontal	54

4.9	Prototypisches Gesamtsystem – seitlich	55
5.1	Von Vuforia SDK unterstützter Frame Marker	58
5.2	Von Vuforia SDK unterstütztes Image Target	58
5.3	Image Target mit visuell markierten Virtual Buttons	59
5.4	Beispielhafte Verwendung von Virtual Buttons zur Interaktion des Benutzers mit dem System	60
5.5	Übersicht über die Komponenten einer mit dem Vuforia SDK entwickelten mobilen Applikation	62
5.6	Finales Bild zur Verwendung als Image Target	63
5.7	Erste Gehversuche der Implementation: Animierte Digitaluhr als 3D-Objekt überlagert Image Target	63
5.8	Aktivitätsdiagramm eines Render-Durchlaufs der Applikation	64
5.9	Kartenausschnitt überlagert Image Target, noch ohne Möglichkeiten zur Manipulation des Kartenausschnitts	66
5.10	Image Target mit visuellen Überlagerungen entsprechend dem definierten Raster aus Virtual Buttons und deren Status, verdeckte Virtual Buttons werden grün angezeigt, nicht verdeckte rot	67
5.11	Eigens zur Unterstützung der Entwicklung entstandene Hilfsapplikation	68
5.12	Trigger-Symbol für Pan-Gesten	72
5.13	Trigger-Symbol für Rotate- und Pinch-to-zoom-Gesten	72
5.14	Auftauchendes Trigger-Symbol für Pan-Gesten bei Marker-Verdeckung durch einen Finger innerhalb der finalen Applikation	73
5.15	Auftauchendes Trigger-Symbol für Rotate- und Pinch-to-zoom-Gesten bei Marker-Verdeckung durch zwei Finger innerhalb der finalen Applikation	73
6.1	Statischer Kartenausschnitt zur Verwendung während des Benutzertests, Bildgröße: ca. 2000 * 1500 Pixel	77
6.2	Initialer Zustand der Applikation während des Benutzertests, bei erfolgreichem Tracking des Markers	78
6.3	Benutzertest unter Verwendung der Videobrille als Display	79
6.4	Benutzertest unter Verwendung eines Monitors statt der Videobrille als Display	80
6.5	Konstruktion zur Befestigung des Markers an Außenseite des Handgelenks	81
6.6	Konstruktion zur Befestigung des Markers an Innenseite des Handgelenks	82
6.7	Ergebnisse des zweiten Fragenblocks: Vorwissen und Kenntnisse	87
6.8	Ergebnisse des Fragenblocks 3.1: Bearbeitung der Aufgabe und Benutzbarkeit des Systems – Teil 1	88

6.9	Ergebnisse des Fragenblocks 3.2: Bearbeitung der Aufgabe und Benutzbarkeit des Systems – Teil 2	89
6.10	Ergebnisse des Fragenblocks 3.3: Position des Markers	90
6.11	Ergebnisse des Fragenblocks 4: Tragen des Gesamtsystems	92
6.12	Ergebnisse des Fragenblocks 5.1: Potential des evaluierten Systems	93
6.13	Ergebnisse des Fragenblocks 5.2: Potential eines optimalen Systems	94

1 Einleitung

1.1 Motivation

Mobile Geräte haben sich seit ihrer Erfindung mit rasender Geschwindigkeit weiterentwickelt. Vor allem die Art der Nutzung und der Interaktion mit mobilen Geräten hat sich innerhalb kürzester Zeit grundlegend verändert, und mittelfristig werden neue Technologien noch besser auf uns Menschen zugeschnittene Lösungen ermöglichen.



Abbildung 1.1: Google's Project Glass (Bild verfügbar unter [Goo12])

Das aktuell viel Aufmerksamkeit erregende, sogenannte „Project Glass“ (siehe Abbildung 1.1) der Firma Google zeigt beispielsweise, dass die längst in Vergessenheit geratene Head-Mounted-Display-Technologie noch lange nicht ausgereizt ist und mobile Geräte in Zukunft wahrscheinlich einen immer wichtigeren Platz in unserem Leben einnehmen werden, wobei die Interaktion mit ihnen dabei gleichzeitig immer intuitiver werden wird.

Allerdings bringen solch ubiquitäre User-Interfaces auch neue Herausforderungen mit sich. Wichtige unbeantwortete Fragen sind dabei: Welche Möglichkeiten der Interaktion mit ubiquitären User-Interfaces gibt es? Welche davon eignen sich besser als andere? Wie schafft man es, diese Interaktion in sicherheitskritischen Situationen nicht zu einem Risiko werden zu

lassen? Wie können solche Interaktionskonzepte besser in das alltägliche Leben integriert werden?

1.2 Zielsetzung

Ziel dieser Arbeit ist es, eben diese Fragen zu bearbeiten. Hierzu soll ein Konzept einer neuartigen Interaktion mit User-Interfaces entwickelt und evaluiert werden. Darüber hinaus soll eine prototypische Umsetzung einer Augmented Reality-Anwendung mit Hilfe bestehender Head-Mounted-Display-Lösungen realisiert werden. Es soll Benutzern ermöglicht werden, mit ihrem Smartphone über das Head-Mounted-Display zu interagieren, ohne dass das Smartphone dabei aus der Tasche genommen werden muss. Mit Hilfe dieses vollständigen, prototypischen Systems sollen grundlegende Forschungsergebnisse erzielt werden.

1.3 Aufbau dieser Arbeit

Im nachstehenden Abschnitt 'Grundlagen' wird der Grundstein für das Verständnis dieser Arbeit gelegt, indem die Grundlagen von *Augmented Reality*-Systemen und die wichtigsten Methoden der Interaktion mit solchen Systemen beschrieben werden. Im darauf folgenden Abschnitt 'Konzept' werden das Grundkonzept dieser Arbeit sowie die entworfenen Konzepte für Anwendung und Interaktion vorgestellt. Anschließend werden im Abschnitt 'Aufbau des prototypischen Systems' die Auswahl der Hardware-Komponenten und die Erstellung des Prototyps geschildert, während der darauf folgende Abschnitt 'Implementierung' erläutert, wie und unter Zuhilfenahme welcher vorhandenen Software-Bibliotheken die festgelegten Aufgaben bearbeitet wurden und in welchen Bereichen Probleme auftraten. Im weiteren Verlauf wird im Abschnitt 'Evaluation' das Vorgehen zur Evaluation dargestellt und die gewonnenen Erkenntnisse veranschaulicht und bewertet, woraufhin im Abschnitt 'Fazit und Ausblick' abschließend das Fazit dieser Arbeit und ein Ausblick folgt.

2 Grundlagen

Nachfolgend werden die Grundlagen zum Verständnis der vorliegenden Arbeit gegeben. Während die Technik der Augmented Reality zuerst im Allgemeinen vorgestellt wird, werden anschließend nacheinander die für die Arbeit relevanten Bereiche der Displays, des Trackings und der Eingabe und Interaktion im Kontext der Augmented Reality betrachtet.

2.1 Augmented Reality

Unter *Augmented Reality* (deutsch: Erweiterte Realität) versteht man eine Technik, welche es erlaubt, die reale Welt mit computergenerierten Informationen zu kombinieren und damit die Realitätswahrnehmung zu erweitern. Die computergenerierten Informationen werden dabei meist visuell dargestellt, sodass man unter Augmented Reality in den häufigsten Fällen die Überlagerung von Bildern oder Videos der Realität mit virtuellen Objekten unter Zuhilfenahme von Computergrafik versteht.

Dabei entsteht eine sogenannte *Mixed Reality* (deutsch: Gemischte Realität), welche einen wichtigen Teil des Reality-Virtuality Continuums nach Milgram et al. [MTUK94] (siehe Abbildung 2.1) ausmacht. Bei Letzterem handelt es sich um eine kontinuierliche Skala, welche es erlaubt, Systeme, die die reale Umgebung mit einer virtuellen vermischen, ihrem diesbezüglichen Mischverhältnis zuzuordnen. An einem Extrem des Reality-Virtuality Continuums befindet sich dabei die reale Welt, am anderen Extrem eine komplett virtuelle Umgebung – auch *Virtual Reality* genannt. Während der Begriff *Augmented Virtuality* (zu deutsch etwa: Erweiterte Virtualität) auf der Skala eher auf Seiten des letztgenannten Extrems angesiedelt ist und Systeme beschreibt, welche eine virtuelle Umgebung mit realen Informationen anreichert, befindet sich die Augmented Reality auf der gegenüberliegenden Seite der Skala und beschreibt daher das genaue Gegenteil.

Weiterhin ist anzumerken, dass Augmented Reality – vor allem in der Fachliteratur – häufig dadurch charakterisiert wird, dass zusätzlich die Anforderung einer Interaktion in Echtzeit erfüllt ist (vgl. [Azu97]). Dies bedeutet, dass die computergenerierten Überlagerungen nicht statisch sind, sondern dynamisch auf die reale Umwelt des Augmented Reality-Systems reagieren. Ein Beispiel für eine statische Überlagerung von virtuellen Objekten findet man im Kino wieder: Die Postproduktion von Filmen, bei der nachträglich mit viel Aufwand computergenerierte Objekte in den Film eingefügt werden, ist keine Interaktion in Echtzeit, weil zur Bearbeitung jedes einzelnen Bildes ein relativ großer Zeitraum von bis zu mehreren Ta-

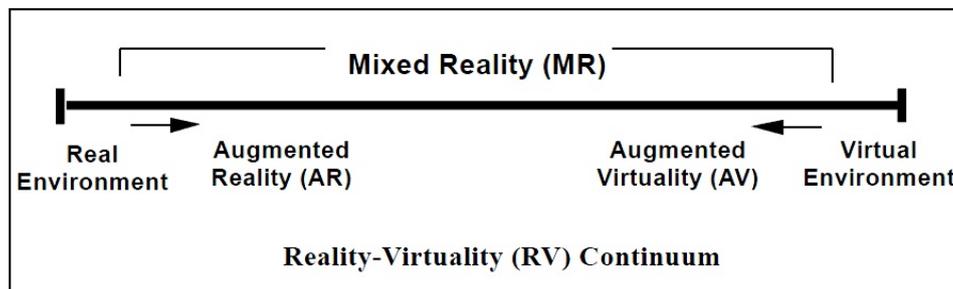


Abbildung 2.1: Vereinfachte Darstellung des Reality-Virtuality Continuums nach Milgram et al. (siehe [MTUK94], S. 283)

gen zur Verfügung steht oder zumindest theoretisch zur Verfügung stehen könnte.

Augmented Reality-Anwendungen hingegen müssen jedoch in der Lage sein, die computergenerierten Überlagerungen – inklusive ihrer meist nötigen perspektivischen korrekten Verzerrung bezogen auf Objekte der realen Welt – in einer Zeit zu berechnen, die geringer ist als der zeitliche Abstand zum nächsten Bild, damit eine flüssig erscheinende Interaktion mit den virtuellen Objekten möglich ist.

Augmented Reality wird heutzutage schon in einigen Bereichen verwendet. Bekannte Beispiele sind die in Echtzeit berechneten und eingeblendeten virtuellen Linien und Markierungen bei Sportübertragungen: Beim Fußball werden beispielsweise bei Freistößen die jeweiligen Entfernungen bis zum Tor angezeigt. Andere Einsatzgebiete sind die des Militärs, der Medizin und der Werbe- sowie Spieleindustrie (siehe Abbildung 2.2 und Abbildung 2.3).

Jedoch könnte Augmented Reality mit fortschreitender Entwicklung der Technologie in unzähligen Bereichen des Alltags zum Einsatz kommen. Monteure könnten sich den nächsten Arbeitsschritt direkt in ihr Sichtfeld einblenden lassen; Soldaten oder Katastrophenhelfer könnten sich Ziele und Gefahrenzonen im Gelände anzeigen lassen und Designer könnten mit tatsächlich und virtuell anwesenden Kollegen am selben dreidimensionalen Modell arbeiten.

Bezüglich der Voraussetzungen bei der Erstellung von Augmented Reality-Anwendungen lässt sich allerdings sagen, dass eine Vielzahl mathematischer und technischer Problemstellungen bearbeitet werden muss, um dem Benutzer ein benutzbares System zu liefern. Diese Problemstellungen lassen sich gut anhand der für Augmented Reality-Anwendungen notwendigen Komponenten beschreiben, welche sich in folgende drei Bereiche einteilen lassen: Darstellung, Tracking und Interaktion. (vgl. [Tön10],

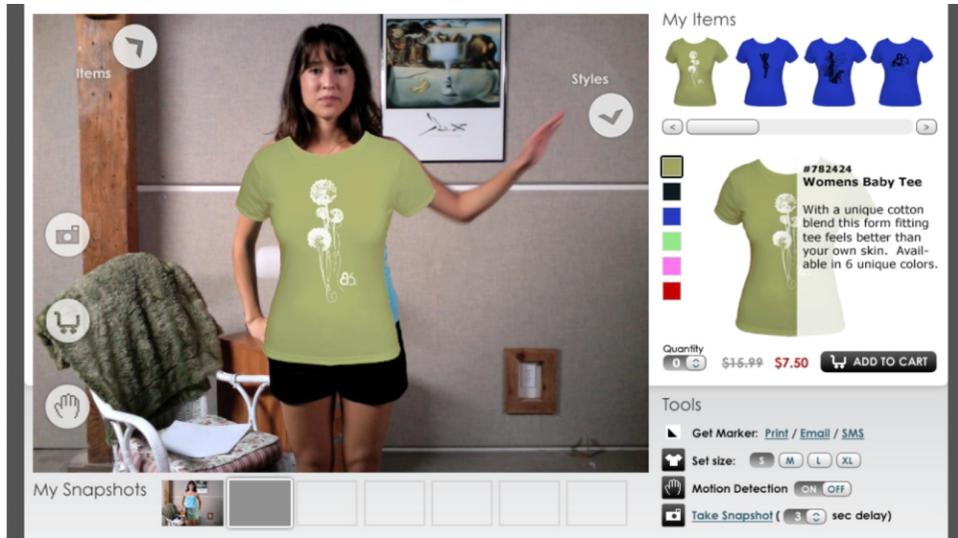


Abbildung 2.2: Augmented Reality Web-Anwendung *Social Shopper* (siehe [Zug12])

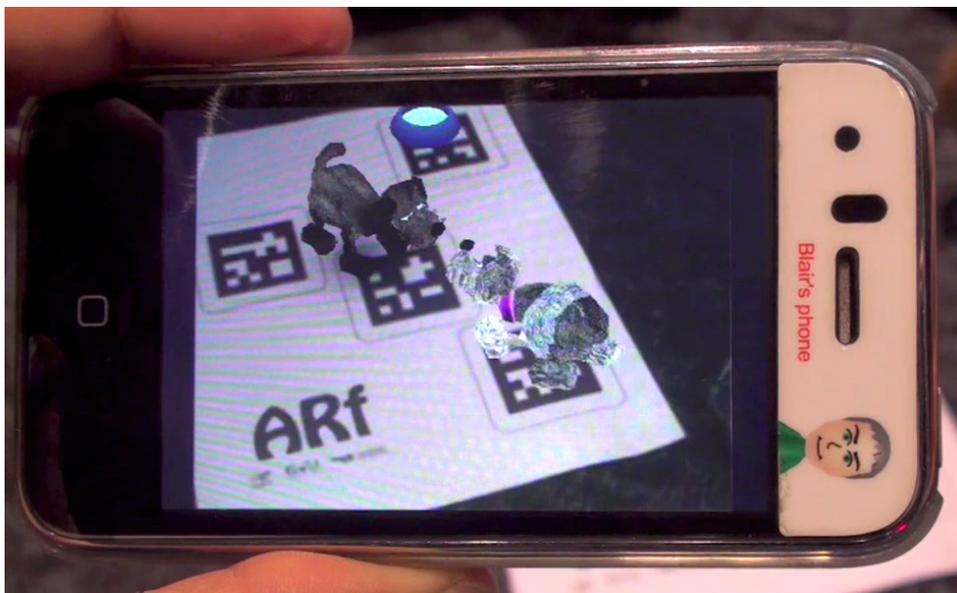


Abbildung 2.3: Augmented Reality Game Demo für das Smartphone (siehe [Tou12])

S. 4)

Die Darstellung virtueller Inhalte erfolgt zumeist optisch, daher werden andere Arten der Darstellung hier nicht näher beleuchtet. Es sei an dieser Stelle jedoch gesagt, dass prinzipiell jeder der menschlichen Sinneskanäle angesprochen werden kann, um Informationen zu transportieren; somit existieren auch im Bereich der Augmented Reality neben der optischen Darstellung sowohl Möglichkeiten als auch Umsetzungen – bereits in allen Bereichen – der folgenden Darstellungsformen:

- akustische Darstellung,
- haptische Darstellung (den Tastsinn betreffend),
- gustatorische Darstellung (den Geschmackssinn betreffend) und
- olfaktorische Darstellung (den Geruchssinn betreffend).

Der Bereich der optischen Darstellung lässt sich wiederum in die Bereiche *Software* und *Displays* einteilen. Im Rahmen dieser Arbeit wird der Bereich *Software*, bezogen auf die Darstellung der virtuellen Inhalte, jedoch nicht weitergehend behandelt; auch wird das Tracking nicht in seiner möglichen Tiefe behandelt, da der Fokus auf der Interaktion liegt.

2.2 Displays

Displays zur Anzeige von Augmented Reality-Anwendungen können Daten- bzw. Computerbrillen, Smartphones, konventionelle Monitore oder Videoprojektionsleinwände sein. Trotz dieser unterschiedlichen Display-Arten gibt es nur zwei Prinzipien, die beschreiben, wie die Realität mit virtuellen Objekten überlagert werden kann: Entweder wird das freie Sichtfeld des Benutzers oder aber nur ein Videobild dessen überlagert. Die entsprechenden Displays werden *Optical See-Through* bzw. *Video See-Through* Displays genannt und nachfolgend jeweils genauer betrachtet.

Auf den genannten Display-Prinzipien basierend existieren darüber hinaus eingangs erwähnte Display-Arten, die je nach Anwendungsfall eingesetzt werden. Diese lassen sich nach Tönnis (vgl. [Tön10], S. 22ff.) in sogenannte *Head-Mounted Displays*, *raum- oder umgebungsfixierte Displays*, *bewegliche Displays* und *Handheld Displays* unterteilen. Im Folgenden soll einzig die Gattung der *Head-Mounted Displays* genauer betrachtet werden.

2.2.1 Optical See-Through Displays

Optical See-Through Displays ermöglichen den direkten Blick auf die umgebende reale Welt; es kommt ein halbdurchlässiger Spiegel – der soge-

nannte *Combiner* – zum Einsatz, durch den gleichzeitig ein Computerdisplay sichtbar wird. Dieses Prinzip hat zum Vorteil, dass die direkte Sicht auf die Umgebung nicht beeinträchtigt wird und der Benutzer sich somit bei der Verwendung des Augmented Reality-Systems nicht bzgl. der Größe und Flexibilität seines Sichtfeldes umgewöhnen muss.

Allerdings muss gesagt werden, dass die reale Welt ohne jegliche zeitliche Verzögerung wahrgenommen wird, jeder noch so schnelle Computer jedoch eine ebensolche in der Berechnung des Ausgabebildes nicht verhindern kann: Er muss für die computergenerierten Objekte zunächst die Bewegung verfolgen (*tracken*), darauf aufbauend die neue Position und Orientierung berechnen und das Ausgabebild inklusive Beleuchtungssimulation berechnen (*rendern*). Somit hat dieses Display-Prinzip den Nachteil der zeitlichen Verzögerung in der Darstellung der computergenerierten, überlagernden Inhalte. Dabei bemerkt der Mensch selbst kleinste Verzögerungen von einigen Millisekunden, und das virtuelle Bild erscheint ihm verschwommen.

Außerdem können die virtuellen Überlagerungen bei Optical See-Through Displays nie vollständig opak angezeigt werden, da das reale Umfeld aufgrund des halbdurchlässigen Spiegels zu jeder Zeit zu einem bestimmten Teil durchscheint, was in den meisten Fällen jedoch einen gewünschten Nebeneffekt darstellt.

2.2.2 Video See-Through Displays

Video See-Through Displays verwenden eine oder mehrere Videokameras – etwa eine für jedes Auge des Benutzers –, welche das Bild bzw. die Bilder für die Hintergrundanzeige des Displays liefern. Die virtuellen Objekte werden in jedem Anzeigeschritt kurz nach dem Hintergrundbild, also aus Benutzersicht davor, gezeichnet. Diese Vorgehensweise bewirkt, dass der reale Hintergrund immer um den gleichen Zeitraum verzögert erscheint wie die virtuellen Objekte. Allerdings hängen beide Bildteile und somit das gesamte Bild, welches der Benutzer sieht, den realen Bewegungen des Benutzers und seinem Umfeld um ebendiesen Zeitraum hinterher.

Zudem besteht ein wesentlicher Nachteil von Video See-Through Displays darin, dass die Sicht auf die Realität schlechter wird, da das Bild der Umgebung in reduzierter Qualität und mit weniger Informationen an den Seiten des Sichtfeldes erscheint. Dies liegt daran, dass Videokameras aufgrund ihrer Auflösung und ihrem Öffnungswinkel meist nicht in der Lage sind, die Realität so aufzuzeichnen und mittels Displays wiederzugeben, dass die Möglichkeiten des menschlichen Auges dabei imitiert werden. Während beispielsweise das Blickfeld (engl.: *field of view*, abgekürzt: FOV) des

Menschen über einen horizontalen Öffnungswinkel von ca. 180° und einen vertikalen Öffnungswinkel von annähernd 150° verfügt, bringen es Video See-Through Displays je nach Anwendungsfall maximal auf Öffnungswinkel von 100° respektive 65°.

2.2.3 Head-Mounted Displays

Head-Mounted Displays (kurz: HMDs) sind am Kopf getragene visuelle Ausgabegeräte in Form von Brillen, welche zumeist einen oder zwei Bildschirme enthalten. Diese befinden sich nahe der menschlichen Augen und stellen virtuelle Inhalte dar. Ist die Brille teilweise durchsichtig, handelt es sich um erwähntes Optical See-Through Display, das virtuelle Inhalte halbtransparent sichtbar werden lassen kann und somit Augmented Reality ermöglicht; kann man nicht durch die Brille hindurchsehen, handelt es sich zunächst um ein Virtual Reality-Display (zur Unterscheidung von Augmented Reality und Virtual Reality siehe vorangegangenen Unterabschnitt 'Augmented Reality'). Soll mittels Letzterem jedoch ebenfalls Augmented Reality realisiert werden, kann das Display mittels ein oder zwei Kameras zu erwähntem Video See-Through Display erweitert werden: Dabei wird das Kamerabild in den Displays angezeigt. Abbildung 2.4 zeigt ein Head-Mounted Display aus dem Jahr 2009, welches keine Möglichkeit der Einblendung der realen Welt mit sich bringt und daher von Haus aus nur zur Anwendung von Virtual Reality verwendet werden kann.



Abbildung 2.4: i-Glasses High Resolution Head Mounted Display (siehe [Sew09])

Im Falle eines einzelnen Displays spricht man zudem von einem *monokularen* Display; bei zwei Displays kann hingegen ein dreidimensionaler Eindruck der virtuellen Inhalte erzeugt werden und man spricht von einem

stereoskopischen Display. Dies lässt sich analog auf die Einblendung von Kamerabildern im Falle von Video See-Through Displays übertragen: Wird nur eine einzelne Kamera verwendet, wird die reale Welt zweidimensional wahrgenommen; bei Verwendung je einer Kamera für beide Augen des Trägers entsteht ein stereoskopischer Eindruck der realen Welt wie beim natürlichen Sehen.

Beim Tragen eines Head-Mounted Displays soll dem Träger zudem der Eindruck entstehen, das Gesehene würde sich mehrere Dezimeter vor ihm abspielen. Dies ist notwendig, da die menschlichen Augen nur mit großer Mühe Dinge fokussieren können, die sich unmittelbar – beispielsweise nur wenige Zentimeter – vor ihnen befinden. Auf Dauer würden sich Ermüdungserscheinungen einstellen. Wie Abbildung 2.5 zeigt, wird das virtuelle Bild daher durch eine Optik aus Linsen in eine größere Entfernung verschoben. Der Abstand dieser sogenannten fokalen Ebene, auf der sich dem Träger die Inhalte präsentieren, beträgt bei HMDs meist zwischen 60 und 80 cm.

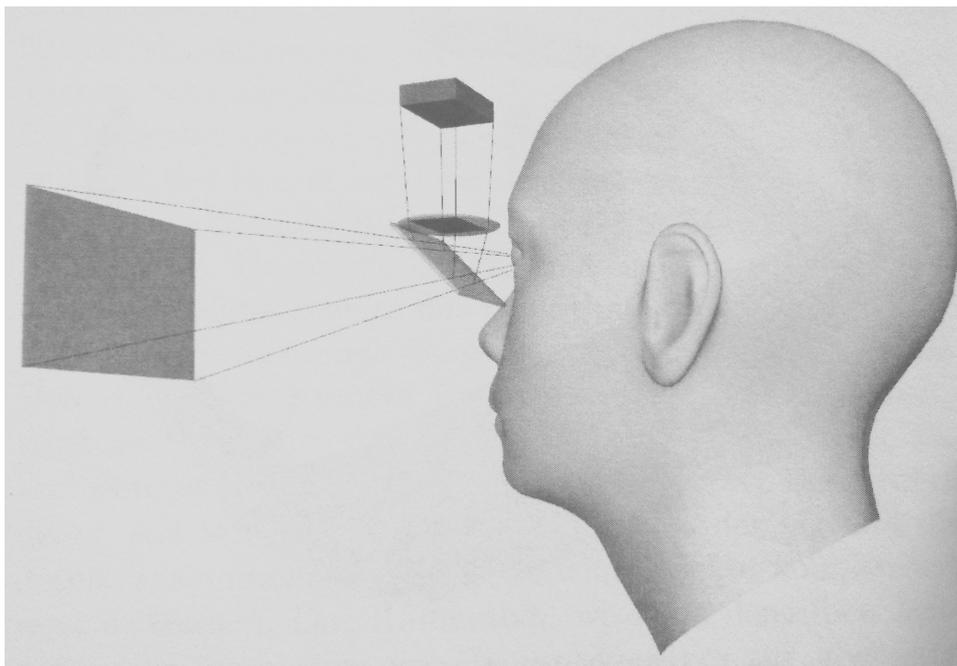


Abbildung 2.5: Verschiebung der fokalen Ebene bei Head-Mounted Displays (siehe [Tön10], S. 24)

Neuere Modelle von Head-Mounted Displays versuchen indes, die darzustellenden virtuellen Inhalte über entsprechende Optiken direkt auf die Netzhaut des menschlichen Auges zu projizieren. Die entsprechende Technologie wird meist *Virtual Retinal Display*, abgekürzt VRD, genannt. Dabei befinden sich die erwähnten Optiken und der bilderzeugende Projektor des

Displays beispielsweise seitlich oder oberhalb des menschlichen Auges. Ein Beispiel einer solchen Konstruktion zeigt Abbildung 2.6.



Abbildung 2.6: AiRScouter, ein Virtual Retinal Display von Brother (siehe [Bro11])

Die Vorteile einer solchen Konstruktion sind, dass sie potenziell sehr leicht ist und wenig Platz einnimmt, da keine flächigen Bildschirme zum Einsatz kommen. Zudem wird – im Gegensatz zu obigen HMD-Varianten – zur Realisierung von Augmented Reality kein teilweise durchsichtiges Brillenglas mehr benötigt, da der Träger seine Umwelt auf direktem und natürlichem Wege wahrnimmt. Die virtuellen Inhalte können dabei einen hinsichtlich Größe und Position beliebigen Teil des Sichtfeldes überblenden. Einzig die Position der Konstruktion im Bezug zum Auge und die Ausrichtung der Projektion schränken diese Möglichkeiten etwas ein (siehe Abbildung 2.7).

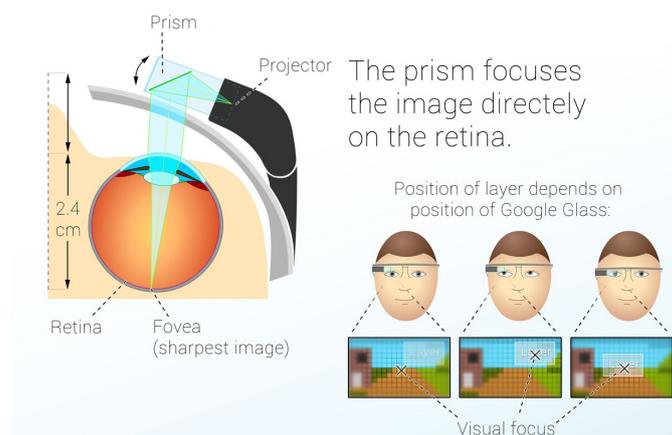


Abbildung 2.7: Prinzip der Projektion von Bildern auf die Retina mittels VRDs (siehe [Mis13])

Das natürliche Sichtfeld des Trägers ist überdies solange keinerlei Beeinflussung ausgesetzt, wie keine Projektion auf die Netzhaut stattfindet, das

System also ausgeschaltet ist. Werden die virtuellen Inhalte nur auf ein Auge projiziert, führt dies dazu, dass das Sichtfeld an gegebener Stelle – durch die Zusammensetzung der Bilder beider Augen im Sehzentrum des Gehirns – automatisch halbtransparent überlagert wird. Abbildung 2.8 soll dies verdeutlichen. Werden beide Augen mit Licht aus jeweils eigenen Projektoren bestrahlt, kann hier zudem ebenfalls ein stereoskopisches Bild der virtuellen Inhalte vermittelt werden.

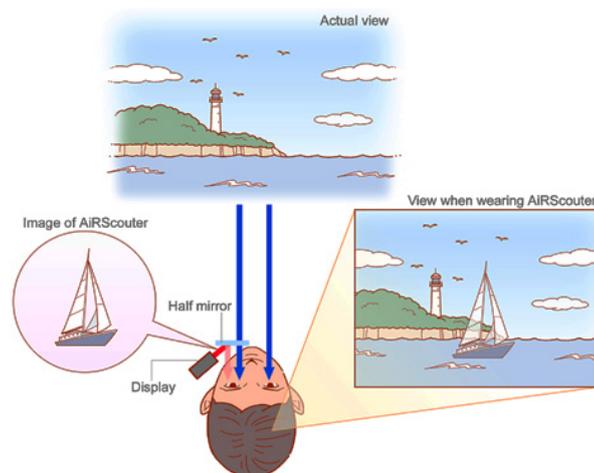


Abbildung 2.8: Funktionsweise des Virtual Retinal Displays AiRScouter (siehe [Bro11])

2.3 Tracking

Da sich Augmented Reality-Systeme dadurch auszeichnen, dass virtuelle Inhalte dynamisch auf die reale Umwelt des Systems reagieren – also eine möglichst realistische Interaktion in Echtzeit stattfindet –, kann die Lagebestimmung, neben der Darstellung der Inhalte, als zweiter zentraler Bestandteil angesehen werden. Die Lage des Displays zu wichtigen Gegenständen in der Umgebung muss dem System dabei zur Verfügung stehen, um virtuelle Inhalte korrekt anzeigen zu können. Je nach Art des Displays ist zusätzlich die Information notwendig, wie sich die Lage des Betrachters zur der vom Display verhält. Der Prozess der Lagebestimmung wird im Allgemeinen als Tracking bezeichnet.

Im Folgenden wird zuerst ein Überblick über die Prinzipien des Trackings und die dafür verwendeten Technologien gegeben. Daraufhin wird – aufgrund der Relevanz für diese Arbeit – genauer auf die Technologie des optischen Trackings eingegangen und im Anschluss daran werden Probleme und Herausforderungen des Trackings im Allgemeinen und bezüglich des optischen Trackings im Besonderen erläutert.

Einen weiteren großen Bereich von Virtual und Augmented Reality, den man zum Tracking zählen kann, stellen Registrierung und Kalibrierung dar, welche je nach Art des Systems eine wichtige Rolle spielen, um das Tracking überhaupt möglich zu machen. Der Fokus dieser Arbeit liegt jedoch auf der Verwendung eines Head-Mounted Displays in Kombination mit einer Smartphone-Kamera zur Realisierung eines Augmented Reality-Systems. In diesem Fall ist keine Registrierung unterschiedlicher statischer Koordinatensysteme notwendig. Einzig eine Kamera-Kalibrierung wäre sinnvoll, um z.B. die Verzerrung der im Smartphone verbauten Linsen zu kompensieren und das Tracking somit genauer zu machen. Aufgrund des prototypischen Charakters des Systems wird darauf jedoch verzichtet. Daher wird auch in diesem Grundlagenteil auf das Thema der Kalibrierung nicht weiter eingegangen; an dieser Stelle sei auf [Tsa87] verwiesen.

2.3.1 Technologien und Prinzipien

Es existiert eine Vielzahl von verschiedenen Trackingsystemen, welche jeweils in unterschiedlichen physikalischen Domänen arbeiten. All diese konkreten Systeme bauen dabei jeweils auf einem von folgenden Prinzipien auf: *Inside-Out*, *Outside-In* und *Inside-In*. Beim *Inside-Out*-Tracking nimmt das bewegte Objekt das Tracking selbst vor und benutzt die statische Umgebung als Referenz, während beim *Outside-In*-Tracking das bewegte Objekt selbst passiv ist und keinerlei Kenntnis über seine Lage besitzt. Das Trackingsystem ist dabei fest in der Umgebung montiert und schaut dabei von außen auf das zu trackende Objekt. Das Prinzip des *Inside-In*-Trackings kommt meist in mobilen Augmented Reality-Anwendungen zum Einsatz, bei denen es nicht notwendig ist, statische Referenzen in der Umgebung zu nutzen: Alle zu trackenden Objekte sind selbst beweglich und befinden sich meist am Körper des Trägers des Systems, mit der weiteren Umwelt wird nicht interagiert. Abbildung 2.9 veranschaulicht die Unterschiede der drei genannten Prinzipien.

Beim *Inertialtracking* wird mittels Beschleunigungs- oder Neigungsmesser in diskreten Zeitintervallen die Veränderung der Lage oder des Rotationswinkels, relativ zum letzten Stand, ermittelt. Beide genannten Formen können zur Komplettierung auch problemlos gemeinsam eingesetzt werden. Weiterhin sind solche Trackingsysteme unabhängig von externer Hardware; sie benötigen keinerlei Bezugspunkt. Falls jedoch Bezugspunkte herangezogen werden, können aus den Messungen der jeweiligen Veränderung durch sukzessives Aufaddieren über alle bisherigen Messungen seit der letzten bekannten Position respektive Orientierung absolute Werte ermittelt werden. Jedoch entstehen durch minimale Messfehler sowie durch die diskrete Abtastung, welche das Computersystem vornimmt, Abweichun-

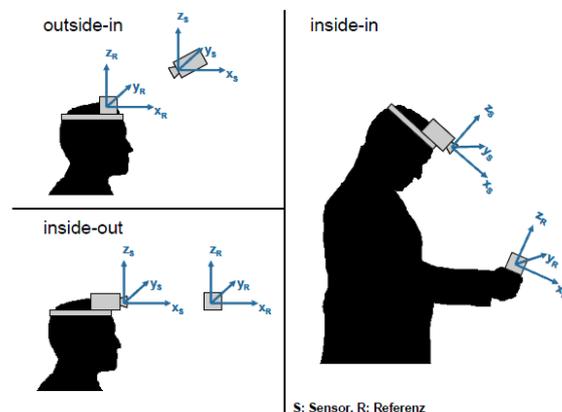


Abbildung 2.9: Prinzipien des Trackings (siehe [Rei09], S. 67)

gen zu der tatsächlichen Position bzw. Orientierung. Diese Abweichungen und die damit verbundene Ungenauigkeiten der absoluten Werte nehmen mit der Laufzeit der Messung zu und sollten daher nicht als alleiniges Mittel zur Lagebestimmung genutzt werden. In der Praxis werden Inertial-trackingsysteme daher meist unterstützend eingesetzt, um eine höhere Genauigkeit zu erreichen.

Im Falle von magnetischem Tracking erhält man hingegen direkt absolute Angaben zur aktuellen Orientierung. Ein Kompass richtet sich dabei an einem Magnetfeld aus – entweder am natürlichen Erdmagnetfeld oder an einem künstlich aufgebauten Magnetfeld. Das Erdmagnetfeld betreffend lässt sich sagen, dass dieses einzig die Rotation um zwei Achsen liefert, während man bei künstlichen Magnetfeldern mittels zusätzlich bekannten Daten über Form und Stärke des Magnetfeldes auch Daten bezüglich der dritten Rotationsachse und in komplett bekannten Umgebungen sogar Daten zur absoluten Position im Raum ermitteln kann. Ein weiteres positives Merkmal von magnetischem Tracking ist die fehlende Notwendigkeit von direktem Sichtkontakt unterschiedlicher Hardware-Komponenten im Raum: Dadurch stellt Verdeckung kein Problem dar.

Weiterhin existiert die Möglichkeit von laufzeitbasiertem Tracking. Entsprechende Systeme nutzen das umgebende Medium zum Übertragen eines ausgesendeten Signals, messen dessen Laufzeit und bestimmen daraus die aktuelle Lage. Als prominentes System dieser Art sei an dieser Stelle das *Global Positioning System* (kurz: GPS) genannt, bei dem in der Erdumlaufbahn befindliche Satelliten ihre aktuelle Position inklusive sehr genauer Zeitstempel kontinuierlich aussenden und GPS-Empfänger auf der Erde diese Signale empfangen, auswerten und zur Positionsbestimmung bezüglich der geographischen Lage und Höhe über dem Meeresspiegel nutzen können. Zur eindeutigen Bestimmung dieser Werte reichen Verbindungen

zu drei unterschiedlichen Satelliten aus. Wird zusätzlich das Signal eines vierten Satelliten empfangen, kann die eigene Position über die Zeit verfolgt und über die Berechnung der mathematischen Ableitung der Position eine eventuelle Bewegungsrichtung ermittelt werden. Während die Genauigkeit von standardmäßigem GPS jedoch nur im Meterbereich liegt und sich außerdem nicht innerhalb von Gebäuden nutzen lässt, existieren weitere laufzeitbasierte Trackingsysteme, die sich auch in Räumen nutzen lassen. Dazu zählen akustische sowie optische Systeme, welche die Laufzeit oder die Phasenverschiebung von ausgesendeten und empfangenen Signalen messen und versuchen, daraus die Lage im Raum zu ermitteln.

Zu guter Letzt gibt es Systeme, welche direkte mechanische Verbindungen zwischen dem zu trackenden Gegenstand und der Umgebung nutzen, um dessen Lage zu bestimmen. Solche Systeme eignen sich gleichzeitig auch zu sogenanntem *Force Feedback*, also zur Rückmeldung von Kraft an den Benutzer des Systems, um mechanische Widerstände während einer Benutzereingabe zu simulieren.

Die im Bereich der Augmented Reality am häufigsten verwendete Technologie stellt allerdings das optische Tracking dar, welches im Folgenden genauer beleuchtet wird.

2.3.2 Optisches Tracking im Detail

Beim optischen Tracking wird meist im Bereich des sichtbaren Lichtes ($380nm - 780nm$) gearbeitet. Man unterscheidet überdies zwischen *Markertracking* und *markerlosem Tracking*.

Beim Markertracking werden vorher bekannte visuelle Muster an dem zu trackenden Objekt angebracht. Diese Muster werden *Marker* genannt und können dann von Bildverarbeitungssystemen erkannt und ihre Position und Orientierung im Raum berechnet werden. Bei der Berechnung von Position und Orientierung – was man zusammengefasst als *Pose* kennt – stößt man auf je drei unbekannte Variablen. In diesem Fall ist auch oft die Rede von *Freiheitsgraden*. Bei der Berechnung der Pose gibt es also sechs Freiheitsgrade: Drei Koordinaten für die 3D-Position und drei Winkel für die Rotation im Raum (um alle drei Achsen). Marker gibt es prinzipiell in zwei Formen: Räumlich angeordnete, reflektierende Marker und flächige Marker mit Mustern.

Reflektierende Marker bestehen meist aus einer spezifischen räumlichen Anordnung von mehreren kleinen Kugeln, welche mit retroreflektivem Material beschichtet sind, welche eintreffendes Licht immer in die Richtung reflektieren, aus der es kommt. Abbildung 2.10 zeigt einen solchen

Marker. Durch die stark reflektierenden Kugeln in Verbindung mit Foto-
blitzen oder durchgängig eingeschalteten und an der Kamera befestigten
Lichtern entstehen in Kamerabildern entsprechende helle Punkte, die sich
mittels Bildverarbeitung relativ leicht im Bild lokalisieren lassen. Dadurch,
dass die Kugeln nicht auf einer Fläche angebracht sind, sondern eine räum-
liche Struktur bilden, ergeben sich aus zwei Kamerabildern, aufgenommen
aus hinreichend unterschiedlichen Richtungen bezogen auf einen Marker,
auch unterschiedliche Abstände der hellen Punkte im Bild zueinander.
Wenn die Abstände der Kugeln dieses Markers zueinander sowie die La-
ge der Kamerapositionen zueinander bekannt ist, ist es somit möglich,
allein aus den X- und Y-Positionen der hellen Punkte in diesen zwei Ka-
merabildern die Position und Orientierung des Markers im Raum, also die
vollständige Pose, zu berechnen.

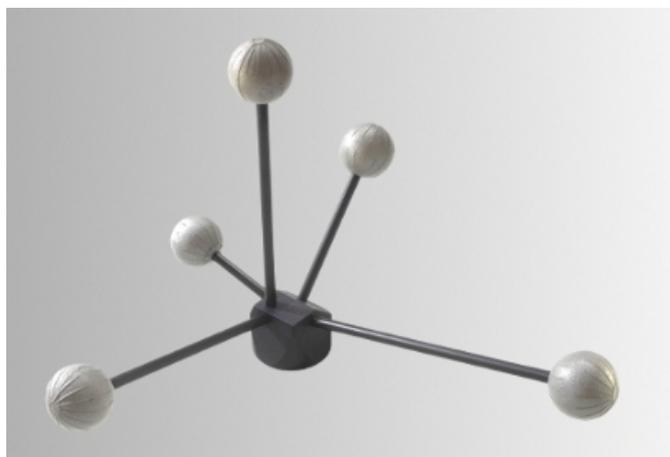


Abbildung 2.10: Beispiel eines reflektierenden Baum-Markers (siehe [Adv])

Bei flächigen Markern mit Mustern handelt es sich meist um auf handels-
übliches Papier gedruckte, oft rechteckige Muster. Auch diese Muster müs-
sen dem Trackingsystem naturgemäß bekannt sein, damit die Berechnung
der Pose des Markers erfolgen kann. Abbildung 2.11 zeigt den wahrschein-
lich bekanntesten flächigen Marker überhaupt, den sogenannten „Hiro“-
Marker des ARToolKit, einer weit verbreiteten, kostenlosen und quellof-
fenen Softwarebibliothek für Markertracking im Rahmen von Augmented
Reality-Anwendungen.

Im Rahmen von ArToolKit können auch all solche Marker verwendet wer-
den, die statt dem „Hiro“ ein beliebiges anderes schwarz-weißes Muster
in der Mitte des rechteckigen Randbereichs enthalten. Das Erlernen eines
solchen, selbst erstellten Markers wird über ein eigens dafür mitgeliefertes
Programm ermöglicht; dieses extrahiert aus dem ihm übergebenen Marker-
Bild eine Datei, welche durch das ArToolKit selbst zur Laufzeit eingele-



Abbildung 2.11: Flachmarker des ARToolKit (siehe [KB99])

sen und zum Markertracking verwendet werden kann. Am Beispiel des in Abbildung 2.11 gezeigten „Hiro“-Markers lässt sich die Pose-Bestimmung dann z.B. mittels folgendem Algorithmus realisieren:

1. Zunächst wird im Kamerabild nach starken Kanten, also Stellen zwischen sehr unterschiedlich hellen Pixeln, gesucht.
2. Aus diesen Kanten wird der rechteckige, schwarze Randbereich des Markers bestimmt, woraufhin in einem weiteren Schritt – ebenfalls aus den zuvor gefundenen Kanten – die Eckpunkte des Rechtecks ermittelt werden können.
3. Aus den Eckpunkten lässt sich die Drehung der Markerfläche gegenüber der Kamera bzgl. aller drei möglichen Rotationswinkel berechnen.
4. Da das schwarze Rechteck für sich selbst genommen jedoch rotationssymmetrisch ist, ist die Erkennung der groben Ausrichtung des Musters im Inneren des Markers zusätzlich notwendig, um Rotationen des zweidimensionalen Markers um die zur Oberfläche orthogonale Achse (entspricht dem Normalenvektor der vom schwarzen Rechteck aufgespannten Ebene) über 90° hinaus zu bestimmen.
5. Die verbleibenden drei Freiheitsgrade der Position des Markers können weiterhin nur dann eindeutig bestimmt werden, wenn die Kantenlänge des rechteckigen Markers vorher bekannt ist. Ohne Kenntnis der Kantenlänge bleibt der Freiheitsgrad der Entfernung des Markers von der Kamera unbestimmt.

Im Allgemeinen existieren darüber hinaus auch Methoden zur Pose-

Bestimmung von Markern, welche beispielsweise nicht rechteckig sind, keine festgelegte Struktur besitzen oder auch Farben enthalten können. Neuere Systeme bieten sogar die Funktion, beliebige vorher eintrainierte Bilder als Marker zu nutzen. Der Aufwand für robustes Tracking kann in solchen Fällen allerdings beliebig komplex werden.

Flächige Marker haben den Vorteil, dass sie relativ einfach herzustellen sind und man somit für ein komplettes Trackingsystem nichts weiter als eine Kamera und ein bedrucktes Blatt Papier benötigt. Allerdings kann die Kamera einen flächigen Marker nur dann erkennen, wenn er ihr gegenüber um maximal 90° gekippt ist. Möchte man ein zu trackendes Objekt beliebig im Raum bewegen können, kann man sechs flächige Marker zu einem Markerwürfel zusammenfügen – ein Beispiel dessen zeigt Abbildung 2.12). Dieser kann nun aus jeder Lage getrackt werden. Ein weiterer Nachteil von flächigen Markern ist, dass ihre Erkennung, im Gegensatz zu reflektierenden Markern, relativ stark von der Beleuchtungssituation abhängt: Sobald z.B. sehr dunkler Schatten auf den Marker fällt oder die Helligkeit der Beleuchtung aufgrund von Sonneneinstrahlung plötzlich stark zunimmt, ist die Erkennung gerade bei einfacheren Systemen nicht mehr robust.

Über die hier beschriebenen passiven Marker hinaus existieren aktive Marker für optische Trackingsysteme, welche selbst Licht aussenden und, vor allem durch die Kenntnis der Art des ausgesendeten Lichtes seitens des Systems, leichter gefunden werden können. Solche Marker unterscheiden sich in der Möglichkeit des Trackings sonst jedoch nicht von den bereits beschriebenen flächigen oder räumlich angeordneten Markern. Die weiteren technischen Feinheiten der Pose-Bestimmung nach erfolgreicher Erkennung von Markern im Kamerabild und die eingesetzte Methodik zur Umrechnung vom Kamerakoordinatensystem in das des Markers sollen an dieser Stelle nicht weiter vertieft werden.

Im Falle von markerlosem Tracking soll kein Bedarf für zusätzliche Marker bestehen; stattdessen muss allein das Abbild der Umgebung ausreichen. Dabei wird meist versucht, natürliche visuelle Merkmale – sogenannte *Features* – von zu trackenden Objekten aus Kamerabildern zu extrahieren und die Anordnung der Features mit dem vorher vorhandenen Wissen über die zu trackenden Objekte zu vergleichen, um Korrespondenzen zu finden und letztendlich Pose der Objekte zu bestimmen. Sowohl für die Extraktion der Features als auch für den strukturellen Vergleich bietet die Bildverarbeitung ein breites Spektrum an geeigneten Methoden. An dieser Stelle sei auf die in Fachkreisen bekannten *SIFT-Features* (vgl. [Low04]) sowie auf Verfahren des *Point Pattern Matching* (vgl. z.B. [GS85] oder [Mur92]) verwiesen.

Überdies existieren weitere Ansätze zum markerlosen Tracking. Neuere Systeme verwenden beispielsweise interne dreidimensionale Modelle

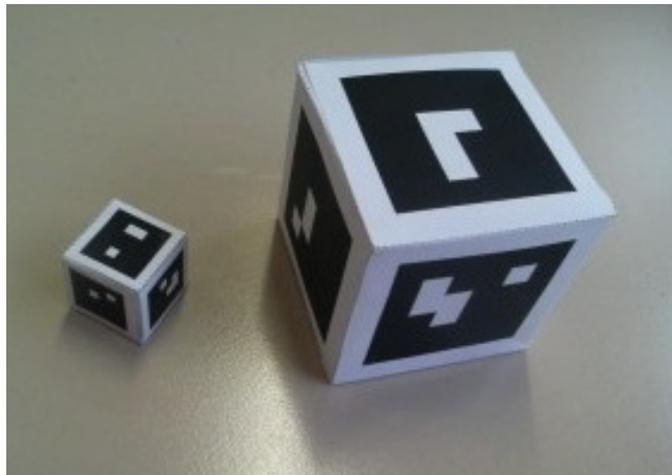


Abbildung 2.12: Würfel aus Markern des ARToolKit (siehe [KB99])

der zu trackenden Objekte und versuchen, Referenzbilder zum Vergleich mit dem Kamerabild zu berechnen. Dazu sind jedoch eine geschätzte Beleuchtungssituation (Winkel des eintreffenden Lichts etc.) sowie initial geschätzte oder letzte bekannte Posen der Objekte notwendig. Letztere werden durch Optimierung zur Bestimmung der neuen Pose herangezogen. Dieser Ansatz nennt sich *Analyse durch Synthese* und ist Gegenstand aktueller Forschung.

2.3.3 Herausforderungen im praktischen Einsatz

Bei der Verwendung von Trackingsystemen stößt man auf eine Reihe von Problemen, die sich nur teilweise lösen lassen und daher noch immer in vielen Augmented Reality-Anwendungen erkennbar sind.

Eine Ausprägung solcher Probleme sind Schwimmeffekte. Dies meint die Erscheinung, dass das Bild der Realität und die überlagerten virtuellen Bilder nicht vom selben Zeitpunkt stammen und damit versetzt sein können. Dieser Effekt entsteht im Allgemeinen durch die Laufzeit der Signale durch das Trackingsystem, die Verarbeitung der Signale und die benötigte Zeit zum Rendern. Bewegt sich nichts in der aktuellen Szene und steht das Display still, macht sich dieser Effekt nicht bemerkbar. Sobald jedoch ein Marker oder das Display bewegt wird, scheint das virtuelle überlagerte Bild etwas verzögert zu folgen. Dieser Umstand wurde im Unterabschnitt 'Displays' schon angesprochen und die Konsequenzen für Optical See-Through sowie Video See-Through Displays erläutert. Ist zudem ein Head-Mounted Display im Einsatz, können Schwimmeffekte vor allem Kippbewegungen – also Nicken oder seitlichem Neigen des Kopfes – den Gleichgewichtssinn

des Trägers stören. Dies kann wiederum zu Schwindelgefühl und Übelkeit führen. In diesem Fall spricht man von der sogenannten *Cybersickness* oder auch visuell induzierter Bewegungskrankheit (*Visually induced motion sickness*).

Eine weitere Herausforderung von Augmented Reality-Anwendungen im Allgemeinen und von Trackingsystemen im Speziellen ist falsche Verdeckung. Das virtuelle Bild liegt zu jeder Zeit vor dem realen Bild, unabhängig von der Verwendung von Optical See-Through oder Video See-Through Displays. In der Folge wird ein realer Gegenstand – z.B. ein Körperteil –, welcher eigentlich vor einem virtuellen Objekt liegen sollte, von ebendiesem verdeckt. Dadurch entsteht ein verfälschter Tiefeneindruck der Szenerie, der Benutzer wird irritiert und die Arbeit mit dem System erschwert. Besonders stark tritt dieser Effekt ein, wenn z.B., wie in Abbildung 2.13, die komplette Tischfläche, neben kleineren Objekten auf dem Tisch, zusätzlich als virtuelles Objekt überlagert werden soll, um einen Marker zu verstecken.

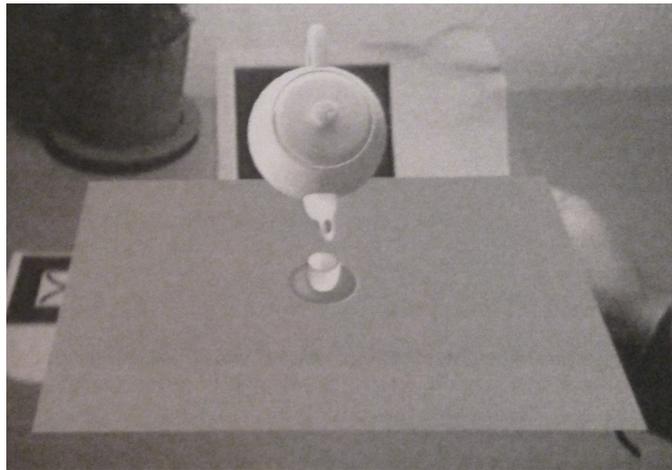


Abbildung 2.13: Die Tischoberfläche wird als virtuelles Objekt ebenfalls gerendert, um den linke Marker möglichst zu verstecken. (siehe [Tön10], S. 85)

Werden statische Objekte mit bekannter Position verdeckt, wäre die einfachste Lösung, sie nicht mit virtuellem Inhalt zu überlagern, indem man an entsprechender Stelle nichts rendert. In obigem Beispielbild könnte sich unter der grauen virtuellen Fläche beispielsweise noch eine Serviette befinden, welche nie bewegt wird. Sind Ausmaße und Position derer relativ zur Position des Markers auf dem Tisch bekannt, kann der entsprechende Teil der verdeckenden Fläche nicht vom Rendering ausgeschlossen werden, sodass die Serviette wieder zum Vorschein kommt. Im Falle von dynamischen Objekten könnte man diese zusätzlich selbst tracken, um dem Pro-

blem einer falschen Verdeckung entgegenzutreten. Ist das Objekt jedoch, wie der Hand im obigen Beispielbild, zusätzlich noch deformierbar, lautet die einzig akzeptable Lösung, mittels Tiefenmessungen z.B. durch einen nahe der Kamera befindlichen Laserscanner, bestenfalls pro Pixel des virtuellen Bildes, herauszufinden, ob sich reale Objekte aus Kamerasicht vor der durch den zweidimensionalen Marker aufgespannten Ebene befinden. Falls dem so ist, kann das virtuelle Objekt an dieser Stelle verborgen werden.

Über die beschriebenen Herausforderungen hinaus gibt es noch ein weiteres prominentes Problem von bildverarbeitenden Systemen: *Jitter*. Dies bezeichnet das Phänomen, dass sich eine virtuelle Überlagerung entweder stärker bewegt als der getrackte Marker, oder, dass sich die Überlagerung bewegt, obwohl sich die Sicht auf die reale Welt inklusive der Position des entsprechenden Markers gar nicht zu ändern scheint. Das Phänomen lässt sich meist auf die Nutzung von nicht sehr robusten Verfahren der Bildverarbeitung innerhalb des Trackingsystems zurückführen: Dadurch, dass die Auflösung der Kamera und damit auch die Auflösung des zu verarbeitenden Bildes der Realität begrenzt ist, die Realität also diskret abgetastet wird, entstehen durch Rundung der Positionen im Bild zu jeder Zeit – meist minimale – Fehler. Je nach Algorithmus des Markertrackings genügt manchmal allerdings nur eine kleine Änderung der realen Position im Bild, um die berechnete Pose um mehrere Einheiten springen zu lassen.

Ein Ansatz zur Lösung dieses Problems besteht darin, die berechneten Posen über mehrere Schritte hinweg zu mitteln, um plötzliche Ausreißer etwas zu mildern. Es existieren weiterhin auch Methoden zur spezifischen Erkennung von Ausreißern, die daraufhin ignoriert werden können. All diese Ansätze basieren jedoch auf der Annahme, dass Ausreißer bestimmter Größen keine reale Entsprechung haben, und reduzieren damit die absolute Genauigkeit des Trackingsystems. Darüber hinaus entsteht durch kontinuierliche Glätten bzw. Mitteln der Posen eine zusätzliche zeitliche Verzögerung des Trackings, was wiederum zu erwähnten Schwimmeffekten führt.

2.4 Eingabe und Interaktion

Die bisher behandelten Grundlagen bezogen sich rein auf die Ausgabe von Informationen; in diesem Unterabschnitt soll nun erläutert werden, wie die Eingabe von Daten in Augmented Reality-Systeme erfolgen kann. Zunächst erfolgt eine Übersicht über vorhandene Möglichkeiten zur Eingabe, daraufhin wird auf die konkreten Techniken der Interaktion eingegangen und anschließend die grundlegenden Anforderungen an neue User-Interfaces kurz zusammengefasst. Zur Begrifflichkeit: Die Interaktion mit

einem System setzt die Möglichkeit zur Eingabe voraus und verarbeitet die ankommenden Eingabedaten auf eine spezifische Art und Weise.

2.4.1 Eingabe

Die Eingabegeräte, welche manche Menschen tagtäglich nutzen, um ihren Computer zu bedienen, sind für Eingaben in ein Augmented Reality-System aus vielerlei Gründen zumeist nicht geeignet. Zum einen sind Geräte wie Tastaturen und Mäuse schlicht unhandlich, wenn es darum geht, den Schreibtisch zu verlassen und Eingaben möglich zu machen, während der Benutzer des AR-Systems sich bewegt. Zum anderen sind die meisten Eingabegeräte – beispielsweise auch berührungsempfindliche *Touchpads* respektive *Touchscreens*, welche heutzutage verbreitet zur Nutzung mobiler Computer oder Smartphones verwendet werden – meist nur dazu in der Lage, zweidimensionale Daten zu liefern. Mit diesen Daten lässt sich ohne Weiteres keine Interaktion mit dem dreidimensionalen Raum bewerkstelligen. Daher sind meist andere Konzepte notwendig, um Daten in AR-Systeme einzugeben. Im Folgenden werden einige Alternativen vorgestellt.

Eingabe mittels Marker

Im Falle von Markertrackingsystemen werden Marker gerne zur Eingabe verwendet. Eine Möglichkeit besteht darin, einen Marker wie einen Button zu nutzen: Sobald ein Nutzer den Marker z.B. mit seiner Hand für einen bestimmten Zeitraum verdeckt, das System den Marker also für diese Dauer nicht im Kamerabild findet, kann eine vorher definierte Aktion ausgelöst werden. Dies soll die Funktionsweise eines Knopfdrucks simulieren.

Sollen Zahlenwerte eingegeben werden können, kann eine Kombination aus zwei Markern als Schieberegler (engl.: *Slider*) dienen. Dazu wird einer der beiden Marker genutzt, um ihn entlang einer festgelegten Achse mit dem grafischen Modell einer Schieberegler-Bahn zu überlagern. Der zweite Marker wird mit dem virtuellen Schieberegler selbst überlagert und die Entfernung der beiden Marker entlang der dargestellten Achse wird als Zahlenwert des Sliders interpretiert.

Der Einsatz mehrerer Marker birgt jedoch die Gefahr, den Benutzer zu verwirren, weil nicht klar ist, welcher Marker welche Funktion erfüllt. Daher sollten bei gleichzeitiger Verwendung mehrerer Marker kleine Icons neben den Markern eingeblendet werden, damit der Benutzer zu jeder Zeit erkennen kann, welche Eigenschaft ein Marker hat.

Weiterhin existiert die Möglichkeit, Marker zum Ausdruck von Gesten zu verwenden. Dabei könnte das System beispielsweise erkennen, wenn ein

Marker gedreht oder geschüttelt wird. Auch eine in sich abgeschlossene – beispielsweise kreisförmige – Bewegung eines Markers im Raum könnte als solche erkannt und entsprechende Kommandos an die Einheit zur Darstellung der virtuellen Inhalte geschickt werden. Jedoch muss der Benutzer zunächst wissen, welche Gesten möglich sind. Um dem Benutzer dies zur Laufzeit des Systems automatisch mitzuteilen, gibt es den Ansatz, animierte Hilfestellungen, welche die möglichen Gesten nachbilden, neben die Marker einzublenden (vgl. dazu [WLF07]). Abbildung 2.14 zeigt eine solche Animation, welche aus dem wiederholten, mit steigendem Transparenzwert dargestellten, Abbild des zu bewegenden Markers besteht. Dadurch soll der zeitliche Verlauf der Geste deutlich werden.

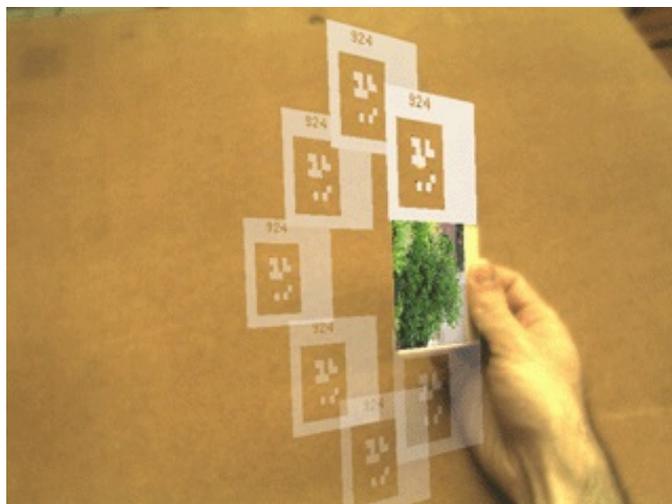


Abbildung 2.14: Visueller Hinweis auf mögliche Geste (siehe [WLF07], S. 2)

Tangible User-Interfaces

Im Gegensatz zur Eingabe mittels Markern wird im Falle von *Tangible User-Interfaces* – zu deutsch etwa „greifbare Benutzerschnittstelle“ – versucht, die gegebene Umwelt der Menschen in ein User-Interface zu verwandeln. Anders gesagt sollen die inhärenten Eigenschaften von realen Gegenständen in der durch die AR-Anwendung erweiterten Realität übernommen werden. Ishii und Ullmer [IU97] prägten den Begriff der sogenannten *Tangible Bits*, also greifbarer digitaler Informationen, welche die Lücke zwischen der virtuellen Welt und der physikalischen Umgebung schließen und die digitalen Informationen direkt wahrnehm- und manipulierbar machen sollen. Übergeordnetes Ziel dabei ist, das User-Interface einer Augmented Reality-Anwendung möglichst intuitiv bedienbar zu machen und die Interaktion mit dem AR-System somit als natürlich und selbstverständlich erscheinen zu lassen.



Abbildung 2.15: Reactable – ein Tangible User-Interface (Bild verfügbar unter [Wik13b])

In Abbildung 2.15 ist ein Beispiel für ein Tangible User-Interface zu sehen. Die Tischplatte ist gleichzeitig ein Display und zeigt den Zustand des dahinter stehenden Softwaresystems an, einem modularen Synthesizer, mit dem sich elektronische Musik durch Aneinanderreihung verschiedener Module erzeugen lässt. Reale Objekte repräsentieren diese verschiedenen Module und können auf der Tischplatte platziert werden. Diese werden dann vom System erkannt und die entsprechenden virtuellen Module werden dem Softwaresystem hinzugefügt – der Systemzustand ändert sich also entsprechend der Existenz der physikalischen Objekte. Zusätzlich werden die hinzugefügten, virtuellen Module an der Stelle des Tisch-Displays angezeigt, an der sich die jeweiligen physikalischen Repräsentanten befinden.

Ullmer und Ishii [UI00] stellten darüber hinaus eine Liste von vier Merkmalen vor, welche ein Tangible User-Interfaces kennzeichnen. Diese Merkmale, welche sich anhand des obigen Beispiels besser nachvollziehen lassen, sind:

1. Physische Objekte sind mit digitaler Information verbunden.
2. Physische Objekte enthalten und realisieren Mechanismen zur interaktiven Kontrolle.
3. Physische Repräsentationen sind wahrnehmbar verbunden mit ihrer

digitalen Repräsentation.

4. Der Zustand des physischen Objektes verkörpert Schlüsselzustände des digitalen Systems.

Motion-Tracking

Über die bereits vorgestellten Ansätze hinaus gibt es die Möglichkeit, den menschlichen Körper selbst zu nutzen, um Eingaben für ein Augmented Reality-System zu tätigen. Sogenannte *Motion-Tracking*-Systeme versuchen, die Bewegung einzelner Körperteile oder des gesamten Körpers mittels geeigneter Trackingsysteme zu erkennen und in Eingabekommandos umzuwandeln. Als Beispiele seien an dieser Stelle Hand-, Kopf- sowie Körpertracking erwähnt.

Im Falle von Handtracking wird versucht, auf einer festgelegten Fläche (z.B. bei Multitouch-Systemen) oder im freien Raum Gesten zu erkennen. Dazu ist es meist notwendig, neben der Hand selbst auch einzelne Finger erfolgreich zu tracken. Gelingt dies, können z.B. mit einer Fingerspitze Selektionen und mittels zwei Fingerspitzen Greifgesten durchgeführt werden. Meist werden Marker zum Tracking einzelner Finger eingesetzt; der ältere aber unkomfortablere Ansatz ist der eines Datenhandschuhs, zu sehen in Abbildung 2.16.



Abbildung 2.16: Der Datenhandschuh CyberGlove II (Bild verfügbar unter [Vir])

Kopftracking kann ebenfalls hinsichtlich der Eingabe verwendet werden. Dies geschieht meist über erkannte Nickbewegungen oder Kopfschütteln. Meist werden diese Möglichkeiten zur Eingabe dann verwendet, wenn ein Head-Mounted Display im Einsatz ist und dieses für das funktionierende Augmented Reality-Gesamtsystem ohnehin getrackt werden muss.

Schließlich gibt es die Möglichkeit des Körpertrackings, welches beispielsweise Informationen darüber liefern kann, ob der Benutzer des Systems

gerade sitzt oder steht. Derartige Informationen können dazu genutzt werden, den Modus des Systems zu ändern: Wenn der Benutzer aufsteht, könnte zum Beispiel ein Navigationssystem erscheinen; setzt er sich hin, könnten benötigte Dokumente ins Sichtfeld eingeblendet werden.

Eingabe mittels Gaze-Tracking

Gaze-Tracking meint die Verfolgung der Blickrichtung der Augen und kann dazu dienen, ein bestimmtes Objekt mittels Fokussierung selektieren zu können. Dabei wird meist ein bestimmter Zeitraum zum Auslösen der Selektion festgelegt; sobald der Benutzer ein Objekt über diesen Zeitraum hinweg anschaut, wird es erfolgreich ausgewählt. Somit kann das Drücken virtueller Knöpfe realisiert werden.

Spracheingabe

Einen weiteren Ansatz stellt die Spracheingabe dar. Dieser ist vor allem dann eine gute Alternative, wenn die Hände nicht zur Eingabe verwendet werden können, weil sie z.B. bereits ausgelastet sind, oder wenn die Beleuchtungssituation optische Eingaben nicht zulässt. Des Weiteren können Spracheingaben immer parallel zu anderen Aktivitäten getätigt werden. Ist der visuelle Kanal jedoch stark gefordert, sollten keine zu komplizierten Sprachanweisungen gefordert werden, da der Benutzer beim Nachdenken über seine Aussagen meist die visuelle Informationsaufnahme vernachlässigt.

Es existieren unterschiedliche Erkennungssysteme je nach Art der Spracheingaben: Es können sowohl prägnante Kommandos als auch numerische Eingaben und sogar eine mehr oder minder freie Sprache genutzt werden. Bei den beiden erstgenannten Arten ist die Erkennung der jeweiligen Sprachmuster mittels aktueller Verfahren relativ einfach zu bewerkstelligen; eine relativ freie Spracheingabe setzt hingegen einen hinreichend großen hinterlegten Sprachschatz voraus.

2.4.2 Interaktion

Im letzten Unterabschnitt wurden vorhandene Möglichkeiten zur Eingabe von Daten besprochen. Im Folgenden soll nun erläutert werden, wie ankommende Eingabedaten zur Kontrolle und Steuerung von Augmented Reality-Systemen genutzt werden können. Dazu wird eine Übersicht der für diesen Zweck notwendigen Interaktionstechniken gegeben. Die dazu getroffene Aufteilung der Interaktionstechniken richtet sich nach Bowman et al. [BKLP04], welcher bei der Verwendung von 3D-User-Interfaces

zwischen Manipulation, Bewegung (des Benutzers), Signaletik, Systemeinstellungen und symbolischen Eingaben unterscheidet. Allerdings wird der Bereich der Bewegung des Benutzers im Rahmen dieser Arbeit nicht weiter beachtet, da dieser eher in Virtual Reality-Systemen von Bedeutung ist; im Falle von Augmented Reality kann der Benutzer sich frei in der realen Welt bewegen, das Tracking übernimmt im konkreten Fall die entsprechende Anpassung des Systems und der virtuellen Inhalte, falls notwendig.

Manipulation

Manipulation im Kontext von Augmented Reality meint meist die Positionierung und Drehung eines virtuellen Objektes. Zusätzlich kann es im konkreten Fall notwendig sein, weitere Parameter von Objekten – wie dessen Skalierung oder Form – verändern zu können. Solche Anforderungen werden an dieser Stelle jedoch nicht behandelt.

Für das Problem, die Position eines virtuellen Objektes im Raum zu verschieben, existiert eine einfache Lösung: Falls das Objekt nicht ohnehin schon mittels einem Marker mit der realen Welt verbunden ist, wird eine gedankliche Verbindung mit einem Marker für die Zeit der Manipulation hergestellt; daraufhin wird das Objekt mithilfe des Markers an die gewünschte Stelle verschoben und ebenjene Verbindung, falls notwendig, anschließend wieder entfernt.

Der Marker sollte sich für diese Zeit naturgemäß an der Stelle des virtuellen Objektes befinden, sodass die Position des Markers direkt auf die des virtuellen Objektes übertragen werden kann. Es kann jedoch vorkommen, dass das Erreichen des virtuellen Objektes im Raum aus der Position des Benutzers nicht möglich ist. In solchen Fällen muss also eine Technik gefunden werden, Objekte zu verschieben, ohne dass der Benutzer sich dazu selbst bewegen muss.

Eine mögliche Vorgehensweise ist, das virtuelle Objekt durch eine relative Steuerung zu bewegen. Voraussetzung dazu ist, dass das System auch ohne räumliche Korrespondenz eines Markers mit einem virtuellen Objekt weiß, wann welches Objekt manipuliert werden soll – eine Möglichkeit zur Selektion von entfernten Objekten muss dazu bereit stehen. Dann kann die relative Steuerung in Kraft treten: Die erste Pose des zur Steuerung verwendeten Markers bestimmt die Nullposition, bei der das zu steuernde Objekt an seinem Ort verharret. Wird der Marker nun in einer Richtung verschoben, verschiebt sich das virtuelle Objekt in derselben Richtung.

Der Betrag der Verschiebung des Markers kann dabei ohne Umwege zur Verschiebung des Objektes genutzt werden: Eine Verschiebung des Markers um *10cm* würde das Objekt somit ebenfalls um *10cm* in dieselbe Richtung verschieben. Sind in jedem Fall nur kleine Wege zurückzulegen,

stellt dies einen geeigneten Weg dar. Sollen jedoch beliebig große Verschiebungen möglich sein, müsste die relative Steuerung im Falle dieser Technik zwischendurch immer wieder deaktiviert, der Marker zurück bewegt und das Verfahren wiederholt werden. Eine bessere Lösung stellt folgender Ansatz dar: Die Änderungen der Position des Markers fließen in eine nichtlineare Transferfunktion ein, welche die Positionsänderung des virtuellen Objekts berechnet (vgl. dazu [PBWI96]). Dabei kann die Berechnung beispielsweise in Abhängigkeit vom bereits zurückgelegten Abstand des Markers zu seinem Ausgangspunkt erfolgen. Im Ergebnis hieße das, dass die Position des virtuellen Objektes mit steigendem Abstand des Markers vom Ausgangspunkt immer größere Änderungen erfährt. Dadurch stellt die Reichweite der Verschiebung virtueller Objekte kein Problem mehr dar.

Soll das virtuelle Objekt überdies auch gedreht werden können, bringt die Verwendung von flächigen Markern Schwierigkeiten mit sich: Zweidimensionale Marker können nicht beliebig rotiert und dabei fortwährend erkannt werden. Im Falle von in Unterunterabschnitt 'Optisches Tracking im Detail' bereits erwähnten Markerwürfeln oder räumlichen angeordneten Markern (z.B. Baum-Markern, s.o.) muss hingegen umständlich umgegriffen werden, um dem System beliebige Rotationen als Eingabe zu liefern. Bei der Verwendung von Hand- bzw. Fingertracking wird stattdessen die Dehnfähigkeit des Benutzers auf die Probe gestellt.

Darüber hinaus erscheint es in vielen Fällen als sinnvoll, kleine Schwellwerte zu verwenden, die sicherstellen, dass beispielsweise Rotationsbewegungen von Markern nur dann an die virtuelle Welt weitergegeben werden, wenn sie eine Schwelle von wenigen Grad überschreiten. Damit sollen minimale, unvermeidbare Bewegungen des Benutzers, die dieser nicht als Eingabe verstanden haben möchte, herausgefiltert werden. Zudem können durch diese Technik auch bereits beschriebene Jitter-Effekte unterdrückt werden.

Signaletik

Die sogenannte *Signaletik* ist ein kognitiver Prozess, der der räumlichen Orientierung von Menschen in komplexen Umgebungen dient. Dazu wird neues räumliches Wissen aufgenommen und zu einem mentalen Modell der Umgebung verarbeitet sowie bereits vorhandenes Wissen zur Planung von Wegen durch die Umgebung genutzt. Um die Signaletik bei der Benutzung von Augmented Reality-Systemen zu unterstützen, können zwei Gruppen von Maßnahmen getroffen werden, nutzerbezogene und umgebungsbezogene.

Nutzerbezogene Unterstützung der Signaletik beinhaltet, bezogen auf

Augmented Reality, meist technische Faktoren wie möglichst geringe Einbußen in der Informationsaufnahme durch die Verwendung von Head-Mounted Displays – die menschlichen Sinneskanäle wie Augen und Ohren sollten durch das Tragen dieser Systeme möglichst wenig eingeschränkt sein. Zudem gilt es, in Unterunterabschnitt 'Herausforderungen im praktischen Einsatz' beschriebene Schwimmeffekte zu vermeiden und die Existenz virtueller Objekte durch eventuelle multimodale Systemausgaben zu stützen. Die zusätzliche Verwendung von räumlichen akustischen Signalen z.B. kann das Richtungs- und Entfernungsverständnis virtueller Objekte fördern.

Umgebungsbezogene Unterstützung der Signaletik indes meint Maßnahmen zur besseren Entwicklung eines mentalen Modells der Umgebung, z.B. durch Schilder oder anderweitige umgebungsbezogene Hinweise. Im Kontext von Augmented Reality kann unter anderem versucht werden, den Tiefeneindruck, hauptsächlich bei Verwendung von monoskopischer Darstellung, zusätzlich durch die Illustration physikalischer Begebenheiten innerhalb der virtuellen Welt zu stärken. Durch Schatten oder andere Beleuchtungseffekte kann beispielsweise die Lage eines Objektes leichter nachvollziehbar gemacht werden. Darüber hinaus können in dafür geeigneten AR-Anwendungen auch erwähnte umgebungsbezogene Hinweise, virtuelle Wegpunkte oder Übersichtskarten zur Unterstützung der Orientierung eingesetzt werden.

Systemeinstellungen

Augmented Reality-Systeme benötigen oftmals auch die Möglichkeit, zusätzliche Einstellungen zu tätigen. Diese Einstellungen lassen sich meist eher der Metaebene der AR-Anwendung zuordnen – wie die Auswahl der zu verwendenden virtuellen Szene, das Setzen der Systemlautstärke etc. Graphische Menüs aus Desktop-Anwendungen können dabei nicht ohne Weiteres übernommen werden, da die klassischen Eingabegeräte nicht vorliegen und das Sichtfeld der AR-Anwendung nicht durch in Desktop-Programmen übliche Fenster eingeschränkt werden soll.

Sollen graphische Menüs trotz dessen in AR-Anwendungen verwendet werden, stellt sich – vor allem bei Verwendung von Head-Mounted Displays – die Frage nach der Platzierung. Graphische Menüs allgegenwärtig z.B. am oberen Rand des Sichtfeldes des Benutzers statisch anzuzeigen, würde die visuelle Wahrnehmung der Realität einschränken und zudem den Grad der Natürlichkeit der Benutzung des Systems schmälern. Weiterhin wäre das Problem der Interaktion mit einer virtuellen Schnittstelle, welches sich nicht in die reale Umgebung einfügt, zu lösen.

Dementsprechend scheint es effektiver, solche Menüs direkt in den umge-

benden Raum zu verlagern. Je nach Kontext des Menüs könnte es relativ zum Kopf, zu einzelnen Körperteilen oder auch zu Objekten der Umgebung angezeigt werden. Auch wurden Experimente mit einem persönlichen Interaktionspanel getätigt (siehe Abbildung 2.17), welches der Benutzer mit sich führte und auf dem mittels Tracking das Menü platziert wurde. Aus heutiger Sicht scheint jedoch ein Smartphone oder ein Tablet, welches die Anzeige von und die Interaktion mit Menüs komfortabel und robust durchführt und mittels drahtloser Übertragung mit dem eigentlichen Augmented Reality-System verbunden werden könnte, die bessere Lösung zu sein, wenn ohnehin zusätzliches Gerät eingesetzt werden soll. Meist möchte man jedoch auf zusätzliche Hardware am Körper weitestgehend verzichten.



Abbildung 2.17: Das Personal Interaction Panel (siehe [SG97], S. 8)

Ganz gleich, welcher der genannten Orte im Raum genutzt wird, um ein graphisches Menü anzuzeigen, konventionelle Menüs sind zweidimensional und die Eingabe basiert auf zwei Freiheitsgraden – den Positionen von Berührungen mit der Interaktionsebene des Menüs. Da AR-Trackingsysteme jedoch sechs Freiheitsgrade liefern, müssen diese für eine funktionierende Interaktion mit dem Menü passend auf die Interaktionsebene projiziert werden. Zudem sollte diese Interaktionsebene so im Raum platziert werden, dass die Interaktion im Hinblick auf anthropometrische Aspekte durchführbar ist und der Benutzer nicht zu sehr belastet wird.

Die konkrete Interaktion mit dem Menü kann also über einfaches Drücken

von Buttons, realisiert durch Marker, erfolgen; variabler ist jedoch die Nutzung von Sprache oder Gesten. In diesen Fällen sei jedoch angemerkt, dass der Benutzer zunächst wissen muss, welche Gesten oder Sprachbefehle er verwenden kann. Somit muss er das notwendige Interaktionsvokabular entweder vor der Nutzung erlernen oder hilfreiche Anleitungen müssen ihm visuell während der Benutzung des Systems angezeigt werden.

Auch benötigt das System die Information, wann es angesprochen wird. Dazu kann ein Button (z.B. ebenfalls realisiert als Marker) eingeführt werden, der eine „push to talk“- respektive „push to act“-Funktionalität bereitstellt, den der Benutzer also drücken muss, um mit der Interaktion beginnen zu können. Vom Benutzer unbeabsichtigte Anweisungen werden dadurch ausgeschlossen. Zudem sollte der Benutzer eine Rückmeldung erhalten, sobald das System seine Arbeit aufnimmt und die Eingaben des Benutzers wahrnimmt. Erfolgt eine solche Rückmeldung nicht innerhalb kurzer Zeit, wirkt dies verwirrend, da der Benutzer nicht einschätzen kann, weshalb das System auf – für es eventuell nicht erkenn- bzw. interpretierbare – Eingabeversuche nicht reagiert.

Die Eingabe mittels Sprach- und Gestensteuerung sind für den Benutzer anstrengend, geistig sowie, im Falle von Gesten vor allem, körperlich. Dies sollte bei der Konzeptionierung von Augmented Reality-Systemen unter Verwendung dieser Interaktionsformen entsprechend beachtet werden, gerade dann, wenn es sich um zentrale und häufig wiederkehrende Tätigkeiten handelt.

Symbolische Eingaben

Anwendungen der Augmented Reality benötigen in manchen Fällen zusätzlich die Eingabe von Texten oder Zahlen. Vor allem vor dem Hintergrund, dass AR in Zukunft mehr und mehr Einzug in den Alltag halten wird, wo symbolische Eingaben eine wichtige Rolle spielen, lohnt sich eine genauere Betrachtung des Themas. Nachfolgend werden die gängigsten Techniken kurz vorgestellt.

Der nächstliegende Ansatz für symbolische Eingaben lautet, die Tastatur portabel zu machen. Dabei existieren vielfältige Möglichkeiten bzgl. Anordnung und Zuordnung von Tasten. Um sich der Tastatur-Hardware zu entledigen, können Trackingsysteme benutzt werden, um symbolische Eingaben mittels virtueller Tastatur zu tätigen. Auf diese Art können auch stiftbasierte Eingaben erkannt und umgesetzt werden: Der Benutzer vollzieht dazu die notwendigen Bewegungen zum Schreiben einzelner Buchstaben, und zwar mit einem getrackten Stift oder Finger und frei in den Raum oder – der Ergonomie wegen – auf einer Fläche. Das Trackingsystem kann diese Bewegungen nun erkennen und in einem weiteren Schritt

eine Schriftzeichenerkennung durchführen, um den finalen Text zu erhalten.

Auch im Falle symbolischer Eingaben sind darüber hinaus sprach- und gestenbasierte Systeme denkbar. Dafür notwendige Gesten können in diesem Fall mit Gebärdensprache verglichen werden; das zu erlernende Vokabular wäre entsprechend umfangreich. Dieser Bereich wird daher heute kaum noch weiter verfolgt. Sprachbasierte Systeme sind im Gegensatz dazu immer noch Gegenstand aktueller Forschung. Der Grund dafür könnte in der Unabhängigkeit der Sprache vom optischen Medium zu finden sein. Gerade in sicherheitskritischen Situationen wie dem Führen eines PKWs oder eines Fahrrads im Straßenverkehr, welche die volle Aufmerksamkeit des visuellen Kanals benötigen, bietet Spracheingabe eine interessante Alternative.

2.4.3 Anforderungen an User-Interfaces

Da in den meisten Augmented Reality-Systemen auf den Einsatz von Tastatur und 2D-Maus verzichtet wird, sind, wie bereits erwähnt, andere Interaktionsformen zur Steuerung des Systems notwendig. Im Bereich konventioneller Anwendungen stehen Entwicklern umfangreiche Bibliotheken mit Interface-Elementen zur Verfügung. In Augmented Reality-Anwendungen hat sich jedoch noch kein einheitlicher Standard durchgesetzt. Die grundlegenden Richtlinien für das User Interface Design konventioneller Anwendungen gelten jedoch ebenso für solche Systeme. Demzufolge sollte ein User-Interface daher:

- intuitiv und leicht erlernbar sein,
- schnell und einfach zu bedienen sein,
- plausibles und verständliches Feedback bieten und
- den Benutzer nicht zu sehr von der eigentlichen Anwendung ablenken.

3 Konzept

In diesem Abschnitt erfolgt die Vorstellung der unterschiedlichen Konzepte, welche notwendig waren, um frühzeitig eine genaue Vorstellung vom zu entwickelnden System zu bekommen. Im Folgenden wird zunächst das Grundkonzept des prototypischen Systems präsentiert; daraufhin werden die Konzepte für die Anwendung und die Interaktion mit dem System vorgestellt.

3.1 Grundkonzept

Für die Entwicklung eines Anwendungs- und Interaktionskonzeptes war zunächst ein Grundkonzept notwendig, welches im Folgenden vorgestellt wird. Bei diesem Grundkonzept wurde angenommen, dass der Benutzer eine sogenannte *Datenbrille* trägt, also eine Brille, welche in irgendeiner Form über einen mobilen Computer – intern oder extern – und somit bzgl. Rechenkapazität und Konnektivität über die Möglichkeiten eines aktuell durchschnittlichen Smartphones verfügt. Des Weiteren verfügt die Datenbrille gemäß Konzept über eine Farbbildkamera sowie ein optisches Trackingsystem, wodurch die Brille in der Lage ist, das durch sie sichtbare Bild der realen Welt mit virtuellen visuellen Inhalten zu überlagern. Es handelt sich somit um ein Head-Mounted Display, welches Augmented Reality ermöglicht. Die weiteren Spezifikationen der Datenbrille bzgl. der in Unterabschnitt 'Displays' genannten möglichen Arten und Formen spielen an dieser Stelle keine Rolle, da das Konzept unabhängig von den späteren Hardware-Ausprägungen erarbeitet wurde.

Mithilfe eines solchen Systems können dem Benutzer somit nun Informationen in Form computergenerierter Bilder und Texte vor seinem Auge eingeblendet werden, wodurch die reale Welt an dieser Stelle seines Sichtfeldes teilweise überlagert wird. Somit wird Augmented Reality möglich und mittels entsprechender Anwendungen können dem Benutzer dieses Systems nun hilfreiche Informationen und Inhalte eingeblendet werden. Weiterhin wurde angenommen, dass das System dem Benutzer mithilfe geeigneter Interaktionsformen die Möglichkeit bietet, mit der laufenden Anwendung zu interagieren und sie zu steuern.

Im Rahmen dieser Arbeit liegt der Fokus auf der Grundlagenforschung bzgl. der Interaktion mit Augmented Reality-fähigen Datenbrillen; in dieser ersten Stufe soll sich die Darstellung umfassender virtueller Inhalte daher in Grenzen halten. Man könnte im Falle des hier vorgestellten Grundkonzeptes, bezogen auf die Menge an virtuellen Inhalten, somit auch von einem *Head-up-Display* (kurz: HUD) sprechen, bei welchem dem Träger der

Datenbrille einzig auf das Wesentliche reduzierte Informationen in sein Sichtfeld projiziert werden. Allerdings kann der Benutzer mit Inhalten eines Head-up-Displays gemeinhin nicht direkt interagieren respektive die Darstellung in aktiver Weise steuern, weshalb der Begriff der Augmented Reality bezogen auf das hier vorgestellte Grundkonzept letztlich passender erscheint.

Durch die genannten Merkmale hat das System das Potential, das Arbeitsleben spezieller Berufe wie auch das alltägliche Leben vieler Benutzer in unzähligen vorstellbaren Situationen zu vereinfachen. Dies beinhaltet beispielsweise die Situation eines medizinischen Eingriffs: Während der ausführende Arzt des Eingriffs hochkonzentriert mit beiden Händen beschäftigt ist, könnte ihm ein solches System virtuelle Zusatzinformationen oder Hilfestellungen geben, ohne dass der Arzt dabei auf einen externen Monitor schauen muss und dabei den Fokus auf den Patienten verliert. Diese Funktionsweise lässt sich zugleich auf all solche Aufgabenbereiche übertragen, bei denen Menschen zur Ausübung der Tätigkeit ihre Hände nicht ständig dazu zur Verfügung haben, zusätzliche, ihm wichtige Informationen einzuholen. Ein alltägliches Beispiel dessen ist das Kochen: Während man Essen zubereitet, könnte das gewünschte Rezept mithilfe eines solchen Systems als virtueller Inhalt stückweise angezeigt werden und sich der Fokus dabei, mit bestimmten Kommandos an das System, auf den gerade wichtigen Teil des Rezepts ändern lassen. Auf die gleiche Weise und mittels der Einblendung von Anleitungen kann ein solches System auch Monteuren oder Installateuren bei der Arbeit helfen.

Ebenso kann das beschriebene System dazu genutzt werden, Navigationshinweise während dem Fahrrad- oder Autofahren anzuzeigen. Auch könnte das System im Falle der Erkundung fremder Orte zu Fuß auf besondere Dinge in der näheren Umgebung hinweisen oder die Navigation mittels entsprechender Anzeigen erleichtern. Überdies kann ein solches System im Grunde alle Aufgaben erfüllen, die ein aktuell erhältliches Smartphone mittels Internetkonnektivität ebenfalls schon erfüllt: (Video-)Telefonie, Abspielen von Medien, Terminplanung, Kommunikation über E-Mails oder Internet-Chat (sogenanntes *Instant Messaging*) sowie das Surfen im Internet. Einzig die Anzeige würde vom Smartphone-Display in das Display der Datenbrille wandern und die Interaktion ist statt mittels Touchscreen über geeignete Interaktionsformen mit der Datenbrille zu erbringen. An dieser Stelle sei jedoch darauf hingewiesen, dass Letzteres – also die Entwicklung von Möglichkeiten zur Interaktion mit Datenbrillen oder ähnlichen Systemen – keine triviale Aufgabe darstellt. Es bedarf daher umfangreicher Forschungsarbeit, um geeignete Konzepte ausfindig zu machen. Die vorliegende Arbeit soll, wie erwähnt, einen Beitrag dazu leisten.

Die oben beschriebenen Einsatzmöglichkeiten des hier vorgestellten Sys-

tems implizieren jedoch die Anforderung, durch die Einblendung virtueller Inhalte die Sicht des Benutzers auf die reale Welt möglichst wenig einzuschränken. Bezogen auf den beschriebenen Anwendungsfall des Systems während medizinischer Eingriffe ist diese Anforderung wesentlich, in anderen Fällen eventuell weniger. Zusammenfassend stellt die Einblendung der virtuellen Inhalte an Stellen des Sichtfeldes, die der Benutzer nicht verdeckt wissen möchte, jedoch ein mehr oder weniger großes Problem dar, welches ein Sicherheitsrisiko mit sich bringen kann. Als Beispiel sei genannt, dass sowohl Fahrradfahrer als auch Autofahrer darauf angewiesen sind, ihr Sichtfeld nur zu einem möglichst geringen Teil und nur an einem dafür geeigneten Ort des eigenen Sichtfeldes überlagert zu sehen, um sich und andere im Straßenverkehr nicht zu gefährden. Somit sind sowohl die Größe als auch die Stelle der Überlagerung kritisch. Das Sicherheitsrisiko kann zudem eingeschränkt werden, wenn die virtuellen Inhalte ausschließlich dann eingeblendet werden würden, wenn der Benutzer sie benötigt.

Um dem genannten potenziellen Sicherheitsrisiko bei der Überlagerung der realen Welt mit virtuellen visuellen Inhalten entgegenzuwirken, wurde folgende Idee entwickelt: Die überlagernden Inhalte werden einzig an der Stelle seines Sichtfeldes angezeigt, an der er sein Handgelenk sieht. Damit ist sichergestellt, dass der Benutzer größtenteils selbst entscheidet, wann und an welcher Stelle sein Sichtfeld mit Inhalten überlagert wird. Es kann naturgemäß auch in diesem Fall zu der Situation kommen, dass der Benutzer seinen Unterarm, nicht jedoch die virtuellen Inhalte sehen möchte. Allerdings wird dem daraus entstehenden Sicherheitsrisiko zusätzlich dadurch vorgebeugt, dass die Größe der Inhalte während ihrer Einblendung durch die Größe des Unterarms des Benutzers – bezogen auf die aktuelle Perspektive im realen Sichtfeld – beschränkt ist.

Der ebenso wichtigen Komponente der Intuitivität dieses Grundkonzeptes wird dabei dadurch Rechnung getragen, dass der Benutzer eben erwähnte Entscheidung, wann und wo die Einblendung stattfindet, allein mittels der Bewegung seines Unterarms tätigt. Zusammengefasst wird, aus Sicht des Benutzers, eines seiner Handgelenke mit visuellen Inhalten überlagert, sobald das Handgelenk in sein Sichtfeld gelangt. Abbildung 3.1 stellt einen frühen Konzeptentwurf dar und veranschaulicht das Prinzip.

Nach der Ausarbeitung dieses Grundkonzeptes galt es, eine Anwendung für ein solches mögliches System zu konzeptionieren, um mit dieser Anwendung wiederum die Interaktion – welche zuvor ebenso eines Konzeptes bedarf – mit dem System evaluieren zu können. Im Folgenden werden die Konzepte für Anwendung und Interaktion vorgestellt.



Abbildung 3.1: Konzeptentwurf: Visuelle Informationen überlagern Handgelenk des Benutzers

3.2 Anwendung

Bei der Entwicklung des Anwendungskonzeptes erschien es hilfreich, möglichst viel der für die Anwendungsentwicklung verfügbaren Zeit in die Entwicklung einer einzigen, jedoch vorher gründlich konzeptionierten Anwendung zu investieren. Dazu wurden mehrere Kriterien aufgestellt, anhand derer die einzelnen Anwendungsideen bewertet und gegeneinander aufgewogen wurden. Folgende Kriterien wurden dabei berücksichtigt:

- Machbarkeit,
- Simplizität des Anwendungsfalls,
- hohe Relevanz im alltäglichen Leben,
- intuitive Bedienung,
- erkennbarer Mehrwert

Die Kriterien *Machbarkeit* und *Simplizität des Anwendungsfalls* zielen dabei darauf ab, die Durchführbarkeit im Allgemeinen sicherzustellen und zu komplexe Funktionalitäten aufgrund des damit verbundenen erhöhten Potentials für unerwartet auftauchende Probleme während der Realisierung auszuschließen. Letzteres meint Schwierigkeiten, welche nicht im Kontext

der Interaktion stehen und dessen Bearbeitung daher keinen wesentlichen Teil dieser Arbeit ausmachen sollte.

Die Punkte *hohe Relevanz im alltäglichen Leben* und *intuitive Bedienung* haben hingegen zum Ziel, auch nicht besonders technikaffinen Personen die Nutzbarkeit der Anwendung einfach und schmackhaft zu gestalten. Das letzte Kriterium *erkennbarer Mehrwert* bezieht sich auf die Sinnhaftigkeit der Anwendung und den Mehrwert gegenüber bereits vorhandenen Anwendungen, welche in Verbindung mit bestehenden Systemen und bekannten User-Interfaces funktionieren. Es gilt hierbei zudem, durch einen – später auch durch den Benutzer selbst erfahrbaren – Mehrwert der Anwendung, die natürliche Barriere eines Benutzers, sich auf neue Interaktionskonzepte einzulassen, zu durchbrechen.

Im Verlauf des Prozesses der Ideenfindung kamen folgende Vorschläge für Anwendungen auf, welche im Anschluss unter Zuhilfenahme obiger Kriterien beurteilt wurden:

- Eventmanagement (inklusive Benachrichtigung über eingehende E-Mails, Chat-Nachrichten, Anrufe, SMS etc.),
- Terminplanung,
- Navigation für Fußgänger,
- Videotelefonie,
- E-Mail-Klient,
- Musikplayer

Während all diese Funktionen von aktuell erhältlichen Smartphones schon mehr oder minder gut bewältigt werden, gibt es für ein System, bei dem der Benutzer virtuelle Inhalte auf sein Handgelenk projiziert sieht, erhebliche Unterschiede bezüglich der oben erwähnten Kriterien für potenzielle Anwendungen.

Einem E-Mail-Klient und einer Anwendung zur Terminplanung mangelt es beispielsweise an Möglichkeiten für eine entsprechend intuitive Bedienung, da in diesen Fällen präzise Texteingaben notwendig sind, welche ohne entsprechend präzise Interaktionsmöglichkeiten schwer zu realisieren scheinen. Mit einer solchen Anwendung würde dem neu zu entwickelnden User-Interface die zusätzliche Bürde auferlegt, exakte Texteingaben zu ermöglichen.

Eine Eventmanagement-Anwendung hingegen spiegelt keinen simplen Anwendungsfall wider. Zudem ist der Benutzer bei der Benachrichtigung über eingehende E-Mails o.ä. auch gewillt, auf solche Ereignisse zu reagieren, was in der Konsequenz wiederum dazu führt, dass die Anwendung

noch komplexer und weniger machbar zu werden droht. Auch wäre an dieser Stelle zu diskutieren, ob es überhaupt allgemein gewollt oder für den einzelnen von Vorteil ist, bei Tragen des beschriebenen Systems zu jeder Zeit und an jedem Ort über eingehende Nachrichten informiert zu werden. Selbst Smartphones verfügen heute schon über die Möglichkeit, zu jeder beliebigen Zeit auf sich aufmerksam zu machen, sei es über das Display, die Lautsprecher oder den Vibrationsmotor – sobald es sich in der Nähe des Benutzers befindet und keine entsprechenden Restriktionen (lautlos, keine Vibration) eingestellt sind, kann das Smartphone den Benutzer von wichtigen Dingen ablenken und unwichtige Dinge in den Fokus rücken. Somit scheint eine Eventmanagement-Anwendung, welche über eingehende Nachrichten und Anrufe benachrichtigt, aus den genannten Gründen nicht sehr passend.

Bezogen auf die Idee des Musikplayers scheint das Kriterium der hohen Relevanz im alltäglichen Leben für manche Menschen zwar gegeben, doch erscheint eine solche Anwendung eher eine technische Spielerei darzustellen, der es an sinnvollem Mehrwert im Allgemeinen mangelt. Eine Videotelefonie- sowie eine Navigationsanwendung scheinen indes in sinnvollerer Weise nutzbar zu sein; weiterhin sind die Anwendungsfälle simpel und die Bedienung der Anwendungen vermutlich vergleichsweise intuitiv realisierbar.

Betrachtet man allerdings die Kriterien der Machbarkeit und des Mehrwertes genauer, hat die Idee der Videotelefonie gegenüber dem Vorschlag der Navigationsanwendung das Nachsehen: Da ein System wie in Unterabschnitt 'Grundkonzept' vorgestellt auch für unterwegs nützlich erscheint, jedoch aktuell nicht an jedem Ort und zu jeder Zeit ausreichend mobile Bandbreite garantiert werden kann – und sich dieses Problem auch kurz- bis mittelfristig nicht komplett beheben lassen wird – ist die Funktion der Videotelefonie, welche stark auf Internetkonnektivität angewiesen ist, eher nicht realisierbar. Darüber hinaus kann eine Navigationsanwendung in Verbindung mit dem oben beschriebenen System eine wirkliche Hilfe für Menschen darstellen, welche aufgrund körperlicher Handicaps nicht in der Lage sind, aktuell bestehende User-Interfaces wie Touchscreens zu bedienen.

Die Beurteilung aller Ideen anhand der obigen Kriterien ergab somit im Ergebnis, dass eine Navigationsanwendung erstellt werden soll. Bezug nehmend auf die Use-Cases einer solchen Navigationsanwendung wurde sodann festgelegt, dass der Benutzer einzig in der Lage sein soll, mittels einer zweidimensionalen Übersichtskarte zu navigieren. Das impliziert, dass der Benutzer den sichtbaren Kartenausschnitt bewegen können soll, um seine eigene Position auf der Karte ausfindig machen und ein oder mehrere, vorher anderweitig festgelegte, Ziele auf der Karte sehen und mögliche Routen

finden zu können.

Nachdem die Entscheidung für eine Navigationsanwendung feststand, wurde ein erster Konzeptentwurf angefertigt (siehe Abbildung 3.2). Dieser Entwurf zeigt unter anderem eine Satellitenansicht eines Kartenausschnitts inklusive Überlagerung mit Kartenmaterial wie Namen von Straßen und Gewerbebetrieben, wie sie aus Web-Services wie „Google Maps“ bekannt ist. Zu diesem Zeitpunkt wurden jedoch noch keinerlei Überlegungen bezüglich des konkreten User-Interfaces der Anwendung und der Möglichkeiten zur Interaktion mit selbigem angestellt. Somit galt es in einem nächsten Schritt, ein User-Interface zu entwerfen und ein entsprechendes Interaktionskonzept auf dessen Basis zu entwickeln.



Abbildung 3.2: Erster Konzeptentwurf: Navigationsanwendung

3.3 Interaktion

Im Anschluss an die Erstellung des Anwendungskonzeptes wurde nun zunächst versucht, ein prototypisches User-Interface für eine Navigationsanwendung zu entwerfen. Abgesehen von eventuell notwendigen symbolischen Eingaben – beispielsweise zur Angabe eines Navigationszieles – und allgemeiner Möglichkeiten zur Interaktion mit einem graphischen System-Menü – wie das Setzen von System-Einstellungen mittels einfachem Drücken von Buttons – sind gerade die notwendigen Möglichkeiten zur Steuerung der zweidimensionalen Kartenansicht interessant und zudem vom im vorigen Unterabschnitt 'Anwendung' bereits beschriebenen

Use-Case gefordert. Diese Steuerungs- respektive Manipulationsmöglichkeiten beinhalten:

1. das Verschieben,
2. das Drehen sowie
3. das Vergrößern/Verkleinern

des aktuellen Kartenausschnitts.

Mittels dieser drei Möglichkeiten zur Manipulation lassen sich im Allgemeinen zudem jegliche zweidimensionalen Inhalte wie Fotos, Texte oder Webseiten beliebig ausrichten und in ihrer Größe anpassen. Während die Entscheidung zur Fokussierung auf diese Manipulationsmöglichkeiten frühzeitig getroffen wurde – und weitere symbolische Eingabemöglichkeiten oder System-Menüs damit, auch aus Zeitgründen, nicht weiter verfolgt wurden –, war vorerst nicht klar, wie die Interaktion im Detail erfolgen sollte. Zu Anfang wurde daher das in Abbildung 3.3 gezeigte Konzept erstellt.



Abbildung 3.3: 1. Entwurf eines Interaktionskonzeptes: Buttons zur Manipulation des Kartenausschnitts

Darauf ist zu sehen, wie eine Vielzahl an Buttons eingeführt wurde, um die oben beschriebenen Manipulationen des Kartenausschnitts bewerkstelligen zu können: Es existieren Buttons für das Verschieben um einen festen Versatz in alle vier Richtungen, für das Erhöhen und Verringern der Zoom-Stufe (= Vergrößern/Verkleinern um vorher festgelegte Faktoren) und für

das Drehen der Karte um einen festen Winkel im Uhrzeigersinn.

Dieses Konzept wurde daraufhin einer Präevaluation unterzogen. Dazu wurden vier Personen auserwählt, welche die Gemeinsamkeit haben, sich als nicht sehr technikaffin einzuschätzen. Auf diese Eigenschaft wurde geachtet, um die Meinung eines Durchschnittsnutzers ohne signifikante technologische Vorkenntnisse anzunähern. Somit sollte eine grundlegend kritische Auseinandersetzung mit dem vorgestellten Konzept – ohne Rücksicht auf sonst eventuell feststellbare Schwierigkeiten bezüglich der technischen Realisierbarkeit – sichergestellt werden. Die Personen sollten anhand des in Abbildung 3.3 abgebildeten Konzeptentwurfs beurteilen, inwiefern sie an ein solches Interaktionskonzept für gut befinden und warum.

Die einhellige Meinung der befragten Personen innerhalb der Präevaluation war, dass eine solche Lösung mittels vielerlei Buttons nicht sehr intuitiv und eine Lösung wie im Falle der Bedienung von Smartphone-Anwendungen stattdessen wünschenswert sei. Darüber hinaus berichtete jede der befragten Personen, ein Smartphone einem solchen System mit derart „kompliziertem User-Interface“ vorziehen zu wollen. Eine gewisse Neugier und ein grundlegendes Interesse für das Konzept ließ sich jedoch im Falle aller Befragten feststellen. Der erhaltenen Kritik wurde sich sodann angenommen.

In Bezug auf Interaktion lässt sich sagen, dass die drei oben erwähnten Manipulationsmöglichkeiten von aktuell erhältlichen Smartphones, wie sie die befragten Personen – auch ohne technikaffin zu sein – allesamt kennen und zumindest bereits mal benutzt haben, mittels Touchscreens und einfachen Multitouch-Gesten überaus intuitiv umgesetzt werden. Dabei werden mehrere, potenziell gleichzeitige Berührungen des Touchscreens seitens des Benutzers vom Smartphone erkannt und bezüglich ihrer Position auf dem Touchscreen getrackt. Die entsprechenden Gesten werden folgendermaßen bezeichnet:

1. verschieben = *pan*
2. drehen = *rotate*
3. vergrößern/verkleinern = *pinch to zoom*

Somit ist von Interesse, inwieweit sich die durch Smartphones eingeführten und etablierten Multitouch-Gesten auch als Interaktionsformen mittels optischem Tracking im Raum umsetzen lassen. Eine Idee an dieser Stelle lautet, diese Multitouch-Gesten als solche auf flächigen, also zweidimensionalen Markern nutzbar zu machen. Dabei gibt die Verdeckung von Markern – oder die Verdeckung von Teilen von Markern – durch die Finger des Benutzers Hinweise auf potenzielle Berührungen mit der Markeroberfläche. Durch geschickte Kombination dieser Hinweise lassen sich idealerweise ro-

buste Rückschlüsse auf ausgeführte Gesten ziehen. Die detaillierte Ausarbeitung dieser Idee hängt jedoch von den Möglichkeiten den Trackings ab und wird daher in Abschnitt 'Implementierung' durchgeführt.

Eine Schwierigkeit könnte dabei jedoch sein, dass sich im Falle von optischem Tracking zweidimensionaler Marker aus der Draufsicht keine genauen Zeitpunkte der Berührung der Markeroberfläche ausfindig machen lassen. Die Möglichkeiten dieser Idee gilt es daher im weiteren Verlauf der Arbeit zu überprüfen. Als Grundlage dessen und als Antwort auf die Präevaluation wurde sodann das finale Interaktionskonzept entwickelt, welches in Abbildung 3.4 zu sehen ist.



Abbildung 3.4: 2. Entwurf des Interaktionskonzeptes: Manipulation des Kartenausschnitts ohne Buttons

Wie man erkennen kann, wurde das User-Interface, auch im Hinblick auf die in Unterunterabschnitt 'Anforderungen an User-Interfaces' aufgezählten Anforderungen an User-Interfaces, um sämtliche, vorher existente Buttons zur Regelung der Zoom-Stufe, der Rotation und der Position des Kartenausschnitts bereinigt. Dadurch soll eine möglichst einfache, leicht erlernbare und intuitive Interaktion wie im Falle der Bedienung von Smartphone-Anwendungen ermöglicht werden, welche somit mittels der genannten, bewährten zweidimensionalen Gesten *Pan*, *Rotate* und *Pinch-to-zoom* erfolgt. Im Folgenden werden diese genauer beleuchtet.

Bei einer *Pan*-Geste (vgl. Abbildung 3.5) wird nur ein Finger benutzt, um die Ansicht des zweidimensionalen Inhalts zu verschieben. Sobald die Geste aktiviert ist – also eine Fingerkuppe auf der Eingabefläche erkannt wird

– und sich der Finger bewegt, bewegt sich der „unter“ der Fingerkuppe liegende Inhalt in absoluten Abständen, analog zur Fingerkuppe, mit.

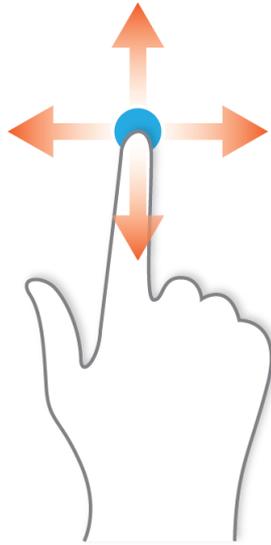


Abbildung 3.5: Veranschaulichung einer Pan-Geste (siehe [Wik13a])

Im Falle einer Rotate-Geste, welche in Abbildung 3.6 dargestellt ist, werden hingegen zwei Finger zur Eingabe benötigt. Sobald eine Erkennung zweier Fingerkuppen initial stattfindet, wird der Vektor, den die Positionen der Fingerkuppen auf der Eingabefläche aufspannen, in Relation zu einem internen Koordinatensystem gesetzt und ein initialer Winkel – beispielsweise zwischen diesem Vektor und der horizontalen Achse der Eingabefläche – bestimmt. Der Benutzer kann nun Drehbewegungen einer oder beider Fingerkuppen um das Zentrum beider Fingerkuppen vornehmen. Gleichzeitig wird seitens des Systems der initial bestimmte Winkel nun als Referenz für die veränderten Positionen der Fingerkuppen verwendet: Zu jedem beliebigen Zeitpunkt dieser Folge von erkannten Positionen lässt sich nun die Differenz zwischen dem Referenzwinkel und dem aktuell berechneten Winkel dazu einsetzen, den Inhalt entsprechend zu drehen.

Eine Pinch-to-zoom-Geste (zu deutsch etwa: Kneifen zum Vergrößern/-Verkleinern) benötigt ebenfalls den Einsatz zweier Finger. Dabei wird der Abstand der Fingerkuppen zueinander aktiv verändert, wodurch sich die Zoom-Stufe des Inhalts entsprechend ändert: Bei Vergrößerung des Abstandes der Fingerkuppen zueinander vergrößert sich der Inhalt, bei Verkleinerung des Abstandes verkleinert er sich analog. Durch den Ein-

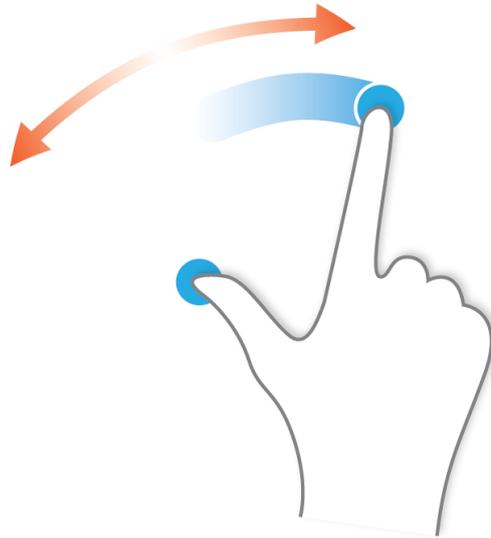


Abbildung 3.6: Veranschaulichung einer Rotate-Geste (siehe [Wik13a])

satz möglichst vieler, sich kaum merklich voneinander unterscheidender Zoom-Stufen wird der Eindruck einer kontinuierlichen Vergrößerung respektive Verkleinerung des Inhalts erzeugt. In der Literatur existiert darüber hinaus keine einheitliche Benennung dieser Geste; oft wird sie auch in zwei Bestandteile aufgeteilt: Eine *Pinch-Open*-Geste zur Verkleinerung und eine *Pinch-Close*-Geste zur Vergrößerung des Inhalts. Letztere sind in Abbildung 3.7 und Abbildung 3.8 jeweils veranschaulicht.

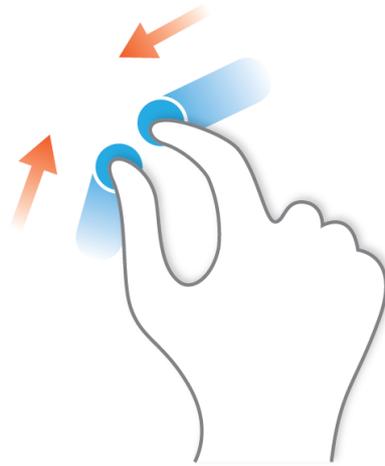


Abbildung 3.7: Veranschaulichung einer Pinch-Close-Geste (siehe [Wik13a])

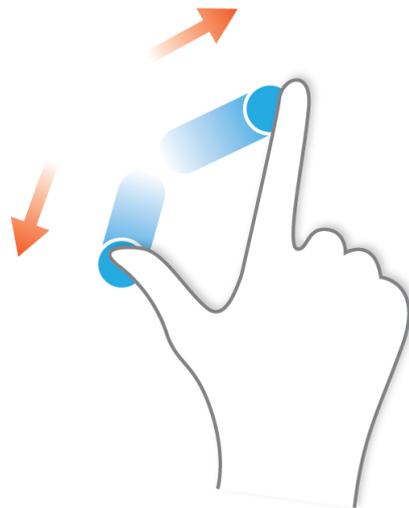


Abbildung 3.8: Veranschaulichung einer Pinch-Open-Geste (siehe [Wik13a])

4 Aufbau des prototypischen Systems

Dieser Abschnitt beschäftigt sich mit dem Bau des prototypischen Augmented Reality-Systems. Zuerst werden die Möglichkeiten der Verwendung unterschiedlicher Hardwarekomponenten diskutiert und dabei die Entscheidung für die zum Bau des Prototyps verwendete Kombination aus Komponenten nachvollzogen. Anschließend wird der Aufbau des Systems inklusive auftretender Herausforderungen erläutert und mittels Fotos abgebildet, um auch einen visuellen Eindruck des Prototyps zu geben.

4.1 Komponentendiskussion

Nachdem die in Unterabschnitt 'Grundkonzept' vorgestellte Grundidee für das Augmented Reality-System feststand, wurden die unterschiedlichen notwendigen Hardwarekomponenten für den Bau eines solchen Systems ermittelt. Zunächst geht aus dem Grundkonzept hervor, dass das System eine Brille als Ausgabe-Komponente beinhalten soll, welche zusammen mit einer Kamera und einem optischen Trackingsystem in der Lage ist, durch die Kamera erfasste Bilder der realen Welt mit virtuellen Inhalten zu überlagern und sich so ergebende Bilder einer erweiterten Realität (Augmented Reality) für den Träger der Brille sichtbar zu machen. Über die weiteren Ausprägungen des Systems wird nachfolgend diskutiert.

4.1.1 Webcam und Notebook im Rucksack vs. Smartphone-Kamera und Android-Applikation

Neben einem zentralen Rechner zur Aufnahme, Verarbeitung und Weitergabe der Daten an die Ausgabe-Komponente wird eine Kamera-Komponente benötigt, welche in der Lage sein soll, die Umgebung möglichst hochauflösend und in möglichst hoher Bildfrequenz zu erfassen und an die Recheneinheit weiterzugeben. Durch diese Anforderungen wird sichergestellt, dass eine Augmented Reality-Umgebung mit flüssig animierten virtuellen Objekten theoretisch umzusetzen ist. Zudem soll die Kamera-Komponente durch eine vorzugsweise geringe Gesamtgröße am Kopf des Benutzers montiert werden können. Somit ist gewährleistet, dass die Kamera bei der Erfassung der realen Umwelt diese annähernd aus derselben Richtung betrachtet wie die Augen des Benutzers des Systems, wodurch das Überlagern des realen Sichtfeldes durch virtuelle Inhalte erst ermöglicht wird.

Die zentrale Rechner-Komponente hingegen soll einzig die Anforderung nach ausreichend Leistung – welche zu dieser frühen Zeit der Komponenten-

tendiskussion allerdings nicht exakt beziffert werden kann – erfüllen und über die notwendigen Schnittstellen zur Datenein- und ausgabe verfügen. Wünschenswert ist darüber hinaus die Möglichkeit zur Verwendung einer bekannten Software-Plattform, um den Grundstein der Software-Entwicklungsarbeit auf diesem System zu legen und die eventuelle Wiederverwendung bereits existierender Teillösungen zu ermöglichen.

Hinsichtlich des Gesamtsystems scheinen des Weiteren ein geringes Gewicht und eine hohe Mobilität als erforderliche Kriterien, um das System später eventuell auch im mobilen Bereich testen zu können. Außerdem wäre es von Vorteil, wenn sich das ganzheitliche System später eher schnell als langsam aus seinen einzelnen Komponenten aufbauen ließe.

Zieht man all diese Anforderungen in Betracht, kristallisieren sich zwei Kombinationen aus Rechner- und Kamera-Komponenten heraus: Eine, bei der die Rechner-Komponente durch ein Notebook, welches in einem Rucksack auf dem Rücken getragen wird, und die Kamera durch eine handelsübliche eigenständig arbeitende Kamera abgebildet wird; und eine zweite, bei der ein Smartphone gleichzeitig als Rechner- und – mittels integrierter Kamera – als Kamera-Komponente fungiert, mit der Voraussetzung, dass die nötigen Schnittstellen von der Bilddatenverarbeitung hin zur Ausgabe der Daten existieren und nutzbar sind.

Eine Konstruktion eines vollständigen Augmented Reality-Systems aus erstgenannter Kombination von Komponenten zeigt Abbildung 4.1.



Abbildung 4.1: Konstruktion eines AR-Systems mittels Notebook, Rucksack, AR-Brille und Webcam (siehe [Ins])

Die Vorteile eines solchen Systems liegen u.a. in der Performance, da man in dem Notebook durchaus die aktuell leistungsstärkste, auf dem

Consumer-Markt verfügbare Hardware verbauen könnte, und in der Vielfältigkeit verwendbarer Software, bezogen auf den Einsatz von Unix- oder Windows-basierten Plattformen. Zudem wiegt man sich mit einem solchen System bezüglich der Lösbarkeit eventuell auftretender, komplexer softwaretechnischer Problemstellungen auf der sicheren Seite, da sowohl das Notebook als auch die eigenständig arbeitende und flexibel austauschbare Kamera hinsichtlich dessen maximale Möglichkeiten bieten. Als nachteilig lassen sich jedoch das relativ hohe Gewicht des Notebooks, welches sich negativ auf den Faktor der Mobilität auswirkt, und der geringe Tragekomfort sehen.

Indes liegen einige Vorteile einer ausschließlich aus einem Smartphone bestehenden Rechner- und Kamera-Komponente auf der Hand: Anstatt umständlicher Verkabelungswege muss einzig das Smartphone zwecks Bildausgabe mit der Brille verbunden werden; die Bilderfassung sowie -verarbeitung erfolgt vollständig durch das Smartphone selbst. Abbildung 4.2 veranschaulicht die Umsetzung dieser Idee.

Als Software-Plattform des Smartphones wäre darüber hinaus das „Android Operating System“ (kurz: Android OS) von Google zu wählen, da es für eine große Bandbreite von mobilen Geräten verfügbar ist, eine gut dokumentierte *API* (Abkürzung für: *application programming interface*) für die verschiedenen Aspekte dieser Geräte bietet und bereits viel Funktionalität mit sich bringt, beispielsweise eine umfassende Rendering-Umgebung und den Zugriff auf die Bilddaten der integrierten Kamera. Der daraus resultierende Wegfall der Notwendigkeit, die eventuell fehleranfällige Kommunikation von Kamera zu Notebook – im Falle der erstgenannten Variante – selbst zu implementieren, ist an dieser Stelle nicht zu unterschätzen. Das zu tragende Gewicht eines Rucksacks inklusive Laptop entfällt weiterhin, und das Gewicht des Smartphone, befestigt am Kopf des Trägers, hält sich im Vergleich dazu in Grenzen. Insgesamt verfügt die Variante mit Smartphone somit über eine höhere Mobilität und Simplizität bezüglich des Gesamtsystems.

Die weiteren, oben genannten Kriterien – wie die der Möglichkeit zur Befestigung der Kamera-Komponente an die Brille bzw. den Kopf des Benutzers – erfüllen beide Hardware-Kombinationen. Zu der Forderung nach hinreichend guten Leistungen der gewählten Kamera-Komponente bezüglich Auflösung und Bildfrequenz ist zu sagen, dass diese Werte naturgemäß stark vom jeweiligen Gerät abhängen. Allerdings sollten sowohl gegenwärtige, günstige Webcams als auch integrierte Kameras aktuell erhältlicher Smartphones für den hier beschriebenen Zweck in jedem Fall ausreichen. Auch die Leistung von Smartphones sollte in Zeiten von darin verbauten Mehrkern-Prozessoren kein Hindernis darstellen, wenngleich die Verwendung eines Notebooks hingegen erheblich weniger Risiko bezüglich des



Abbildung 4.2: Konstruktion eines AR-Systems mittels Smartphone und AR-Brille

Erreichens von Grenzen der Rechenleistung bietet.

Vergleicht man jedoch die Vorteile beider Varianten erneut miteinander und schaut genauer hin, fällt auf, dass die Vorzüge der Kombination aus Notebook und Webcam einzig aus Entwicklersicht positiv ins Gewicht fallen. Bezüglich der Eigenschaften, die ein Benutzer des Systems als wichtig empfindet, schneidet die Smartphone-Variante besser ab – allen voran bezogen auf den Tragekomfort und die Einfachheit des Systemaufbaus.

Hinzu kommt, dass aktuelle Entwicklungen wie das bereits erwähnte „Project Glass“ von Google in dieselbe Richtung gehen: Hochmobile tragbare Systeme sollen den Alltag der Menschen bereichern. Dazu ist es mehr als notwendig, dass diese Systeme komfortabel zu tragen und zu nutzen sind. Bei den Möglichkeiten, die sich in dieser Hinsicht aktuell auftun, scheint die Verwendung traditioneller Computer-Hardware ein Schritt in die verkehrte Richtung. Unter diesen Gesichtspunkten fiel die Wahl auf die vorgestellte Variante mittels Smartphone und der Programmierung einer Applikation unter Android OS.

4.1.2 Optical See-Through vs. Video-See-Through

Wie in Unterabschnitt 'Displays' erläutert, gibt es im Falle von Head-Mounted Displays sowohl Lösungen mit Optical See-Through als auch welche mit Video See-Through Displays. Die Vor- und Nachteile beider Konzepte wurden an jener Stelle ebenfalls aufgezeigt.

Danach scheinen Optical See-Through Displays als ideal, sofern keine

merkliche zeitliche Verzögerung in der Darstellung der computergenerierten, überlagernden Inhalte auftritt. Ebenso stellt die Verwendung eines Optical See-Through Displays den oder die Entwickler des Systems vor die Herausforderung, die Kamera und das Trackingsystem derart präzise zu kalibrieren und mit dem reellen Sichtfeld des Benutzers zu registrieren, dass die Position der überlagernden Inhalte für den Benutzer hinreichend plausibel und exakt erscheint.

Video See-Through Displays haben diese Nachteile hingegen nicht und sind einfacher in der technischen Handhabung, da keine Registrierung vorgenommen werden muss und das Bild der Kamera inklusive der virtuellen Überlagerungen komplett – anstatt ausschließlich der virtuellen Inhalte an den jeweils richtigen Bildpositionen – dargestellt werden kann. Allerdings bieten Video See-Through Displays eine zeitlich verzögerte und vor allem reduzierte Sicht auf die reale Umwelt. Dadurch erhält der Träger weniger Informationen über die reale Welt als gewohnt, was den wahrgenommenen Sehkomfort meist sinken lässt.

Unter Berücksichtigung des prototypischen Charakters des zu entwickelnden Systems sowie der sehr beschränkten zeitlichen Ressourcen zur Entwicklung des Systems wurde das Prinzip des Video See-Through Displays an dieser Stelle bevorzugt. Obgleich im vorigen Abschnitt – bezüglich der Entscheidung für Rechner- und Kamera-Komponenten – mehr Wert auf die Nutzererfahrung als auf die Vorteile auf Seiten der System-Entwicklung gelegt wurde, muss gesagt werden, dass sich eine solche Fokussierung auf die Nutzererfahrung bezogen auf die hiesige Auswahl der Display-Technologie nicht umsetzen ließ: Die Verwendung eines Optical See-Through Displays im Zusammenhang mit einem glaubhaften Eindruck der virtuellen Überlagerungen wäre zwar wünschenswert, würde jedoch eine hinreichend komplexe Registrierung erforderlich machen, welche den verfügbaren Rahmen mit dessen zusätzlichem Aufwand gesprengt hätte. Gerade die eigentlich beabsichtigte Konzentration auf die Evaluation neuer Interaktionsformen hätte dadurch gelitten.

Die verfügbare Hardware – bezogen auf die Komponente des Head-Mounted Displays – bestand ferner aus einer Video-Brille aus dem Jahr 1998 mit der Bezeichnung „Sony Glasstron PLM-A55“ (siehe Abbildung 4.3). Diese Brille zeigt eingehende Bildsignale auf zwei Displays an (eins für jedes Auge) und stellt dabei zwar prinzipiell ein Optical See-Through Display dar, welches mit halbdurchlässigen Spiegeln arbeitet, bietet jedoch auch einen Video-Modus, sobald der sogenannte *Shutter*, also die Verschlussvorrichtung hinter den Spiegeln, geschlossen wird. Die Brille verfügt weiterhin über zwei in der Diagonalen 0,7 Zoll große LCD-Bildschirme mit jeweils 800 * 225 Bildpunkten und Öffnungswinkeln von horizontalen 30° respektive vertikalen 22,5° (vgl. [Son98]).



Abbildung 4.3: Die Videobrille Sony Glasstron PLM-A55

Nach obiger Entscheidung wurde diese Brille somit im Video-Modus betrieben und zur Darstellung des mit virtuellen Inhalten überlagerten Kamerabildes genutzt, welches vom eingesetzten Smartphone ausgegeben wurde.

4.2 Konstruktion und Aufbau

Zur Konstruktion des prototypischen Systems wird, neben der bereits vorgestellten Video-Brille „Sony Glasstron PLM-A55“, ein Smartphone benötigt. Für die Verwendung innerhalb des Systems am geeignetsten erschien ein Gerät, welches relativ aktuelle Hardware verbaut hat und die Ausgabe des eigenen Bildschirminhalts an ein externes Display nativ unterstützt. Zusätzlich muss die Verfügbarkeit des Geräts über den Zeitraum der Arbeit gewährleistet sein.

Die Wahl fiel daher auf das seit kurzem im Handel erhältliche Smartphone „Nexus 4“, welches in Zusammenarbeit der Firmen Google und LG Electronics gefertigt wird. Dieses Smartphone wird mit der bis dato aktuellsten Version des Android OS (Version 4.2) ausgeliefert und verfügt über 2 Gigabyte RAM und einen Vierkern-Prozessor, welcher mit 1,5 Ghz taktet. Dies sollte genug Ressourcen auch für algorithmisch komplexe Anwendungen bieten. Der Bildschirm des Smartphones besitzt eine Diagonale von 4,7 Zoll und eine Auflösung von 1280 * 768 Bildpunkten. Die Kamera löst im Falle der kontinuierlichen Bilderfassung überdies mit 1920 * 1080 Bildpunkten auf und besitzt dabei eine Bildrate von 30 fps (Bilder pro Sekunde). Die angesprochene native Unterstützung der Übertragung des Bildschirminhalts ist im Falle des „Nexus 4“ ebenfalls gegeben – die verwendete Technologie nennt sich „Slimport“ und setzt ein entsprechendes Adapterkabel voraus. Insgesamt eignet sich dieses Smartphone somit gut für Entwicklung der angestrebten Anwendung.

Im Folgenden wird zunächst der Aufbau des prototypischen Systems unter Verwendung der beschriebenen Hardware-Komponenten beleuchtet. Über die elektronische Hardware hinaus ist zur Realisierung des Anwendungskonzeptes weiterhin eine Möglichkeit zur Befestigung eines flächigen Papier-Markers am Unterarm des Benutzers notwendig. Auch dafür wurde eine entsprechende Konstruktion geschaffen, welche im Anschluss vorgestellt wird. Daraufhin wird das sich ergebende Gesamtsystem schließlich kurz in Bildern veranschaulicht.

4.2.1 Helm mit Smartphone-Halterung

Wie in Unterabschnitt 'Komponentendiskussion' bereits gefordert, soll das Smartphone so am Kopf des Benutzers des Systems befestigt werden, dass das Sichtfeld der integrierten Kamera des Smartphones ungefähr dem des Benutzers entspricht. Um dieser Anforderung gerecht zu werden, wurde ein Fahrradhelm mit einer eigens entwickelten Smartphone-Halterung ausgestattet; der finale Helm ist in Abbildung 4.4 abgebildet.



Abbildung 4.4: Fahrradhelm mit Smartphone-Halterung

Das Smartphone wird dabei in die dafür vorgesehene flexible Schutzhülle – in diesem Fall ausschließlich passend für das verwendete Smartphone „Nexus 4“ – gedrückt, welche über mehrere Winkeleisen am Helm befestigt ist. Darüber hinaus ist es möglich, die Ausrichtung des Smartphones zu ändern, um das System an den jeweiligen Träger und dessen Kopfgröße anpassen zu können. Dabei wurde darauf geachtet, dass der Helm inklusive der Befestigung des Smartphones – nach gewünschter Ausrichtung und anschließender Arretierung – verwindungssteif ist. Dies bedeutet, dass die einzelnen Teile der Konstruktion möglichst wenig anfällig sind gegenüber

eventuell heftigen Kopfbewegungen des Benutzers. Die Vorrichtung zur Fixierung des Smartphones und dessen Ausrichtung ist in Abbildung 4.5 genauer zu sehen.



Abbildung 4.5: Smartphone-Halterung im Detail

4.2.2 Bildübertragung Smartphone zu Videobrille

Für die Übertragung des Bildschirminhalts des Smartphones an die Videobrille sind mehrere Adapter bzw. Konverter notwendig. Dies liegt an den entsprechenden Aus- bzw. -Eingabe-Schnittstellen der konkreten Komponenten:

- Das Smartphone „Nexus 4“ gibt das Bild mittels „Slimport“-Adapter als digitales HDMI-Signal aus.
- Die Videobrille „Sony Glasstron PLM-A55“ erwartet das Bild als analoges FBAS- bzw. Composite-Signal.

Neben dem erwähnten „Slimport“-Adapter ist also die Umwandlung eines digitalen in ein analoges Bildsignal erforderlich, wobei letzteres nur durch eine aktive Konvertierung des Signals erfolgen kann. Dazu wurde der in Abbildung 4.6 abgebildete Konverter eingesetzt.

4.2.3 Marker-Befestigung am Unterarm

Zur Befestigung eines Papier-Markers am Unterarm des Benutzers wurde folgende Idee verfolgt: Es sollte eine Art Kleidungsstück gefunden werden, welches den Unterarm umschließt und sich zum An- und Ausziehen



Abbildung 4.6: Ligawo HDMI zu Composite/S-Video AV-Konverter

über die Hand des Benutzers streifen lässt, sodass der Marker auf flexible Art und Weise am Unterarm angebracht und wieder entfernt werden kann. Solche Kleidungsstück stellen Schweißbänder dar. Durch die Verwendung eines solchen, hinreichend eng anliegenden Schweißbandes kann zudem erreicht werden, dass ein – daran ausreichend gut befestigter – Marker nicht verrutscht, sobald der Benutzer seinen Arm bewegt. Weiterhin sollte das Papier des Markers in sich möglichst verwindungssteif sein, also sich nicht biegen, sobald es z.B. der geringen Kraft eines Luftzuges ausgesetzt wird.



Abbildung 4.7: Konstruktion zur Befestigung eines Papier-Markers am Unterarm

Somit wurde ein passendes Schweißband herangezogen und der Papier-Marker zunächst auf ein entsprechend geformtes Stück relativ dicken Kartons geklebt. Anschließend wurde das Stück Karton mittels mehrerer Heftklammern mit dem Stoff des Schweißbandes verbunden. Das Resultat ist in Abbildung 4.7 zu sehen und stellt eine erstaunlich flexible und gleichzeitig verwindungssteife Möglichkeit zur Befestigung des Markers am Unterarm des Benutzers dar.

4.2.4 Gesamtsystem

In Abbildung 4.8 sowie Abbildung 4.9 ist das prototypische Gesamtsystem illustriert, um einen Eindruck der Handhabung dessen zu vermitteln.



Abbildung 4.8: Prototypisches Gesamtsystem – frontal



Abbildung 4.9: Prototypisches Gesamtsystem – seitlich

5 Implementierung

Im Folgenden werden die zentralen Aspekte zur Realisierung der Augmented Reality-Anwendung erläutert. Zunächst werden die genutzten Bibliotheken und erste Vorbereitungen vorgestellt. Daraufhin wird ein grober Systemüberblick gegeben, in dessen Zuge der schematische Ablauf der finalen Applikation übersichtlich präsentiert wird. Anschließend wird dargestellt, wie die geforderte Gestenerkennung zur Interaktion mit der Anwendung umgesetzt wurde. Dabei werden zudem auftretende Probleme und Herausforderungen beleuchtet und eventuelle Lösungsansätze besprochen.

5.1 Verwendete Bibliotheken

Im Rahmen der Implementationsarbeiten für die konzeptionierte Augmented Reality-Anwendung wurden bestimmte Softwarebibliotheken und APIs verwendet. Im Folgenden werden diese einzeln vorgestellt.

5.1.1 Android OS und OpenGL ES

Zunächst ist zu sagen, dass die Applikation, wie bereits erwähnt, auf Basis von Googles mobilem Betriebssystem Android OS betrieben wird. Dieses bietet unter anderem einfachen Zugriff auf die Bilddaten der Smartphone-Kamera und die Möglichkeit zu 3D-Rendering unter Verwendung von „OpenGL ES“ (Abkürzung für: „Open Graphics Library for Embedded Systems“), einer plattformunabhängigen und recht verbreiteten API zur Entwicklung von 3D-Computergrafik auf eingebetteten bzw. mobilen Geräten. Die hier zur Darstellung der virtuellen Inhalte genutzte Version von „OpenGL ES“ ist Version 2.0. Darüber hinaus wird zur Nutzung der entwickelten Anwendung ein Smartphone mit Android OS in der Version 2.2 als Minimum vorausgesetzt. Dadurch wird sichergestellt, dass die benötigten Bibliotheken auf dem konkreten Gerät zur Verfügung stehen, wodurch die Funktionalität der Applikation gewährleistet werden kann.

5.1.2 Vuforia SDK

Ferner galt es, das Tracking flächiger Papier-Marker zu implementieren. Dies ist einerseits laut Anwendungskonzept (vgl. Unterabschnitt 'Anwendung') nötig, um die virtuellen Inhalte – in diesem Fall eine zweidimensionale Übersichtskarte – perspektivisch korrekt an der Stelle des Unterarms des Benutzers im aufgenommenen Bild einzublenden. Andererseits

soll, wie im Interaktionskonzept (vgl. Unterabschnitt 'Interaktion') erläutert, etwa durch die Kombination des Trackings mehrerer flächiger Marker eine Interaktion mit dem System mittels der ebenda beschriebenen Finger-Gesten ermöglicht werden.

An dieser Stelle wurde das „Qualcomm Vuforia SDK“ (ehemals „Qualcomm AR SDK“, künftig: Vuforia SDK) eingesetzt, eine von Qualcomm entwickelte, unter [Qua13b] kostenlos vertriebene und frei nutzbare Softwarebibliothek zur Realisierung von Augmented Reality-Applikationen auf mobilen Geräten. Das Vuforia SDK unterstützt dabei sowohl Android OS-basierte Geräte als auch Geräte der Firma Apple, welche dessen proprietäres Betriebssystem „iOS“ nutzen.

Ferner nutzt das Vuforia SDK Technologien der Bilderkennung, um zur Laufzeit bekannte planare Bilder oder Muster in dem von der Kamera kommenden Video-Stream zu erkennen und deren Position und Orientierung zu tracken. Die verwendbaren flächigen Marker-Typen untergliedern sich in:

- *Frame Markers*, welche über einen Rand, bestehend aus einer Kombination aus schwarzen und weißen Rechtecken, verfügen müssen, und
- *Image Targets*, also einfache Bilder, die grundsätzlich keine weiteren Kriterien erfüllen müssen.

Ein Beispiel eines Frame Markers zeigt Abbildung 5.1. Wie man sieht, kann der Bereich innerhalb des schwarz-weiß kodierten Randes beliebige Farben und Formen enthalten. Um nun mehrere verschiedene Frame Marker gleichzeitig in einer Applikation verwenden und eindeutig voneinander unterscheiden zu können, enthält das SDK ein Menge von 512 Bildern mit je verschiedenen Randkodierungen, welche man mit Inhalt beliebig füllen und ausgedruckt zusammen mit einer laufenden Applikation verwenden kann. Das Vuforia SDK erkennt und unterscheidet diese 512 Frame Marker zur Laufzeit und weist jedem erkannten Marker seine zugehörige Identifikationsnummer (von 0 bis 511) zu, die dann von der eigentlichen Applikation weiterverwendet werden kann.

In Abbildung 5.2 ist indes ein in dem Vuforia SDK bereits enthaltenes Beispiel eines Image Targets abgebildet. Wie man erkennen kann, können diese Marker beliebige Bildinhalte besitzen und ebenfalls farbig sein. Weitere harte Kriterien existieren nicht. Allerdings nimmt der Eignungsgrad des Bildes als zu trackendes Image Target zu, je markanter und kontrastreicher eventuell existente Strukturen des Bildes sind und je weniger sich Teile dieser Struktur im Bild wiederholen. Insofern lässt sich das abgebildete Beispiel eines Image Targets relativ gut tracken, während ein Bild mit weitgehend homogenen Farbflächen ebenso wenig geeignet scheint wie

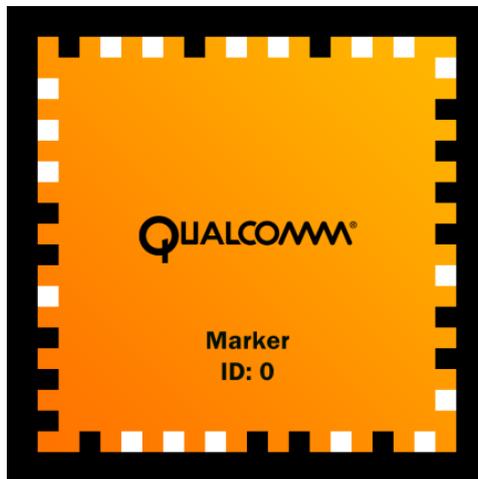


Abbildung 5.1: Von Vuforia SDK unterstützter Frame Marker

ein Bild aus sich vielfach wiederholenden Strukturen (z.B. Schachbrettmuster).



Abbildung 5.2: Von Vuforia SDK unterstütztes Image Target

Pro eingehendes Bild wird nun also das Trackingsystem des Vuforia SDKs durchlaufen und im Bild nach dem System bekannten Markern gesucht. Für jeden erkannten Marker wird sodann die jeweilige Pose berechnet und an die umgebende Applikation weitergegeben, welche daraufhin in der Lage ist, beliebige virtuelle Objekte so zu rendern, dass diese das eingehende Kamerabild an der Stelle der Marker – entsprechend perspektivisch verzerrt – überlagern. Somit entsteht der Eindruck, virtuelle Objekte befänden sich in der realen Umgebung. Somit ist Augmented Reality möglich.

Über das Markertracking hinaus bietet das Vuforia SDK ein weitere nütz-

liche Funktion: Sogenannte *Virtual Buttons*. Dies meint vorher definierte, rechteckige Bereiche innerhalb von Image Targets (Frame Marker werden an dieser Stelle nicht unterstützt), welche als Buttons dienen und zur aktiven Interaktion des Benutzers mit dem System verwendet werden können. In Abbildung 5.3 ist ein mit solchen Buttons auch visuell ausgestattetes Image Target zu sehen. Die visuelle Markierung dient an dieser Stelle einzig zur Orientierung des Benutzers; das System liest die Daten zur Angabe der Virtual Button-Positionen und -Größen hingegen über in bestimmter Weise aufgebaute XML-Dateien ein.

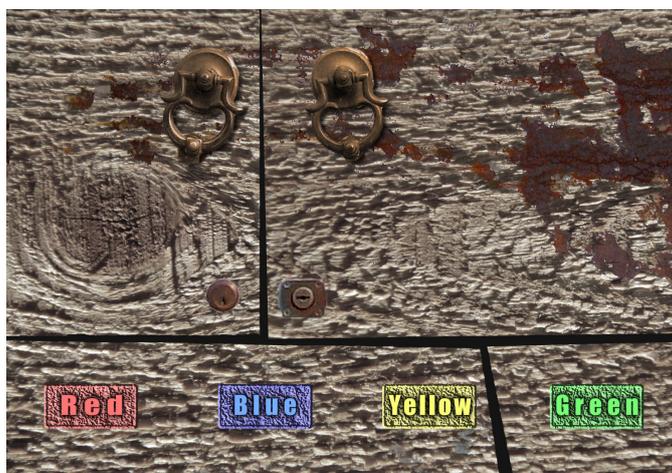


Abbildung 5.3: Image Target mit visuell markierten Virtual Buttons

Zur Laufzeit und sobald ein Image Target mit enthaltenen Virtual Buttons erkannt wurde, kann der Benutzer – idealtypisch mittels einem seiner Finger – einen solchen Virtual Button auf dem Papier-Marker zu „drücken“ versuchen. Dabei wird der definierte Bereich des Image Targets aus Kamerapersicht optimalerweise in der Art verdeckt, dass das Trackingsystem dies bemerkt und den veränderten Status des verdeckten Virtual Buttons entsprechend an das restliche System weitergeben kann. Abbildung 5.4 veranschaulicht das Prinzip; in diesem Beispiel wird die Farbe der Teekanne entsprechend verändert, sobald einer der vier definierten Buttons gedrückt wird.

Zusammenfassend lässt sich sagen, dass das Vuforia SDK initial eingesetzt wurde, um das Teilproblem des Markertrackings zu lösen. Im weiteren Verlauf stellte sich jedoch heraus, dass man durch das Tracking eines einzelnen Image Targets gleichzeitig in der Lage ist, die Grundlagen für die geforderte Erkennung von Finger-Gesten zu schaffen. Genauer dazu in den folgenden Abschnitten.

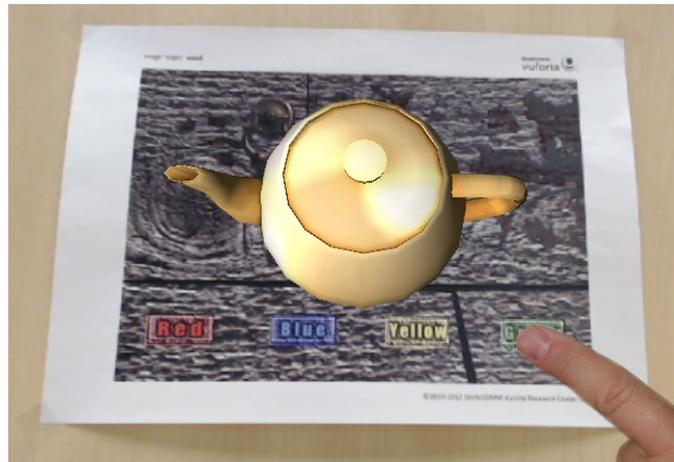


Abbildung 5.4: Beispielhafte Verwendung von Virtual Buttons zur Interaktion des Benutzers mit dem System

5.1.3 Google Maps API

Zunächst war überdies angedacht, den Kartenausschnitt, welcher mittels der Applikation an der Stelle des Unterarms des Benutzers angezeigt werden soll, mittels der „Google Maps API“ (siehe [Goo13]) dynamisch aus dem Internet nachzuladen. Somit könnte der Benutzer durch die umfangreichen weltweiten Kartendaten von „Google Maps“ navigieren und wäre dabei nicht auf einen statischen Kartenausschnitt beschränkt.

Diese Idee war jedoch nicht in der angedachten Art und Weise umsetzbar. Ohne an dieser Stelle technisch zu detailliert zu werden, wird der Grund nachfolgend kurz erläutert: Die „Google Maps API“ bietet native Unterstützung für Android-Applikationen. Es existiert ein eigens dafür verwendbares graphisches Element namens *MapView* – eine Spezialisierung einer Standard-Komponente des Android-User-Interfaces, der *View* –, welches es erlaubt, das Kartenmaterial von „Google Maps“ in ausreichender Geschwindigkeit über das Internet anzufordern und den Benutzer intuitiv durchforsten zu lassen. Jedoch muss in der im Rahmen dieser Arbeit zu entwickelnden Applikation der Kartenausschnitt als virtuelles 3D-Objekt mittels „OpenGL ES“ gerendert werden, auch, wenn es letztlich nur ein flächiges Rechteck ist. Dies bedeutet, dass der Inhalt einer im Hintergrund angelegten *MapView* in jedem gezeichneten Frame der Applikation in einen Zwischenspeicher abgelegt werden müsste, um dann beim Rendern mittels „OpenGL ES“ darauf zugreifen zu können.

Obwohl eine solche Vorgehensweise laut Nutzungsbestimmungen der „Google Maps API“ nicht erlaubt ist, wurde der Ansatz testweise implementiert. Allerdings wurde schnell die Erkenntnis gewonnen, dass die

zusätzliche Latenz der Zwischenspeicherung und des erneuten Zugriffs auf diesen Speicher zu einer eklatanten Verzögerung des Zeichens jedes einzelnen Frames führt und die Darstellung des Kartenausschnitts als 3D-Objekt somit der eigentlich aktuellen Ansicht hinterher hängt. Da dieses Phänomen nicht mit einer intuitiven und erwartungskonformen Anwendung vereinbar ist, wurde letztendlich doch ein statischer Kartenausschnitt erstellt, welcher als Bild beim Start der Applikation geladen wird. Dieser Kartenausschnitt wurde letztendlich aufgrund verschiedener Vorteile aus dem Kartenmaterial von Microsofts „Bing Maps“ (siehe [Mic13]) manuell zusammengefügt.

5.2 Vorbereitungen

Um das Vuforia SDK zum Tracking von Image Targets nutzen zu können, muss zu dem Bild, welches sich auf Papier gedruckt als Marker tracken lassen soll, ein maschinenlesbarer Datensatz existieren, den das Trackingsystem zu Laufzeitbeginn einlesen und als Grundlage des Trackings verwenden kann.

Es werden zwar bereits Beispiele für Image Targets samt ihrer zugehörigen Datensätze mit dem SDK mitgeliefert; sollen jedoch eigens ausgewählte Bilder als Image Targets verwendet werden, muss ein dafür verfügbarer Webservice der Firma Qualcomm genutzt werden. Dieser Webservice gibt nach Erstellung des, dem zuvor hochgeladenen Bild zugehörigen, Datensatzes – welcher sodann zum Download angeboten wird – zudem eine Bewertung der Eignung des Bildes als Image Target aus.

Abbildung 5.5 zeigt eine grafische Übersicht über die Komponenten einer mobilen Applikation, die das Vuforia SDK verwendet, und veranschaulicht die Zuständigkeiten innerhalb einer solchen Applikation und die Erstellung des maschinenlesbaren Target-Datensatzes über den Webservice. Die Grafik wurde von [Qua13a] bezogen und dahingehend verändert, dass die im Rahmen dieser Arbeit nicht genutzten Komponenten des SDKs nun ausgegraut erscheinen. Dies beinhaltet unter anderem einen *Cloud-Service*, welcher es ermöglichen soll, zur Laufzeit der Anwendung mittels Internetzugang auf eine Datenbank von bis zu 1 Million Image Targets zuzugreifen und diese Menge an Targets simultan zu tracken (siehe ebd.).

Im Rahmen der vorliegenden Arbeit wurde nach einem möglichst gut für das Tracking geeigneten Bild gesucht. Während einer manuellen Suche im Internet inklusive dem Ausprobieren vieler Bilder als Image Target wurde ein Bild gefunden, welches sich nach eigener Erfahrung und zudem laut der Bewertung des erwähnten Webservices besser zum Tracking eignet als jegliche mitgelieferten Beispielbilder. Dieses Bild, zu sehen in Ab-

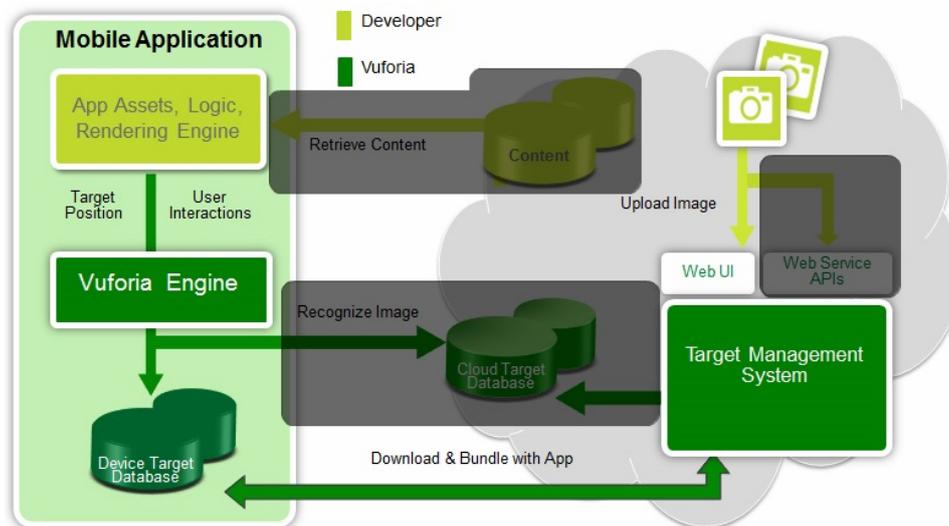


Abbildung 5.5: Übersicht über die Komponenten einer mit dem Vuforia SDK entwickelten mobilen Applikation (verfügbar unter [Qua13a]), ausgegraute Bereiche werden nicht genutzt

bildung 5.6, wurde sodann durchgehend zur Entwicklung und Evaluation des Systems eingesetzt.

Sodann wurde das Vuforia SDK in eine neue Android-Applikation eingebunden und bald erste Versuche des Trackings und der Überblendung des Image Targets mit virtuellen Objekten durchgeführt. Abbildung 5.7 zeigt ein Resultat dieser ersten Gehversuche. Zu sehen ist die aktuelle, vom Smartphone-Betriebssystem gelieferte Uhrzeit als selbst erstelltes, dreidimensionales Objekt, welches entsprechend der Pose des getrackten Image Targets perspektivisch korrekt an dessen Stelle gerendert und sekundenweise aktualisiert wird, sodass es einer Digitaluhr-Animation gleicht.

5.3 Systemüberblick

Wie eingangs erwähnt, wird an dieser Stelle der schematische Ablauf der finalen Applikation kurz veranschaulicht.

Wie in Abbildung 5.8 verdeutlicht, ist die implementierte Applikation so aufgebaut, dass sich im Kern folgender Ablauf pro Render-Durchlauf (auf englische auch *rendercycle* genannt) ergibt:

1. Zunächst wird das aktuellste Kamerabild von der entsprechenden Software-Schnittstelle abgefragt.



Abbildung 5.6: Finales Bild zur Verwendung als Image Target (verfügbar unter [Tra])

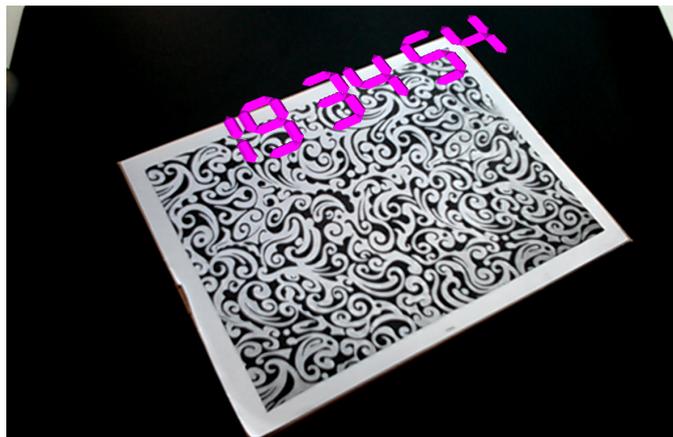


Abbildung 5.7: Erste Gehversuche der Implementation: Animierte Digitaluhr als 3D-Objekt überlagert Image Target

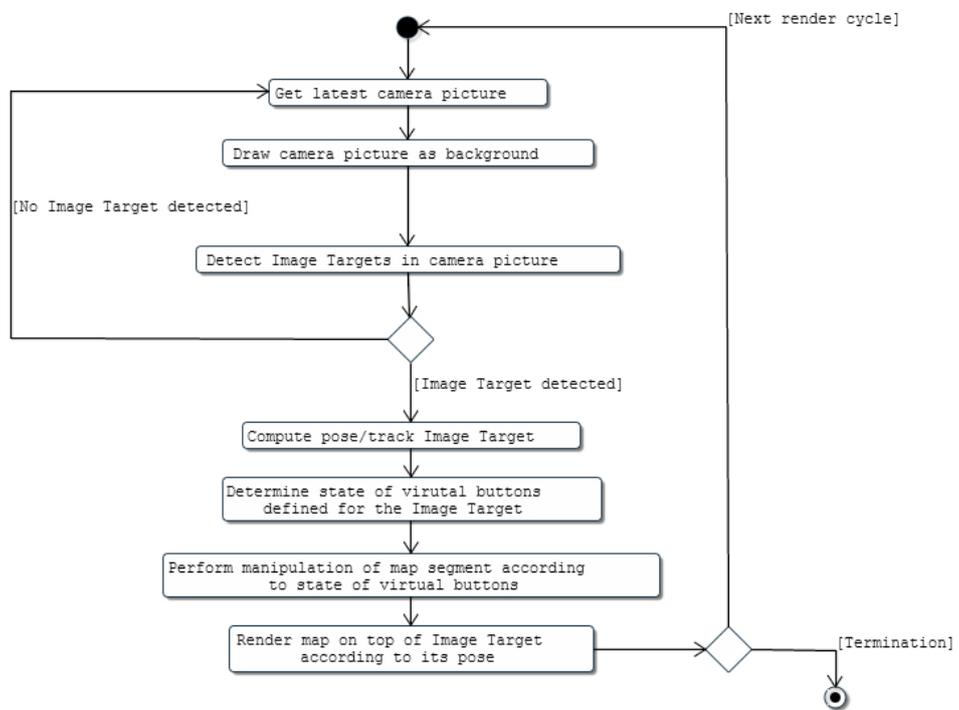


Abbildung 5.8: Aktivitätsdiagramm eines Render-Durchlaufs der Applikation

2. Dieses wird ohne Umwege – als eine Art Hintergrund – angezeigt.
3. Daraufhin erfolgt die Detektion des zur Verwendung definierten Image Targets im Kamerabild. Alle nun folgenden Schritte erfolgen nur, falls das Image Target gefunden wurde.
4. Im Anschluss wird die Pose eines neu detektierten Image Targets berechnet bzw. ein im vorherigen Render-Durchlauf schon detektiertes Image Target bzgl. seiner Pose verfolgt (getrackt).
5. Danach werden die für das Image Target definierten Virtual Buttons einzeln auf Verdeckung untersucht.
6. Anhand des Status der einzelnen Virtual Buttons wird eine eventuelle Manipulation des Kartenausschnitts vorgenommen (hier mittels Gestenerkennung, mehr dazu später).
7. Schließlich wird der Kartenausschnitt perspektivisch korrekt über das Image Target gerendert.

Die Schritte 3 bis 5 sind dabei Teil des Trackingsystems des Vuforia SDKs und werden von diesem erledigt. Allerdings arbeitet das Trackingsystem in diesem Fall als in sich geschlossener und unzugänglicher Software-Baustein – in der Art einer sogenannten *Blackbox* – mit einer wohldefinierten Schnittstelle nach außen. Das heißt, es besteht leider keine Möglichkeit, die verwendeten Methoden und Techniken des Trackingsystems und seiner Bestandteile zu erfahren. Über die genauen Ausprägungen der Schritte 3 bis 5, über die groben Tätigkeitsbeschreibungen der obigen Liste hinaus, lässt sich somit nur spekulieren.

Schritt 6 hingegen wird im folgenden Unterabschnitt 'Gestenerkennung' genauer erläutert. Abbildung 5.9 zeigt die laufende Applikation noch ohne eine Möglichkeit zur Interaktion bzw. Manipulation des Kartenausschnitts, jedoch mit dem in Schritt 7 erwähnten Rendering.

5.4 Gestenerkennung

Um nun die im Interaktionskonzept (vgl. Unterabschnitt 'Interaktion') erläuterten Finger-Gesten mittels Tracking eines Image Targets zu ermöglichen, kam die Idee auf, die vom Vuforia SDK bereitgestellte Funktionalität der Virtual Buttons einzusetzen.

Der weitere Gedanke war, auf dem verwendeten Image Targets eine rechteckige Gitternetzstruktur von hinreichend vielen Virtual Buttons zu definieren. Würde die Verdeckung jedes einzelnen Virtual Buttons dieses Rasters durch das Vuforia robust erkannt werden, könnte man auf Basis dessen



Abbildung 5.9: Kartenausschnitt überlagert Image Target, noch ohne Möglichkeiten zur Manipulation des Kartenausschnitts

eine Gestenerkennung ähnlich der von kapazitiven Multitouch-Displays, wie sie in vielen Smartphones eingesetzt werden, realisieren.

Mit der Anzahl der Virtual Buttons im Raster lässt sich dabei die visuelle Abtastung des Papier-Markers variieren und mit zunehmender Anzahl lässt sich solch ein Raster mit einem rechteckigen Pixelraster vergleichen, das wiederum als Bild – in diesem Fall jedoch mit dem *boolschen* Wertebereich [*false, true*]. – interpretiert werden kann. Auf Bildern kann man bekanntlich Techniken der Bildverarbeitung anwenden und damit Mustererkennung (engl.: *pattern matching*) betreiben. Somit war die Idee geboren, Finger-Gesten, wie sie bei heutigen Smartphones üblich sind, mittels optischem Markertracking, auf Basis von virtuellen Buttons auf einem Papier-Marker und Mustererkennung, im Raum zu ermöglichen.

5.4.1 Vorbereitungen

Nun wurden zunächst die per XML-Datei definierbaren Virtual Buttons in Rastern verschiedener Ausmaße (bezogen auf die Anzahl der Virtual Buttons) getestet und immer wieder geschaut, wie robust die Erkennung der Verdeckung der einzelnen Virtual Buttons seitens des Vuforia SDKs ist. Somit wurde die – auf Erfahrungswerten basierend – optimalen Ausmaße des Rasters pragmatisch ermittelt. Da es sinnvoll erscheint, die Virtual Buttons als quadratische Flächen zu definieren und das verwendete Image Target ein Seitenverhältnis von 4 zu 3 (Breite zu Höhe) besitzt, musste auch die Anzahl der Virtual Buttons in der Horizontalen gegenüber der in der Vertikalen im Verhältnis 4 zu 3 stehen. Die optimalen Ausmaße des Rasters betragen somit $12 * 9$; genauer gesagt besteht das Raster aus 12 Zeilen und

9 Spalten von Virtual Buttons. Abbildung 5.10 zeigt das Raster als virtuelle Überlagerung des Markers.



Abbildung 5.10: Image Target mit visuellen Überlagerungen entsprechend dem definierten Raster aus Virtual Buttons und deren Status, verdeckte Virtual Buttons werden grün angezeigt, nicht verdeckte rot

Um bei dem darauf folgenden Schritt der Implementierung der Gestenerkennung auf Basis des Rasters die Entwicklung zu unterstützen und Fehlerdiagnose einfach betreiben zu können, wurde eine zweite Android-Applikation entwickelt. Mit dieser ist es möglich, den jeweils aktuellen Stand der Gestenerkennung mittels einfacher Multitouch-Eingaben über ein Smartphone-Display zu testen. Dazu wurde das Raster aus Virtual Buttons durch ein Raster von standardmäßigen Android-Checkboxes simuliert. Der Code zur Gestenerkennung ist derselbe wie der der Augmented Reality-Applikation selbst, ist entsprechend abgekapselt und arbeitet unabhängig von der Eingabe; daher kann die beschriebene Hilfsapplikation jederzeit durch die eigentliche Anwendung ersetzt werden. Abbildung 5.11 zeigt die Arbeit mit der Hilfsapplikation.

Die Hilfsapplikation kann den aktuellen Systemzustand der Gestenerkennung visuell erfahrbar machen; z.B. werden Buttons, welche von dem Teil des Algorithmus, der für die Regionenextraktion zuständig ist, zur selben Region gezählt werden, auch in derselben Farbe angezeigt. Die Hilfsapplikation bietet darüber hinaus die Möglichkeit, bestimmte Finger-Eingaben – also die Aktivierung bestimmter Buttons in bestimmter zeitlicher Abfolge – beliebig oft wiederzugeben und die Geschwindigkeit der Wiedergabe dabei so zu verringern, dass die visuellen Diagnose-Ausgaben dabei ge-

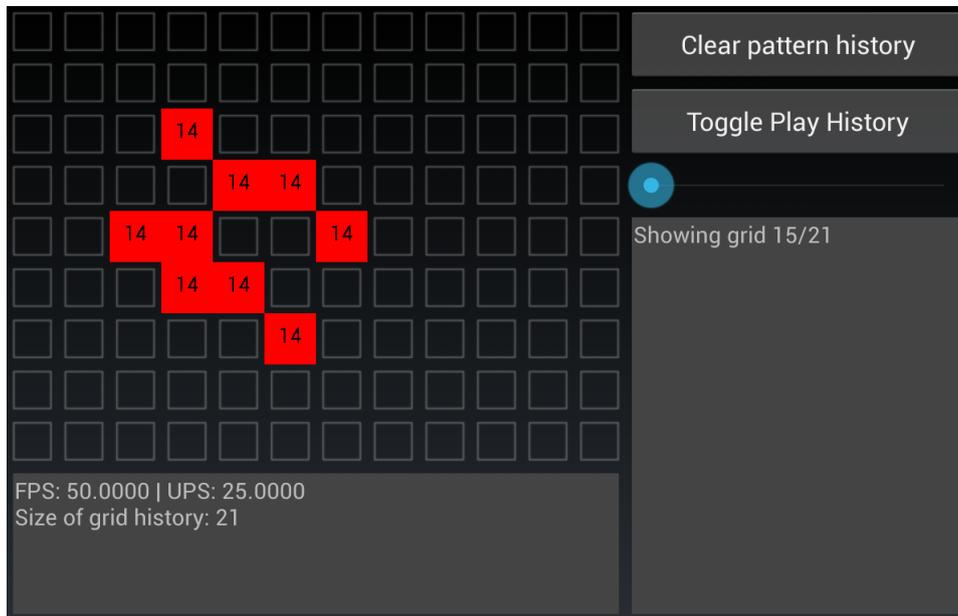


Abbildung 5.11: Eigens zur Unterstützung der Entwicklung entstandene Hilfsapplikation

nau beobachtet werden können, um damit wiederum eine möglichst exakte Vorstellung des inneren Systemzustands der Gestenerkennung zu erlangen.

5.4.2 Algorithmus

Der sodann entwickelte Algorithmus zur Erkennung der geforderten Finger-Gesten auf Basis des Virtual Button-Rasters verfolgt die grundlegenden Ideen aus einem Patent von Hotelling et al. zur Gestenerkennung auf Touch-sensitiven Eingabegeräten (siehe [HSH⁺06]).

Der entwickelte Algorithmus arbeitet auf einem eingehenden Raster an booleschen Werten. D.h. pro Durchlauf der kompletten Gestenerkennung – welche 25 mal die Sekunde auf den jeweils aktuellen bzw. seit dem letzten Durchlauf gesammelten Daten des Trackingsystems und der Virtual Buttons aufgerufen wird – wird der Algorithmus erneut ausgeführt. Als Ausgabe des Algorithmus dienen letztlich Aufrufe von Funktionen zur Manipulation des gezeigten Kartenausschnitts. Falls also im aktuellen Durchlauf eine Geste zum Verschieben des Kartenausschnitts inklusive der entsprechenden Parameter erkannt wurde, wird dies an die entsprechende Stelle weitergeleitet.

Die wesentlichen Datentypen, welche innerhalb des Algorithmus von

Nöten sind, sind *Regionen* und *Gestenzüge*. Eine Region definiert einen zusammengehörigen Bereich aus aktiven Punkten des Rasters zu einer bestimmten Zeit und soll im Resultat die Verdeckung mehrerer Virtual Buttons durch einen Finger ermitteln und die Informationen dieses Fingers bündeln – zu einem bestimmten Zeitpunkt. Sie besitzt daher also eine Menge von zweidimensionalen Koordinatenpunkten innerhalb des Rasters, den Schwerpunkt dieser Menge von Koordinatenpunkten und eine ID. Ein Gestenzug hingegen stellt die Bewegung eines einzelnen aktiven Rasterbereichs über die Zeit dar und enthält dem entsprechend eine geordnete Liste von Regionen-Objekten, die alle dieselbe ID besitzen. Ein solcher Gestenzug entspricht im Resultat somit der Bewegung eines Fingers.

Der Algorithmus selbst führt innerhalb jedes Durchlaufs eine Liste der derzeit gefundenen Regionen, überdies hat er zu jeder Zeit Zugriff auf alle bisherigen Listen von gefundenen Regionen und auf die Liste aller bisherigen Gestenzüge. Er lässt sich weiterhin grob in den folgenden Ablauf gliedern:

1. Zunächst werden aus dem Raster von teilweise aktivierten Buttons zusammengehörige Regionen ermittelt, extrahiert und mit einer bezüglich dieses Durchlaufs eindeutigen ID versehen. Zudem wird der Schwerpunkt einer jeden so gefundenen Region berechnet; anschließend werden diese Regionen in der Liste der aktuellen Regionen zusammengefasst.
2. Nun wird diese Liste der gerade detektierten Regionen mit der Regionenliste des vergangenen Durchlaufs verglichen und Korrespondenzen gesucht. Es findet also ein Tracking der Regionen über die Zeit statt. Wird eine Regionen-Korrespondenz gefunden, wird die ID der neueren Region auf die der älteren gesetzt, um die Zusammengehörigkeit der Regionen für später abrufbar zu machen.
3. Daraufhin gilt es, die Gestenzüge zu aktualisieren bzw. zu erstellen: Für jede Region, zu der keine Korrespondenz einer vorherigen Region gefunden wird, wird ein neuer Gestenzug angelegt. Für alle anderen Regionen mit Vorgängern wird der zugehörige Gestenzug anhand der ID gefunden und letzterer durch die neue Region erweitert.
4. Danach werden alle Gestenzüge, welche aktuelle Regionen enthalten – und somit als aktiv anzusehen sind – gezählt. Diese Anzahl repräsentiert nun die Anzahl der Finger, welche aktuell den Marker verdecken.
5. Entsprechend dieser Anzahl wird nun entweder die Berechnung der zweidimensionalen Verschiebung (im Falle nur eines Fingers; Pan-Geste) oder des gleichzeitigen Drehens und Skalierens (im Falle zweier Finger; Rotate- und Pinch-to-zoom-Gesten) des Kartenausschnitts

weiterverfolgt. Die Rotate- und Pinch-to-zoom-Gesten funktioniert somit nur zusammen. Wurde keiner oder wurden mehr als zwei Finger erkannt, wird nichts weiter unternommen.

Im Falle einer Pan-Geste (nur ein Gestenzug) werden die aktuelle Region des Gestenzuges sowie deren Vorgänger herangezogen: Es erfolgt eine Verschiebung um den Abstand der Schwerpunkt beider Regionen. Im Falle der Rotate-Geste werden ebenfalls aktuelle Regionen und deren Vorgänger genutzt – dieses Mal jedoch von beiden erkannten Gestenzügen: Zwischen den aktuellen Regionen beider Gestenzüge und zwischen den beiden Vorgänger-Regionen werden jeweils Vektoren gebildet und der Winkel zwischen diesen beiden Vektoren berechnet. Dieser dient sodann als Parameter der Drehung. Bei der Pinch-to-zoom-Geste werden dieselben Vektoren zur Berechnung herangezogen. Dieses Mal werden jedoch deren Längen bestimmt und das Verhältnis zwischen der Länge des neueren Vektors (im Zähler) und der des Vorgänger-Vektors (im Nenner) wird als Skalierungsfaktor verwendet.

An dieser Stelle soll Schritt 1 der obigen Aufzählung erneut beleuchtet werden. Die Regionenbildung erfolgt an dieser Stelle über eine rekursive Funktion, welche bei einem aktiven Koordinatenpunkt startet und nacheinander alle horizontalen, vertikalen und diagonalen Nachbarnpunkte nach deren Aktivitätsstatus abfragt. Jeder der aktiven Nachbarnpunkte wird zur aktuellen Region hinzugefügt, falls er es nicht schon einer Region angehört. Daraufhin wird die beschriebene rekursive Funktion für jeden der Nachbarnpunkte aufgerufen. Mittels weniger weiterer Abbruchbedingungen stellt dieses Prinzip eine simple Art der Regionenbildung dar.

5.4.3 Herausforderungen

Noch während der Entwicklung der Gestenerkennung zeichneten sich jedoch auch einige Probleme und Herausforderungen ab.

Zunächst fiel auf, dass die Rotation des Kartenausschnitts relativ ungenau zu steuern ist. Dadurch passiert es häufig, dass der Benutzer nach dem Ausführen einer der Zwei-Finger-Gesten (Rotation und/oder Skalierung) – er also entweder skalieren wollte und den Kartenausschnitt dabei versehentlich etwas gedreht hat, oder er den Kartenausschnitt bewusst gedreht hat – nicht mehr in der Lage ist, den Kartenausschnitt durch eine kurze Geste (d.h. mit vertretbarem Aufwand) wieder in die exakte, initiale Waagerechte (0° Rotation) zu bringen.

Der Grund für die ungenaue Rotation liegt in der relativ geringen Anzahl der Virtual Buttons und der damit verbundenen schlechten Abtastung

des zur Eingabe verwendeten Markers. Diese Unterabtastung verschlechtert nun zwar alle unterstützten Gesten in ihrer Genauigkeit, jedoch fällt sie im Falle der Rotation gefühlt noch stärker ins Gewicht. Zudem entstehen bei der Berechnung der Winkel mittels trigonometrischer Funktionen immer sehr unterschiedliche Fließkommawerte mit vielen Nachkommastellen, welche sich durch wiederholte Ausführung von Rotationsgesten schlecht reproduzieren lassen, sodass man selten wieder die exakte Ausgangsausrichtung des Kartenausschnitts erreicht.

Eine erste Lösung des beschriebenen Problems wurde dadurch angegangen, dass eine automatische Ausrichtung an bestimmte Winkelgrößen – z.B. alle Vielfachen von 45° – erfolgt, sobald der eigentliche Winkel in die Nähe einer solchen Ausrichtung gelangt. Damit war es nun wieder einfach möglich, einen einmal beliebig rotierten Kartenausschnitt wieder in seine initiale Ausrichtung von 0° zu bringen.

Daraufhin tauchte bald schon eine weitere Schwierigkeit auf: Es passiert häufig, dass man während der Benutzung der Applikation zusammen mit dem Handgelenk-Marker versehentliche Gesten auslöst, indem z.B. der Teile des Arms ohne Marker aus beliebigen Gründen zwischen die Smartphone-Kamera und den Handgelenk-Marker geraten und eine Verdeckung des Markers darstellen. In diesem Fall wird die Gestenerkennung sofort gestartet und meist eine große Region gefunden, die dann bei Bewegung eine Verschiebung des Kartenausschnitts auslöst.

Diesem Problem wurde daraufhin mit der Einführung sogenannter *Trigger* in die Applikation begegnet. Dies meint einen internen Countdown, der gestartet wird, sobald Teile des Markers von neuem verdeckt werden; Erst nach Ablauf dieses Countdowns startet dann die eigentliche Gestenerkennung und die Manipulation des Kartenausschnitts kann beginnen. Damit dieser Vorgang für den Benutzer transparent ist, geht man normalerweise den Weg und blendet einen Hinweis ein, der den aktuellen Countdown-Wert visualisiert. Somit kann der Benutzer abschätzen, wie lang er mit dem Ausführen von Gesten warten muss, bevor das System darauf reagiert. Dieser Punkt ist wichtig, da eine fehlende Reaktion des Systems auf eigentlich gültige Eingaben sonst sehr schnell dazu führen, dass der Benutzer frustriert ist und dem System eine schlechte Benutzbarkeit attestiert.

Somit wurden Trigger-Symbole eingeführt, die Anhand ihrer mit dem Countdown sinkenden Ausmaße eben diesen visualisieren. Das Trigger-Symbol wird also bei erkannter Verdeckung mit einer bestimmten Größe an der Stelle der entdeckten Verdeckung angezeigt und verkleinert sich dann kontinuierlich, bis es nach einer vorher festgelegten Zeit ganz verschwunden ist und der Benutzer mit der weiteren Interaktion beginnen kann.

Diese Trigger-Symbole können dabei natürlich ebenso visualisieren, welche der unterschiedlichen unterstützten Gesten ausgeführt wird, sobald das Symbol verschwunden ist. Im Falle dieser Applikation erscheint jedoch einzig die Unterscheidung anhand der Anzahl der zur Geste verwendeten Finger sinnvoll, da das System vor Beginn der Gestenerkennung naturgemäß nicht wissen kann, ob der Benutzer mit seinen beispielsweise gerade eingesetzten zwei Fingern nun rotieren oder skalieren möchte. Das erstellte Trigger-Symbol für Pan-Gesten (mittels einem Finger) ist in Abbildung 5.12, dasjenige für die Rotate- und Pinch-to-zoom-Gesten (mittels zweier Finger) in Abbildung 5.13 zu sehen. Als Wartezeit bis zur Gestenerkennung hat sich ein initialer Countdown-Wert von einer Sekunde als sinnvoll und pragmatisch erwiesen.

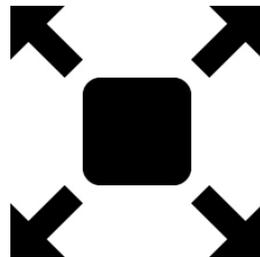


Abbildung 5.12: Trigger-Symbol für Pan-Gesten

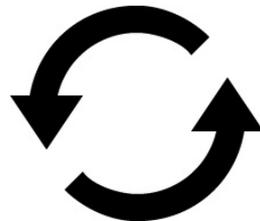


Abbildung 5.13: Trigger-Symbol für Rotate- und Pinch-to-zoom-Gesten

Abbildung 5.14 und Abbildung 5.15 zeigen die Trigger-Symbole – zusammen mit der finalen Applikation – in Aktion. Wie zu erkennen ist, wird der Trigger für die Zwei-Finger-Gesten Rotate und Pinch-to-zoom im Schwerpunkt der erkannten Verdeckungen, also in der Mitte beider Finger, angezeigt. Diese Positionierung stellte sich als die am sinnvollsten heraus.

Bezogen auf die beschriebenen Probleme wurde inmitten der Applikationsentwicklung zudem erneut darüber nachgedacht, einzelne Virtual Buttons außerhalb des Rasters z.B. für das sonst per Finger-Geste relativ ungenaue Rotieren einzuführen. Dazu müsste auch für diese einzelnen Buttons

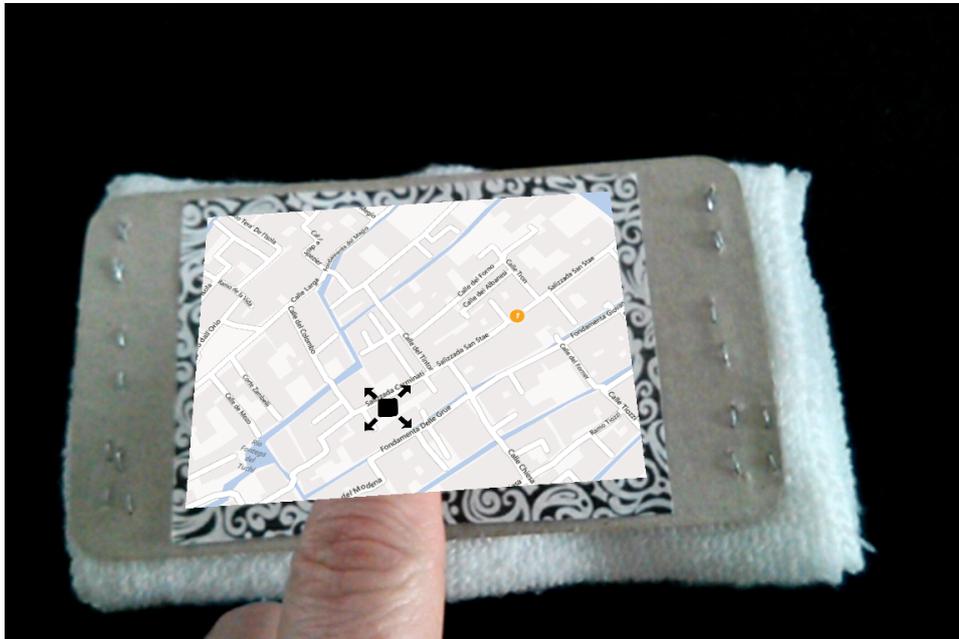


Abbildung 5.14: Auftauchendes Trigger-Symbol für Pan-Gesten bei Marker-Verdeckung durch einen Finger innerhalb der finalen Applikation

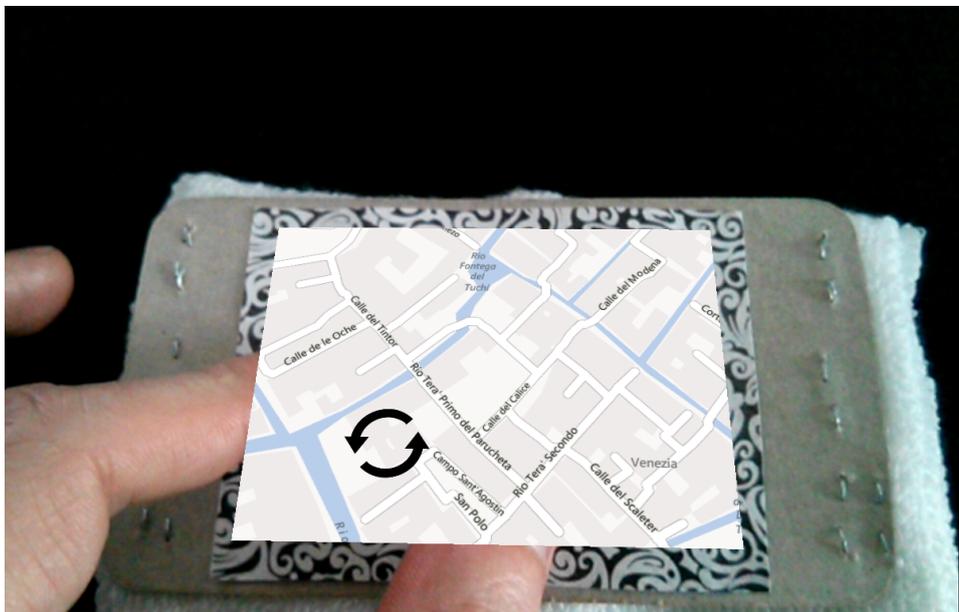


Abbildung 5.15: Auftauchendes Trigger-Symbol für Rotate- und Pinch-to-zoom-Gesten bei Marker-Verdeckung durch zwei Finger innerhalb der finalen Applikation

ein Countdown implementiert werden: Sobald einer dieser Buttons für eine bestimmte Zeit lang verdeckt ist, wird dies als in dem Moment einmaliges Drücken des Buttons akzeptiert. Die Hoffnung war zudem, die Gestenerkennung dadurch einfacher werden zu lassen; der Benutzer könnte sich bei der Verwendung zweier Finger auf eine Geste konzentrieren, da die andere nicht mehr als Geste, sondern mittels einzelner dazu ausgewiesener Virtual Buttons beispielsweise am seitlichen Rand des Rasters zur Gestenerkennung unterstützt wird.

Erste Umsetzungen dieser Idee verhalfen allerdings nicht zum erhofften Resultat. Im Gegenteil – ganz gleich, wo die zusätzlichen Buttons auf dem vorhandenen Handgelenk-Marker angebracht wurden, ihre Benutzung endete häufig in versehentlichen Verdeckungen des Rasters aus Virtual Buttons und damit wieder in ungewollten, aber vom System akzeptierten Gesten. Auch der umgekehrte Fall – das Drücken eines selbstständigen Buttons wurde teilweise unbeabsichtigt während der Ausführung von Gesten ausgelöst. Somit wurde diese Idee letztlich wieder verworfen und weiter an der reinen Gestenerkennung als einzige Möglichkeit der Manipulation des Kartenausschnitts festgehalten.

6 Evaluation

In diesem Abschnitt wird die Evaluation des prototypischen Systems erläutert. Die Basis zur Durchführung dessen bildet die Definition einer Zielsetzung, um die Evaluation zielorientiert gestalten zu können. Anschließend wird die Planung und Durchführung der Evaluation vorgestellt und letztlich eine Auswertung der Ergebnisse vorgenommen.

6.1 Zielsetzung

Die Durchführung und Auswertung der Evaluation des entstandenen prototypischen Systems soll Aufschluss darüber geben, inwiefern das im Rahmen dieser Arbeit entwickelte Konzept zur Interaktion mit ubiquitären User-Interfaces geeignet ist. Dabei sollen mittels eines persönlichen Fragebogens die Empfindungen der Probanden erfasst werden und somit unter anderem Rückschlüsse auf die Intuitivität der Interaktion und deren einzelne Bestandteile gezogen werden. Ein besonderes Augenmerk liegt zudem auf der Frage nach dem möglichen Potential des vorgestellten Prinzips zur Interaktion auf Basis zukünftiger, ausgereifter Technologien. Unter weiterer Zuhilfenahme eines praktischen Benutzertests soll daher sichergestellt werden, dass sich die Probanden intensiv mit dem System und dessen Möglichkeiten auseinandersetzen.

Anhand der resultierenden Ergebnisse sollen weiterhin potentielle Einsatzgebiete eines solchen Systems ermittelt werden – sowohl bezogen auf das konkrete, im Rahmen dieser Arbeit entstandene System als auch auf ein zukünftiges, in vielen Belangen optimales System dieser Art. Mithilfe eigens dafür angelegter Fragenblöcke sollen darüber hinaus Antworten auf folgenden Fragen gefunden werden: Welche Formen könnte die Interaktion mit allgegenwärtigen User-Interfaces in Zukunft annehmen? Inwieweit erscheinen aktuelle Interaktionsformen überhaupt sinnvoll? Welche Möglichkeiten der Verbesserung diesbezüglich existieren mittel- bis langfristig? Dabei sollen die Probanden zur kritischen Reflektion ihrer Gedankengänge angeregt werden.

6.2 Planung und Durchführung

Im Verlauf der Planung der Evaluation wurde zunächst – auf Basis bestimmter Vorüberlegungen – eine Auswahl möglicher Probanden getroffen. Daraufhin wurde der bereits angesprochene praktische Benutzertest entwickelt und im Anschluss daran der persönliche Fragebogen erstellt.

Nachfolgend sind diese drei Schritte daher einzeln erläutert; des Weiteren wird die Durchführung der Evaluation als Gesamtes kurz vorgestellt.

6.2.1 Auswahl der Probanden

In Anbetracht des geschätzten zeitlichen Aufwandes einer jeden Evaluation und des verfügbaren Rahmens innerhalb der vorliegenden Arbeit wurde eine Mindestanzahl von zehn verschiedene Evaluationen anvisiert. Es galt somit, diese Anzahl an Probanden zu finden. Dabei wurde es als sinnvoll erachtet, die Auswahl der Probanden selbst vorzunehmen und für geeignet befundene Personen direkt anzusprechen.

Dazu war es notwendig, Vorüberlegungen zu den Merkmalen geeigneter Probanden zu treffen. In der Folge wurde festgelegt, dass die Auswahl der Probanden hinsichtlich der Affinität gegenüber technischer Gerätschaften und der Vorkenntnisse im Bereich der Augmented Reality grob einen Querschnitt der Bevölkerung darstellen sollte. Das bedeutet konkret, es sollten sowohl Personen mit als auch ohne grundsätzlichem Interesse an den im Rahmen dieser Arbeit behandelten Themen für die Evaluation in Betracht gezogen werden. Auch das Geschlecht, das Alter und der Beruf der Personen war ein Kriterium bei der Probandenauswahl: Es sollte – im Rahmen des Möglichen – versucht werden, eine gewisse natürliche Streuung bezüglich der genannten Merkmale zu erreichen.

Allgemein sollen die genannten Maßnahmen helfen, die Resultate der Evaluation trotz der relativ geringen Anzahl der Probanden statistisch weitgehend unabhängig von Vorlieben und Vorkenntnissen einzelner Probanden geraten zu lassen, um die Aussagekraft der Ergebnisse zu erhöhen.

6.2.2 Definition des praktischen Benutzertests

Zur Entwicklung des praktischen Benutzertests wurde ein möglicher Anwendungsfall der implementierten Navigationsanwendung als Aufgabe für die Probanden ausgewählt, im Rahmen dessen also eine Ausgangssituation und ein Ziel der Aufgabe definiert, welche den Probanden dazu verleiten sollen, sich eine gewisse Zeit und auf eine intensive Art und Weise mit dem prototypischen System – und vor allem der Interaktion mit selbigem – auseinanderzusetzen.

Der Benutzertest startet mit dem Tragen des Gesamtsystems und dem initialen Zustand der Android-Applikation, welcher einen relativ hochauflösenden Kartenausschnitt der italienischen Stadt Venedig auf mittlerer

Zoom-Stufe anzeigt. Der Proband sieht also vorerst nur einen kleinen, zentralen Teil des gesamten Kartenausschnitts, jedoch aus einer Entfernung, aus der sich keine Straßennamen erkennen lassen. In Abbildung 6.1 ist der statische Kartenausschnitt zu sehen; Abbildung 6.2 zeigt indes den initialen Zustand nach Start der Applikation und erfolgreichem Tracking des verwendeten Markers am Handgelenk des Probanden. Anhand dieser Abbildung lässt sich nachvollziehen, inwieweit der sichtbare Kartenausschnitt bezüglich der wahrnehmbaren Details eingeschränkt ist.

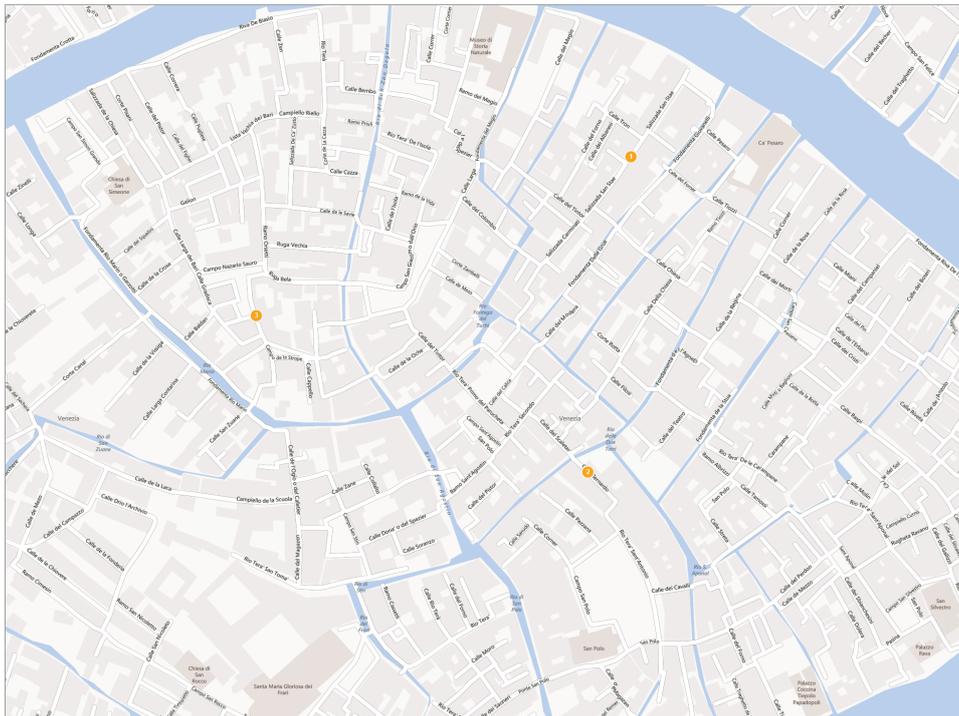


Abbildung 6.1: Statischer Kartenausschnitt zur Verwendung während des Benutzertests, Bildgröße: ca. 2000 * 1500 Pixel

Die definierte Aufgabe des Benutzertests beinhaltet zunächst, drei auf dem dargestellten, statischen Kartenausschnitt befindliche, nummerierte Markierungen mittels Manipulation der Kartenansicht – unter Nutzung von Verschieben, Drehen und Vergrößern/Verkleinern mit Hilfe der unterstützten Finger-Gesten – ausfindig zu machen. Diese nummerierten Markierungen stellen Routenwegpunkte dar; in einem nächsten Schritt soll nun eine mögliche Route vom ersten zum zweiten und von da aus zum dritten Wegpunkt in dieser Reihenfolge gefunden werden. Zusätzlich sollen die Stationen der gefundenen Route zur Verifikation sukzessive laut ausgesprochen werden.

Zur Bewältigung der Aufgabe muss der Proband zunächst durch Verschie-

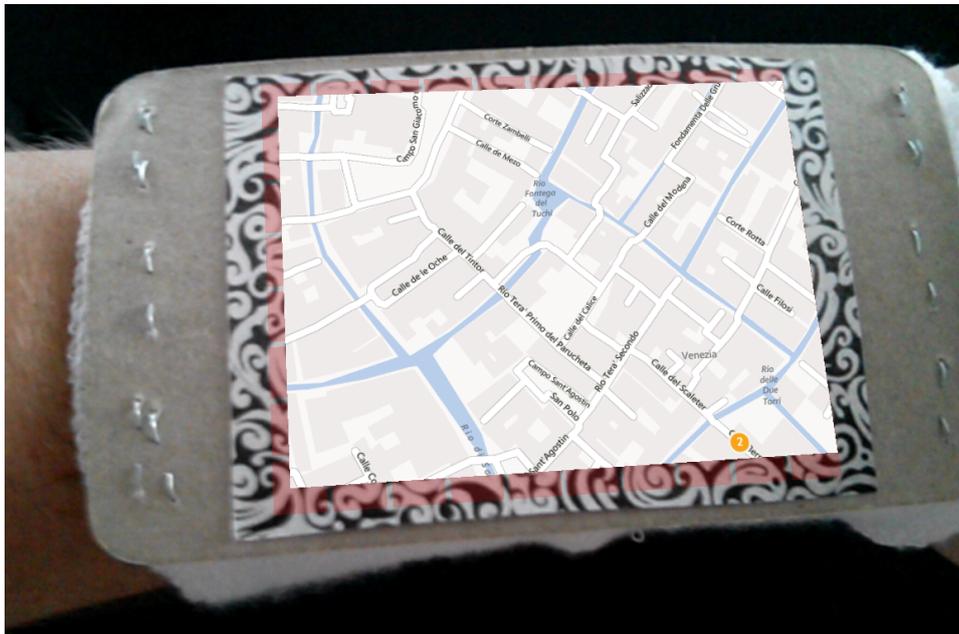


Abbildung 6.2: Initialer Zustand der Applikation während des Benutzertests, bei erfolgreichem Tracking des Markers

ben des sichtbaren Kartenausschnitts die durch farbige Hervorhebung auch in kleinerer Zoom-Stufe – d.h. bei größerer wahrgenommener Entfernung zur virtuelle Kartenebene – sichtbaren Routenwegpunkte lokalisieren und durch anschließendes temporäres Vergrößern der Ansicht die Nummer des jeweiligen Wegpunktes identifizieren. Um daraufhin eine mögliche Route zu finden und deren einzelne Stationen wie Straßennamen lesen und vortragen zu können, muss die Ansicht immer wieder gedreht werden. Insgesamt muss der Proband also vielerlei Manipulationen an der Kartenansicht vornehmen, um die Aufgabe zu lösen. Die Aufgabe gilt dann als bestanden, wenn der Proband die Navigation bis zum dritten Wegpunkt vollendet hat.

Um repräsentative Ergebnisse zu erhalten, musste zudem weitgehend ausgeschlossen werden, dass Teile der Probanden den im Kartenausschnitt dargestellten Ort kennen und somit Vorteile bei der Erfüllung der Aufgabe besitzen. Eine im Entwicklungsprozess der Applikation verwendete Karte der Koblenzer Altstadt wurde daher, wie bereits erwähnt, durch eine Karte der Innenstadt von Venedig ersetzt – basierend auf der Annahme, dass keiner der Probanden Kenntnis über diese besitzt. Die Auswahl des verwendeten Ortes richtete sich außerdem danach, ob die dortigen Straßennamen für deutschsprachige Personen einfach les- und aussprechbar sind; dies ist im Falle der italienischen Sprache zumeist gegeben.

Bei der initialen Erstellung des Benutzertests wurde davon ausgegangen, dass die gestellte Aufgabe während des Tragens des Gesamtsystems – bestehend aus Helm inklusive befestigtem Smartphone, Marker am Handgelenk und der aufgesetzten Videobrille zur Anzeige der Bildschirmausgaben des Smartphones – von den Probanden bewältigt werden kann. Abbildung 6.3 veranschaulicht diese Situation.



Abbildung 6.3: Benutzertest unter Verwendung der Videobrille als Display

Erste Testdurchläufe der Evaluation des Gesamtsystems samt Bearbeitung der Aufgabe des Benutzertests zeigten jedoch, dass die Auflösung der verwendeten Videobrille, der „Sony Glasstron“, mit den ihrerseits verfügbaren $800 * 225$ Bildpunkten nicht ausreichend ist, um die Straßennamen während des Benutzertests lesen zu können. Nach Absprache mit dem Betreuer der vorliegenden Arbeit wurde daher entschieden, auf die Verwendung der Videobrille während des Benutzertests zwischenzeitlich zu verzichten. Stattdessen soll der Bildschirminhalt des am Kopf befestigten Smartphones zeitweise auf einem Monitor ausgegeben werden, welchem der Proband frontal zugewandt ist und auf dem er sodann die virtuellen Überlagerungen des Kamerabildes sehen kann.

Im Detail bedeutet dies, dass der Proband die ihm gestellte Aufgabe zur Navigation gänzlich nur mittels dem aufgesetzten Helm zur Halterung des Smartphones löst. Diese Vorgehensweise ist jedoch nicht weiter bedauerlich, da der Fokus des Benutzertests in der Evaluierung des Interaktionskonzeptes liegt. Im Anschluss an die gelöste Aufgabe folgt dann ein weiterer Teil des Benutzertests, bei dem die Videobrille schließlich aufgesetzt wird und der Benutzer sich nochmals einige Zeit mit dem System und der Anwendung befassen soll, um das Gesamtsystem hinsichtlich Tragekomfort und weiterer Kriterien zu evaluieren. Der Benutzertest wurde dadurch

in zwei Bausteine aufgeteilt. Obwohl im Voraus nicht in der Art angedacht, erscheint diese Zweiteilung unter den gegebenen Umständen als sinnvoll respektive im Allgemeinen als nicht störend, da sich die gesammelten Erfahrungen des Probanden mit der Interaktion mit dem System einerseits und dem Tragen des Gesamtsystems andererseits nicht zwingend überschneiden und gegenseitig beeinflussen. Eine statistische Unabhängigkeit der Ergebnisse zur Interaktion mit dem System von der Frage, ob eine Videobrille oder ein eigens dazu eingesetzter Monitor zum Erleben der erweiterten Realität eingesetzt wurde, scheint also gegeben zu sein. Abbildung 6.4 veranschaulicht die Situation des Lösens der dem Probanden gestellten Aufgabe unter Verwendung eines Monitors, also ohne das Tragen der Videobrille.

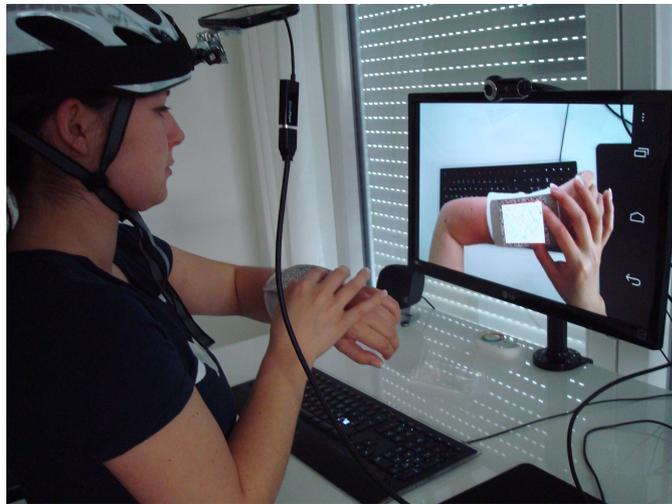


Abbildung 6.4: Benutzertest unter Verwendung eines Monitors statt der Videobrille als Display

Überdies wurde während der ersten Testdurchläufe der Evaluation festgestellt, dass die Positionierung des Markers wie angedacht (und in Abbildung 6.5 dargestellt) am äußeren Handgelenk des Benutzers zu Ermüdungserscheinungen hauptsächlich der Oberarmmuskulatur führen kann. Daraufhin wurde ein zweites Schweißband verwendet, um eine der eigentlichen Marker-Konstruktion ähnliche zu erstellen, bei der der Papier-Marker jedoch um 90° um seinen Mittelpunkt gedreht am Schweißband befestigt wurde. Abbildung 6.6 verdeutlicht die Idee. Dadurch kann das Schweißband so über die Hand gestreift werden, dass der Papier-Marker sich an der Innenseite des Handgelenks befindet; dem entsprechend muss der Benutzer die Innenseite seines Handgelenks jederzeit seinem Gesicht zuwenden, um das System benutzen zu können bzw. das Tracking des Markers zu erlauben. Eine Verwendung des Systems auf diese Weise könnte unter anthropometrischen Gesichtspunkten die bessere Alternative dar-

stellen und eventuell zu weniger Ermüdungserscheinungen auf Seiten des Körpers des Benutzers führen. Zur Ermittlung der besseren Alternative wurde dem persönlichen Fragebogen ein eigenständiger Fragenblock hinzugefügt.



Abbildung 6.5: Konstruktion zur Befestigung des Markers an Außenseite des Handgelenks

6.2.3 Erstellung des persönlichen Fragebogens

Der finale persönliche Fragebogen, welcher von jedem Probanden während der Evaluation ausgefüllt werden soll, ist in Anhang A der vorliegenden Arbeit einzusehen. Wie dort zu sehen, wurde der persönliche Fragebogen in sechs Fragenblöcke eingeteilt, wobei der erste Teil einzig obligatorische Angaben zur Person abfragt. Dies beinhaltet Angaben über Alter, Geschlecht, Beruf – zudem wird abgefragt, ob der Proband Brillenträger ist. Die Intention dieser Frage war, eventuelle visuelle Schwierigkeiten seitens des Probanden bei der Nutzung des Systems gegebenenfalls mit dem Sehvermögen der Person in Zusammenhang bringen zu können.

Die Fragenblöcke 2 bis 5 beinhalten geschlossene Aussagen – auch Hypothesen genannt –, welche anhand einer *Ordinalskala* mit einer von fünf möglichen Bewertungen versehen werden. Statistisch werden die verschiedenen Hypothesen als Variablen oder, im Falle von Ordinalskalen, als *Merkmale* bezeichnet, die Ausprägungen heißen sodann *Merkmalsausprägungen* und implizieren eine natürliche Rangordnung. Mit Hilfe dieser Rangordnung lassen sich den Merkmalsausprägungen entsprechende Rangzahlen



Abbildung 6.6: Konstruktion zur Befestigung des Markers an Innenseite des Handgelenks

zuordnen. Dadurch können die Merkmale später einzeln und im Verbund bewertet, Relationen untereinander hergestellt und Mittelwerte berechnet werden.

Die verwendete Ordinalskala besteht im Falle fast aller Fragenblöcke aus den folgenden Merkmalsausprägungen, (die entsprechenden Rangzahlen stehen in Klammern):

- trifft nicht zu (-2)
- trifft eher nicht zu (-1)
- trifft teilweise zu (0)
- trifft eher zu (1)
- trifft zu (2)

In Falle des Fragenblocks 3.1 werden hingegen folgende Merkmalsausprägungen verwendet (Rangzahlen ebenfalls entsprechend in Klammern):

- schlecht (-2)
- eher schlecht (-1)
- neutral (0)
- eher gut (1)
- gut (2)

Der zweite Fragenblock fragt den Erfahrungsschatz des Probanden bezüglich technischer Geräte im Allgemeinen und Augmented Reality-Anwendungen im Speziellen ab. Zudem erfolgt an dieser Stelle eine Selbsteinschätzung der Technikaffinität und es wird das Interesse an neuartigen tragbaren Computern generell und an der Idee des im Rahmen dieser Arbeit entwickelten Systems abgefragt. Letztere zwei Fragen sollen zur Einschätzung der Antworten auf spätere Fragen zur emotionalen Einstellung gegenüber dem dann getesteten System dienen.

Im dritten Fragenblock sollen die Benutzbarkeit des Systems und die Lösbarkeit der Aufgabe des Benutzertests bewertet werden. In Fragenblock 3.1 wird dazu unter anderem nach den Bewertungen der einzelnen Interaktionsmöglichkeiten gefragt. Dazu sollen die oben erwähnten Ausprägungen von „schlecht“ bis „gut“ in die – in diesem Falle lückenhaften – Aussagen eingesetzt werden. In Fragenblock 3.2 soll besonders die Intuitivität der gebotenen Interaktionsmöglichkeiten bewertet. Fragenblock 3.3 hingegen fragt nach den Bewertungen der zwei getesteten, unterschiedlichen Marker-Konstruktionen (Marker an der Innen- bzw. Außenseite des Handgelenks).

Fragenblock 4 beschäftigt sich mit dem Gesamtsystem. Dazu werden der Tragekomfort und ähnliche Parameter bewertet; außerdem sollen an dieser Stelle Emotionen des Probanden, der Benutzung des Systems gegenüber, und der Spaßfaktor ermittelt werden.

Der fünfte Fragenblock ist erneut in zwei Bausteine aufgeteilt. Der erste dient der Ermittlung der Bereitschaft des Probanden, das getragene System in seiner derzeitigen Form im Alltag oder in besonderen Situationen zu nutzen; überdies soll der Proband die Nützlichkeit dieses Systems einschätzen. Der zweite Teil enthält dieselben Hypothesen zur Bewertung, jedoch bezogen auf ein nach Sicht des Probanden optimales System.

Der sechste und letzte Fragenblock beinhaltet indessen keine Hypothesen zur Bewertung, sondern allgemein Fragen zur schriftlichen Beantwortung. Dort soll der Proband beispielsweise niederschreiben, was ihm an dem getesteten System gefällt und was nicht. Zudem soll er Verbesserungsvorschläge, falls vorhanden, äußern und durch die Beantwortung einer weiteren Frage dazu angeregt werden, über mögliche Einsatzzwecke eines solchen Systems nachzudenken.

6.2.4 Durchführung der Evaluation

Die Durchführung der Evaluation soll hier nur kurz angeschnitten werden; zur genaueren Betrachtung empfiehlt sich die Lektüre der eigens erstellten Checkliste zur Evaluation, welche den Ablauf ebenjener – auch hinsichtlich des angestrebten zeitlichen Rahmens – genauestens beschreibt und während der Durchführung der Evaluation stets herangezogen wurde. Diese Checkliste befindet sich in Anhang B.

Grob zusammengefasst wurde folgender Ablauf der Evaluation angestrebt:

1. Proband wird kurz in das Thema der Evaluation eingeführt.
2. Proband beantwortet den ersten und zweiten Fragenblock des persönlichen Fragebogens.
3. Proband wird mit dem Gesamtsystem und den Interaktionsmöglichkeiten vertraut gemacht.
4. Eine der beiden Marker-Konstruktionen (für Innen- oder Außenseite des Handgelenks) wird angebracht.
5. Helm mit Smartphone-Halterung wird dem Probanden aufgesetzt, die Ausgabe des Systems ist auf einem vor ihm stehenden Monitor zu sehen.

6. Proband wird die Aufgabenstellung des Benutzertests erläutert; die Bearbeitung soll begonnen werden.
7. Nachdem der Proband die Hälfte der Aufgabe gelöst hat, wird die bisher getragene Marker-Konstruktion durch die jeweilige Alternative ersetzt.
8. Nach Ende des Benutzertests beantwortet der Proband den dritten Fragenblock des persönlichen Fragebogens.
9. Nun wird zusätzlich zum Helm mit Smartphone-Halterung die Videobrille aufgesetzt und als Ausgabe des Systems verwendet.
10. Proband wird zur erneuten Auseinandersetzung mit der Anwendung und dessen Interaktion aufgefordert; an dieser Stelle wird dem Probanden keine spezielle Aufgabe gestellt, der Proband kann die Dauer der Auseinandersetzung daher selbst bestimmen.
11. Anschließend beantwortet der Proband die restlichen Fragenblöcke 4 bis 6 des persönlichen Fragebogens. Dies markiert das Ende der Evaluation.

Die Marker-Konstruktion, welche jeweils als erstes zum Einsatz kommt, wird dabei hinsichtlich mehrerer Evaluationsdurchläufe derart rotiert, dass jede Konstruktion in ca. der Hälfte aller Durchläufe als erstes zum Einsatz kommt. Somit wird eine Beeinflussung der Bewertung der einzelnen Marker-Konstruktionen durch die Reihenfolge der Verwendung bestmöglich abgewendet.

6.3 Auswertung

Die Evaluation wurde schließlich an 14 Personen aus verschiedenen Tätigkeitsbereichen und Altersgruppen durchgeführt. Die Rohdaten der Auswertung aller Fragenblöcke inklusive der Antworten auf die allgemeinen Fragen aus dem sechsten Fragenblock sind – bei Bedarf – Anhang C zu entnehmen. Im Folgenden sollen jedoch die Ergebnisse der einzelnen Fragenblöcke genauer untersucht werden.

Zunächst soll Tabelle 6.1 einen groben Überblick über die Auswahl der Probanden geben. Die gewünschte Streuung hinsichtlich Geschlecht, Alter und Beruf wurde demnach nicht in dem gewünschten Maße erreicht, jedoch stellt die Streuung hinsichtlich unterschiedlich großer Affinität gegenüber technischer Gerätschaften und unterschiedlicher Vorkenntnisse im Bereich der Augmented Reality den weitaus wichtigeren Faktor dar, um aussagekräftige Ergebnisse zu erhalten.

Geschlecht	Alter	Beruf	Brillenträger
w	23	Studentin	nein
m	28	Programmierer, Student	nein
w	24	Studentin	nein
m	26	wiss. Mitarbeiter	nein
m	25	Student	nein
w	26	Studentin	nein
m	26	Student	ja
m	24	Student	nein
m	22	Student	ja
m	22	Student	nein
w	47	Physiotherapeutin	ja
m	56	Bankkaufmann	ja
w	52	Bürokauffrau	ja
w	24	Bürokauffrau	ja

Tabelle 6.1: Übersicht der Probanden anhand der getätigten Angaben zur Person

In diesem Sinne zeigt Abbildung 6.7 die Ergebnisse des zweiten Fragenblocks, welcher das Vorwissen der Probanden abfragen sollte. Doch vorerst soll die Darstellung kurz erläutert werden: Wie man an diesem Diagramm sieht, werden Säulendiagramme zur Darstellung der Ergebnisse verwendet. Auf der y-Achse wird dabei die absolute Anzahl der Probanden angezeigt und auf der x-Achse sind die gestellten Hypothesen aufgezeichnet. Die verschiedenen Farbtönen symbolisieren die fünf bereits beschriebenen Merkmalsausprägungen, die dem Probanden bei der Bewertung der einzelnen Hypothesen zur Auswahl standen. Der dunkelgrüne Farbton beschreibt dabei die vollständige Zustimmung der Hypothese, der rote Farbton bedeutet indes, dass ihr gar nicht zugestimmt wurde.

6.3.1 Fragenblock 2: Vorwissen und Kenntnisse

Im Falle des zweiten Fragenblocks lauteten die Hypothesen wie folgt:

- a) Ich würde mich als technikaffin bezeichnen.
- b) Meine Anpassungsgabe im Umgang mit neuen technischen Geräten und Umgebungen ist gut.
- c) Ich hatte bereits Erfahrungen mit praktischen Anwendungen auf dem Gebiet der Augmented Reality.
- d) Das Projekt „Google Glass“ ist mir bekannt.

- e) Ich finde die Idee, Kontaktlinsen oder Brillen als Displays für mobile Computer zu nutzen, gut.

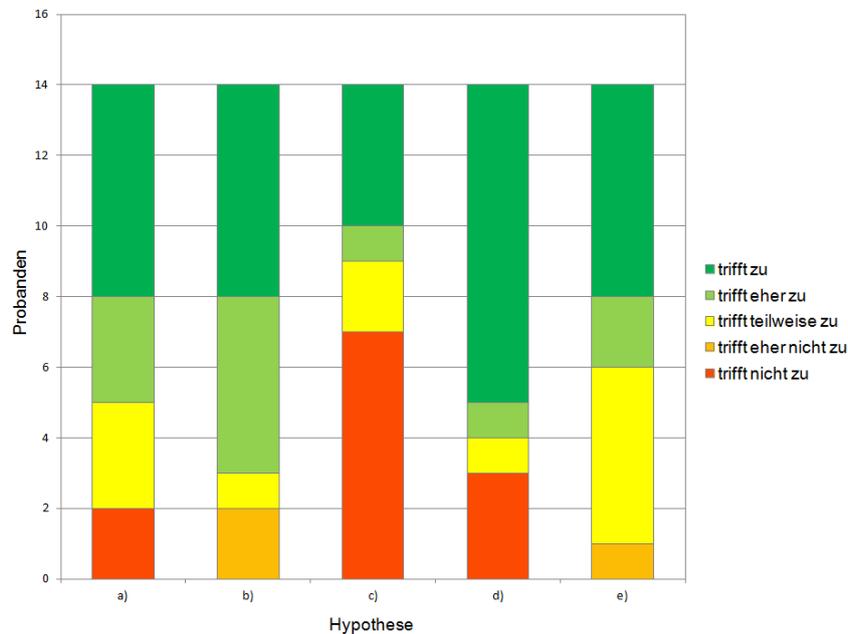


Abbildung 6.7: Ergebnisse des zweiten Fragenblocks: Vorwissen und Kenntnisse

Wie man also Abbildung 6.7 entnehmen kann, herrscht genügend Diversität hinsichtlich der Technikaffinität und vor allem auch bezüglich der bereits gesammelten Erfahrungen mit Augmented Reality. Bezüglich der Anpassungsgabe betreffs technischer Geräte schätzt sich die Mehrheit indes überraschend positiv ein, während der Bekanntheitsgrad von Googles „Project Glass“ deutlich schwankt. Die letzte Frage hat keiner der Probanden vollständig negativ beantwortet – dies sollte von grundsätzlichem Interesse für das Themengebiet dieser Arbeit zeugen.

6.3.2 Fragenblock 3.1: Bearbeitung der Aufgabe und Benutzbarkeit des Systems – Teil 1

Hypothesen:

- Mit der Bedienung des Systems kam ich ... zurecht.
- Das Verschieben der Karte hat ... funktioniert.
- Das Rotieren der Karte hat ... funktioniert.
- Das Vergrößern / Verkleinern der Karte hat ... funktioniert.

e) Die mir gestellte Aufgabe war mit diesem System ... realisierbar.

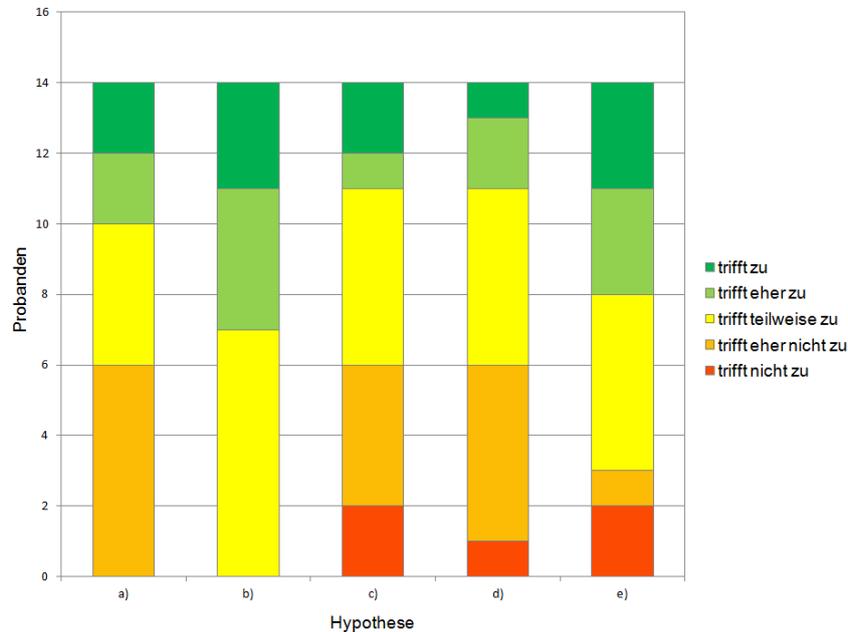


Abbildung 6.8: Ergebnisse des Fragenblocks 3.1: Bearbeitung der Aufgabe und Benutzbarkeit des Systems – Teil 1

In diesem Fragenblock geht es, wie weiter oben bereits erwähnt, um die Beurteilung der Interaktionsmöglichkeiten. Wie man anhand Abbildung 6.8 sieht, fällt diese Beurteilung eher neutral aus, mit einem Schlag in die negative Richtung. Einzig die Möglichkeit zum Verschieben des Kartenausschnitts wurde eher positiv bewertet, wohingegen die Möglichkeiten des Rotierens und Skalierens ein scheinbar relativ hohes Frustrpotential besitzen, welches sich grundsätzlich auch mit den Erfahrungen aus den Evaluationssituationen deckt. Leider ließ sich aufgrund mangelnder Möglichkeiten seitens des Vuforia SDKs die Abtastung des Markers (also die Anzahl der Virtual Buttons) nicht weiter anheben, ohne starke Einbrüche in der Erkennung hinnehmen zu müssen. Durch eine solche Maßnahme in Übereinkunft mit einer allgemein robusteren Erkennung der Virtual Buttons – ebenfalls ein Teilbereich des Vuforia SDKs, der nicht zugänglich ist – hätte das Rotieren und Skalieren womöglich verbessert.

Viele Probanden haben laut den obigen Ergebnissen weiterhin das Gefühl, nicht mit der Bedienung des Systems zurechtzukommen und fühlten sich daher eventuell etwas überfordert mit der an sie gestellten Aufgabe. Möglicherweise sind diese Ergebnisse ein Anzeichen dafür, dass die Aufgabe des Benutzertests zu ambitioniert angelegt war und man den Probanden ihren Einstieg in das System mit simpleren Anwendungsbeispielen hätte

erleichtern können.

6.3.3 Fragenblock 3.2: Bearbeitung der Aufgabe und Benutzbarkeit des Systems – Teil 2

Hypothesen:

- a) Das System tat das, was ich von ihm erwartete.
- b) Die Möglichkeit zum Verschieben der Karte ist intuitiv.
- c) Die Möglichkeit zum Rotieren der Karte ist intuitiv.
- d) Die Möglichkeit zum Vergrößern / Verkleinern der Karte ist intuitiv.

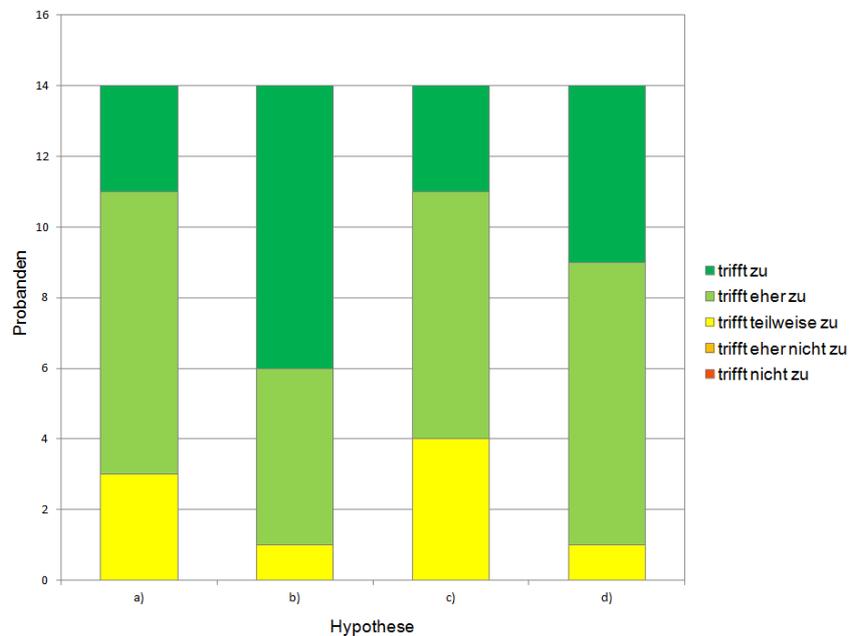


Abbildung 6.9: Ergebnisse des Fragenblocks 3.2: Bearbeitung der Aufgabe und Benutzbarkeit des Systems – Teil 2

Wie man Abbildung 6.9 entnehmen kann, werden die hinter der Interaktion stehenden Ideen bezüglich ihrer Intuitivität recht gut bewertet, wobei auch hier die Verschiebung der Karte durch die simpel anmutende Geste besser abschneidet als die anderen Gesten. Dabei ist zudem sichtbar, dass die Geste des Rotierens knapp schlechter abschneidet als die des Skalierens; dies deckt sich mit den bereits in Unterabschnitt 'Gestenerkennung' erläuterten Problemen bezüglich dieser Geste. Weiterhin schneidet Hypothese a) – die Erwartungskonformität der Anwendung – hinsichtlich der im vorigen Fragenblock erzielten Ergebnisse erstaunlich gut ab.

6.3.4 Fragenblock 3.3: Position des Markers

Hypothesen:

- a) Das Tragen des Markers an der oberen Seite des Handgelenks (im Stil einer Armbanduhr) empfand ich als angenehm.
- b) Ich könnte mir vorstellen, dass das Tragen in letzterer Form bei längerer Benutzung Schmerzen verursacht.
- c) Das Tragen des Markers an der unteren Seite des Handgelenks empfand ich als angenehm.
- d) Ich könnte mir vorstellen, dass das Tragen in letzterer Form bei längerer Benutzung Schmerzen verursacht.

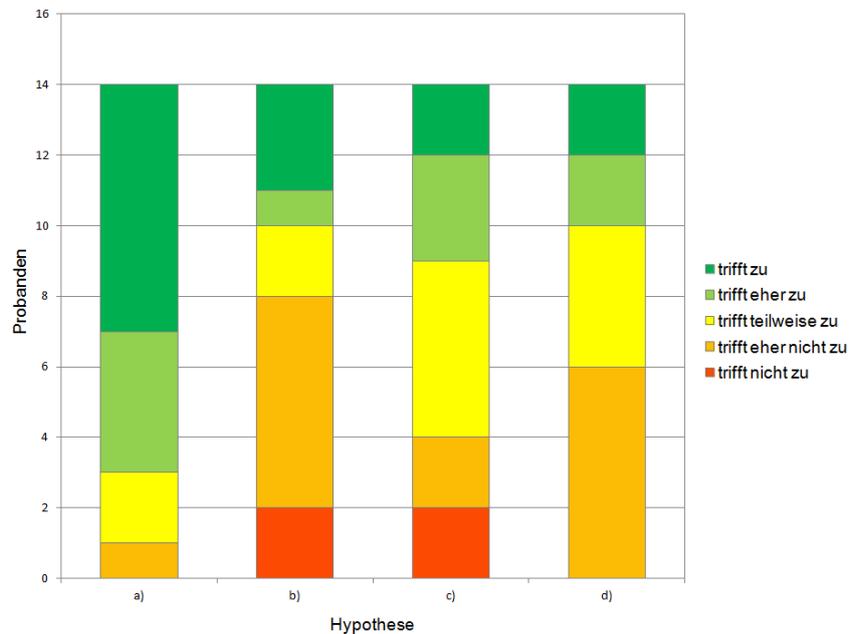


Abbildung 6.10: Ergebnisse des Fragenblocks 3.3: Position des Markers

Obwohl die in Abbildung 6.10 veranschaulichten Ergebnisse den Eindruck erwecken, dass das Tragen des Markers im Stil einer Armbanduhr deutliche Vorteile gegenüber der anderen Variante hat, sie an dieser Stelle angemerkt, dass fragwürdig ist, ob die Ergebnisse der unterschiedlichen Marker-Konstruktionen für die Innen- oder Außenseite des Handgelenks als solche verwendbar sind: Während der Evaluation hat sich nämlich herausgestellt, dass der verwendete Marker unabhängig von seiner Position am Handgelenk vom Probanden meist sehr nah an die Smartphone-Kamera gehalten werden musste, um eine robuste Erkennung der Virtual

Buttons seitens des Vuforia SDKs zu ermöglichen. Die schlechte Erkennung auf Entfernungen von ca. 30 cm lässt sich dabei eventuell auf situativ schlechtere Bedingungen bezüglich der Lichtverhältnisse im Raum oder Ähnliches zurückführen.

Jedenfalls kommt der begründete Verdacht auf, dass die Ergebnisse dieses Fragenblocks durch den Zwang, den Marker fast auf Augenhöhe zu halten, teilweise massiv verfälscht werden. Dieser Verdacht besteht einerseits, da die bei der Konstruktion der Marker angedachte Stellung des Armes während der Benutzung des Systems in der Evaluation nur in seltenen Fällen überhaupt derart möglich war, und andererseits, da die unerwünschte Anpassung der Haltung des Handgelenks und des kompletten Ober- sowie Unterarms zusätzliche Ermüdungserscheinungen der Muskulatur mit sich zog, die von den Probanden oft mitgeteilt wurden. Während der Evaluationssituationen wurde zudem der Eindruck erweckt, dass die Variante des Tragens des Markers an der Innenseite des Handgelenks von der geforderten unnatürlichen Haltung deutlich benachteiligt wurde.

Insgesamt lässt sich somit zu der Frage nach der besseren der beiden Marker-Konstruktionen unter kritischer Betrachtung der Umstände leider keine aussagekräftige Antwort finden.

6.3.5 Fragenblock 4: Tragen des Gesamtsystems

Hypothesen:

- a) Beim Tragen des Systems fühlte ich mich hinsichtlich der Orientierung sicher.
- b) Beim Tragen des Systems fühlte ich mich wohl.
- c) Das Tragen des Systems ist komfortabel.
- d) Ich empfand das Tragen der Brille und der Kopfhalterung als angenehm.
- e) Ich konnte mich beim Tragen des Systems gut in meiner Umgebung orientieren.
- f) Das Tragen machte mir Spaß.
- g) Ich finde das System cool.
- h) Ich finde das System spannend.

Die Fragen dieses Blocks beziehen sich auf das Tragen des gesamten prototypischen Systems inklusive der Videobrille. Die Hypothesen a) und e)

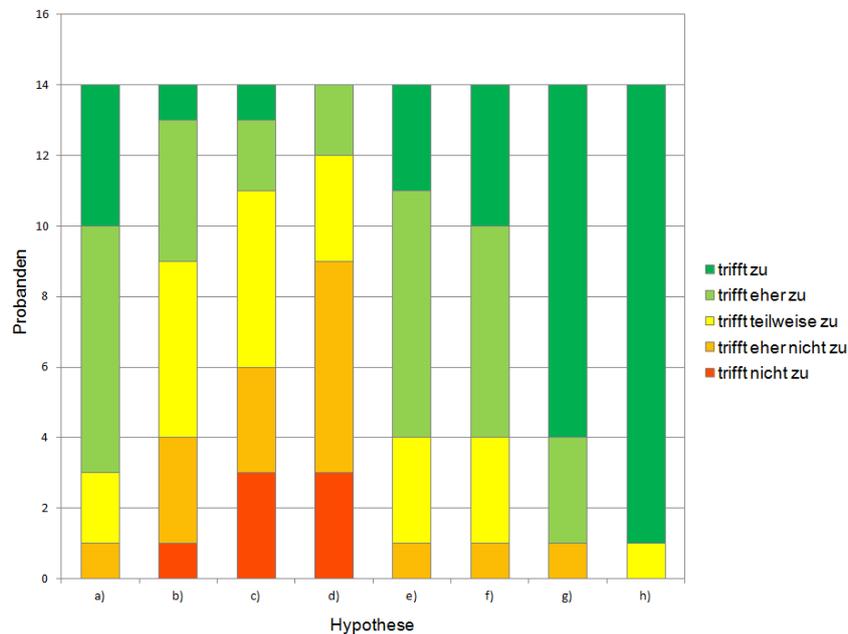


Abbildung 6.11: Ergebnisse des Fragenblocks 4: Tragen des Gesamtsystems

wurden dabei unter der Annahme ausgewählt, dass die Probanden während der Evaluation und beim Tragen des Systems auch mal aufstehen und umher gehen sollen. Der Verdacht lautete, dass die Orientierung in diesem Fall – gerade wegen der fest auf den Unterarm ausgerichteten Smartphone-Kamera, welche in den seltensten Fällen mit der natürlichen Blickrichtung während des Gehens übereinstimmen dürfte – sehr eingeschränkt wird. Leider wurde ein umher Gehen als Teil der Evaluation des Systems aufgrund von Zeitmangel und der Notwendigkeit, das System jederzeit am Stromnetz anschließend zu müssen, nicht durchgeführt. Die in Abbildung 6.11 zu sehenden Ergebnisse bezüglich dieser Hypothesen haben somit kritisch betrachtet leider wenig Aussagekraft.

Weiterhin fällt anhand der Ergebnisse auf, dass das Tragen des Gesamtsystems weder sehr komfortabel noch angenehm zu sein scheint – beides war aufgrund des Gewichts des Helms und der Brille genau so zu erwarten. Die positiven Ergebnisse der Hypothesen f), g) und h) hingegen lassen hoffen, dass die Probanden trotz der Teilnahme an dieser Evaluation künftig nicht vor ähnlichen Ideen und Systemen zurückschrecken werden, da das Prinzip eines solchen Systems solche Emotionen wie die oben stehenden wohl trotz eines dominierenden prototypischen Charakters auszulösen im Stande ist. Die Ergebnisse dieser bewusst emotional gestellten Hypothesen haben die Erwartungen dahingehend sogar übertroffen.

6.3.6 Fragenblock 5.1: Potential des evaluierten Systems

Hypothesen:

- a) Ich könnte mir vorstellen, ein solches System im Alltag zu benutzen.
- b) Ich erachte ein solches System generell als sinnvoll.
- c) Ein solches System könnte für bestimmte Personen eine Bereicherung darstellen.
- d) Ein solches System hat Zukunft.
- e) Ich könnte mir vorstellen, dass ein solches System irgendwann die Benutzung von Smartphones überflüssig macht.

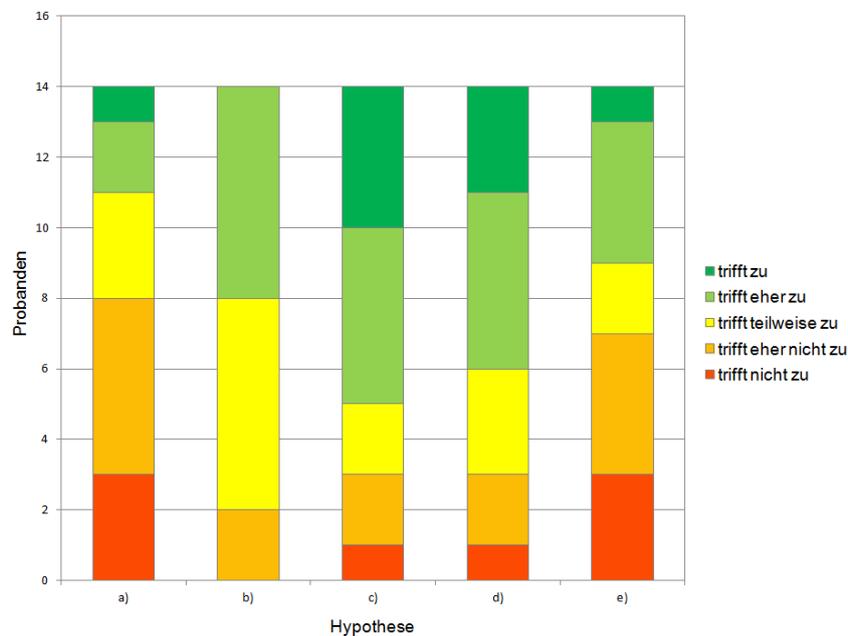


Abbildung 6.12: Ergebnisse des Fragenblocks 5.1: Potential des evaluierten Systems

Wie man den Ergebnissen in Abbildung 6.12 entnehmen kann, kommt das evaluierte System bzgl. seiner Alltagstauglichkeit nicht gut weg. Die Frage nach der Bereicherung für bestimmten Personen ergab zwar einige positive Bewertungen, jedoch besteht der Anspruch solcher System – zumindest seit der aktuellen internationalen Entwicklungen auf diesem Gebiet – gerade darin, den Alltag eines jeden Menschen zu verbessern und das Smartphone hinsichtlich der Intuitivität seiner Bedienung zu schlagen und es auf Dauer komplett überflüssig zu machen.

Bezüglich Hypothese d) erhält das entwickelte System immerhin noch eine im Durchschnitt positive Bewertung, nach kritischer Auseinandersetzung mit der Formulierung der Hypothese lässt sich allerdings sagen, dass diese dazu verleitet, an ein künftiges, besseres Gerät zu denken, ohne, dass dies so gefordert war. Daher ist es denkbar, dass auch viele der Probanden an dieser Stelle gute Bewertungen abgaben, weil sie dem Prinzip des Systems im Allgemeinen eine rosige Zukunft voraussagen. Dies sollte eigentlich jedoch erst im kommenden Fragenblock bewertet werden.

Insofern sprechen die hier erzielten Ergebnisse eine klare Sprache; das entwickelte System ist so im Alltag nicht sinnvoll einsetzbar.

6.3.7 Fragenblock 5.2: Potential eines optimalen Systems

Dieser Fragenblock beinhaltet dieselben Hypothesen wie der vorherige Fragenblock 5.1. Dieses Mal beziehen sich die Hypothesen jedoch auf ein optimales System.

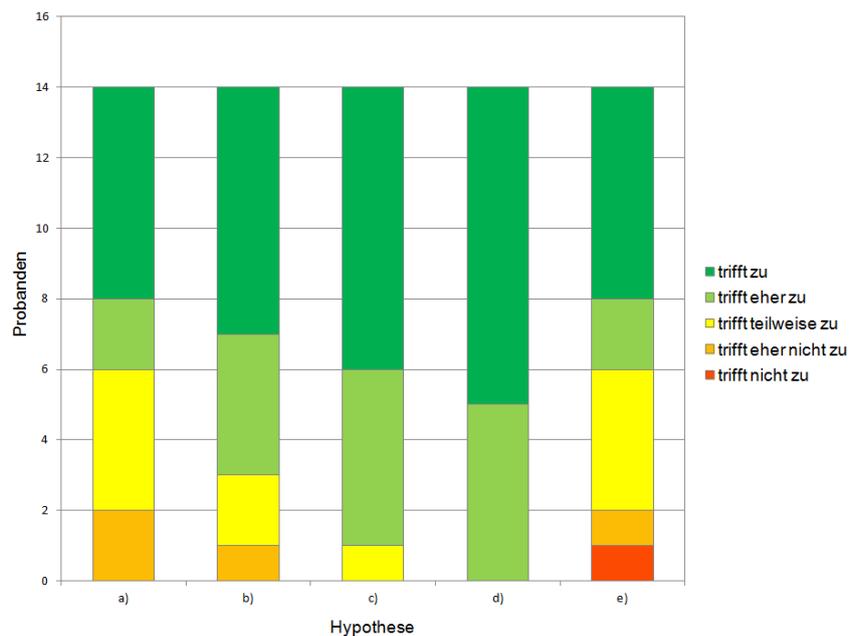


Abbildung 6.13: Ergebnisse des Fragenblocks 5.2: Potential eines optimalen Systems

Schaut man sich die in Abbildung 6.13 gezeigten Ergebnisse im Zusammenhang mit denen des vorigen Fragenblocks an, kommt man schnell darauf, dass die große Mehrheit der Probanden in der grundlegenden Idee des Systems viele Verwendungsmöglichkeiten sehen oder auch einfach nur

sehr interessiert daran sind, was ein solches System leisten könnte, wenn es optimal funktionieren würde. Die Hypothese d) zeigt dies am deutlichsten: Jeder der Probanden glaubt, dass ein funktionierendes und komfortables System dieser Art großes Potenzial für die Zukunft besitzt, auch wenn einige sich mit dem Gedanken, das Smartphone durch ein solches System zu ersetzen, nicht anfreunden wollen.

6.3.8 Fragenblock 6: Allgemeine Fragen

Dieser Fragenblock enthält einzig offene Fragen zur schriftlichen Beantwortung. An dieser Stelle wird eine manuell getroffene Auswahl der gegebenen Antworten herangezogen, um weitere eventuelle Ergebnisse daraus zu extrahieren.

Auf die Frage nach den positiven Eigenschaften des getesteten Systems wurden unter anderem folgende Antworten gegeben:

1. „Ich brauche mein mobiles Gerät nicht aus der Hosentasche holen.“
2. „Schritt in die nächste Stufe zukünftiger mobiler Systeme“
3. „die Integration von Informationen in den Sichtbereich, ohne explizit ein Gerät in der Hand halten zu müssen“
4. „Das Warten, bevor die Bedienung funktioniert, finde ich gut gelöst.“

Die Antworten 1 bis 3 geben den Tenor der Äußerungen recht gut wieder und sind ein weiteres Zeichen für das Potential solcher Systeme. Antwort 4 bestätigt den Nutzen des eingeführten Triggers für die Gestenerkennung. Da es zu diesem Punkt keine negativen Äußerungen gab, lässt sich diese Vorgehensweise zumindest dahingehend als Erfolg verbuchen, dass Systeme, welche ebenso die schwierige Aufgabe der Gestenerkennung allein auf Basis von Farbinformationen lösen wollen, die Technik dieser Art von Triggern ohne Bedenken in Betracht ziehen können.

Auf die Frage nach den negativen Gesichtspunkten des Systems wurde unter anderen folgendes geschrieben:

1. „Helm zu schwer“
2. „Kamera der Brille schlechte Auflösung“
3. „Helm mit Aufbau relativ schwer“
4. „Bei längerer Benutzung tut der Arm weh.“
5. „Anstrengung beim längeren Tragen“

6. „... dass ich eine Kamera auf dem Kopf habe. Wer weiß, wer alles mit schaut.“

Schaut man sich diese Auswahl an Antworten genauer an, fällt sofort auf, dass der Unmut über die mäßige Hardware bezüglich deren Leistung und Komfort bei den negativen Aspekten des Systems überwiegt. Eine Ausnahme bildet Antwort 6; hier wurden zum ersten und einzigen Mal Bedenken hinsichtlich der Privatsphäre geäußert. Aufgrund aktueller Entwicklungen liegt die Vermutung nahe, dass dieses Thema in Zukunft viel Raum einnehmen wird. Die geringe Anzahl der Äußerungen über dieses Thema innerhalb dieser Evaluation lässt sich wahrscheinlich mit der Tatsache begründen, dass solche sehr prototypischen Systeme solche Bedenken im Allgemeinen nicht fördern, da man sich schlecht vorstellen kann, die Systeme im Alltag zu nutzen.

Antwort 4 bestätigt überdies die bereits geäußerte Vermutung, dass die Umstände der schlechten Erkennung der Virtual Buttons und die daraus resultierende Notwendigkeit, den Marker am Handgelenk hoch zu halten, zu zusätzlichen Ermüdungserscheinungen führen.

Weiterhin wurden die Probanden nach ihren Verbesserungsvorschlägen für das getestete System und nach den Komponenten, die ihrer Meinung nach am ehesten einer Verbesserung bedürfen, gefragt. Im Zuge dessen wurden folgende Antworten gegeben – die auftretenden Häufigkeiten stehen in Klammern:

1. Tragekomfort (10x)
2. Auflösung (6x)
3. Reaktionszeit (5x)
4. Bedienung (5x)
5. Sichtfeld (2x)
6. Tracking (1x)

Dies verdeutlicht ebenfalls ganz gut, dass die Hauptprobleme des Systems in der derzeit verfügbaren Hardware liegen, wobei auch die Bedienung – also die Gestensteuerung – häufig als verbesserungswürdig bezeichnet wurde.

Schließlich folge die Frage nach den möglichen Einsatzgebieten eines solchen Systems, worauf folgende Antworten gegeben wurden:

1. „Maps“
2. „Telefon“
3. „Tourismus“

4. „unterwegs in unbekanntem Städten“
5. „in allen Anwendungsbereichen eines Smartphones“
6. „alle Bereiche, in denen heute Smartphones verwendet werden“
7. „medizinischer Bereich“
8. „touristische Besucherführungen“
9. „Vergleiche bei Reparaturen/Operationen richtig/falsch“

Hieran ist zu erkennen, dass touristische Einsatzgebiete oft genannt wurden, direkt danach folgt die Nennung aller Aktivitäten, die man heutzutage mit dem Smartphone erledigt. Auch der medizinische Bereich wurde mehrfach genannt.

7 Fazit und Ausblick

7.1 Fazit

Im Rahmen dieser Arbeit wurde ein prototypisches System vorgestellt und entwickelt, welches unter Zuhilfenahme eines Head-Mounted Displays und eines Smartphones grundlegende Augmented Reality-Anwendungen ermöglicht. Dazu wurde eine Kopfhalterung für das Smartphone eigens entwickelt und die Übertragung der nötigen Daten von Smartphone zum Head-Mounted Display sichergestellt.

Zudem wurde ein Interaktionskonzept erstellt, welches zum Ziel hat, neue Pfade zu beschreiten und zu evaluieren, welche Formen der Interaktion mit künftigen allgegenwärtigen User-Interfaces in Frage kommen und welche Ausprägungen dessen sinnvoll erscheinen. Es wurde zudem die Idee einer Interaktionsform entwickelt, welche sich im Idealfall eines funktionierenden Systems nahezu nahtlos in Alltagsaktivitäten integrieren lässt. Dazu war es notwendig, eine intuitive Benutzung im Fokus zu haben, dabei gleichzeitig die Anforderungen sicherheitskritischer Bereiche des Alltags in die Konzeption mit einzubeziehen und Lösungen für die Einblendung virtueller Inhalte in solchen Situationen zu finden, damit die Nutzung moderner und viel versprechender Technologien nicht zwangsweise mit der Akzeptanz einer Reihe von Kompromissen einhergeht.

Diese Anforderungen wurden dadurch angegangen, dass der Nutzer des Systems vollständig selbst entscheidet, ob und welcher Teil seines Blickfeldes mit virtuellen Inhalten überlagert wird. Zudem bietet das Interaktionskonzept die Möglichkeit der intuitiven Steuerung des Systems mittels Finger-Gesten, wie sie von Smartphones nahezu perfekt umgesetzt werden. Allerdings ist an dieser Stelle anzumerken, dass ein Smartphone, mit einem kapazitiven Sensor heutiger Technik ausgestattet, die Zeit und die Intensität der Berührung seiner Oberfläche bis ins Kleinste feststellen kann.

Die Interaktion über rein optische Wege – mittels Tracking – erlauben hingegen den Vorteil, keinerlei Gerät zur Interaktion in der Hand haben zu müssen. Jedoch stellt sich dadurch das Problem der Ferndiagnose: Ohne örtliche Nähe und sonstiger künstlicher Datenverbindungen sollen Dinge erfasst und klassifiziert werden. Der Zeitpunkt von Start und Ende der Interaktion ist mit aktuellen Mitteln vom Computersystem nicht hinreichend gut ermittelbar, was Ungenauigkeiten impliziert. Das Problem der grundlegenden Machbarkeit ist jedoch auf einem guten Weg zur Lösung, wobei die Genauigkeit mit der Verbesserung der technischen Gegebenheiten immer weiter ansteigen wird; jedoch kommen bei dieser Art von Interaktion zusätzliche Störfaktoren ins Spiel. Während im Falle von Touch-Displays

die örtliche Nähe gegeben ist und Störfaktoren nur dann zu einem echten Problem werden, wenn sie systeminhärenter Natur sind, existiert bei dem Ansatz des optischen Trackings eine Vielzahl an externen Störfaktoren, die nur eingeschränkt beeinflussbar sind; dazu gehört vor allem die Beleuchtung. Weitere Störfaktoren im Falle von optischem Tracking sind einzig durch bestimmte Maßnahmen und Einschränkungen auf Nutzerseite zu bekämpfen. Hierzu zählen auch die im Rahmen dieser Arbeit aufgetretenen Probleme, welche nachfolgend noch erläutert werden.

Zur Evaluation des im Rahmen dieser Arbeit entwickelten Interaktionskonzeptes war es überdies notwendig, eine beispielhafte Augmented Reality-Anwendung zu entwickeln, bei dessen Konzeptionierung erneut darauf geachtet werden sollte, dass die Anwendung – im Rahmen des zeitlich Realisierbaren – möglichst einen Mehrwert gegenüber aktuellen Lösungen besitzt.

Im weiteren Verlauf der Arbeit wurde die Realisierung der Konzepte angegangen. Dabei kam es immer wieder zu unverhofften neuen Herausforderungen, die es zu lösen oder zumindest zu umgehen galt. Dabei wurden zahlreiche neuartige Erfahrungen gesammelt. Bezug nehmend auf die eben angesprochenen Schwierigkeiten ist vor allem das Problem der ungewollten Interaktion hervorzuheben: Ungewollte optische Reize lösen auf Seiten des Computersystems Dinge aus, die vom Nutzer nicht beabsichtigt sind – im Falle des Markertrackings werden diese Reize durch die meist versehentliche Verdeckung von Teilen von Markern dargestellt.

Das beschriebene Problem wurde während der Phase der Implementierung bekannt und es wurde daraufhin versucht, dieses Problem mittels der Verwendung von Triggern möglichst weitgehend zu lösen. Trigger sind systeminterne Countdowns, welche den Start der Interaktion mit dem System verzögern, um damit die meist kurzweiligen Störfaktoren – die unbeabsichtigten Verdeckungen – von der Interaktion mit dem System auszuschließen. Betrachtet man diesen Ansatz jedoch genauer und versucht die Nebenwirkungen für die Interaktion zu ermitteln, kommt man darauf, dass diese Art der Lösung nur eine Verschiebung des Problems auslöst. Denn durch die Nutzung von Triggern wird mit der gewohnten Interaktionsmetapher des direkten und erwartungskonformen Feedbacks seitens des Systems auf physische Eingaben des Nutzers gebrochen. Der Nutzer muss diese vergleichsweise umständliche Art der Benutzung in Kauf nehmen – erneut ein Kompromiss, der vermieden werden sollte, um die Interaktion insgesamt zu verbessern.

Dieses und andere Probleme sowie die Einschränkungen bezüglich der verwendbaren Hardware führten dazu, dass die Ergebnisse der Evaluation bezüglich den prototypischen Systems ernüchternd ausfallen: Kaum einer der Probanden würde das System freiwillig nutzen. Jedoch konnten Erfol-

ge hinsichtlich der Bewertung der entwickelten Ansätze erzielt werden; ein Großteil der Nutzer sieht eine große Zukunft für ähnliche Systeme, sobald diese funktionieren und nicht störend wirken. Zudem scheint das Interesse für solche Systeme im Allgemeinen sehr groß zu sein, sowohl vor der Erfahrung der Evaluation als auch danach.

Konkludierend lässt sich sagen, dass das entwickelte prototypische System aufgrund vielfältiger Ursachen keine Bereicherung im Alltag darstellt. Jedoch war das oberste Ziel, neue Erkenntnisse auf diesem Gebiet zu erlangen. Es war daher also gewünscht, möglichst schnell zu einem nutzbaren Prototypen zu kommen, mit dem man sodann evaluieren konnte, inwiefern diese Art der Interaktion sinnvoll ist und welche Dinge man in diesem recht unerforschten Gebiet der Interaktion mit allgegenwärtigen User-Interfaces beachten sollte, um die Technik möglichst gut und universell benutzbar zu machen.

7.2 Ausblick

Anhand der durch diese Arbeit gesammelten Erkenntnisse lässt sich sagen, dass nicht mehr viel fehlt, bis diesem Prototyp ähnliche Systeme massenkompatibel werden. Die technischen Voraussetzungen sind derweil meines Erachtens schon größtenteils erfüllt, das Interesse zumindest der jüngeren Bevölkerung steigert sich gefühlt von Monat zu Monat; jedoch fehlt es noch an den richtigen Nutzungskonzepten und einer entsprechenden Aufklärung. Während Google mit dem „Project Glass“ eine große Zahl an möglichen Einsatzbereichen bedient und Google mit diesem Gerät – falls denn gewollt – innerhalb kürzester Zeit in Massenproduktion gehen könnte, wird in sämtlichen Teilen der Welt Angst vor Veränderung geschürt. Diese, gemischt mit dringend notwendigen aber nur oberflächlich vollzogenen Diskussionen zum Datenschutz der Zukunft, behindern die aktuelle Entwicklung.

Im Hinblick auf die technischen Ausprägungen ist trotz dessen gut vorstellbar, dass die Leistung und vor allem die Sensorik tragbarer Computer in Zukunft massiv vorangetrieben wird. Die im Rahmen dieser Arbeit entwickelte Gestenerkennung könnte von minimalen Verbesserungen schon profitieren. Man könnte die implementierte Anwendung zu einem freien und einfach zu erweiternden SDK werden lassen, welches anderen Entwicklern die Arbeit hinsichtlich der Eingabe von Daten innerhalb eigener Augmented Reality-Anwendungen abnimmt. Mit besserem Tracking dank besserer Sensorik und mehr Leistung sind vielfältige Einsatzgebiete denkbar. Beispielsweise könnten eingesetzte Stereokameras die Möglichkeiten zur Interaktion mit solchen Systemen vervielfältigen; der Computer könnte unsere Handbewegungen mit entsprechender Technik sodann soweit er-

kennen und verstehen, dass er Gebärdensprache direkt in digital erzeugte Laute umwandeln könnte.

Abschließend lässt sich gerade mit Blick auf die angesprochenen aktuellen Entwicklungen auf dem internationalen Markt des *Wearable* respektive *Ubiquitous Computing* festhalten, dass eine – nach dem letzten großen Erfolg des Smartphones – erneute radikale Veränderung der Art und Weise, wie Menschen mit Computern umgehen und kommunizieren, kurz bevorsteht.

Literatur

- [Adv] ADVANCED REALTIME TRACKING GMBH: *Tree target*. <http://www.ar-tracking.com/products/markers-targets/targets/passive/>, Abruf: 20.04.2013
- [Azu97] AZUMA, Ronald T.: A survey of augmented reality. In: *Presence: Teleoperators and Virtual Environments* 6 (1997), August, Nr. 4, S. 355–385
- [BKLP04] BOWMAN, Doug A. ; KRUIJFF, Ernst ; LAVIOLA, Joseph J. ; POUPYREV, Ivan: *3D User Interfaces: Theory and Practice*. Redwood City, CA, USA : Addison Wesley Longman Publishing Co., Inc., 2004. – ISBN 0201758679
- [Bro11] BROTHER INDUSTRIES, LTD.: *Brother announces commercialization of "AiRScouter" see-through type head-mounted display*. Version: 2011. <http://www.brother.com/en/news/2011/airscouter/>, Abruf: 20.04.2013
- [DGN09] DUBOIS, Emmanuel ; GRAY, Philip ; NIGAY, Laurence: *The Engineering of Mixed Reality Systems*. 1st. Springer, 2009. – ISBN 1848827326, 9781848827325
- [Goo12] GOOGLE INC.: *Project Glass - Design study: Emily out and about*. Version: 2012. https://lh3.googleusercontent.com/-quy90x8dQJI/T3xUHhub6PI/AAAAAAAAABU/C_aMHkXPmTw/s5000/glass_photos.jpg, Abruf: 14.12.2012
- [Goo13] GOOGLE INC.: *Google Maps API*. Version: 2013. <https://developers.google.com/maps/>, Abruf: 23.04.2013
- [GS85] GOSHTASBY, Ardeshir ; STOCKMAN, G.: Point Pattern Matching Using Convex Hull Edges. In: *IEEE Trans. Systems, Man, and Cybernetics* Bd. 15, 1985, S. 631–637
- [HSH⁺06] HOTELLING, Steve ; STRICKON, Joshua A. ; HUPPI, Brian Q. ; CHAUDHRI, Imran ; CHRISTIE, Greg ; ORDING, Bas ; KERR, Duncan R. ; IVE, Jonathan P.: *Gestures for touch sensitive input devices*. Version: 02 2006. http://www.patentlens.net/patentlens/patent/US_2006_0026521_A1/en/
- [Ins] INSTITUTE FOR COMPUTER GRAPHICS AND VISION: *Indoor Modeling and Tracking for Augmented Reality*. <http://www.icg.tugraz.at/project/indoorar/>, Abruf: 20.04.2013

- [IU97] ISHII, Hiroshi ; ULLMER, Brygg: Tangible bits: towards seamless interfaces between people, bits and atoms. In: *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*. New York, NY, USA : ACM, 1997 (CHI '97). – ISBN 0–89791–802–9, 234–241
- [KB99] KATO, Hirokazu ; BILLINGHURST, Mark: Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System. In: *Proceedings of the 2nd International Workshop on Augmented Reality (IWAR 99)*. San Francisco, USA, Oktober 1999
- [Low04] LOWE, David G.: Distinctive Image Features from Scale-Invariant Keypoints. In: *International Journal of Computer Vision* 60 (2004), Nr. 2, 91-110. <http://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf>
- [Mic13] MICROSOFT CORPORATION: *Bing Maps*. Version: 2013. <http://www.bing.com/maps/>, Abruf: 23.04.2013
- [Mis13] MISSFELD, Martin: *Die Google-Brille (Infografik)*. Version: Februar 2013. <http://www.brille-kaufen.org/google-brille/>, Abruf: 20.04.2013
- [MTUK94] MILGRAM, Paul ; TAKEMURA, Haruo ; UTSUMI, Akira ; KISHINO, Fumio: Augmented Reality: A Class of Displays on the Reality-Virtuality Continuum. In: *Proceedings of SPIE 2351, Telemanipulator and Telepresence Technologies*, 1994, S. 282–292
- [Mur92] MURTAGH, Fionn: A feature - based $O(N^2)$ approach to point pattern matching. In: *Pattern Recognition, 1992. Vol.II. Conference B: Pattern Recognition Methodology and Systems, Proceedings., 11th IAPR International Conference on, 1992*
- [PBWI96] POUPYREV, Ivan ; BILLINGHURST, Mark ; WEGHORST, Suzanne ; ICHIKAWA, Tadao: The Go-Go Interaction Technique: Non-Linear Mapping for Direct Manipulation in VR. In: *ACM Symposium on User Interface Software and Technology*, 1996, S. 79–80
- [Qua13a] QUALCOMM AUSTRIA RESEARCH CENTER GMBH: *Developing with Vuforia | Vuforia Developer Portal*. Version: 2013. <https://developer.vuforia.com/resources/dev-guide/getting-started>, Abruf: 23.04.2013
- [Qua13b] QUALCOMM AUSTRIA RESEARCH CENTER GMBH: *Qualcomm Vuforia SDK*. Version: 2013. <https://www.vuforia.com/>, Abruf: 23.04.2013

- [Rei09] REIF, Rupert: *Entwicklung und Evaluierung eines Augmented Reality unterstützten Kommissioniersystems*, Technische Universität München, Diss., 2009
- [Sew09] SEWELL DEVELOPMENT CORPORATION: *i-Glasses High Resolution Head Mounted Display*. Version: 2009. <http://sewelldirect.com/i-Glasses-HMD.asp>, Abruf: 20.04.2013
- [SG97] SZALAVARI, Zsolt ; GERVAUTZ, Michael: The Personal Interaction Panel - a Two-Handed Interface for Augmented Reality. In: *Computer Graphics Forum*, 1997, S. 335–346
- [Shu09] SHUMAKER, Randall (Hrsg.): *Virtual and Mixed Reality, Third International Conference, VMR 2009, Held as Part of HCI International 2009, San Diego, CA, USA, July 19-24, 2009. Proceedings*. Bd. 5622. Springer, 2009 (Lecture Notes in Computer Science). – ISBN 978-3-642-02770-3
- [Son98] SONY CORPORATION: *Glasstron PLM-A55 – Manual*. 1998
- [Tön10] TÖNNIS, Marcus: *Augmented Reality – Einblicke in die Erweiterte Realität*. Springer, 2010. – ISBN: 978-3-642-14178-2, DOI: 10.1007/978-3-642-14179-9
- [Tou12] TOUCHARCADE.COM, LLC: *Touch Arcade - Impressive Augmented Reality Game Possibilities on the iPhone*. Version: 2012. <http://toucharcade.com/2009/03/26/impressive-augmented-reality-game-possibilities-on-the-iphone/>, Abruf: 14.12.2012
- [Tra] TRANSFORMIT AUSTRALIA: *Black & White Swirl Overlay*. <http://transformit.com.au/assets/images/Table%20Decor/Black%20%20White%20Swirl%20Overlay.JPG>, Abruf: 20.04.2013
- [Tro98] TROUVÉ, Alain: Diffeomorphisms Groups and Pattern Matching in Image Analysis. In: *International Journal of Computer Vision* 28 (1998), Nr. 3, 213-221. <http://dx.doi.org/10.1023/A:1008001603737>. – DOI 10.1023/A:1008001603737. – ISSN 0920-5691
- [Tsa87] TSAI, Roger Y.: A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. In: *IEEE Journal of Robotics and Automation* 3 (1987), Nr. 4, 323-344. http://www.vision.caltech.edu/bouguetj/calib_doc/papers/Tsai.pdf

- [UI00] ULLMER, B. ; ISHII, H.: Emerging frameworks for tangible user interfaces. In: *IBM Syst. J.* 39 (2000), Juli, Nr. 3-4, 915–931. <http://dx.doi.org/10.1147/sj.393.0915>. – DOI 10.1147/sj.393.0915. – ISSN 0018–8670
- [Vir] VIRTUAL REALITIES, LTD.: *CyberGlove II*. <http://www.vrealities.com/cyber.html>, Abruf: 20.04.2013
- [Wik13a] WIKIPEDIA: *Multi-touch* — *Wikipedia, The Free Encyclopedia*. Version: 2013. <http://en.wikipedia.org/w/index.php?title=Multi-touch&oldid=550448225>, Abruf: 20.04.2013
- [Wik13b] WIKIPEDIA: *Reactable* — *Wikipedia, The Free Encyclopedia*. Version: 2013. <http://en.wikipedia.org/w/index.php?title=Reactable&oldid=548385050>, Abruf: 20.04.2013
- [WLF07] WHITE, Sean ; LISTER, Levi ; FEINER, Steven: Visual Hints for Tangible Gestures in Augmented Reality. In: *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. Washington, DC, USA : IEEE Computer Society, 2007 (ISMAR '07). – ISBN 978–1–4244–1749–0, 1–4
- [Zug12] ZUGARA: *We Are Organized Chaos - 3 Questions To Ask Yourself Before Using Augmented Reality In Marketing And Advertising*. Version: 2012. <http://weareorganizedchaos.com/index.php/tag/augmented-reality-shopping-online/>, Abruf: 14.12.2012

A Fragebogen zur Evaluation

Fragebogen zur Evaluation eines prototypischen Augmented Reality-Systems im Rahmen der Master-Arbeit „Interaktion mit ubiquitären User-Interfaces“

1 Persönliche Angaben

Bitte angeben.

Geschlecht: weiblich männlich

Alter: _____

Beruf: _____

Sind Sie Brillenträger? ja nein

2 Vorwissen und Kenntnisse

Schätzen Sie folgende Aussagen bitte anhand der vorliegenden Skala ein.

	trifft nicht zu	trifft eher nicht zu	trifft teilweise zu	trifft eher zu	trifft zu
a) Ich würde mich als technikaffin bezeichnen.	<input type="checkbox"/>				
b) Meine Anpassungsgabe im Umgang mit neuen technischen Geräten und Umgebungen ist gut.	<input type="checkbox"/>				
c) Ich hatte bereits Erfahrungen mit praktischen Anwendungen auf dem Gebiet der Augmented Reality.	<input type="checkbox"/>				
d) Das Projekt ‚Google Glass‘ ist mir bekannt.	<input type="checkbox"/>				
e) Ich finde die Idee, Kontaktlinsen oder Brillen als Displays für mobile Computer zu nutzen, gut.	<input type="checkbox"/>				

3.1 Bearbeitung der Aufgabe und Benutzbarkeit des Systems – Teil 1

Bitte setzen Sie gedanklich eine der Bewertungen in die jeweilige Aussage ein.

	schlecht	eher schlecht	neutral	eher gut	gut
a) Mit der Bedienung des Systems kam ich ... zurecht.	<input type="checkbox"/>				
b) Das Verschieben der Karte hat ... funktioniert.	<input type="checkbox"/>				
c) Das Rotieren der Karte hat ... funktioniert.	<input type="checkbox"/>				
d) Das Vergrößern / Verkleinern der Karte hat ... funktioniert.	<input type="checkbox"/>				
e) Die mir gestellte Aufgabe war mit diesem System ... realisierbar.	<input type="checkbox"/>				

3.2 Bearbeitung der Aufgabe und Benutzbarkeit des Systems – Teil 2

Schätzen Sie folgende Aussagen bitte anhand der vorliegenden Skala ein.

	trifft nicht zu	trifft eher nicht zu	trifft teilweise zu	trifft eher zu	trifft zu
a) Das System tat das, was ich von ihm erwartete.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
b) Die Möglichkeit zum Verschieben der Karte ist intuitiv.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
c) Die Möglichkeit zum Rotieren der Karte ist intuitiv.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
d) Die Möglichkeit zum Vergrößern / Verkleinern der Karte ist intuitiv.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

3.3 Position des Markers

Schätzen Sie folgende Aussagen bitte anhand der vorliegenden Skala ein.

	trifft nicht zu	trifft eher nicht zu	trifft teilweise zu	trifft eher zu	trifft zu
a) Das Tragen des Markers an der oberen Seite des Handgelenks (im Stil einer Armbanduhr) empfand ich als angenehm.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
b) Ich könnte mir vorstellen, dass das Tragen in letzterer Form bei längerer Benutzung Schmerzen verursacht.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
c) Das Tragen des Markers an der unteren Seite des Handgelenks empfand ich als angenehm.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
d) Ich könnte mir vorstellen, dass das Tragen in letzterer Form bei längerer Benutzung Schmerzen verursacht.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

4 Tragen des Gesamtsystems

Schätzen Sie folgende Aussagen bitte anhand der vorliegenden Skala ein.

	trifft nicht zu	trifft eher nicht zu	trifft teilweise zu	trifft eher zu	trifft zu
a) Beim Tragen des Systems fühlte ich mich hinsichtlich der Orientierung sicher.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
b) Beim Tragen des Systems fühlte ich mich wohl.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
c) Das Tragen des Systems ist komfortabel.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
d) Ich empfind das Tragen der Brille und der Kopfhaltung als angenehm.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
e) Ich konnte mich beim Tragen des Systems gut in meiner Umgebung orientieren.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
f) Das Tragen machte mir Spaß.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
g) Ich finde das System cool.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
h) Ich finde das System spannend.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

5.1 Potential des evaluierten Systems

Schätzen Sie folgende Aussagen bitte anhand der vorliegenden Skala ein.

	trifft nicht zu	trifft eher nicht zu	trifft teilweise zu	trifft eher zu	trifft zu
a) Ich könnte mir vorstellen, ein solches System im Alltag zu benutzen.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
b) Ich erachte ein solches System generell als sinnvoll.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
c) Ein solches System könnte für bestimmte Personen eine Bereicherung darstellen.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
d) Ein solches System hat Zukunft.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
e) Ich könnte mir vorstellen, dass ein solches System irgendwann die Benutzung von Smartphones überflüssig macht.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

5.2 Potential eines optimalen Systems

Schätzen Sie folgende Aussagen bitte anhand der vorliegenden Skala ein.

	trifft nicht zu	trifft eher nicht zu	trifft teilweise zu	trifft eher zu	trifft zu
a) Ich könnte mir vorstellen, ein solches, optimales System im Alltag zu benutzen.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
b) Ich erachte ein solches, optimales System generell als sinnvoll.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
c) Ein solches, optimales System könnte für bestimmte Personen eine Bereicherung darstellen.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
d) Ein solches, optimales System hat Zukunft.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
e) Ich könnte mir vorstellen, dass ein solches, optimales System irgendwann die Benutzung von Smartphones überflüssig macht.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

6 Allgemeine Fragen

Bitte nehmen Sie sich für die Beantwortung der folgenden Fragen ausreichend Zeit.

a) Was gefällt Ihnen an diesem System?

b) Was gefällt Ihnen nicht an diesem System?

c) Was könnte man Ihrer Meinung nach verbessern?

d) Welche Teile dieses Systems bedürfen Ihrer Meinung nach am ehesten einer Verbesserung (z.B. Tragekomfort, Bedienung, Auflösung, Sichtfeld, Reaktionszeit, Tracking)?
Mehrfachnennungen sind erwünscht.

e) Welche Anwendungen könnten mit seinem solchen System eine Bereicherung darstellen?

f) Welche weiteren Anmerkungen haben Sie?

Vielen Dank für Ihre Teilnahme.

B Checkliste zur Evaluation

Evaluationscheckliste

Start- und Endzeit der Tätigkeit [hh:mm – hh:mm]	Tätigkeit
00:00 – 00:02	Begrüßung und thematische Kurzeinführung
00:02 – 00:07	Persönlicher Fragebogen – Seite 1: <ul style="list-style-type: none"> - Proband erhält persönlichen Fragebogen und wird freundlich dazu aufgefordert, die 1. Seite auszufüllen.
00:07 – 00:15	Einweisung: <ul style="list-style-type: none"> - Proband setzt sich an Schreibtisch, auf dem alle zur Evaluation nötigen Gegenstände bereit liegen. Vor ihm steht ein Monitor bereit. - Datenbrille vorstellen - Handgelenk-Marker vorstellen - Gestenerkennung und mögliche Interaktionsformen mit dem System erläutern: <ul style="list-style-type: none"> o Ansicht verschieben (pinch), o Ansicht vergrößern/verkleinern (zoom), o Ansicht drehen (rotate) - Helm mit Smartphone-Aufsatz vorstellen - Probanden darauf aufmerksam machen, dass bei eventuellem Unwohlsein während des Tragens von Brille oder Helm der Versuch sofort unterbrochen werden kann.
00:15 – 00:20	Versuchsaufbau Teil 1 – Evaluation der Interaktion: <ul style="list-style-type: none"> - Einen der beiden verfügbaren Handgelenk-Marker am Handgelenk der schwachen Hand des Probanden anbringen - Smartphone in vorgesehene Helmhalterung einsetzen - Smartphone mit Monitor verbinden und Versuchsprogramm auf Smartphone starten - Probanden selbst den Helm aufsetzen lassen, dabei helfend eingreifen und den Sitz des Helmes überprüfen - In Kommunikation mit dem Probanden den Smartphone-Aufsatz des Helms so ausrichten, dass der Proband die vor ihm befindliche Tischfläche durch das Display der Datenbrille sieht - Probanden dazu auffordern, seinen Unterarm mit dem Handgelenk-Marker in seinen aktuellen Sichtbereich zu halten; dabei eigens überprüfen, ob der Marker vom Smartphone erkannt und das Overlay bereits angezeigt wird, gegebenenfalls Ausrichtung korrigieren und falls nötig, mittels Eingabe am Smartphone den Auto-Fokus der Smartphone-Kamera manuell auslösen

00:20 – 00:25	<p>Einführung in die Aufgabenstellung - folgende Punkte sind wichtig:</p> <ul style="list-style-type: none"> - Aktuell sichtbar ist ein Kartenausschnitt der italienischen Stadt Venedig. - Innerhalb des Kartenausschnitts befinden sich drei orange markierte Kreise, welche nummeriert sind. - Diese Kreise stellen Routenwegpunkte dar. - Übergeordnetes Ziel ist es, eine mögliche Route von Wegpunkt 1 zu Wegpunkt 2 und von da aus zu Wegpunkt 3 (in dieser Reihenfolge) zu finden und die entsprechenden Navigationsbefehle (z.B. „an nächster Kreuzung von Straße X in Straße Y abbiegen) dabei laut auszusprechen. - Diese Aufgabe soll durch Manipulation der Kartenansicht unter Nutzung der erläuterten Interaktionsmöglichkeiten mittels der starken Hand des Probanden gelöst werden. - Die ausgesprochenen Navigationsbefehle des Probanden können derweil vom Versuchsleiter niedergeschrieben werden, wenn der Proband eine spätere Prüfung seiner Route wünscht. - Der Versuch endet, wenn der Proband die Navigation bis Wegpunkt 3 vollendet hat. - Nach Ende des Versuchs kann gemeinsam mit dem Probanden an einem Desktop-Rechner eine Online-Kartenansicht geladen und die Navigation anhand der aufgeschriebenen Befehle nachvollzogen werden. - Dabei kann geprüft werden, ob die Route des Probanden gültig und sinnvoll ist.
00:25 – 00:40	<p>Durchführung des 1. Versuchs:</p> <ul style="list-style-type: none"> - Durchführung nach zuvor erläuterter Aufgabenstellung - Nach Fund des Wegpunkt 2 kurz unterbrechen und den aktuell getragenen Marker mit dem jeweils anderen auswechseln, danach Versuch fortführen - Nach Ende des Versuchs Kopfhaltung abnehmen
00:40 – 00:45	<p>Persönlicher Fragebogen – Seite 2:</p> <ul style="list-style-type: none"> - Proband wird freundlich dazu aufgefordert, die 2. Seite des persönlichen Fragebogens auszufüllen.
00:45 – 00:50	<p>Versuchsaufbau Teil 2 – Evaluation des Gesamtsystems:</p> <ul style="list-style-type: none"> - Aktuell getragenen Handgelenk-Marker durch den favorisierten des Probanden ersetzen - Probanden selbst die Datenbrille aufsetzen lassen, eventuell helfend eingreifen - Smartphone in Helmhalterung mit Datenbrille verbinden und Versuchsprogramm auf Smartphone starten - Probanden erneut selbst den Helm aufsetzen lassen, dabei helfend eingreifen und den Sitz des Helms überprüfen - Shutter der Datenbrille schließen und Datenbrille einschalten - Probanden nochmals den Sitz von Datenbrille und Helm prüfen lassen; dabei sollte das Display der Datenbrille für den Probanden vollständig sichtbar sein - In Kommunikation mit dem Probanden den Smartphone-Aufsatz des Helms erneut so ausrichten, dass der Proband die vor ihm

	<p>befindliche Tischfläche durch das Display der Datenbrille sieht</p> <ul style="list-style-type: none"> - Probanden erneut dazu auffordern, seinen Unterarm mit dem Handgelenk-Marker in seinen aktuellen Sichtbereich zu halten; dabei eigens überprüfen, ob der Marker vom Smartphone erkannt und das Overlay bereits angezeigt wird (hinter Proband stellen und das Smartphone-Display beobachten), gegebenenfalls Ausrichtung korrigieren und falls nötig, mittels Eingabe am Smartphone den Auto-Fokus der Smartphone-Kamera manuell auslösen
00:50 – 01:00	<p>Durchführung des 2. Versuchs:</p> <ul style="list-style-type: none"> - Um die Prozedur abzukürzen, soll der Proband sich dieses Mal nur 1-2 Minuten mit dem System und der Interaktion auseinandersetzen, um ein Gefühl für das Gesamtsystem zu bekommen. - Nach Ende des Versuchs Kopfhaltung und Datenbrille abnehmen
01:00 – 01:10	<p>Persönlicher Fragebogen – Seite 3-5:</p> <ul style="list-style-type: none"> - Proband wird freundlich dazu aufgefordert, die restlichen Seiten des persönlichen Fragebogens auszufüllen. - Auch für die offenen Fragen gegen Ende des Fragebogens sollte sich zur Beantwortung Zeit genommen werden
01:10 – 01:15	<p>Danke:</p> <ul style="list-style-type: none"> - Als kleines Dankeschön für die Mühe des Probanden darf dieser einen Teil der dafür vorgesehenen Süßigkeiten mitnehmen.

C Auswertung der Evaluation – Rohdaten

Auswertung – Fragenblöcke 1-5

Fragenblock: Nr.	1					2					3.1					3.2					3.3					4					5.1					5.2					
	Geschlecht	Alter	Beruf	Brillenträger		(a)	(b)	(c)	(d)	(e)	(a)	(b)	(c)	(d)	(e)	(a)	(b)	(c)	(d)	(e)	(a)	(b)	(c)	(d)	(e)	(a)	(b)	(c)	(d)	(e)	(a)	(b)	(c)	(d)	(e)	(a)	(b)	(c)	(d)	(e)	
1	w	23	Studentin	nein		-2	-1	-2	1	-1	0	0	0	0	0	1	2	0	1	-1	2	0	0	0	-1	-1	-2	-1	-1	0	-2	-1	-1	0	-1	-1	0	1	0		
2	m	28	Programmierer, Student	nein		2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	w	24	Studentin	nein		0	1	-2	2	2	1	-1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
4	m	26	wiss. Mitarbeiter	nein		2	1	2	0	-1	0	-1	0	1	1	1	2	-2	-1	2	1	1	-1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	
5	m	25	Student	nein		2	2	2	2	0	1	0	-1	0	2	2	1	2	0	2	1	0	0	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
6	w	26	Studentin	nein		1	1	0	-1	0	-2	0	0	1	1	1	1	1	0	1	1	0	1	1	0	-1	0	1	2	-1	0	1	1	1	0	1	1	-1	0	1	2
7	m	26	Student	ja		2	2	2	2	-1	1	-2	-2	1	2	2	1	0	2	-1	-2	2	1	1	2	-2	2	1	1	2	-2	0	-1	-2	0	-1	-2	2	1	2	
8	m	24	Student	nein		2	2	1	2	1	1	0	-1	-2	0	1	1	0	2	2	1	0	0	-1	1	2	2	0	0	1	2	2	0	0	1	0	1	1	1	0	
9	m	22	Student	ja		2	2	0	2	1	0	-1	0	1	1	0	1	0	0	2	-1	1	0	-1	1	0	-1	1	2	-1	1	2	-1	2	-1	2	2	2	2		
10	m	22	Student	nein		1	2	-2	0	-1	0	-1	-1	0	2	1	1	2	-1	1	1	0	0	1	0	0	1	0	1	2	0	0	1	1	0	1	-1	0	1	1	
11	w	47	Physiotherapeutin	ja		0	1	-2	0	2	0	1	0	1	1	1	0	1	-1	-1	0	0	1	1	2	2	1	1	2	2	1	1	2	2	2	2	2	2	2	2	
12	m	56	Bankkaufmann	ja		1	1	-2	0	1	2	2	1	2	1	2	2	1	2	-1	-1	2	0	-1	0	2	0	2	2	0	1	2	2	1	2	2	2	2	2	2	
13	w	52	Bürokauffrau	ja		-2	-1	-2	-2	0	0	0	0	0	0	1	0	0	2	-1	2	0	1	1	-1	0	2	-1	0	1	1	0	0	2	-1	0	1	1	0	2	
14	w	24	Bürokauffrau	ja		0	0	-2	-2	1	0	1	-1	0	1	2	1	2	0	1	0	1	2	2	-2	1	1	2	-1	1	2	-1	0	0	-1	-1	0	2	2		

Auswertung – Fragenblock 6

6.a)

- Es ist innovativ und ausbaufähig.
- Cutting-Edge-Technologie
- Die Idee ist gut es mal in so eine Art umzusetzen.
- Ich brauche mein mobiles Gerät nicht aus der Hosentasche holen.
- Modische Accessoires könnten als Marker verwendet werden.
- Schritt in die nächste Stufe zukünftiger mobiler Systeme
- stabiles Tracking
- Anwendungsmöglichkeiten
- zukunftsorientiert
- intuitiv in Benutzung
- neue Interaktion
- Auf den Arm zu schauen, um mit einem virtuellen Bildschirm zu interagieren finde ich gut.
- Das Warten, bevor die Bedienung funktioniert, finde ich gut gelöst.
- die Integration von Informationen in den Sichtbereich, ohne explizit ein Gerät in der Hand halten zu müssen
- Es ist etwas Neues.
- Die Bedienung ist interessant.
- Zusätzliche nützliche Informationen im Sichtfeld
- Dass man beim Bedienen die Hände frei hat.

6.b)

- Das Anheben des Armes ist anstrengend und unbequem (Tragen wie eine Armbanduhr). Die zweite Variante ist (im Sitzen) angenehmer.
- Tracking der Gesten geht besser.
- Helm zu schwer
- Kamera der Brille schlechte Auflösung
- ... dass ich eine Kamera auf dem Kopf habe. Wer weiß, wer alles mit schaut.
- das fehlende haptische Feedback
- Ungenauigkeiten der Steuerung
- Anstrengung beim längeren Tragen
- in diesem Zustand wenig mobil
- Tragekomfort
- Tragekomfort
- nicht genau genug
- teilweise umständliche Bedienungsgesten

- Die Bedienung ist schwierig.
- Bei längerer Benutzung tut der Arm weh.
- Tragekomfort
- Genauigkeit der Erkennung
- Die Steuerung könnte besser sein.
- Kopf während der Bedienung muss fixiert sein.
- Helm mit Aufbau relativ schwer
- Der Helm mit Handy ist zu schwer.

6.c)

- die Verwendung mit mehreren Fingern
- die Schnelligkeit der Reaktion auf die Körperbewegungen
- die Tragevorrichtung sollte leichter sein
- „Einrasten“ der Funktionalität schneller machen
- mehr Leistung im Handy (ruckelfreier)
- bessere Auflösung beim Tracking
- Kamera der Brille
- besserer Tragekomfort -> Helm zu schwer
- Hand als Marker
- Robustere Gesten, z.B. aus dem Handgelenk, anstatt aus den Fingern: Den Vorteil nutzen, dass man nicht nur eine begrenzte Fläche zur Verfügung hat.
- visuelle Objekte, die über den Marker hinausragen
- Genauigkeit der Steuerung
- Verkleinerung des Kopfaufbaus
- Geschwindigkeit des Systems
- Tracking
- Gestensteuerung
- Erkennung
- gesamte Hardware
- Größeres Bild auf dem Arm, um auch, wenn der Arm weiter weg ist, das System komfortabel bedienen zu können
- Tragekomfort
- Reaktionszeiten
- Steuerung/Bedienung
- die Möglichkeit der Drehung
- Leichtere Handhabung (Gewichtsreduzierung)
- Reaktionszeit teilweise zu langsam
- Besserer Tragekomfort

6.d)

- Tragekomfort: 10x
- Auflösung: 6x

- Reaktionszeit: 5x
- Bedienung: 5x
- Sichtfeld: 2x
- Tracking: 1x

6.e)

- Maps
- Telefon
- Kontakte
- generell alle einfachen Sachen
- Tourismus
- unterwegs in unbekanntem Städten
- in allen Anwendungsbereichen eines Smartphones
- alle Bereiche, in denen heute Smartphones verwendet werden
- Simulationsanwendungen
- allgemeine Anwendung (Handy im Alltag)
- Spiele
- Navigation (bei Richtungsänderungen)
- Prüfen von Nachrichten/Notifications
- Steuern von Musikplayer
- Bedienung des Smartphones im Alltag
- Alltag
- medizinischer Bereich
- touristische Besucherführungen
- Vergleiche bei Reparaturen/Operationen richtig/falsch
- medizinischer Bereich

6.f)

- SMS, Spiele etc. gehören nicht zu den möglichen Anwendungen.
- Man hat bemerkt, dass die Kameraauflösung auf dem PC-Bildschirm wesentlich besser war.
- Sehr gute Idee. Es fehlt halt an Geld um Prototypen optimal zu gestalten.