



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

René Schramowski

GeoMeta - Augmented Reality auf dem Weg zum Stadtplan 2.0

*Fakultät Technik und Informatik
Studiendepartment Informatik*

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

René Schramowski

GeoMeta - Augmented Reality auf dem Weg zum Stadtplan 2.0

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

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

Betreuender Prüfer: Prof. Dr. Philipp Jenke
Zweitgutachter: Prof. Dr. Bernd Kahlbrandt

Eingereicht am: 12. Januar 2018

René Schramowski

Thema der Arbeit

GeoMeta - Augmented Reality auf dem Weg zum Stadtplan 2.0

Stichworte

Microsoft HoloLens, Augmented Reality, GeoMeta, Stadtkarte, Anwendungsfall

Kurzzusammenfassung

Diese Ausarbeitung beschäftigt sich mit der Erstellung eines Anwendungsfalls mit der von Microsoft entwickelten HoloLens im Bereich Augmented Reality. Hinter dem Namen GeoMeta steckt eine interaktive Stadtkarte, die 3D Modelle von Point of Interests in Hamburg als virtuelle Objekte mit der HoloLens darstellt, welche sich auf einem A0 Poster von Hamburg befinden. Je nach Distanz zwischen HoloLens und Poster wird ein anderer Detailgrad an Informationen zu Hamburg angezeigt. Zu den einzelnen 3D Modellen werden ebenfalls weiterführende Informationen eingeblendet.

René Schramowski

Title of the paper

GeoMeta - Augmented Reality on its way to city map 2.0

Keywords

Microsoft HoloLens, Augmented Reality, GeoMeta, City map, Use Case

Abstract

This thesis is about creating a Use Case for the HoloLens from Microsoft in the Augmented Reality section. GeoMeta realizes an interactive map of Hamburg which will display virtual 3D objects of points of interest with the HoloLens on a A0 poster of Hamburg. The distance between the HoloLens and the poster will determine the level of detail of displayed information about Hamburg. Every 3D object has their individual information displayed.

Inhaltsverzeichnis

1	Einführung	1
1.1	Motivation	1
1.2	Zielsetzung	3
1.3	Herausforderungen	3
1.4	Ausblick	4
2	Technische Grundlagen	5
2.1	Augmented Reality und die Abgrenzung zu Virtual Reality	5
2.1.1	Definition	5
2.1.2	Displayarten	7
2.1.2.1	Optisches see-through Displays	9
2.1.2.2	Video see-through Displays	9
2.1.2.3	Head-attached Displays	10
2.1.2.4	Hand-attached Displays	11
2.1.2.5	Spatial Displays	13
2.1.3	Historie	14
2.2	Microsoft HoloLens	19
2.2.1	Entwicklung der Brille über die Zeit	19
2.2.2	Aufbau und Funktionsweise der Developer Edition 2016	21
2.2.3	Zukunftspläne	25
2.3	Verwendete Softwarelösungen	26
2.3.1	Blender	26
2.3.2	Unity	26
2.3.3	Vuforia	27
2.3.4	Visual Studio	28
2.4	Related Work	28
3	Idee und Konzeption	30
3.1	Anforderungsanalyse	30
3.1.1	Funktionale Anforderungen	30
3.1.2	Nichtfunktionale Anforderungen	31
3.2	Analoge Stadtkarte	32
3.3	Trackingverfahren	33
3.3.1	Multi-Level Target	33
3.3.2	Single Target Reference	35
3.4	Evaluation der Trackingverfahren	36

4	Implementation GeoMeta	39
4.1	Erstellung der Stadtkarte von Hamburg	39
4.2	Projektaufbau	42
4.3	Modellierung in Unity	44
4.3.1	Marker und 3D Modelle	46
4.3.2	Detail-Mode	48
4.3.3	Distanzebenen	49
4.4	Scripts und Animationen	52
5	Evaluation der Umsetzung	55
5.1	Auswertung	55
5.2	Schwierigkeiten und Grenzen	59
6	Zusammenfassung	61
7	Ausblick	63
7.1	Weiterentwicklung GeoMeta	63
7.2	Die Zukunft von Augmented Reality	64

1 Einführung

1.1 Motivation

Der Bereich Virtual Reality hat durch die Präsenz im Verbrauchermarkt durch Produkte wie der Oculus Rift¹ oder der HTC Vive² den Bekanntheitsgrad drastisch erhöht. Über sogenannte Head-Mounted Displays, also Geräte, die am Kopf befestigt werden und Displays direkt vor den Augen der Anwender positionieren, wird das reale Umfeld abgedeckt, sodass nur noch der Inhalt der Displays zu sehen ist. Dabei wird über die Displays für gewöhnlich virtuelle Welten so dargestellt, dass es für den Anwender erscheint, als wäre dieser in einer anderen Welt. Wie stark dieses Gefühl ist, dass das, was der Anwender sieht, als real empfunden wird, wird als Immersion definiert. Über Eingabegeräte kann der Anwender, während dieser das Gerät trägt, mit der dargestellten virtuellen Welt interagieren. Viele Faktoren wie der generelle Realitätsgrad der dargestellten Elemente, der Möglichkeiten der Interaktion mit der virtuellen Welt oder auch das Sichtfeld sind Indikatoren für das Maß der Immersion.

In diesem Kontext gibt es einen weiteren Bereich der visuellen Darstellung von virtuellen Elementen, den Bereich Augmented Reality. Anders als bei Virtual Reality, wo das reale Umfeld abgeschirmt wird und jeglicher Inhalt computergeneriert ist, nutzt Augmented Reality das reale Umfeld als Basis und erweitert dieses mit virtuellen Elementen, um, worauf der Name schließen lässt, die Realität zu erweitern. Bis vor Kurzem war der Bekanntheitsgrad für diesen Bereich im Vergleich zur Virtual Reality nicht sehr hoch, da es keine speziell dafür gefertigten Geräte im Verbrauchermarkt gab. Dennoch hat die App *Pokémon GO* für Android und iOS den Bereich Augmented Reality bekannt gemacht. Über die Kamera der Smartphones wird dabei das reale Umfeld aufgenommen und die Pokémon auf dieses Bild eingeblendet, sodass es auf dem Smartphonedisplay so erscheint, als sei dieses real. Im August 2016 hatte die App ihren Höhepunkt erreicht und dabei bis zu 100 Millionen Spieler gleichzeitig aufgezeichnet³. Auch

¹<https://www.oculus.com/rift/> (zuletzt aufgerufen am 18.12.2017)

²<https://www.vive.com/de/> (zuletzt aufgerufen am 18.12.2017)

³<http://www.pcgames.de/Pokemon-GO-Spiel-56108/News/60-Millionen-Spieler-1232255/> (zuletzt aufgerufen am 18.12.2017)

andere bekannte Apps wie Snapchat⁴ verwenden Augmented Reality, um den Gesichtern der Anwender computergenerierte Elemente hinzuzufügen. Außerhalb des Verbrauchermarkts gibt es beispielsweise mit der Microsoft HoloLens ein Head-Mounted Display Gerät, welches speziell für die Darstellung und Interaktion mit Augmented Reality Inhalten entwickelt wurde. Dieses Gerät ist zunächst nur für Entwickler erhältlich und wird erst in Zukunft für Endkunden entwickelt.

Die erste und aktuellste Fassung der Microsoft HoloLens ist die Developer Edition 2016. Grundlegend ist dieses Gerät eine Brille, die es dem Anwender über durchsichtige Displays ermöglicht, das reale Umfeld wahrzunehmen. Außerdem können die Displays computergenerierte Objekte darstellen, die sich somit in das reale Umfeld einbinden. Da diese Displays sich direkt vor den Augen des Anwenders befinden, ist die Möglichkeit gegeben, dass unter Berücksichtigung mehrerer Kriterien wie beispielsweise der Qualität der Darstellung virtueller Objekte, der Einbindung dieser in das reale Umfeld und der möglichen Interaktion mit den Objekten eine hohe Immersion zu erreichen ist. Auch hier wird der Begriff Immersion generell dafür verwendet, wie real die gesamte Szene samt zusätzlichen virtuellen Elementen vom Anwender wahrgenommen wird. Da dieses Gerät vergleichsweise neu ist, eine kleine Verbreitung durch Verkauf an Entwickler statt dem Verbrauchermarkt vorliegt und sie viele Features birgt, die es ebenfalls nicht für den Verbrauchermarkt gibt, ist die Masse an implementierten Anwendungen für die HoloLens relativ klein. Somit bietet dieses Gerät die Möglichkeit, viele noch nicht umgesetzte Anwendungsfälle zu realisieren und zu überprüfen, inwiefern eine hohe Immersion erreichbar ist und ob dadurch ein tatsächlicher Nutzen generiert werden kann.

Im Rahmen dieser Ausarbeitung wird die Microsoft HoloLens in der Developer Edition 2016 verwendet, um mit Unity eine Anwendung zu kreieren, in der eine analoge Stadtkarte von Hamburg mit Augmented Reality Inhalten erweitert wird. Je nachdem wie groß die Distanz zwischen der Stadtkarte und dem Anwender mit aufgesetzter HoloLens ist, werden verschiedene Elemente dargestellt. Bei der größten implementierten Distanz werden allgemeine Informationen zu Hamburg dargestellt. Verringert sich diese Distanz, wird statt den allgemeinen Informationen die Namen der jeweiligen Bezirke von Hamburg auf der Stadtkarte selbst angezeigt. In der dichtesten implementierten Distanz werden Marker Objekte eingeblendet, die sich auf Positionen von Sehenswürdigkeiten befinden. Werden diese vom Anwender mit der Brille fokussiert, so erscheint der jeweilige Name und mittels einem AirTap der HoloLens

⁴<https://www.snapchat.com/l/de-de/> (zuletzt aufgerufen am 18.12.2017)

erscheint ein Miniaturmodell der Sehenswürdigkeit. Zuletzt kann ein AirTap auf das Modell ausgeführt werden, wodurch das Modell nahezu auf das zehnfache ihrer Miniaturgröße gesetzt wird und mit passenden Informationen versehen wird. Somit wird der Detailgrad der dargestellten Informationen für den Anwender immer detaillierter, je näher dieser an der Stadtkarte ist.

1.2 Zielsetzung

Für den Prototypen des Anwendungsfalls GeoMeta soll eine erste Iteration implementiert werden, die grundlegend zeigt, wie mit Hilfe der HoloLens eine analoge Stadtkarte erweitert werden kann, dass mehr Informationen für den Anwender generiert werden können, die allein aus der analogen Stadtkarte nicht zu entnehmen sind. Für einige Sehenswürdigkeiten Hamburgs sollen 3D Modelle einblendbar sein, die dem Anwender eine Visualisierung wichtiger Bauten Hamburgs ermöglichen soll. Diese sollen sich auf der Position der Stadtkarte befinden, auf der die jeweilige Sehenswürdigkeit in der Realität zu finden ist. Außerdem sollen zusätzlich zur Darstellung dieser 3D Modelle zusätzlich Informationen eingeblendet werden, damit der Anwender bei der Betrachtung mehr über diese Sehenswürdigkeit erfahren kann. So kann beispielsweise zum Modell der Elbphilharmonie eingeblendet werden, in welchen Zeitraum sie gebaut wurde, wie groß sie ist und was sie gekostet hat. Je nachdem wie nah der Anwender mit der aufgesetzten HoloLens an der Stadtkarte ist, sollen verschiedene Sichten angezeigt werden. So sollen auf der untersten Ebene die Modelle sichtbar sein, weiter entfernt sollen die Bezirke dargestellt werden, damit der Anwender die Stadtkarte besser eingrenzen kann und auf der höchsten Ebene sollen generelle Informationen zu Hamburg dargestellt werden, um allgemeine Aspekte rund um Hamburg abzudecken. Mit der HoloLens ist es möglich über Hand-Gestiken Aktionen auszuführen um so eine Interaktion zu ermöglichen. In diesem Anwendungsfall soll dieser Aspekt mit bedacht werden, damit der Anwender die Möglichkeit hat in dieser Anwendung über Gestiken bestimmte Elemente zu steuern und so grundlegend eine hohe Immersion zu gewährleisten.

1.3 Herausforderungen

Grundlegend stellt sich die Herausforderung der korrekten Positionierung und Skalierung der computergenerierten Objekte auf der analogen Stadtkarte. Wie in der Zielsetzung erläutert, sollen die 3D Modelle der Sehenswürdigkeiten auf der Position auf der Stadtkarte platziert werden, auf der sie sich in der Realität befinden. Dazu muss implementiert werden, dass die

Stadtkarte in irgendeiner Form erkannt wird und basierend auf der Erkennung dieser initiiert wird, dass die passenden Elemente eingeblendet werden sollen. Die Erkennung der Stadtkarte wird zugleich auch dazu verwendet werden müssen, dass auf Basis dieser auch die korrekte Position ermittelt wird. Die Modelle werden, da sie aus verschiedenen Quellen stammen werden, verschiedene Größen aufweisen. Damit das Gesamtbild bei der Darstellung über die HoloLens stimmig ist, müssen die 3D Modelle untereinander angepasst werden, damit das Größenverhältnis in etwa stimmt. Generell müssen die Modelle auch so skaliert werden, dass sie in das Bild mit der Stadtkarte als Basis passen. Würden die Modelle zu groß skaliert werden, so würde ein großer Teil der Stadtkarte bedeckt werden und beispielsweise auch andere Modelle überschneiden, was in Summe die Immersion immens verschlechtern würde. Eine weitere Herausforderung stellt die Darstellung der verschiedenen Ansichten dar. Zur Laufzeit muss gemessen werden, wie groß der Abstand von der vom Anwender aufgesetzten HoloLens zur Stadtkarte ist, damit je nach Entfernung die passende Sicht dargestellt wird. Dabei ist auch zu berücksichtigen, dass nicht jeder Anwender aufgrund dessen Größe die gleiche Entfernung zur Stadtkarte aufweisen wird. So muss sichergestellt werden, dass relativ zur Größe des Anwenders ein Wechsel der Ansichten stattfinden kann und nicht statisch anhand der generellen Entfernung ein Wechsel stattfindet.

1.4 Ausblick

Im Rahmen dieser Ausarbeitung wird der Reihenfolge nach zunächst ein Einblick in die dafür relevanten technischen Grundlagen gewährt. Darunter fallen eine explizite Definition von Augmented Reality und der Zusammenhang zur Virtual Reality. Des Weiteren werden die Displayarten für den Augmented Reality Anwendungsfall und eine Historie beschrieben, die zeigt, wie sich dieser Bereich seit den letzten 50 Jahren entwickelt hat und welche wichtigen Personen und Umsetzungen es in dieser Zeit gegeben hat. Da dieser Anwendungsfall für die Microsoft HoloLens entwickelt wurde, wird primär der Aufbau und die Funktionsweise des Gerätes erläutert. Neben der kurzen Erläuterung verwendeter Softwarelösungen, um diesen Anwendungsfall zu implementieren, werden Ideen und Konzeption erläutert, anhand der die Implementation detailliert beschrieben wird. Zuletzt wird in der Evaluation das Ergebnis besprochen sowie ein Ausblick darauf gewährt, was mit diesem Anwendungsfall alles möglich ist und was im Bereich Augmented Reality in Zukunft noch passieren könnte.

2 Technische Grundlagen

Zum Verständnis der in dieser Ausarbeitung genutzten Technologien werden in diesem Abschnitt die wesentlichen Grundkenntnisse vermittelt. Unter anderem werden die Begriffe Augmented Reality und Virtual Reality aufgearbeitet und speziell für Augmented Reality weitere Aspekte erläutert. Aufbauend darauf wird die Microsoft HoloLens beschrieben sowie die für den Anwendungsfall verwendeten Softwarelösungen vorgestellt.

2.1 Augmented Reality und die Abgrenzung zu Virtual Reality

Der Begriff Augmented Reality heißt übersetzt *Erweiterte Realität* und stellt somit zunächst einen großen Definitionsfreiraum dar. Prinzipiell gibt es viele Aspekte, die zur Erweiterung der Realität führen. Beispielsweise können Geräte genutzt werden, die über Sensoren Messungen vornehmen können, die wir als Menschen mit unseren Sinnen nicht wahrnehmen können. Die exakte Temperatur im Umfeld, die Zusammensetzung der Luft oder auch den Feinstaubanteil können wir als Menschen nicht ohne weitere Hilfsmittel erfassen. Mit solchen Geräten erweitern wir somit unsere Realität, indem wir zusätzliche Informationen erfassen können. Dabei variiert auch die Form der Darstellung der gesammelten Informationen der Geräte. Diese können über Akustik wiedergegeben werden oder auch über Displays angezeigt werden. In dieser Ausarbeitung wird das Thema Augmented Reality über den visuellen Aspekt verwendet.

2.1.1 Definition

Die Technologie Augmented Reality lässt sich anhand des sogenannten *Realitäts-Virtualitäts-Kontinuum* definieren. Wie anhand der Abbildung 2.1 zu entnehmen, beinhaltet das RV Spektrum die beiden Endpunkte reale Umgebung für den Bereich der Augmented Reality und virtuelle Umgebung für Virtual Reality. In einer Virtual Reality Umgebung befindet sich der Anwender in einer komplett virtuellen Welt, die frei gestaltet werden kann. In dieser Umgebung ist es möglich selbst zu entscheiden, welche Eigenschaften aus der realen Welt einbezogen werden sollen. So können unter anderem die physikalischen Gesetze wie die

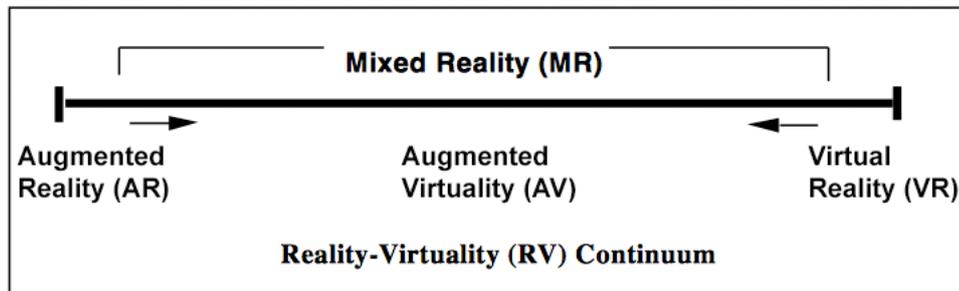


Abbildung 2.1: Das 'Realitäts-Virtualitäts-Kontinuum' [Milgram u. a.]

Schwerkraft außer Kraft gesetzt werden. Außerdem kann die Basis der virtuellen Welt aufgrund realer Gegebenheiten realisiert werden, aber auch aus rein fiktiven Objekten bestehen. Zeit und physische Grenzen können ebenfalls je nach Implementation variieren. Im Gegensatz dazu steht in einer Augmented Reality Umgebung das reale Umfeld im Vordergrund, welches über computergenerierte Objekte erweitert wird. Da hier das reale Umfeld als Basis genutzt wird, ist die freie Gestaltung darauf beschränkt, was zusätzlich dargestellt wird. Die physikalischen Gesetze wie die Schwerkraft sind somit vorgegeben, müssen aber nicht auf die hinzugefügten Objekte zutreffen. Zwischen diesen Endpunkten befindet sich der Mixed Reality Ansatz. Dieser ist eine Mischung beider Konzepte und hat je nach Anteil der Virtualität eine Gewichtung, die eher zu Augmented Reality oder zur Virtual Reality tendiert.

In diesem Themenbereich sind die Aspekte Immersion und Präsenz relevant, damit eine Anwendung im Augmented und Virtual Reality Bereich positiv vom Anwender aufgenommen wird. Das heißt speziell, dass in der Virtual Reality der Anwender das Gefühl hat, ein Teil der virtuellen Welt zu sein und in der Augmented Reality die hinzugefügten Elemente sich als Teil der realen Welt einbinden. Dabei unterscheiden sich die Bedingungen und Ziele von der jeweiligen Technologie. Im Themenfeld Virtual Reality bezeichnet Immersion den objektiven Teil, der dafür verantwortlich ist, dass der Anwender das Gefühl hat, er sei Teil der virtuellen Welt. In einem Artikel von [Bowman und McMahan] werden einige messbare Eigenschaften genannt, die für die Immersion hergezogen werden. Dazu gehören unter anderem die Punkte FOV¹, Displaygröße, Displayauflösung, Bildrate und weitere. Diese Faktoren stellen prinzipiell die Basis für den möglichen Realitätsgrad dar. All diese Eigenschaften ermöglichen es eine virtuelle Welt zu erschaffen, die der realen Welt auf visueller Ebene sehr ähnelt. Entgegen dessen steht die Präsenz, die ein subjektives Level der Akzeptanz der virtuellen Welt darstellt. Menschen haben unterschiedliche Aspekte, die einem wichtig sind, damit etwas als real

¹Field of view / Sichtfeld

erscheint. Das kann sich in verschiedene Richtungen ausprägen wie grundlegende physikalische Eigenschaften oder Bezug auf reale Objekte, Personen oder Ereignisse. Dieser ist nicht messbar, da das von Person zu Person unterschiedlich ist. Die gleiche Virtual Reality Umgebung kann bei verschiedenen Personen eine andere Präsenz ergeben. Zusammen bilden Immersion und Präsenz, also der objektive und subjektive Anteil, den Grad der Akzeptanz und somit das Gefühl, dass der Anwender ein Teil der virtuellen Welt ist.

Betrachtet man diese Aspekte für den Bereich der Augmented Reality, so gibt es keine eindeutige Zuordnung der Begrifflichkeiten. In einem Paper von *William Steptoe, Simon Julier* und *Anthony Steed* [Steptoe u. a.] wird Präsenz in einem ähnlichen Kontext behandelt. Dort wurden virtuelle neben realen Gegenständen platziert und anhand von Umfragen ermittelt, wieviele Personen zwischen beiden unterscheiden konnten. Daraus ergab sich, dass für die gleiche Szene eine unterschiedliche Anzahl an realen und virtuellen Objekten angegeben wurde. Somit definiert sich auch in diesem Kontext die Präsenz als die subjektive Wahrnehmung, wie real die erweiterte Welt erscheint. Der Begriff Immersion wird in diesem Zusammenhang nicht wie in der Virtual Reality explizit für die objektiven Eigenschaften genutzt. In Augmented Reality dient dieser als Gesamtwahrnehmung des Anwenders, also dem was Immersion und Präsenz in der Virtual Reality in der Summe ergeben. Dadurch lässt sich jedoch ableiten, dass unter dem Begriff der Immersion auch objektive Eigenschaften fallen. Fast dieselben objektiven Eigenschaften wie bei der Virtual Reality sind auch hier relevant. FOV, Auflösung der hinzugefügten Objekte, Bildrate und weitere Punkte können gemessen werden und steigern die Gesamtwahrnehmung. Generell wird der Begriff Immersion gekapselt von der jeweiligen Definition für die Technologien im *Realitäts-Virtualitäts-Kontinuum* als Begriff für die Güte genutzt, in der Virtual Reality also wie sehr sich der Anwender als Teil der virtuellen Welt fühlt und in der Augmented Reality wie sehr sich die hinzugefügten Objekte in die reale Welt integrieren. Auch in dieser Ausarbeitung wird der Begriff Immersion in Bezug auf Augmented Reality für diesen Aspekt genutzt.

2.1.2 Displayarten

Es gibt mehrere Varianten, wie sich die Augmented Reality Inhalte darstellen lassen können. Zudem unterscheidet sich grundlegend, in welcher Form die Realität wahrgenommen wird, die als Basis der Augmentierung verwendet wird. Folgend wird dieses Thema auf der Basis von [Milgram u. a.] [Bimber und Raskar] [Doerner u. a.] [Feng Zhou u. a.] zusammengefasst. Grundlegend lassen sich die Displayarten basierend auf unterschiedlichem Fokus kategorisieren.

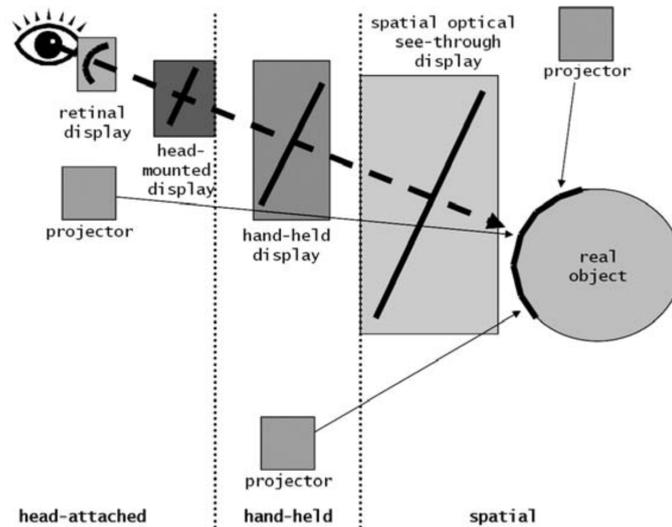


Abbildung 2.2: Ausprägungen der Darstellung von Augmented Reality Inhalten [Bimber und Raskar]

Wie Abbildung 2.2 zu entnehmen ist, ist eine Kategorisierung die Unterscheidung in head-attached, hand-held und spatial. Somit ergeben sich Varianten von Displays, die entweder am Kopf befestigt und direkt vor den Augen des Anwenders positioniert sind, man spricht auch von Head-Mounted Displays, oder in der Hand gehalten werden, diese werden typischerweise durch Smartphones oder Tablets umgesetzt, oder der projektionsbasierten Darstellung, bei der sich das Display vor dem Objekt, auf dem das computergenerierte Element angezeigt wird, befindet und somit losgekoppelt vom Anwender ist. Grundsätzlich ist das Ziel, dass ein Bild, das zur Erweiterung der Realität genutzt werden soll, im Sichtfeld des Anwenders dargestellt wird. Dabei kann das Bild, je nachdem ob es sich um eine head-attached, hand-held oder spatial Umsetzung handelt, irgendwo zwischen den Augen des Anwenders und dem physischen Objekt generiert werden, welches erweitert werden soll. Eine weitere Kategorisierung ist die Unterscheidung in optisches see-through, bei der der Anwender das reale Umfeld über die eigenen Augen wahrnimmt und die zusätzlichen Objekte dazu eingeblendet werden und video see-through, wo das reale Umfeld über eine Kamera aufgenommen und auf einem Display dargestellt wird und auf dieser Basis die zusätzlichen Objekte eingeblendet werden.

2.1.2.1 Optisches see-through Displays

In diesem Ansatz werden die Displays so realisiert, dass es möglich ist durch sie hindurch zu schauen, um das reale Umfeld wahrzunehmen. Dazu sind die Displays in der Lage, computergenerierte Objekte darzustellen, sodass der Anwender beides gleichzeitig sehen kann: Das reale Umfeld mit den eigenen Augen und die hinzugefügten Elemente auf dem Display, die somit in einem gemeinsamen Kontext behandelt werden. Es entstehen hohe Anforderungen an diese Form von Displays, um eine hohe Immersion zu ermöglichen, da das reale Umfeld vom Anwender selbst wahrgenommen wird und die zusätzlichen Elemente direkt im Vergleich zur Realität stehen. Somit müssen sich die zusätzlichen Elemente über gewisse Kriterien anpassen. Eine hohe Genauigkeit und Präzision was die Positionierung der Elemente angeht ist wichtig, damit der Anwender das Gefühl bekommt, dass es sich um ein natives Objekt handelt, das sich an das Umfeld anpasst. Dazu gehört, dass sich die Darstellung der Objekte ändert, wenn sich beispielsweise der Anwender bewegt und das Objekt aus einer anderen Perspektive begutachtet. Hohe Genauigkeit und Präzision ist auch nötig, wenn es um die Erfassung der Position vom Anwender und den dazugehörigen Komponenten wie Kopf und Körper geht. Erst wenn beides zuverlässig funktioniert, ist es grundlegend möglich eine hohe Immersion zu realisieren. Außerdem ist der sogenannte *occlusion effect* eine Herausforderung und von hoher Bedeutung, wenn es um die Immersion geht [Bimber und Frohlich] [Murase u. a.]. Dabei geht es um die richtige Darstellung der virtuellen Objekte in Relation zum realen Umfeld. Ohne weitere Maßnahmen, würden die virtuellen Objekte sich mit den realen Objekten vermischen, sodass es beispielsweise möglich ist, das reale Objekt durch das virtuelle Objekt zu sehen. Das richtige Verhalten dabei wäre, dass basierend auf der Position des virtuellen Objekts in Relation zu den realen Objekten, das virtuelle Objekt entweder vollständig zu sehen ist und die realen Objekte dahinter nicht mehr oder Teile des virtuellen Objekts nicht sichtbar sind, sofern sich diese hinter einem realen Objekt befinden. Weitere Aspekte zur Steigerung der Immersion, die man teilweise bereits in der Definition erwähnt hat, sind unter anderem ein hohes FOV, eine geringe Latenz und eine hohe Auflösung.

2.1.2.2 Video see-through Displays

Wesentlicher Unterschied zum optischen see-through ist die Wahrnehmung des Umfelds. Während beim anderen Ansatz der Anwender das reale Umfeld selber über die Augen wahrnimmt, so wird sie hier über eine Kamera aufgenommen und auf einem Bildschirm dargestellt. Diese Variante wird auch mit *window-on-the-world* betitelt. Vorteil dieses Ansatzes ist es, dass das Tracking in diesem Kontext eine unwesentlichere Rolle einnimmt und somit eine

Vereinfachung der Realisierung darstellt. Das liegt daran, dass für die Darstellung der Objekte zur Erweiterung der Realität das von der Kamera aufgenommene Bild verwendet werden kann. Aufgrund dieser Bilder kann pixelbasiert entschieden werden, wo und wie ein Objekt integriert werden muss, damit eine hohe Immersion ermöglicht werden kann. Auch das Problem des *occlusion effects*, welches für den optischen see-through eine Herausforderung darstellt, kann hier einfacher realisiert werden, da dies ebenfalls pixelbasiert bestimmt werden kann. Dennoch bietet dieser Ansatz grundsätzlich eine geringere Immersion, da das reale Umfeld nicht vom Anwender selbst wahrgenommen wird. Auch wenn das Bild der Kamera der Umgebung entspricht, nimmt der Anwender dies eher als eine virtuelle Welt an, da das von der Kamera resultierende Bild nicht so real wirkt wie das, was das menschliche Auge wahrnimmt.

2.1.2.3 Head-attached Displays

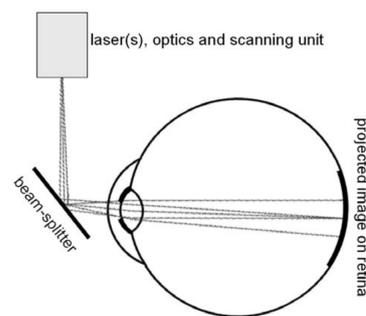


Abbildung 2.3: Basisaufbau eines Retinal Displays [Bimber und Raskar]

Wie bereits erläutert, müssen diese Art Displays am Kopf des Anwenders befestigt werden und positionieren sich direkt vor den Augen des Anwenders. Granularer betrachtet gibt es auch hier noch die weitere Unterteilung. Zunächst gibt es die Retinal Displays, die anders als die anderen Varianten keinen Bildschirm verwendet, auf dem die computergenerierten Objekte dargestellt werden. Über eine Laserdiode, heutzutage über beispielsweise einer Laserlichtquelle, werden diese Objekte direkt auf die Netzhaut des Anwenders projiziert. Das ermöglicht tendenziell eine bessere Darstellung der Objekte hinsichtlich Auflösung und einem größeren FOV. Dennoch können so zumindest bisher keine stereoskopischen Objekte dargestellt werden. Außerdem ist der Fokus nicht variabel, sodass der Anwender selbst auf das Objekt fokussieren muss, um es scharf zu sehen.

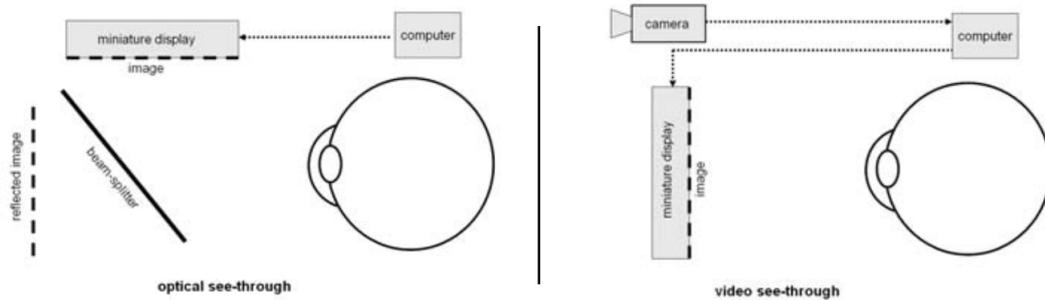


Abbildung 2.4: Optisches und video see-through Varianten für Head-Mounted Display
[Bimber und Raskar]

Eine weitere Variante stellen die Head-Mounted Displays dar. Im optischen see-through Ansatz bestehen die Displays aus halbdurchlässigen Spiegeln, die es zum Einen ermöglichen, dass der Anwender das reale Umfeld wahrnehmen kann und zum Anderen, dass auf dem Spiegel das darzustellende Objekt angezeigt wird, welches über kleine Monitore außerhalb des Sichtbereiches des Anwenders realisiert wird. Das Bild der Monitore spiegelt sich somit in den halbdurchlässigen Spiegeln und ist für den Anwender sichtbar. Ein weiterer Ansatz für den optischen see-through, ist eine projektorbasierte Lösung. Dafür wird statt den Monitoren, die an den Displays gespiegelt und für den Anwender sichtbar gemacht werden, Projektoren platziert, die das Bild über die halbdurchlässigen Spiegel in das Sichtfeld des Anwenders projizieren. Beim video see-through befindet sich vor den Augen des Anwenders ein Bildschirm, der nicht durchsichtig ist. Mit einer Kamera, die frontal am Gerät befestigt ist, die reale Umgebung aufgenommen und auf dem Display angezeigt. Je nach verwendetem Ansatz entstehen spezifische Herausforderungen. Die Auflösung ist an dem Bildschirm gebunden. Somit ist im video see-through die ganze Szene davon betroffen, beim optischen see-through nur die darzustellenden Objekte. Im optischen see-through ist das Tracking der Position des Kopfes elementar, damit es möglich ist die darzustellenden Objekte passend zum realen Umfeld anzuzeigen. Im video see-through ist das prinzipiell einfacher, da das reale Umfeld über eine Kamera aufgenommen wird und die Basis bildet, auf der die Objekte dargestellt werden. Daher kann in diesem Ansatz ein pixelbasiertes Verfahren herausfinden, wie ein Objekt sich in die Szene integrieren kann.

2.1.2.4 Hand-attached Displays

Die bekannteste Variante von hand-attached Displays sind Smartphones. Sie stellen die video see-through Form dar. Über die Kamera des Smartphones wird das reale Umfeld aufge-

nommen und über dem Display dargestellt. Auf dieser Basis kann wie bereits bei anderen video see-through Varianten pixelbasiert die computergenerierten Objekte eingebunden werden. Auch Tablets, die über eine Kamera verfügen, können dazu genutzt werden und bieten dank dem größeren Display auch eine größere Darstellung der Szene auf dem Display. Die Smartphone Variante hat den Vorteil, dass die Nutzung von Smartphones heutzutage einen sehr hohen Anteil hat. Laut einer Statistik auf Statista² nutzen im Jahre 2017 ungefähr 2,32 Milliarden Menschen auf der Welt ein Smartphone. 2020 sollen es sogar um die 2,87 Milliarden Menschen sein. Somit bieten Smartphones die Möglichkeit, dass Augmented Reality Anwendungen genutzt werden können, ohne dass dafür ein zusätzliches Gerät verwendet werden muss.



Abbildung 2.5: Beispielansicht der Applikation *Pokémon GO* - <https://www.gamespot.com/articles/new-pokemon-go-update-arrives-heres-what-it-does/1100-6448876/> (zuletzt aufgerufen am 18.12.2017)

Für einen großen Bekanntheitsgrad des Themas Augmented Reality, hat die Applikation *Pokémon GO*³ für Smartphones gesorgt. Diese Applikation nutzt den video see-through Ansatz um mit der Kamera das reale Umfeld aufzunehmen und platziert basierend darauf die Pokémon. Es gibt auch optische see-through Displays, die hand-attached sind. Ein Beispiel stellt die *Magic Lense* dar [Brown und Hua]. Der Form her ähnelt sie einem Handspiegel und besitzt statt dem Spiegel ein durchsichtiges Display. Je nach Anwendungsfall werden dann basierend

²<https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/> (zuletzt aufgerufen am 18.12.2017)

³<https://www.pokemongo.com/de-de/> (zuletzt aufgerufen am 18.12.2017)

auf dem, was sich unterhalb der *Magic Lense* befindet, die computergenerierten Objekte angezeigt. Wie bereits mehrfach erwähnt, ist die Immersion maßgeblich für die Qualität und der Akzeptanz der Augmented Reality Anwendung. Unter anderem ist der FOV dafür verantwortlich, wie hoch die Immersion ist. Bei der hand-attached Variante wie dem Smartphone ist die Displaygröße klein und ermöglicht somit einen kleineren FOV als beispielsweise der head-attached Modellen, da die hand-attached Geräte auch eine gewisse Entfernung zu den Augen des Anwenders aufweisen. Dennoch profitiert diese Variante von dem sogenannten *Parks Effect* [Parks]. Für den Anwender scheint das effektiv genutzte Display größer, wenn er das Display selbst bewegen kann, um sich die Szene, die sich auf das Kamerabild legt, anschauen zu können. Würde das Display starr sein und die Szene würde sich bewegen, so würde dem Anwender die tatsächlich kleine Größe des hand-attached Displays auffallen. Ohne speziellem Setup, wo das hand-attached Gerät in irgendeiner Form vor dem Anwender platziert wird, bietet sie kein hands-free Arbeiten wie bei den head-attached Varianten.

2.1.2.5 Spatial Displays

Spatial Displays definieren räumliche Displays, also solche, die überwiegend losgekoppelt vom Anwender realisiert werden. Somit ist es grundsätzlich möglich, je nachdem welche Art von Spatial Displays genutzt wird, Augmented Reality Inhalte wahrzunehmen, ohne dafür selber als Anwender irgendeine Art Technologie zu tragen oder zu benutzen. Hier kann ebenfalls in den Varianten optisches oder video see-through unterschieden werden und auch eine projektionsbasierte Variante ist möglich. Für diesen Bereich gibt es eine Reihe an Implementationsmöglichkeiten. Beispielsweise kann ganz rudimentär ein Fernseher dazu verwendet werden, ein reales Umfeld darzustellen, welches von einer Kamera aufgenommen wird und stellt dort die computergenerierten Objekte dar. Bereits zum heutigen Zeitpunkt wird vieles, was in Videos gezeigt wird, digital bearbeitet, sodass dieser Effekt kaum eine Immersion vorweist, da der Anwender hier im Normalfall, ohne einen subjektiven Bezug auf den dargestellten Inhalt zu haben, nicht feststellen kann, dass dort Augmented Reality Inhalte dargestellt werden. Eine andere Form wäre die Projektion von computergenerierten Inhalten auf eine Wand, um einem Betrachter erweiterte Inhalte bereitzustellen. Diese beiden Realisierungen zeigen bereits, dass hier die Interaktion zwischen Anwender und Augmented Reality Inhalten fehlt. Außerdem sind diese Lösungen räumlich gebunden und bieten nativ keine Möglichkeit einer Mobilität an.

2.1.3 Historie

Das Thema Augmented Reality reicht zum heutigen Zeitpunkt knapp 50 Jahre zurück. Bereits 1968 hat Ivan E. Sutherland ein Head-Mounted Display kreiert, welches dem Anwender ein perspektivisches Bild auf das Display anzeigt. Bei einem perspektivischen Bild handelt es sich um ein sich stets dem Blinkwinkel des Anwenders anpassendes Bild, um somit das Gefühl einer realen Betrachtung eines Objekts zu ermöglichen.



Abbildung 2.6: Ivan E. Sutherlands Augmented Reality Headset - <https://www.linkedin.com/pulse/augmenting-reality-phenomenon-pok%C3%A9mon-go-what-means-ar-chloe-price> (zuletzt aufgerufen am 18.12.2017)

Unter anderem besteht das System, wie aus Abbildung 2.6 zu entnehmen ist, aus einer an der Decke befestigten Vorrichtung, die physisch mit dem Headset verbunden ist und die Position des Kopfes misst, damit das perspektivische Bild passend zur Position dargestellt werden kann. Außerdem befinden sich zwei Kathodenstrahlröhren am Headset, die für die Darstellung des Bildes auf dem Display verantwortlich sind. Die Displays bestehen aus halbdurchlässigen Spiegeln, die es ermöglichen, dass gleichzeitig das von den Kathodenstrahlröhren generierte Bild und die reale Umwelt selbst für den Anwender sichtbar sind. Diese Darstellung stellt somit das Grundprinzip der visuellen Augmented Reality dar, wie sie auch heute verwendet wird.

In der Mitte der 1970er Jahre hat Myron Krueger das *Videoplace* ins Leben gerufen [Kalaw-sky]. Mithilfe von Projektoren, Kameras und speziell für diesen Anwendungsfall präparierte Hardware, siehe Abbildung 2.7a, wird die Silhouette des Anwenders an der Wand dargestellt



Abbildung 2.7: *Videoplace* von Myron Krueger - <http://thedigitalage.pbworks.com/w/page/22039083/Myron%20Krueger> (zuletzt aufgerufen am 18.12.2017)

und um diese herum können entweder computergenerierte Inhalte oder eine leere Fläche dargestellt werden. Je nachdem wie das System konfiguriert ist, kann mit verschiedenen Körperteilen wie einem Finger oder gar dem ganzen Körper mit der um der Silhouette befindlichen Umgebung interagiert werden. Ein Anwendungsfall ist es, wie der Abbildung zu entnehmen ist, dass auf der freien Fläche mit dem Finger in der Luft gezeichnet wird, wie in Abbildung 2.7b zu sehen ist. Ein Beispiel mit dem Körper und wie das ganze in Aktion aussieht, kann man aus Abbildung 2.7c entnehmen.

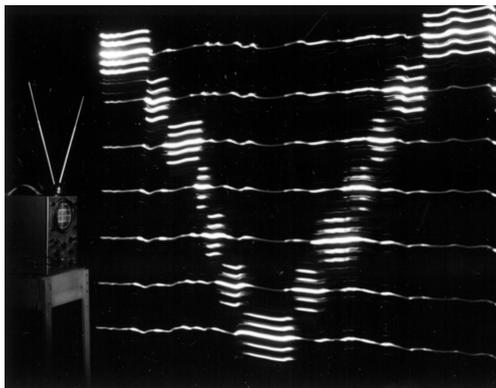
Eine weitere wichtige Person, die den Bereich Augmented Reality vorangebracht hat, ist Steve Mann⁴. Im generellen Kontext der Augmented Reality, also der Erweiterung der Realität in jeglicher Form, hat Mann im Jahre 1974 im Alter von 12 Jahren die sogenannte *Sequential Wave Imprinting Machine* von *The WearTech* gebaut⁵.

Dafür hat Mann eine Stange aus Holz voll mit Glühbirnen besetzt, die über einen tragbaren Verstärker und einem Switch angesteuert worden, womit diese Stange mobil war. Die Radiowellen wurden an der betreffenden Stelle an der Stange gemessen und die Glühbirnen, die sich ebenso an dieser Stelle befanden wurden erleuchtet. Indem Mann diese Stange in der Luft hin- und herschwing, ergaben sich durch das Aufleuchten der jeweiligen Glühbirnen das Bild der Radiowellen. Mit einer Kamera aufgenommen und einigen Anpassungen war es möglich eine Darstellung zu erzeugen, die dem aus Abbildung 2.8a entspricht.

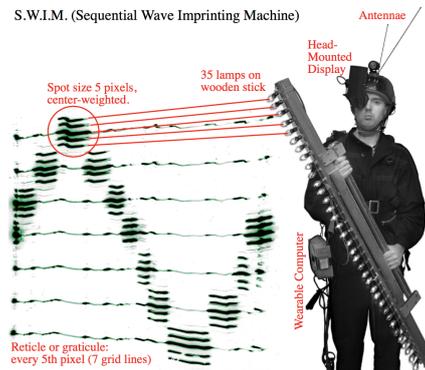
Im Jahre 1978 gab es das erste Gerät, welches als *wearable computer* betitelt und von Mann konzeptioniert und realisiert wurde. Anders als bei der Variante von Sutherland, bei der der

⁴<http://techland.time.com/2012/11/02/eye-am-a-camera-surveillance-and-sousveillance-in-the-glassage/> (zuletzt aufgerufen am 18.12.2017)

⁵<http://wearcam.org/surveillancestudy-36exposures/> (zuletzt aufgerufen am 18.12.2017)



(a) Ein Radio und dessen Radiowellen



(b) Visualisierung der Funktionsweise

Abbildung 2.8: *Sequential Wave Imprinting Machine*, kurz *S.W.I.M.* - <http://www.wearcam.org/ece516/> (zuletzt aufgerufen am 18.12.2017)

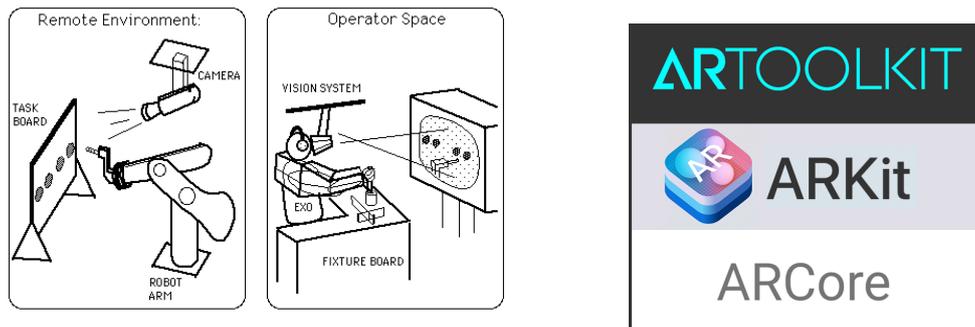
Anwender nur einen kleinen Bewegungsradius dank der an der Decke und dem Headset verbundenen Vorrichtung hatte, befindet sich der am Headset verbundene Computer am Körper des Anwenders und ermöglicht eine freie Bewegung. Für diese Erfindung gibt es keinen einheitlichen Namen. Verwendet wird *Digital Eye Glass*, *Eye Glass*, *Glass Eye* oder auch kurz nur *Glass*. Sie war eines der ersten Modelle, welche Mann im Laufe der Zeit entwickelt hat, die alle eine Variation eines Head-Mounted Displays sind.



Abbildung 2.9: Steve Mann's Augmented Reality Geräte - <https://commons.wikimedia.org/wiki/File:Wearcompevolution.jpg> (zuletzt aufgerufen am 18.12.2017)

Im ersten Bild, welches in [Abbildung 2.9](#) zu sehen ist, handelt es sich um die *Digital Eye Glass*. Wie Steve Mann selber sagte, bewirkt das Gerät, dass das menschliche Auge effektiv gleichzeitig zu einer Kamera und einem Display wird. Über die Zeit hat sich der Name *EyeTap*

durchgesetzt und Mann hat eine gleichnamige Organisation gegründet, die sich um die Weiterentwicklung und die Bewerbung von EyeTap verbundener Technologie kümmert. In ihrer aktuellsten Version besteht sie nur noch aus einem Headset, die vor dem Anwender ein Monokel ähnliches durchsichtiges Display positioniert, auf dem computergenerierte Objekte dargestellt werden können, wie im ganz rechten Bild in Abbildung 2.9 zu sehen ist.



(a) Aufbau der Augmented Reality für die US Air Force von L.B. Rosenberg [Leifer u. a.]

(b) Aktuelle Software Development Kits

Abbildung 2.10: Über Rosenberg zu aktuellen Software Development Kits

Im weiteren Verlauf wurde Augmented Reality für viele Bereiche in einer Form genutzt, die von den bisher entwickelten Varianten inspiriert wurden. Unter anderem hat L.B. Rosenberg für die US Air Force im Jahre 1992 ein System entwickelt [Rosenberg], welches es ermöglicht, dass der Anwender einen Roboter fernsteuern kann, um Aufgaben in einem entfernten Umfeld auszuüben. Wie in Abbildung 2.10a zu sehen, befindet sich der Anwender mit dem Oberkörper in einem Exoskelett um den Roboter zu steuern und hat ein Head-Mounted Display aufgesetzt, wodurch er die entfernte Umgebung wahrnehmen kann. Erweitert wird diese Darstellung durch im Display angezeigte Objekte, die dem Anwender helfen soll die Aufgabe besser durchzuführen. 1999 wurde das Open Source Software Developer Kit AR-ToolKit von Hirokazu Kato veröffentlicht⁶. Dies ermöglicht es, mithilfe von Video Tracking computergenerierte Objekte auf dem Bild einer Kamera zu platzieren. Es ist auch heute noch durch die Erweiterung von vielen Entwicklern eine Library, die in aktuellen Umgebungen wie Smartphone Apps genutzt werden kann⁷. Weitere Software Development Kits im aktuellen Zeitraum stellen ARKit⁸ von Apple für iOS und ARCore⁹ von Google für Android dar. Diese systemspezifischen Lösungen nutzen die Kameras der mobilen Geräte wie Smartphones, um

⁶<https://en.wikipedia.org/w/index.php?title=ARToolKit&oldid=787438905> (zuletzt aufgerufen am 18.12.2017)

⁷<https://www.artoolkit.org/> (zuletzt aufgerufen am 18.12.2017)

⁸<https://developer.apple.com/arkit/> (zuletzt aufgerufen am 18.12.2017)

⁹<https://developers.google.com/ar/> (zuletzt aufgerufen am 18.12.2017)

Augmented Reality Inhalte auf dem Display darzustellen. Kernfeature der beiden Lösungen sind zum Einen die maßstabsgetreue Darstellung der computergenerierten Objekte und zum Anderen die fixe Positionierung dieser Objekte an der realen Stelle im Bild der Kamera.



Abbildung 2.11: Aktuelle Augmented Reality Head-Mounted Display Geräte

Aktuelle Augmented Reality Head-Mounted Display Geräte sind in Abbildung 2.11 zu sehen, dabei sind diese namentlich von links nach rechts, von oben nach unten: Die Google Glass, entwickelt von X, vorher Google X; Magic Leap, Meta 2 und die Microsoft HoloLens. Die Google Glass wurde bereits 2014 in limitierter Anzahl in den USA an großen Firmen verkauft¹⁰. Im selben Jahr wurde diese limitierte Auflage aufgebraucht und es gab auch ein Produktionsstop. Nun im Jahre 2017 wurde die Produktion wieder gestartet und für weitere große Kunden gibt es die Möglichkeit diese Brille zu erhalten. Über die Magic Leap lässt sich von der offiziellen Seite nicht viel entnehmen¹¹. Es gibt weder offizielle Prototypen noch bestätigte Live-Vorführungen. Dennoch ist diese Firma nicht zu unterschätzen: Bis heute wurden fast 1,9 Milliarden Dollar in Magic Leap investiert¹². Die Meta 2 ist ein bereits frei erhältliches Gerät. Sie bietet unter anderem ein 90 Grad FOV, 2.5k Auflösung bei 60Hz und

¹⁰<https://blog.x.company/a-new-chapter-for-glass-c7875d40bf24> (zuletzt aufgerufen am 18.12.2017)

¹¹<https://www.magicleap.com/> (zuletzt aufgerufen am 18.12.2017)

¹²https://www.crunchbase.com/organization/magic-leap/investors/investors_list (zuletzt aufgerufen am 18.12.2017)

bietet Hände Tracking sowie Position Tracking Sensoren. Zuletzt die Microsoft HoloLens, die es momentan als Developer Edition für Entwickler zu kaufen gibt.

2.2 Microsoft HoloLens

Bei der Microsoft HoloLens handelt es sich um ein optisches see-through Head-Mounted Display. Für den Anwendungsfall GeoMeta wird sie als Gerät verwendet, um die Augmented Reality Inhalte darzustellen und mit ihnen zu interagieren. Basierend darauf wird in diesem Kapitel neben generellen Informationen zu diesem Gerät der Aufbau, die Funktionsweise und ein Ausblick auf die Weiterentwicklung behandelt.

2.2.1 Entwicklung der Brille über die Zeit

Am 21. Januar 2015 wurde ein Windows 10 Event von Microsoft veranstaltet¹³. Im Rahmen dieses Events wurde erstmalig das Betriebssystem *Windows Holographic* vorgestellt, welche auf der ebenfalls dort erstmalig erwähnten Microsoft HoloLens laufen wird. Zu diesem Zeitpunkt stand kein Release-Datum fest und auch noch keine Information, inwiefern dieses Gerät erhältlich sein wird. Am 29. April 2015 veröffentlichte Microsoft ein Video auf YouTube über ihren Kanal¹⁴ und gab erste Einblicke zu der HoloLens. Unter Anderem wurden vorherige Prototypen dargestellt.

Über den *Windows Blogs* von Microsoft wurde am 29. Februar 2016 angekündigt, dass am 30. März 2016 eine Developer Edition der HoloLens zum Kauf zur Verfügung stehen wird¹⁵. Der Preis wurde auf 3.000 Dollar gesetzt. Laut dem Blog-Eintrag ist es das erste Gerät zu dieser Zeit, welches es ermöglicht ohne Marker, externe Kameras, Kabel und Verbindung zu einem Smartphone oder Computer *holographic computing* zu realisieren. Dabei handelt es sich um optische see-through Displays, die projektionsbasiert die computergenerierten Objekte in das Sichtfeld des Anwenders positioniert. Über Gestiken, Gaze, also dem Anstarren von Objekten und der Stimme ist es möglich, die HoloLens und die computergenerierten Objekte zu bedienen. Über eingebaute Kameras kann zum einen das reale Umfeld wahrgenommen werden und zum anderen Fotos und Videos von dem erstellt werden, was der Anwender sieht - das reale Umfeld und die dargestellten virtuellen Objekte.

Das in der Abbildung 2.13 zu sehende Head-Mounted Display zeigt die Microsoft HoloLens in der Developer Edition 2016 mit all dem dazugehörigen Zubehör. Zwischen der HoloLens

¹³<https://news.microsoft.com/windows10story/> (zuletzt aufgerufen am 18.12.2017)

¹⁴<https://www.youtube.com/watch?v=AaTyeDtht-8> (zuletzt aufgerufen am 18.12.2017)

¹⁵<https://blogs.windows.com/devices/2016/02/29/announcing-microsoft-hololens-development-edition-open-for-pre-order-shipping-march-30/#pO3ag3si3IBiY0Kl.97> (zuletzt aufgerufen am 18.12.2017)

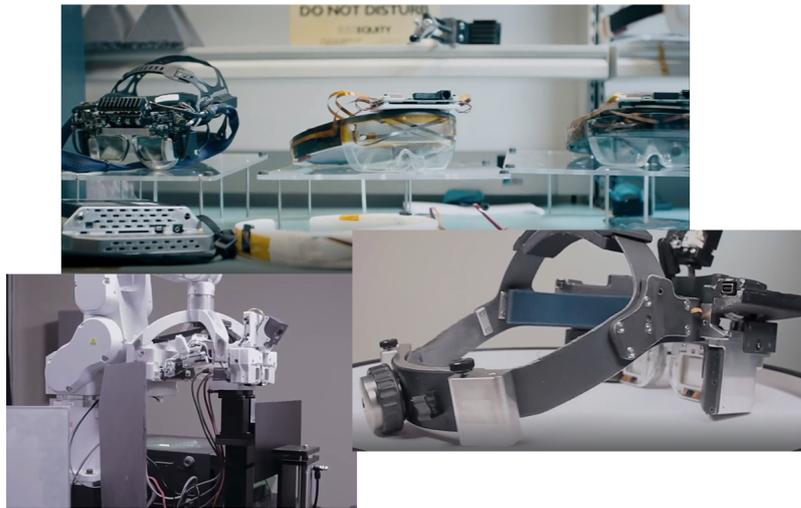


Abbildung 2.12: Einige Prototypen der Microsoft HoloLens - <https://www.youtube.com/watch?v=AaTyeDtht-8> (zuletzt aufgerufen am 18.12.2017)



Abbildung 2.13: Die Microsoft HoloLens in der Development Edition 2016 - <https://blogs.windows.com/devices/2016/02/29/announcing-microsoft-hololens-development-edition-open-for-pre-order-shipping-march-30/#pO3ag3si3IBiY0KI.97> (zuletzt aufgerufen am 18.12.2017)

und der Tragetasche (ganz rechts auf dem Bild) befindet sich ein Klicker, der für die HoloLens genutzt werden kann. Dieser ersetzt die Hand-Gestiken mit physischen Klicks.

2.2.2 Aufbau und Funktionsweise der Developer Edition 2016

Die HoloLens verfügt über mehrere Funktionen, die nur im Zusammenhang mit der dafür konfigurierten Hardware möglich sind. Zunächst über die Features der HoloLens¹⁶:

Spatial Mapping¹⁷ liefert eine detaillierte Struktur der durch die Kameras wahrgenommenen Umgebung. Dies dient bei der Entwicklung zur Darstellung der computergenerierten Objekte in Relation zum Umfeld, was dazu führen soll, dass sich diese dadurch besser in das reale Umfeld integrieren können. Grundlegende Möglichkeiten wie das Platzieren der Objekte auf realen Gegenständen oder der Nicht-Sichtbarkeit der Objekte, wenn sie sich hinter realen Gegenständen befinden, können somit realisiert werden.

Spatial Sound ermöglicht die räumliche Wahrnehmung von Sounds. Die Positionierung der Soundquelle in Relation zur HoloLens lässt den Anwender das Gefühl geben, dass es möglich ist diese Quelle im Raum zu orten.

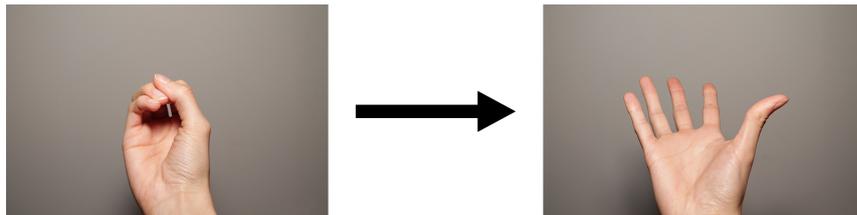


Abbildung 2.14: Die *Bloom* Gestik in der Ausführung

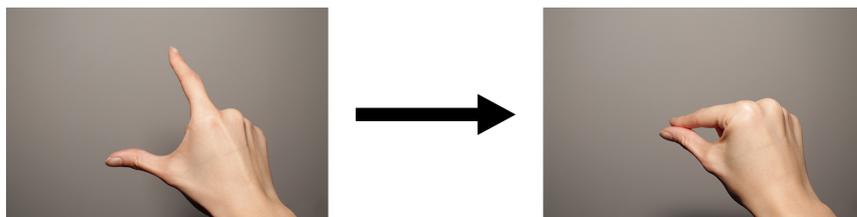


Abbildung 2.15: Die *AirTap* Gestik in der Ausführung

Gestiken ermöglichen die Steuerung der HoloLens. Über die Kameras wird die Hand des Anwenders getrackt und kann somit erkennen, wenn der Anwender mit der Hand eine der

¹⁶<https://www.wearable.com/microsoft/microsoft-hololens-everything-you-need-to-know-about-the-futuristic-ar-headset-735> (zuletzt aufgerufen am 18.12.2017)

¹⁷https://developer.microsoft.com/en-us/windows/mixed-reality/spatial_mapping (zuletzt aufgerufen am 18.12.2017)

vordefinierten Gestiken ausführt und kann darauf reagieren. Zum einen kann das sogenannte *Bloom* verwendet werden, was das Aufgehen einer Blume ähnelt (siehe Abbildung 2.14). Diese wird dafür genutzt um das Hauptmenü des Betriebssystems zu öffnen, um beispielsweise Apps zu starten oder zu den Einstellungen zu wechseln. Außerdem wird sie dazu genutzt die aktuelle Anwendung zu verlassen. Zum anderen gibt es die *AirTap* Gestik, die der Funktion eines Mausclicks bei Windows für den Desktop entspricht (siehe Abbildung 2.15). Desweiteren kann der *AirTap* ausgeführt werden und die Endposition beibehalten werden, was dem Gedrückthalten des Mauszeigers entspricht. Je nach Anwendungsfall kann der dazu genutzt werden Objekte zu verschieben, zu scrollen oder zu zoomen.

Um diese Features zu ermöglichen ist wie bereits erwähnt spezielle Hardware nötig. Auf der 28. *Hot Chips* Konferenz am 22. August 2016, hat Nick Backer von Microsoft einen Vortrag über die HoloLens gehalten und einen technischen Einblick gewährt¹⁸. Dabei wurde nicht explizit die Funktionsweise der einzelnen Bauteile erläutert, wodurch beispielsweise die Erklärung der Projektion der computergenerierten Objekte auf den Displays oder die Zusammenarbeit der Kameras nur auf Vermutungen beruhen, die auf Sachkenntnisse gestützt sind.

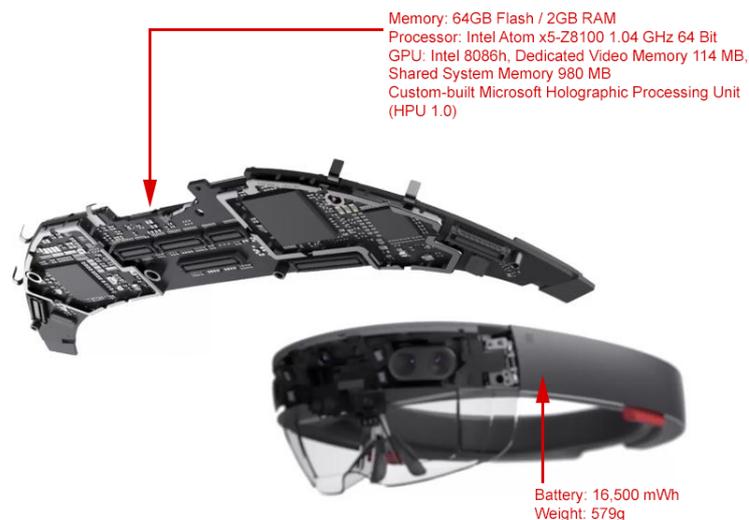


Abbildung 2.16: Das Motherboard der HoloLens

In Abbildung 2.16 ist das Motherboard der HoloLens und dessen grundsätzliche Komponenten dargestellt. Neben den typischen Komponenten wie einer GPU, einer CPU und RAM, wurde für die HoloLens eine sogenannte Holographic Processing Unit entwickelt. Diese küm-

¹⁸https://www.youtube.com/watch?v=u0eBd2m_wEs (zuletzt aufgerufen am 18.12.2017)

merkt sich speziell um Aufgaben, die mit der Verarbeitung? der computergenerierten Objekte in Verbindung stehen. Mit der Akku-Kapazität von 16.500 mWh kann das Gerät bei aktiver Nutzung laut Microsoft¹⁹ bis drei Stunden genutzt werden.



Abbildung 2.17: Die Sensoren der HoloLens

Um unter anderem das *Spatial Mapping* Feature zu ermöglichen, sind besondere Kameras notwendig. In der HoloLens sind in der Gesamtzahl sechs Stück verbaut, wie in [Abbildung 2.17](#) zu sehen ist. Eine der Kameras wird lediglich zur Aufnahme von Videos oder Fotos verwendet. Die fünf restlichen werden in Summe dazu genutzt die Umgebung wahrzunehmen und dabei auch eine Tiefenberechnung zu ermöglichen, womit das *Spatial Mapping* Feature erst möglich ist. Im Anwendungsfall zeigt sich, dass die HoloLens den Cursor stets auf die aufgenommene Umgebung platziert und die Distanz zu diesem relativ präzise zur Realität ist. Desweiteren wird speziell die Tiefenkamera dazu verwendet die Hand des Anwenders zu erkennen, damit die Gestiken möglich sind. Zuletzt befindet sich noch ein *Ambient Light Sensor* auf diesem Bauteil, welches das Umgebungslicht wahrnimmt, um die dargestellten computergenerierten Objekte daraufhin anzupassen.

¹⁹https://developer.microsoft.com/en-us/windows/mixed-reality/hololens_hardware_details (zuletzt aufgerufen am 18.12.2017)

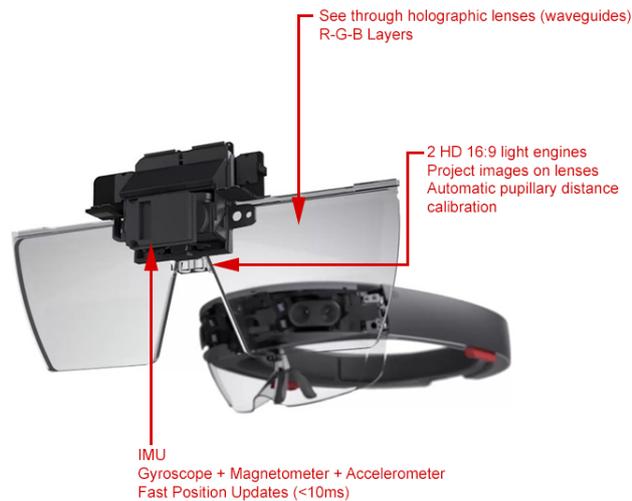


Abbildung 2.18: Die Displays der HoloLens

Wie bereits erwähnt, handelt es sich grundsätzlich um projektionsbasierte optische see-through Displays. Zwischen den Gläsern befinden sich zwei HD 16:9 Projektoren, die für die Darstellung der computergenerierten Objekte zuständig sind (siehe [Abbildung 2.18](#)). Um dies zu ermöglichen, wurden die Gläser speziell angefertigt, damit die Projektion ein Bild erschafft, das für den Anwender klar und als Objekt mit Position im Raum und richtigem Größenverhältnis erscheint. Dazu sind mehrere Faktoren relevant. Für die klare Darstellung der Objekte muss der Abstand zu den Augen gemessen werden, da sich dieser von Anwender zu Anwender ändert. Die Projektion muss an das Auge des Anwenders angepasst werden wofür die Entfernung wichtig ist, ansonsten erscheint das Objekt verschwommen. Damit die Darstellung über die Projektoren so funktioniert, wurden die Gläser so hergestellt, dass eine *total internal reflection* stattfinden kann. Dadurch werden die Strahlen der Projektion nur innerhalb der Gläser reflektiert und bleiben zunächst intern. Über Wellenleiter werden diese schlussendlich gezielt an einen kleinen Bereich auf den Gläsern direkt vor den Augen der Anwender gelenkt und ergeben das Bild des computergenerierten Objekts. Da über die *total internal reflection* ein großes Volumen an Strahlen durch die Gläser gelenkt werden, wurde in der Developer Edition 2016 der HoloLens nur der kleine Bereich auf den Gläsern realisierbar, was den kleinen FOV von 35 Grad des Geräts erklärt.

Um dem dargestellten Bild die Eigenschaft eines realen Objekts zu verleihen, ist es notwendig, dass es sich um ein perspektivisches Bild handelt. Dabei ändert sich die Sichtweise

des Bildes passend zur Perspektive des Anwenders relativ zum Bild. Dies ist in einer hohen Geschwindigkeit nötig, damit die Immersion hoch ist. Zur Realisierung wurde ein *inertial measurement unit* (IMU) verbaut. Hauptsächlich durch einen Gyroscope zusammen mit einem Beschleunigungssensor sind schnelle Positions Updates (schneller als 10 ms) möglich, wodurch die Projektion schnell angepasst werden kann.

Zur weiteren Ausstattung gehören noch Lautsprecher, die sich außen auf der Höhe der Ohren des Anwenders befinden. Eines der Features ist der Spatial Sound, womit erreicht wird, dass der Anwender das Gefühl hat, dass dieser die Soundquelle orten kann. Die Lautsprecher wurden so konzipiert, dass eben dies möglich ist. Außerdem besitzt die HoloLens ein WLAN- und ein Bluetooth-Modul. Neben der Nutzung des Internets über das WLAN-Modul, kann die Verbindung zum Netzwerk dafür genutzt werden, dass Applikationen über diese Verbindung von einem Computer aus gebildet werden können. Dafür müssen sich Computer und HoloLens lediglich im selben Netzwerk befinden und über Visual Studio kann über eine Remoteverbindung eine Applikation direkt auf die HoloLens gebildet werden. Die Bluetooth Verbindung wird für die Nutzung des Clickers benötigt. Dieser kann die Aktionen der Gestiken ausführen. Es können auch weitere bluetoothfähige Geräte mit der HoloLens verbunden werden.

2.2.3 Zukunftspläne

Auf der Microsoft Build Conference 2017²⁰ wurde zuletzt offiziell über die HoloLens von Microsoft aus etwas präsentiert. Dabei wurde unter anderem erwähnt, dass zurzeit keine Rede von einer *Consumer* Version der HoloLens ist. Das läge größtenteils daran, dass dafür der Preis signifikant verringert werden müsste. Erst wenn es möglich ist das Gerät unter 1000 Dollar zu verkaufen, könnte man über eine *Consumer* Version reden. Zum heutigen Zeitpunkt, an dem die HoloLens noch über 3000 Dollar kostet, bleibt es ein Gerät für Entwickler und Unternehmen. Stattdessen hat Microsoft dieses Jahr eine Reihe von Mixed Reality Geräten auf den Markt gebracht, die für Endkunden gedacht ist. Grundsätzlich sind diese Geräte zur Darstellung von Virtual Reality Inhalten gedacht, besitzen aber im Vergleich zu herkömmlichen Virtual Reality Geräten noch Kameras an der Frontseite, womit das reale Umfeld wahrgenommen werden kann. Außerdem wurde die Entwicklung für die Mixed Reality Geräte so konzipiert, dass eine Portierung nahezu eins zu eins auf die HoloLens möglich ist. Das fördert die HoloLens in der Entwicklung, ist aber kein ausschlaggebender Faktor für die

²⁰https://channel9.msdn.com/Events/Build/2017?wt.mc_id=build_hp (zuletzt aufgerufen am 18.12.2017)

Zukunft der HoloLens.

Wie eine technische Newsseite²¹ berichtet, wurde die geplante zweite Version der HoloLens übersprungen und es wird an einer dritten Version gearbeitet, die für 2019 angesetzt wurde. Das wurde damit begründet, dass die geplante zweite Version einen zu kleinen technischen Fortschritt geliefert hätte, als das sich die Veröffentlichung mit all den dazugehörigen Prozessen gelohnt hätte. So wird an einer dritten Version gearbeitet, die einen Generationssprung darstellen soll. Über im April 2017 eingereichte Patente²² soll es den Ingenieuren von Microsoft gelungen sein, dass effektive FOV auf 70 Grad erhöht zu haben. Das wäre eine Verdopplung zur Developer Edition 2016, die lediglich einen FOV von 35 Grad realisiert.

2.3 Verwendete Softwarelösungen

2.3.1 Blender

Bei Blender²³ handelt es sich um eine Open Source 3D Bearbeitungssoftware. Sie wird vielseitig eingesetzt, um unter anderem das Modellieren, Animieren, Simulieren und weiteren 3D Anwendungsbereichen abzudecken.

2.3.2 Unity

Unity ist eine Entwicklungsplattform zur Erstellung von 2D, 3D Anwendungen²⁴. Dabei ist es möglich, Anwendungen für eine Vielzahl von Plattformen zu entwickeln, die entweder nativ in der Entwicklungsumgebung definiert sind oder durch Asset Pakete dahingehend erweitert werden. Integriert ist der Unity Asset Store, der breitgefächert weitere Elemente hinzufügen kann, die für die Entwicklung einer Anwendung hilfreich sein können. Darunter fallen 2D und 3D Modelle, Sound- sowie visuelle Effekte aber auch bereits vorhandene Elemente können durch diesen im Funktionsumfang erweitert werden. Grundsätzlich besteht eine Unity Anwendung aus einer oder mehreren Szenen, die zur Laufzeit dargestellt werden sollen. Jede Szene beinhaltet eine Menge von sogenannten GameObjects, die einzubindende Objekte jeglicher Form repräsentieren. So ist ein in der Anwendung darzustellendes 3D Modell ein

²¹<http://www.thurrott.com/hardware/90780/microsoft-accelerates-hololens-v3-development-sidesteps-v2> (zuletzt aufgerufen am 18.12.2017)

²²<https://hololens.reality.news/news/microsoft-has-figured-out-double-field-view-hololens-0180659/> (zuletzt aufgerufen am 18.12.2017)

²³<https://www.blender.org/>

²⁴<https://unity3d.com/de/public-relations> (zuletzt aufgerufen am 18.12.2017)

GameObject, das durch weitere Komponenten zur Modellierung von gewünschten Eigenschaften erweitert ist. Diese Komponenten können Kollisionsboxen sein, damit zur Laufzeit erkannt wird, wenn etwas dieses GameObject berührt oder auch Skripte, die Funktionalität hinzufügen. Auch Soundeffekte und das Aussehen in Form von Materialien oder Texturen wird über Komponenten realisiert. Außerdem ist es möglich eine Hierarchie in Form von Eltern zu Kind Beziehungen von GameObjects umzusetzen, wodurch die Modellierungsmöglichkeiten erweitert werden. Wird ein GameObject, beispielsweise ein 3D Modell, fertig in einer Szene modelliert, indem weitere Komponenten hinzugefügt wurden, kann dieser als sogenanntes Prefab gespeichert werden. Ein Prefab stellt somit ein vorkonfiguriertes GameObject dar, welches auch so in der Szene verwendet werden kann.

2.3.3 Vuforia

Vuforia²⁵ ist ein Software Development Kit, welches es ermöglicht mithilfe einer Kamera Objekte zu erkennen, die als sogenanntes Target definiert worden sind. Bei einem Target handelt es sich um ein vorher definiertes Objekt, wie beispielsweise eines Bildes. Es können auch weitere Objekte gewählt werden wie einfache Körper, Zylinder oder Quader, oder auch Text. Auf der Vuforia Website können Datensätze angelegt werden, zu denen die Objekte hochgeladen werden können. Die Unterteilung der Datensätze wird dazu genutzt Targets zu kapseln, sodass gezielt die für das Projekt benötigten verwendet werden können. Damit ein Target mit einer Kamera erkannt wird, muss dieses gewisse Kriterien erfüllen:

- Detailreich mit vielen Elementen
- Guter Kontrast durch helle und dunkle Elemente
- Keine wiederholenden Muster
- Das optimale Format

Detailreich wird bei Vuforia mittels sogenannten Features klassifiziert. In einem Bild werden harte Kanten erkannt, sofern der Kontrast bei den Kanten hoch genug ist. Weist ein Bild eine große Menge solcher Features auf, so wird dieses besser erkannt. Die Anzahl Features ist dabei nicht das einzige Kriterium. Es ist wichtig, dass die Features in Relation zum Gesamtbild nicht nur an einer Stelle fokussiert sind, sondern über das gesamte Bild verteilt sind. Außerdem müssen wiederholende Muster vermieden werden, da dies die Erkennung erheblich beeinträchtigt. Diese Analyse findet statt, sobald ein Objekt als Target hochgeladen

²⁵<https://www.vuforia.com/> (zuletzt aufgerufen am 18.12.2017)

wird. Nach der Analyse wird dem Target ein Rating von keinem bis fünf Sternen zugewiesen. Bei keinem Stern wird das Target zur Laufzeit nicht erkannt und je höher das Rating ist, umso besser wird es erkannt. Die Größe des Targets zusammen mit dem Rating, Lichtverhältnis und Kamerafokus entscheiden, auf welcher Distanz das Target erkannt wird. Generell lässt sich die Maximaldistanz aus der Größe des Targets multipliziert mit 10 errechnen, wenn die Gegebenheiten gut sind²⁶.

2.3.4 Visual Studio

Visual Studio²⁷ ist eine integrierte Entwicklungsumgebung, die es ermöglicht für mehrere Plattformen in verschiedenen Programmiersprachen Applikationen zu entwickeln. Unter anderem können Android, iOS, Mac, Windows, Web und Cloud Applikationen entwickelt werden. Visual Studio wird ebenfalls dafür genutzt Applikationen für die HoloLens zu bauen und bereitzustellen. In Unity wird das Projekt in eine Visual Studio Lösung gebaut, die dann über Visual Studio selbst auf die HoloLens gebracht wird. Zur Visualisierung wurden verschiedene Prototypen von Head-Mounted Displays verwendet, die mit einer Kamera versehen sind.

2.4 Related Work

In der Vergangenheit wurde mit dem jeweiligen Stand der Technik bereits erste Implementierungen für die Erweiterung einer Karte realisiert. Im Paper von [Hedley u. a.](#) wurde 2002 bereits auf Basis von geografischen Karten Landschaften in Form von 3D Modellen eingeblendet. So wurden zu vorhandenen Gebirgen auf einer Karte passende 3D Modelle dieser hinzugefügt, sodass ersichtlich ist, wie diese geformt sind. Zu Stadtteilen wurden passende Satellitenbilder eingeblendet. 2005 haben [Reitmayer u. a.](#) einen Prototypen erstellt, der auf projektionsbasierter Augmented Reality setzt. Als Vorlage wurde eine Stadtkarte verwendet, auf der Informationen wie die Position eines Helikopters oder der Verlauf eines Flusses dargestellt wurde. Über der Stadtkarte befand sich eine Kamera, der die Stadtkarte zunächst aufgenommen hat um die korrekte Position der darzustellenden Objekte zu ermitteln. Ein Projektor hat daraufhin zu vorhandenen Daten wie der Position des Helikopters an der jeweiligen Position ein Bild zur Visualisierung projiziert. Ebenfalls 2005 haben [Liarokapis u. a.](#) mobile Devices verwendet, um zu Ausschnitten einer Karte Augmented Reality Inhalte dar-

²⁶<https://library.vuforia.com/articles/Solution/Optimizing-Target-Detection-and-Tracking-Stability.html> (zuletzt aufgerufen am 18.12.2017)

²⁷<https://www.visualstudio.com/de/> (zuletzt aufgerufen am 18.12.2017)

zustellen. Hier wurden zwar auch geografische Elemente wie Gebirge modelliert, aber gab es auch Ausschnitte von Städten, in diesem Fall das Gelände von einem Campus, und dort wurden 3D Modelle von den Gebäuden dargestellt. Dabei wurde nur die einfache Form der Gebäude modelliert und es wurde keine Texturierung vorgenommen. 2009 haben **Schoening u. a.** eine Mischung aus mobile Device Augmented Reality und dem projektionsbasiertem Ansatz entwickelt. Über der Kamera des Devices wurde eine Stadtkarte aufgenommen und über einen kleinen am Device befestigtem Projektor wurden vordefinierte Elemente direkt auf die Karte projiziert. So konnte bei der Erkennung eines Gebäudekomplexes die Umrisse dessen auf der Stadtkarte eingeblendet werden.

3 Idee und Konzeption

Um eine Lösung für die Problemstellung zu finden, sind zunächst Ansätze herauszuarbeiten. In diesem Kapitel werden zunächst die Anforderungen definiert, die die Implementation umsetzen soll. Des Weiteren werden die verschiedenen Trackingmöglichkeiten erläutert und was grundsätzlich in der Implementation enthalten sein soll.

3.1 Anforderungsanalyse

Vor der Implementation eines Prototyps wird eine Anforderungsanalyse durchgeführt. Diese dient dazu, den zu implementierenden Rahmen zu definieren und um eine Dokumentationsgrundlage zu schaffen, auf die zurückgegriffen werden kann. Sie dient außerdem dazu, dass evaluiert werden kann, ob und wie die Anforderungen in der Implementation umsetzbar waren. Es werden in funktionale und nichtfunktionale Anforderungen unterschieden. Funktionale Anforderungen definieren, was das System leisten muss, während nichtfunktionale definieren, wie das System diese Leistung erbringen muss [Pohl].

3.1.1 Funktionale Anforderungen

- Es soll eine analoge Stadtkarte erstellt werden, die als Grundlage der Applikation für die darzustellenden Objekte gilt
- Die Objekte, die durch die Applikation dargestellt werden, sollen auf Basis der Stadtkarte positioniert und skaliert werden
- Die Stadtkarte soll so groß wie möglich realisiert werden, damit die Objekte, die in Relation zur Größe der Karte skaliert werden, gut erkennbar sind
- Alle darzustellenden Objekte sollen anhand der Stadtkarte ausgerichtet und positioniert werden
- Auf verschiedenen Distanzebenen zwischen der HoloLens und der Stadtkarte sollen Informationen dargestellt werden, die proportional zur Distanz detailreicher werden

- Bei der niedrigsten Distanzebene sollen für alle vorhandenen 3D Modelle Marker¹ auf der Stadtkarte angezeigt werden
- Die Marker müssen mit dem Namen der zugehörigen Sehenswürdigkeit versehen sein
- Über eine Gestik soll der Marker ausgeblendet werden, den der Anwender über den Cursor der HoloLens anschaut und das dazugehörige 3D Modell soll an dieser Stelle eingeblendet werden
- Es soll einen Detail-Mode für die 3D Modelle geben, der ausgeführt wird, wenn der Anwender mit dem Cursor der HoloLens auf ein 3D Modell schaut und eine Gestik ausführt
- Im Detail-Mode soll das 3D Modell, welches über die Gestik in diesen Modus versetzt wurde, vergrößert werden und mit Informationen zu der Sehenswürdigkeit erweitert werden
- Befindet sich ein 3D Modell im Detail-Mode, so sollen alle anderen Objekte über alle Distanzebenen hinaus ausgeblendet werden
- Wird das 3D Modell im Detail-Mode mit dem Cursor der HoloLens angeschaut und der Anwender führt eine Gestik aus, so soll der Detail-Mode verlassen werden, das 3D Modell ausgeblendet werden und der dazugehörige Marker wieder eingeblendet werden
- Die 3D Modelle der Sehenswürdigkeiten sollen außerhalb des Detail-Modus auf eine realitätsnahe Skalierung auf Basis der Stadtkarte dargestellt werden, um die Immersion zu vergrößern

3.1.2 Nichtfunktionale Anforderungen

- Intuitiv: Die Applikation soll so realisiert werden, dass die Benutzung dieser mit dem Wissen der Bedienungsweise der HoloLens ohne Einweisung möglich ist
- Reaktionszeit: Die Darstellung der Objekte verschiedener Distanzebenen soll sofort geschehen, sobald eine Distanzebene erreicht wurde
- Performanz: Die Applikation soll bei der Darstellung der Szene mit den dazugehörigen Objekten mindestens 24 FPS² aufweisen

¹Ein Objekt, welches zur Orientierungshilfe für den Anwender dient

²Frames per second / Bilder pro Sekunde

- **Optische Anforderung:** Die verwendeten 3D Modelle sollen den realen Sehenswürdigkeiten nachempfunden werden, sodass eine Vorstellung des realen Erscheinungsbildes der Sehenswürdigkeit vermittelt wird
- **Umgebungsanforderung:** Die von der HoloLens zur Verfügung stehenden Gestiken sollen zur Bedienung verwendbar sein
- **Offline-Fähigkeit:** Die Applikation soll ohne Internetverbindung verwendbar sein, um sie unabhängig nutzen zu können

3.2 Analoge Stadtkarte

Eine analoge Stadtkarte ist die grundlegende Anforderung für die Realisierung der Applikation. Aufgrund des Standorts der Hochschule und dem persönlichen Interesse an der Stadt Hamburg, wird diese auch als Beispiel für eine Implementation genutzt. Eine Großstadt bietet sich als Detail-Level gut an, da aufgrund der Popularität eine Vielzahl an 3D Modellen zur Verfügung stehen wird. Ebenfalls denkbar wäre es eine niedrigere Detail-Stufe auszuwählen wie etwa gesamt Deutschland, Europa oder der Welt und man dann die bekanntesten Bauten in Betracht zieht. Höhere Detail-Stufen, sodass man kleinere Städte oder nur Teile von Großstädten darstellt, wäre in Anbetracht des Interesses und Repräsentation durch bekannte Bauten die schlechtere Wahl. So bietet sich Hamburg als Prototyp gut an, um exemplarisch vorzuführen, was im Nachhinein für andere Städte, Länder etc. implementierbar ist.

Die Größe der Stadtkarte ist wichtig für die Erschaffung der Immersion, da die 3D Modelle der Sehenswürdigkeiten zunächst in Miniaturform passend zur Karte dargestellt werden. Daher benötigt die Karte eine gewisse Mindestgröße, sodass die Relation zwischen Miniaturmodellgröße und Stadtkartengröße halbwegs real umgesetzt werden kann, während die Miniaturmodelle eine Mindestgröße aufweisen, damit sie auch in dieser Größe erkennbar sind. Nützlich ist die Erweiterung nur, wenn der Anwender die dargestellten Elemente durch die HoloLens intuitiv als passende Ergänzung sieht. Dafür müssen vor allem die Skalierung und Positionierung der Elemente mit der Stadtkarte übereinstimmen und es muss gut erkennbar sein. Generell lässt sich mit diesen Punkten ein Szenario für verschiedene Kartengrößen realisieren. In dem Stadium, in dem sich die HoloLens und die softwareseitigen Lösungen zur Zeit der Ausarbeitung befinden, ist eine größere Variante die passendere Lösung. Denkbar wäre eine Lösung für eine Karte in Größe der Reiseführer-Bücher. Perspektivisch wäre der Anwendungsfall somit an jene gerichtet, die als Tourist eine Erweiterung der üblichen

Stadtkarten nutzen würden. Die aktuelle Fassung der HoloLens ist aufgrund der Größe, des Gewichts und des Kostenfaktors jedoch nicht alltagstauglich und der Anwendungsfall hätte einen geringen Realitätsfaktor. Zudem ist die Darstellung so kleiner Objekte zur Wahrung der Immersion nicht optimal. Zu kleine 3D Modelle sind nur schwer zu erkennen und durch die Größe der HoloLens kann man nur bedingt nah an die Karte rangehen. Ein Anwendungsfall, der sich daran richtet ein Setup aufzubauen, um die HoloLens zusammen mit der Stadtkarte an einem Stand zur Vorführung zu nutzen, unterstützt die Wahl einer großen Variante.

3.3 Trackingverfahren

Fundamentale Anforderung der HoloLens Applikation ist es, die darzustellenden Objekte anhand der Stadtkarte auszurichten und zu positionieren. Dazu muss die HoloLens die Stadtkarte erkennen und die Szene dahin ausrichten. Zur Erkennung wird Vuforia genutzt, welches vorher definierte Targets tracken kann und somit alles relativ zur Position des Targets ausrichten kann. Im Rahmen dieser Ausarbeitung werden dabei zwei verschiedene Trackingverfahren in Betracht gezogen, die im Zusammenhang mit der Stadtkarte denkbar sind.

3.3.1 Multi-Level Target

Dieser Ansatz verwendet die Stadtkarte selbst als Target auf verschiedenen Ebenen. Für die in den Anforderungen definierten Distanzebenen sind Target-Gruppen zuständig. Für die höchste Distanzebene wird die gesamte Karte oder zumindest ein Großteil als Target registriert. Je näher die Distanzebene, umso kleiner werden die Ausschnitte, die als Target genutzt werden. Da die Entfernung, auf der ein Target mit der HoloLens erkannt wird, grundsätzlich von dessen Größe abhängig ist, können so die Ebenen realisiert werden. Im Good-Case Szenario, bei dem auf den Targets ausreichend viele Feature-Punkte identifiziert werden können und Vuforia ein hohes Rating vergibt und die Lichtverhältnisse sowie Kamerafokus stimmen, wird ein Target auf einer Distanz getracked, die dem 10-fachen dessen Breite entspricht. Da es möglich ist simultan mehrere Targets zu tracken, könnten sich die Distanzebenen mischen. Damit das nicht passiert, muss bei der initialen Erkennung eines Targets einer Distanzebene auch die entsprechende Distanz gespeichert werden. Dadurch werden die Objekte, die einer anderen Distanzebene angehören, nicht angezeigt, wenn dessen Distanz nicht erreicht wurde. Die näheren Distanzebenen beinhalten immer kleinere Targets. Da der Anwender nicht weiß wohin geschaut werden muss damit die Targets erkannt werden, müssen diese Ebenen so aufgebaut sein, dass die Summe aller Targets einer Ebene wieder die Stadtkarte selbst ergibt. So ist gewährleistet, dass es nicht von Bedeutung ist, wo der Anwender hinschaut, sondern

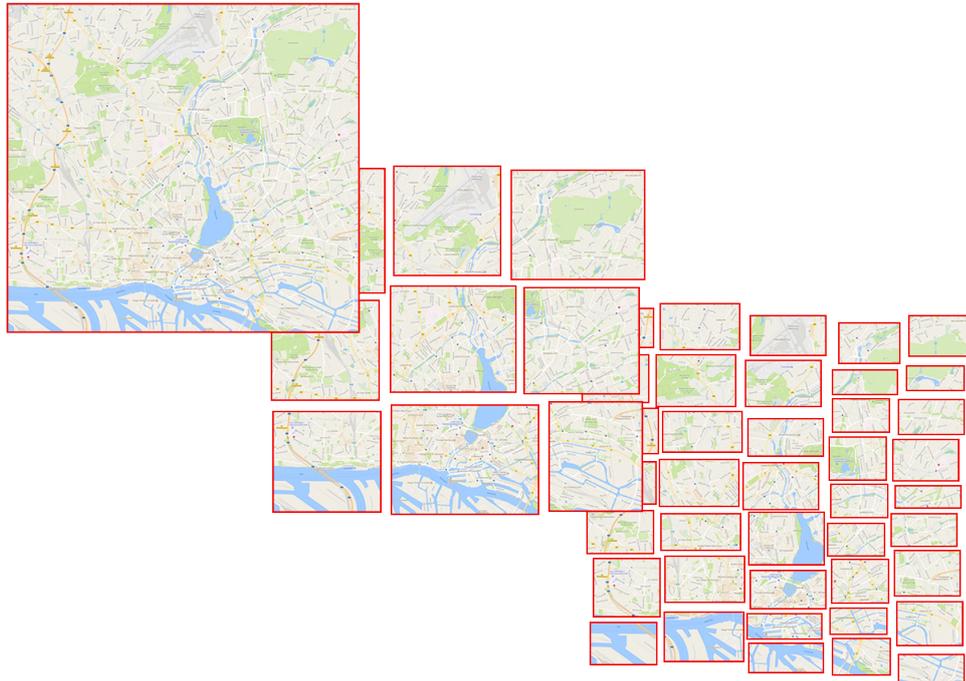


Abbildung 3.1: Darstellung des Multi-Level Target Prinzips

allein die Entfernung ist relevant.

Dieses Trackingverfahren bringt den weiteren Vorteil, dass durch die Tatsache, dass für die nahen Distanzebenen sehr kleine Ausschnitte verwendet werden und dadurch in der Theorie das passende Target auf dem Display eines Smartphones dargestellt werden kann und getracked werden kann. Das würde es erlauben, die Applikation für einen kleinen Teil von der analogen Stadtkarte zu trennen und sie universell auf dem Smartphone zu verwenden.

3.3.2 Single Target Reference

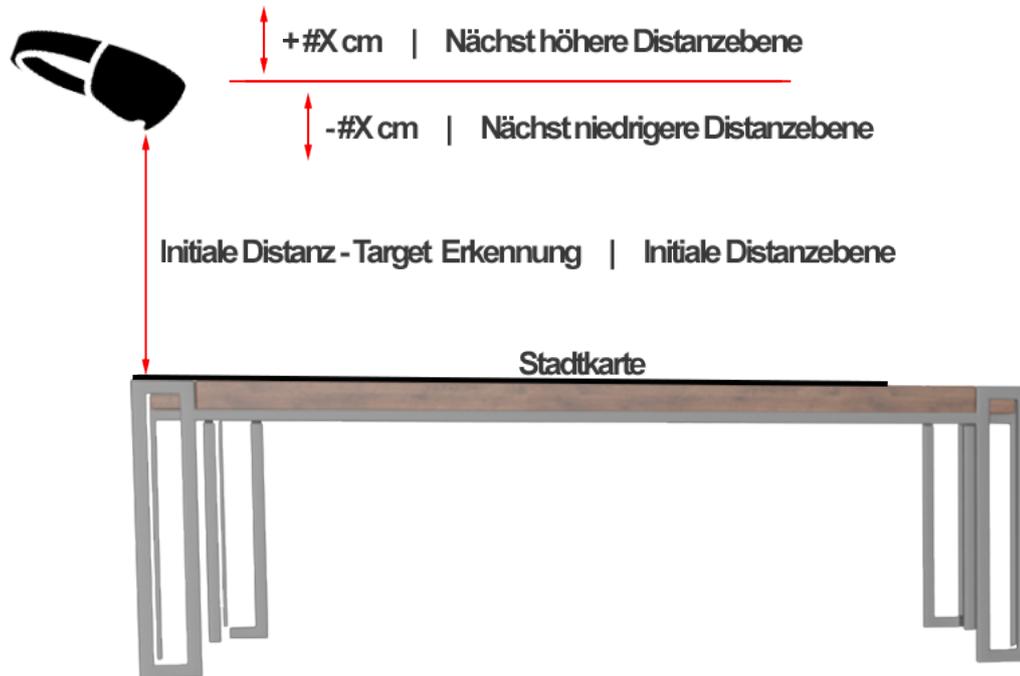


Abbildung 3.2: Darstellung des Single Target Reference Prinzips

Für diesen Ansatz wird nur ein Target genutzt, welches für die Ausrichtung und Realisierung der Distanzebenen zuständig ist. Dieses Target kann entweder ein Ausschnitt der Stadtkarte selbst sein oder ein hinzugefügtes Element sein. Wird das Target mit der HoloLens erkannt, so wird die initiale Distanz gespeichert und bildet eine Distanzebene. Für die weiteren Ebenen werden basierend auf der initialen Distanz feste Entfernungen zu eben dieser definiert, die oberhalb und unterhalb der initialen Distanzebene liegen können. Dadurch ist gewährleistet, dass jede Distanzebene eine definierte Distanz zur HoloLens hat und so gezielt die Objekte angezeigt werden können, die zur jeweiligen Distanzebene gehören. Da nur ein Target verwendet wird um die Szene auszurichten, kann dieser so erstellt werden, dass die optimalen Bedingungen erfüllt sind. Der Ausschnitt der Stadtkarte oder das hinzugefügte Element kann so gewählt werden, dass ein fünf Sterne Rating erreicht wird und dieser auch groß genug ist, dass eine gewünschte Erkennungsdistanz gegeben ist.

3.4 Evaluation der Trackingverfahren

In den ersten Implementierungsversuchen der beiden Verfahren konnten sich klare Vor- und Nachteile ausarbeiten. Zur Wahl der zu implementierenden Lösung wurde auf Erweiterbarkeit, Intuition und vor allem Umsetzbarkeit überprüft.

Multi-Level Target

Erweiterbarkeit:

- Um auf Basis dieses Trackingverfahrens eine Erweiterung in Form von weiteren Distanzebenen zu realisieren, muss die Größenverteilung der Targets innerhalb einer Distanzebene verändert werden, damit weitere Größen verfügbar sind. Falls noch nicht zu viele Distanzebenen verwendet wurden, sind die Größenunterschiede der Targets zwischen den Ebenen groß genug, sodass weitere Größen ohne weiteres verwendbar ist. Dennoch müssen für eine Ebene genügend Targets erstellt werden, damit diese in Summe wieder die Stadtkarte ergeben, damit der Anwender die Freiheit hat an jeder Stelle der Stadtkarte zu schauen.

Intuition:

- Die zu erkennenden Targets sind Ausschnitte aus der Stadtkarte selbst. Für den Anwender ist somit nicht ersichtlich, dass zusätzliche Hilfsmittel benutzt werden. Bei einem guten Rating des Targets bei Vuforia, funktioniert die Erkennung auch je nach Größe des zu erkennenden Targets auf einer guten Distanz, sodass der Anwender nicht erst die Stadtkarte mit der HoloLens absuchen muss. Dennoch könnte es bei schlechten Verhältnissen dazu kommen, dass die Targets nicht sofort erkannt werden und eine zusätzliche Hilfestellung durch Anweisungen erfolgen muss. Dadurch ergibt sich eine generell schlechtere Intuition, dafür aber eine bessere Immersion, da die Stadtkarte allein als Basis reicht um die Applikation zu benutzen.

Umsetzbarkeit:

- Die einzelnen Targets der Distanzebenen können einfach realisiert werden. Jedoch ergeben sich schlechte Ratings von Vuforia, wenn man Teile der Stadtkarte dafür verwendet. Bei der Größe und dem Detailgrad der von der Stadtkarte abgeleiteten Bildern ergeben sich größere Dateien, die die zwei MB Grenze von Vuforia für hochgeladene Bilder überschreitet. Um diese dennoch als Target nutzen zu können, müssen diese

beim Speichern komprimiert werden, wodurch Qualitätsverluste entstehen. Aufgrund dessen ergeben sich nicht genug harte Kanten in den hochgeladenen Bildern, da diese leicht verschwommen sind. In diesem Fall bekommen die Targets kein Rating und können von der HoloLens nicht erkannt werden. Für die nahen Distanzebenen, bei denen die Targets so klein sind, dass die Bildausschnitte nicht komprimiert werden müssen um die zwei MB Grenze einzuhalten, ergeben sich Targets mit verwendbaren Ratings, je nach Ausschnitt. Das reicht leider nicht aus, um diesen Ansatz zu realisieren.

Single Target Reference

Erweiterbarkeit:

- Ein Target je Stadt. Für die Erweiterung einer Stadt mit weiteren Distanzebenen, müssen in diesem Ansatz keine weiteren Targets erstellt werden, sondern nur weitere Distanzhöhen für die neuen Ebenen.

Intuition:

- Sofern ein Teil der Stadtkarte selbst als Target genutzt wird, ergibt sich das gleiche Bild wie beim Multi-Level Target. Wird das Target als weiteres Element der Karte hinzugefügt wie beispielsweise ein QR-Code, so verschlechtert sich die Immersion, da der Anwender weiß, dass explizit etwas getracked werden muss. Bei schlechten Verhältnissen zeigt sich jedoch eine höhere Intuition, da der Anwender weiß, dass der QR-Code genutzt werden muss, damit die Applikation funktioniert und wird diesen explizit anschauen und es müssen keine Hilfestellungen geleistet werden wie beim Multi-Level Target.

Umsetzbarkeit:

- Einen Teil der Stadtkarte als Target für den Single Target Reference Ansatz zu benutzen ist aus dem selben Grund wie beim Multi-Level Target Ansatz nicht realisierbar. Damit der Anwender das Target auf der Stadtkarte auf einer annehmbaren Entfernung automatisch mit dem Blick auf der Stadtkarte tracked, muss das Target ein gutes Verhältnis von Größe und Rating haben. Erst kleine Ausschnitte ermöglichen überhaupt ein Rating und diese sind so klein, dass der Anwender sich bereits sehr nah an der Stadtkarte befinden muss und an die richtige Stelle schauen muss. Außerdem weiß der Anwender nicht, dass ein Target in der Stadtkarte versteckt ist und würde ohne Hilfestellung diesen nicht finden. Mit dem hinzugefügten Target wie beispielsweise dem QR-Code, kann ein volles Rating bei ausreichender Target Größe erzielt werden.

Ein Blick auf den QR-Code mit der HoloLens und dieser wird, während man neben der Stadtkarte steht, erkannt. Somit kann dieser Ansatz umgesetzt werden.

Basierend auf der Erweiterbarkeit und der Intuition beider Ansätze, wäre die Implementation des Multi-Level Target Ansatzes vorteilhafter. Im Bereich Augmented Reality ist die Immersion und die Intuition für den Anwender am wichtigsten. Die Applikation muss einfach zu verwenden und verständlich sein, damit diese auch positiv als Erweiterung der Realität aufgefasst wird. Die Stadtkarte auf verschiedenen Ebenen selbst als Target zu verwenden, ist ein guter Ansatz genau das zu erzielen. Wenn die Targets ein hohes Rating erzielen würden, so wäre es für den Anwender intuitiv und die Immersion wäre aufgrund der Funktionalität ohne weitere Hilfsmittel hoch. Maßgeblich ist jedoch die Umsetzbarkeit, die bei diesem Ansatz in dieser Form nicht gegeben ist. Bis auf die etwas geringere Immersion durch den hinzugefügten Target und die Tatsache, dass es nicht die Möglichkeit gibt eine andere Grundlage wie beispielsweise Google Maps auf dem Smartphone zu nutzen, weist der Single Target Reference Ansatz ansonsten keine Nachteile auf und ist durch den offensichtlichen Target grundlegend intuitiv. Daher wird der Single Target Reference Ansatz als Trackingverfahren für die Implementation genutzt.

4 Implementation GeoMeta

In diesem Abschnitt geht es um die konkrete Umsetzung anhand der erstellten Anforderungen. Bei der Evaluation der Trackingverfahren wurde bereits herausgearbeitet, dass der Single Target Reference Ansatz die zu implementierende Lösung darstellt. Diese wird als Grundlage genutzt, um die Erkennung der Stadtkarte und Ausrichtung aller Objekte an eben dieser zu realisieren. Grundlegend wird die Applikation so implementiert, dass es nicht notwendig ist eine Verbindung zum Internet zu haben. Somit kann diese Applikation einfacher ausgeführt werden, da außer der Stadtkarte nichts notwendig ist.

4.1 Erstellung der Stadtkarte von Hamburg

Wie in den Anforderungen beschrieben, wird die Stadtkarte so groß wie möglich realisiert, damit die darzustellenden Modelle in Miniaturform zum einen gut zu erkennen sind und zum anderen in Relation zur Stadtkarte eine angenehme Größe haben. Die Hochschule hat die Möglichkeit einen A0 Druck anzufertigen, was einer Größe von 84,1cm in der Breite und 118,9cm in der Höhe entspricht. Der zuständige Drucker kann dabei eine Druckauflösung von 300 DPI realisieren. Zusammen mit der Größe ergibt das eine benötigte Auflösung der Stadtkarte Hamburgs von 9933 x 14043 Pixel. Im Rahmen dieses Anwendungsfalls wurde ein eigenes Exemplar gefertigt, um die benötigte Auflösung zu erfüllen, da diese verhältnismäßig hoch ist. Google Maps bietet eine gute Basis zur Erstellung einer Stadtkarte, da sie einen hohen Detailgrad selbst bei hohem Zoom-Level bietet. Die Herausforderung bestand darin, dass mehrere Aufnahmen von Google Maps erstellt wurden, sodass bei der Zusammenführung der relevante Ausschnitt Hamburgs zu sehen ist und durch die Zusammenführung der einzelnen Aufnahmen die benötigte Auflösung erreicht wurde. Zunächst wurde über das Feature My Maps von Google ein Rahmen erstellt, der dem Seitenverhältnis der benötigten Auflösung entspricht und den relevanten Teil Hamburgs umfasst. Basierend darauf muss der richtige Zoom-Level eingestellt werden, sodass die Summe der Aufnahmen die benötigte Auflösung entspricht. Dazu wurde testweise eine Reihe von Screenshots in der horizontalen erstellt, sodass man im ersten Bild am linken Rahmenrand angefangen hat und beim letzten

4 Implementation GeoMeta

Screenshot am rechten Rahmenrand aufgehört hat. In Adobe Photoshop wurde eine Vorlage erstellt, die der benötigten Auflösung entsprach. Die erstellten Aufnahmen wurden horizontal eingereiht und überprüft, ob sie die Breite ausfüllen. Wenn das der Fall war, so passte der Zoom-Level insgesamt, da das Seitenverhältnis im Vorfeld definiert wurde und die Summe der Aufnahmen vertikal eingereiht auch stimmen würde. Daraus ergab sich ein verwendeter Zoom-Level von 200m pro Bildschirmzentimeter und 28 Screenshots, die zusammen die zur Implementation genutzte Stadtkarte darstellen.

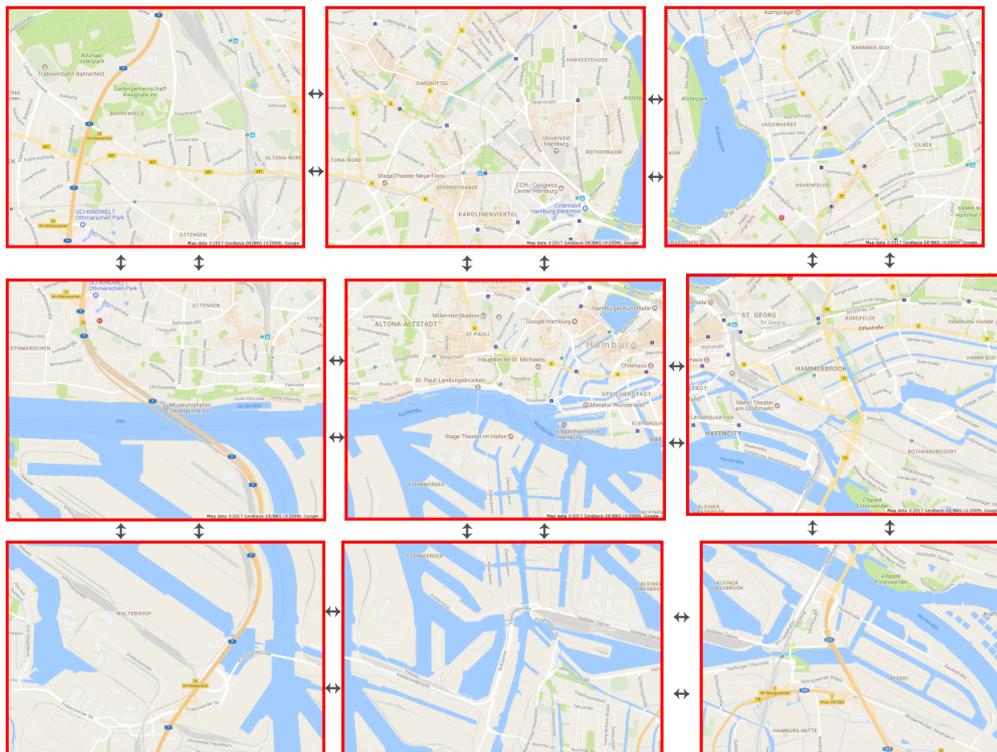


Abbildung 4.1: Die einzelnen Screenshots der Ausschnitte aus Google Maps werden zusammengesetzt

Da ein Teil der Stadtkarte aufgrund des schlechten Ratings nicht als Target verwendbar ist, muss ein Bild hinzugefügt werden. Das Rating zusammen mit der Größe entscheiden, auf welcher Distanz der Marker bei guten Lichtverhältnissen erkannt wird. Bei vollem Rating und guten Lichtverhältnissen ist die Distanz, bei dem das Target getracked wird, in etwa das 10-fache der Breite des Targets. Als Target-Größe wurde 15cm gewählt, was bei guten Verhältnissen zu einer Tracking-Distanz von 1,5m führt. Das ist ausreichend für den Anwendungsfall und ist nicht zu groß, sodass kein wesentlicher Teil der Stadtkarte verdeckt wird. Wie

4 Implementation GeoMeta

in Abschnitt 2.3.3 beschrieben, bestimmen mehrere Kriterien das Rating eines Targets. Ein QR-Code erfüllt diese Kriterien, da viele harte Kanten vorhanden sind, keine wiederholenden Muster vorkommen und der Kontrast hoch ist.

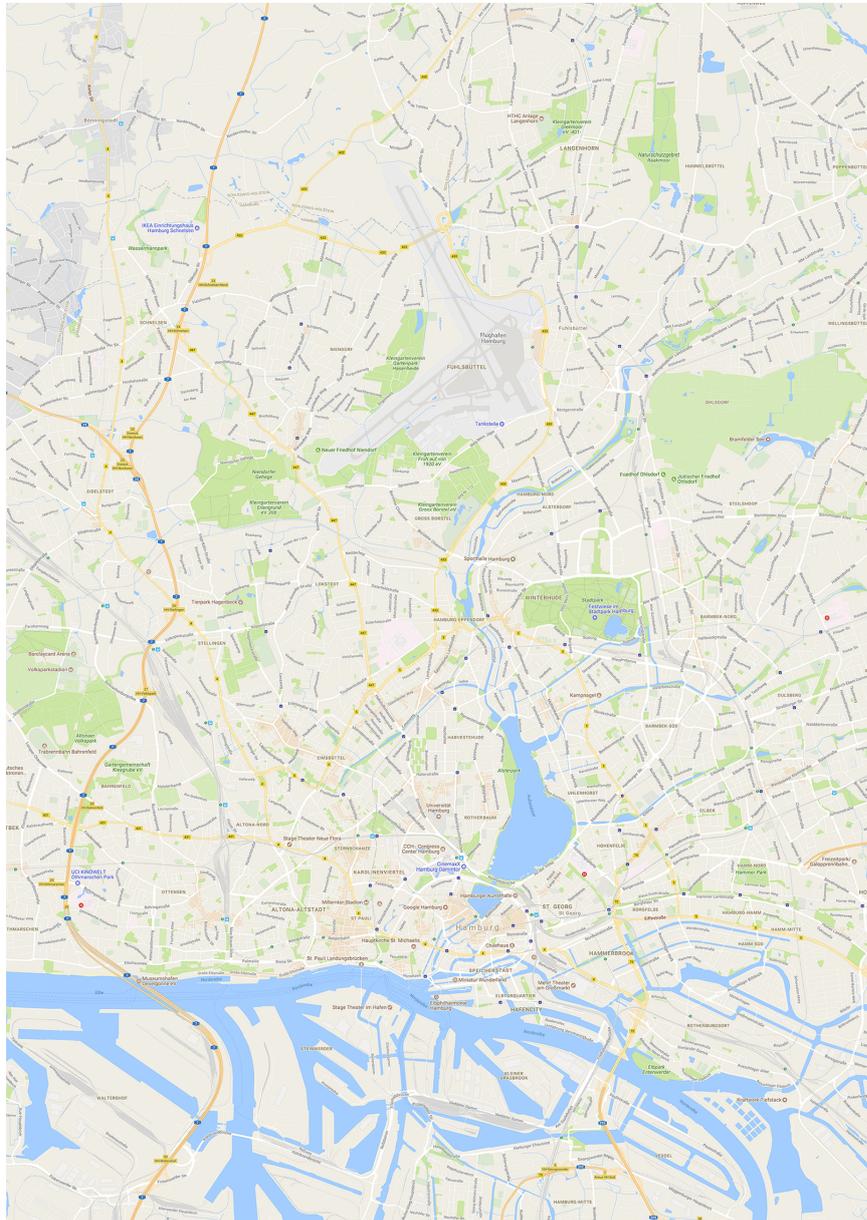


Abbildung 4.2: Stadtkarte Hamburgs als Grundlage für die Orientierung mit der HoloLens und die Darstellung der 3D Modelle

4.2 Projektaufbau

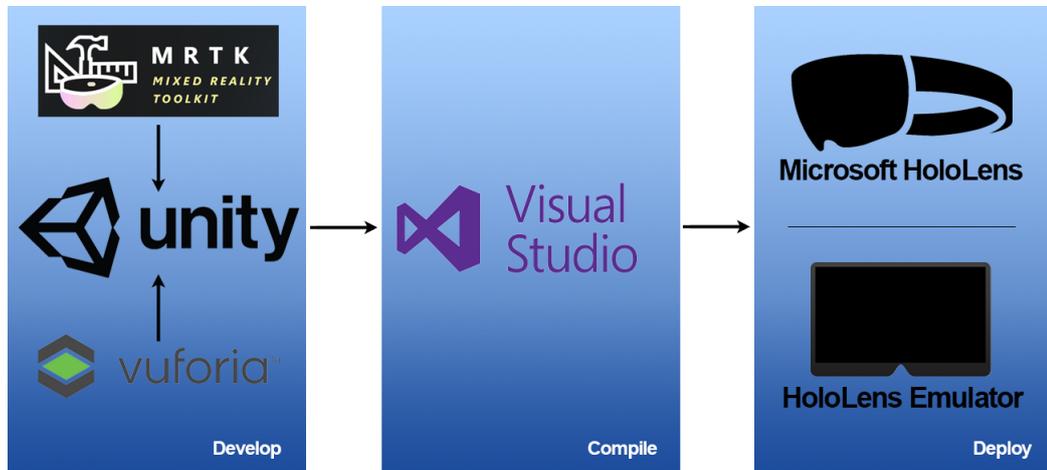


Abbildung 4.3: Aufbau der Implementierung

Zur Entwicklung einer Applikation für die Microsoft HoloLens werden mehrere Softwarelösungen verwendet. Unity ist der Mittelpunkt der Entwicklungskette und ist für den Aufbau der Szene und Erstellung der Funktionalität zuständig. Um eine Applikation für die HoloLens in Unity zu entwickeln, bedarf es an grundlegender Konfiguration¹. Zusätzlich müssen einige Objekte wie die Kamera in der Szene auf die HoloLens abgestimmt werden. Für die vorkonfigurierten Objekte und weiteres gibt es das Mixed Reality Toolkit Asset Paket². Außerdem wird für diese Implementation Vuforia benötigt, damit die Bilderkennung auch in Unity modelliert werden kann. Vuforia ist ebenfalls nicht in Unity integriert und muss über ein Asset Paket importiert werden. Folgende Möglichkeiten bieten die jeweiligen Asset Pakete:

- **Mixed Reality Toolkit:** Dieses Asset Paket bietet grundlegende Prefabs und Skripte, um die möglichen Funktionen der HoloLens in Unity zu modellieren. Außerdem befindet sich dort die HoloLens Scene Camera, die eine auf die HoloLens abgestimmte voreingestellte Kamera darstellt. Der Cursor der HoloLens ist ebenfalls als Prefab vorhanden und kann somit in Unity verwendet werden. Ein großer Satz an vorgefertigten Skripten, die die Gestiken der HoloLens verwenden und welche an Objekte angeheftet werden können, um beispielsweise drag and drop zu ermöglichen, sind vorhanden. Mit das wichtigste Feature ist jedoch das Build Window. Unity wird zur Entwicklung verwendet,

¹https://developer.microsoft.com/en-us/windows/mixed-reality/unity_development_overview (zuletzt aufgerufen am 18.12.2017)

²<https://github.com/Microsoft/MixedRealityToolkit-Unity> (zuletzt aufgerufen am 18.12.2017)

nicht aber zum Compilen der HoloLens Applikation. Visual Studio wird dazu verwendet eine HoloLens Applikation zu compilen und zu deployen. Daher muss aus dem Unity Projekt ein Visual Studio Projekt erstellt werden. Das Build Window des Asset Pakets bietet diese Möglichkeit.

- **Vuforia:** Damit die Bilderkennung möglich ist, muss eine speziell voreingestellte Kamera benutzt werden. In diesem Asset Paket ist diese vorhanden. Die sogenannte AR Camera ist direkt mit den Vuforia Einstellungen gekoppelt. Diese Kamera ersetzt die vorher verwendete Kamera als Main Camera und setzt diese, in unserem Fall die HoloLens Scene Camera, als Ankerpunkt. Die auf der Vuforia Website erstellten Targets sind immer einer Database zugeordnet. Diese Database kann durch das Asset Paket mit eingebunden werden und als Unity Prefab eingesetzt werden. Dieses Prefab ist dann das Target, welches zur Erkennung benutzt wird. Grundlegende Skripte für den Umgang mit Vuforia sind ebenfalls vorhanden. Interfaces die angesprochen werden, wenn ein Target erkannt wird und Basislogik zur Aktivierung von Objekten, wenn eben diese erkannt werden, sind bereits implementiert.

Beide Asset Pakete bietet in deren Umfang noch viele weitere Funktionen an und stellen eine Menge Prefabs bereit, die als Grundlage ohne weitere Konfiguration benutzt werden können. Die Prefabs und Skripte sind eine gute Grundlage, um die Funktionsweise zu verstehen und darauf aufbauend eigene Varianten für seinen Anwendungsfall zu erstellen. Das Skript zur Einblendung eines Objektes bei der Erkennung via Vuforia wurde beispielsweise als Grundlage dafür genutzt eine Abwandlung zu programmieren, die die Distanzmessung zum Target für die verschiedenen Modi beinhaltet.

Das resultierende Unity Projekt wird mithilfe des Mixed Reality Toolkits zu einem Visual Studio Projekt gewandelt. In Visual Studio muss vor dem Compilen und Deployen auf der HoloLens die Konfiguration durchgeführt werden. Hauptsächlich muss eine Verbindung zur HoloLens aufgebaut werden. Zum Einen kann dies über eine physische Verbindung zum PC mit einem USB Kabel realisiert werden oder zum Anderen über eine Remote Verbindung über die zugeordnete, private IP-Adresse. Sobald eine Verbindung steht muss man das Ziel noch von Debug zu Release ändern und von ARM zu x86. Über *Starten ohne debugging* wird das Projekte compiled und auf die HoloLens deployed. Die Applikation kann fortan auf der HoloLens gestartet und genutzt werden. Microsoft stellt einen HoloLens Emulator bereit. Dieser kann, sofern vorher installiert, als Ziel in Visual Studio definiert werden und die Applikation kann darauf ausgeführt werden. Der Emulator hat keinen Zugriff auf Kameras wie die

reale HoloLens, um die Umgebung wahrzunehmen. Im Emulator ist daher der Hintergrund schwarz und es sind nur die Objekte zu sehen, die man in der Szene definiert hat. Dennoch sind vordefinierte Räume, sogenannte *simulated rooms*³ eingebunden, sodass man Zugriff auf ein *Spatial Mapping Mesh* hat, genau wie sie die HoloLens erstellt, wenn ein Raum mit den Kameras aufgenommen wird. So können, obwohl im Emulator nichts zu sehen ist, Objekte in Relation zu dem *Mesh* positioniert werden. Damit die Umgebung im Emulator zumindest wahrnehmbar ist, kann das *Mesh* vom *Spatial Mapping* auf sichtbar gestellt werden.

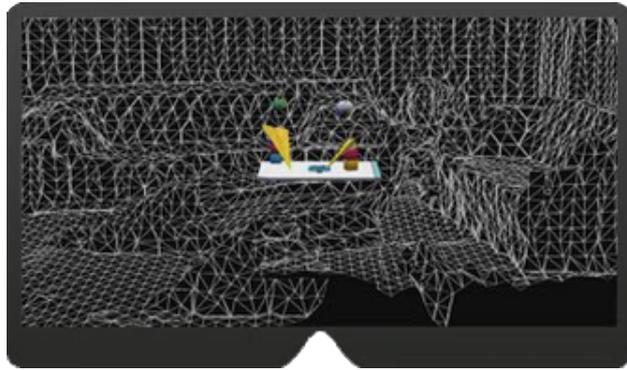


Abbildung 4.4: *Mesh* des Emulators - <https://i.ytimg.com/vi/S-517Y63Cnk/hqdefault.jpg> (zuletzt aufgerufen am 18.12.2017)

4.3 Modellierung in Unity

Damit die einzublendenden Objekte auf der Stadtkarte auch an der richtigen Position und auch richtig skaliert sind, muss die Szene in Unity richtig realisiert werden. Das Koordinatensystem in Unity entspricht dem der HoloLens. Dadurch ist es möglich, basierend auf der Größe der Stadtkarte, alle Elemente so auszurichten, wie sie auf der reellen Stadtkarte sein sollen. Wenn man die reelle Entfernung vom Ursprung der Stadtkarte zu einer Sehenswürdigkeit in X-Achsen Anteil und Y-Achsen Anteil aufspaltet und etwas in Unity auf diesen Punkt platziert, so wird in der HoloLens Applikation das Modell auch genau dort angezeigt. Damit das einfacher umsetzbar ist und nicht jeder Punkt abgemessen werden muss, wurde die Stadtkarte als Bild zur reinen Orientierung maßstabsgetreu in die Unity Szene integriert. Somit müssen die Objekte nur noch auf das Bild gesetzt werden und sind in der Applikation

³https://developer.microsoft.com/en-us/windows/mixed-reality/using_the_hololens_emulator (zuletzt aufgerufen am 18.12.2017)

4 Implementation GeoMeta

auch an eben dieser Stelle. Den Ursprung der Stadtkarte bildet der Punkt, an dem das hinzugefügte Target positioniert wird, welches zum Tracking für Vuforia genutzt wird. Normalerweise befindet sich der Ursprung der Szene aus Unity in der HoloLens Applikation an der Position der HoloLens selbst, als die Applikation gestartet wurde. Damit der Ursprung aber die Position des Targets ist, werden alle einzublendenden Objekte als Kindelement des Targets definiert. Das führt dazu, dass alle Positionen der Kindelement relativ zum Elternelement gesetzt werden und eben diese erst eingeblendet werden, wenn das Target getracked wurde. Damit das auch so funktioniert, muss die Position des Targets auch vorher definiert sein und darf auch nicht geändert werden. In dieser Modellierung befindet sich dieser in der Ecke unten rechts der Stadtkarte.

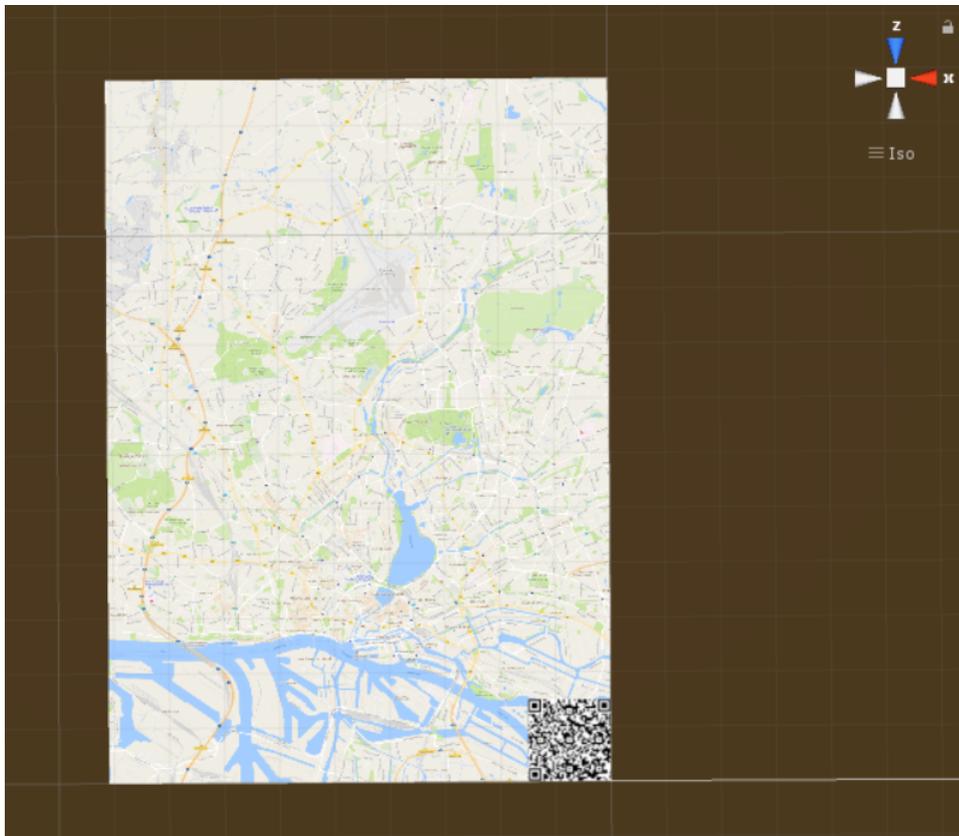
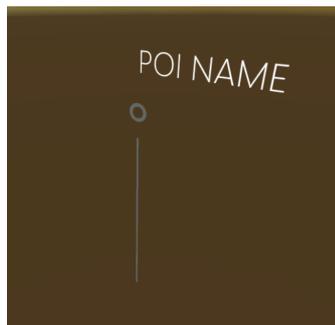


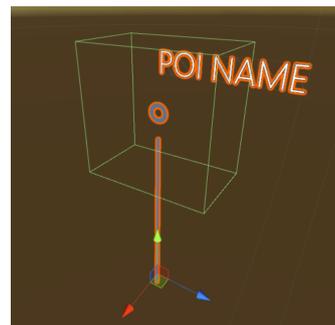
Abbildung 4.5: Die Stadtkarte Hamburgs samt QR-Target in Unity

4.3.1 Marker und 3D Modelle

In der niedrigsten Distanzebene sollen die 3D Modelle der Sehenswürdigkeiten sowie die Marker angezeigt werden. Bei den Markern handelt es sich um ein Modell, welches sich auf der jeweiligen Position der Sehenswürdigkeit befindet und dessen Namen beinhaltet. An allen Positionen, an denen ein Modell der Sehenswürdigkeit vorhanden ist, werden zunächst dessen Marker angezeigt. Diese werden eingesetzt, damit der Anwender nicht sofort alle 3D Modelle sieht und nicht zuordnen kann, um was es sich handelt.



(a) Der reine Marker



(b) Marker mit dessen Kollisionsbox in Grün

Abbildung 4.6: Die Marker, wie sie in Unity abgebildet werden

Erstellt wurde dieser mithilfe von Blender, indem ein Torus als Kopfelement verwendet wurde und ein Zylinder als Körper. POI_NAME ist der Platzhalter für den Namen der dazugehörigen Sehenswürdigkeit, welches über ein Skript zugewiesen wird. Dieser wird erst angezeigt, sobald ein Anwender mit dem Cursor der HoloLens auf den Kopf des Markers schaut. Genau genommen muss dazu wie in 4.6b zu sehen auf die grüne Box, der Kollisionsbox, geschaut werden. Da die Marker objektiv nicht sehr groß sind, ist es schwierig den Kopf des Markers anzuvisieren. Mit der Kollisionsbox wird die sogenannte Hitbox um genau diese Box erweitert, sodass der Anwender im Anwendungsfall nur in die Nähe des Kopfes schauen muss. Die Box selber ist bei der Ausführung nicht sichtbar. Sobald der Anwender auf die Box schaut, wird zusätzlich zur Anzeige des Namens noch ein Hinweiston abgespielt. Wird dann ein *AirTap* ausgeführt (siehe Abbildung 4.7), so wird das dazugehörige Modell eingeblendet und der Marker ausgeblendet. An derselben Position erscheint das Modell in Miniaturform in Relation zur Stadtkarte, um das Gesamtbild stimmig zu halten. Genau wie alle Marker simultan angezeigt werden, können auch alle Marker dem *AirTap* ausgeblendet werden, sodass alle Modelle simultan angezeigt werden. Wird *AirTap* auf ein Modell einer Sehenswürdigkeit ausgeführt, führt dies zum Detail-Mode, welches im nächsten Absatz weiter erläutert wird. Zuletzt führt

ein weiterer *AirTap* auf das Modell im Detail-Mode zur Ausblendung des Modells und zur Einblendung des dazugehörigen Markers und es befindet sich wieder im Ursprungszustand.



Abbildung 4.7: Beispielhafte Ausführung eines *AirTaps* auf den Marker der Elbphilharmonie

4.3.2 Detail-Mode

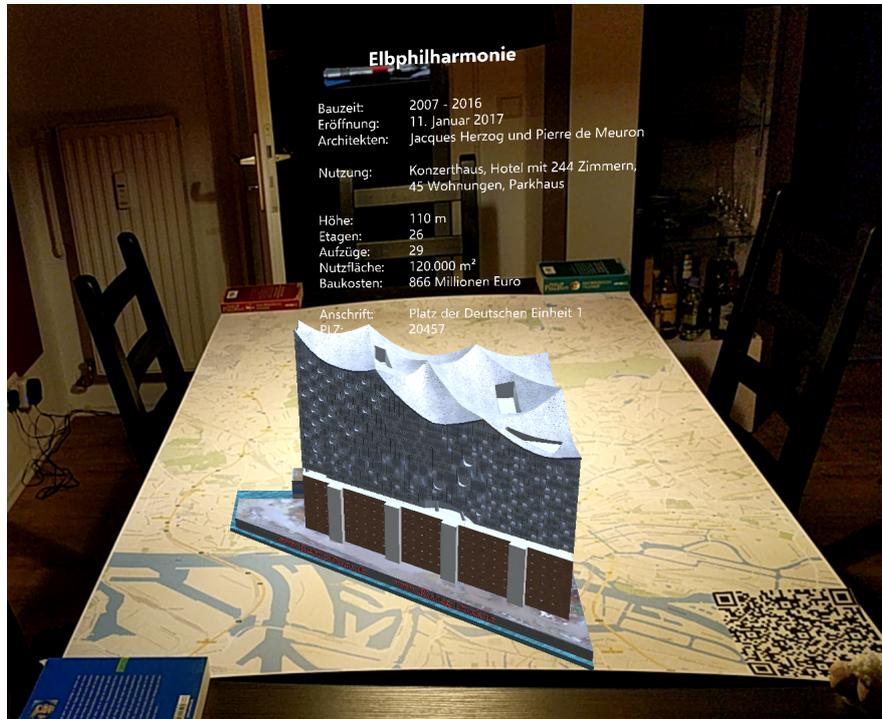


Abbildung 4.8: Die Detail-Mode Ansicht der Elbphilharmonie

Wie bereits kurz in Abschnitt 4.3.1 erwähnt, wird der Detail-Mode eines Modells initiiert, wenn ein *AirTap* auf ein anvisiertes Modell ausgeführt wird. Nachdem ein *AirTap* ausgeführt wurde, werden alle anderen Modelle oder Marker, die zu dieser Zeit noch sichtbar waren, ausgeblendet. Das Modell, welches in den Detail-Mode versetzt wurde, vergrößert sich um fast das 10-fache ihrer vorherigen Größe, sodass sie einen großen Teil der Stadtkarte abdeckt (siehe Abbildung). Zusätzlich werden Informationen zu der Sehenswürdigkeit dargestellt, wie beispielsweise die Größe, Baujahr und individuelle Informationen. Dieser Modus ist daher dafür da, dass der Anwender das Modell genauer betrachten kann und sich darüber informieren kann. Außerdem dreht sich das Modell in diesem Zustand, sodass der Anwender jede Seite betrachten kann ohne sich um die Stadtkarte drehen zu müssen. Ändert der Anwender die Höhe zur Stadtkarte, so ändert sich hier nicht die Distanzebene. Der Detail-Mode deaktiviert alles andere der Applikation und ermöglicht es so dem Anwender, sich frei zu bewegen und das Modell zu betrachten, ohne dass etwas ausgelöst wird. Wird eine weiterer *AirTap* auf das Modell ausgeführt, so minimiert sich das Modell bis es ganz verschwindet. Dies deaktiviert den Detail-Mode und es wird wieder die Distanzebene aktiviert, die zur

aktuellen Höhe des Anwenders zur Stadtkarte gehört. Befindet sich der Anwender wieder in der Distanzebene, in der die Marker und Modelle dargestellt werden, so wird nach dem Detail-Mode wieder der Marker zu der Sehenswürdigkeit angezeigt, der zuvor im Detail-Mode war. Die anderen Modelle oder Marker bleiben davon unberührt und befinden sich wieder in dem Zustand, wie sie es vor dem Detail-Mode waren.

4.3.3 Distanzebenen

Wichtiger Bestandteil der Implementation ist die Umsetzung der Distanzebenen. Wie in den Anforderungen definiert, sollen diese je nach Nähe zur Stadtkarte passende Informationen bereitstellen bis zu niedrigsten Distanzebene, auf der der Anwender die 3D Modelle der Sehenswürdigkeiten betrachten kann. Auf Basis des Single Target Reference Ansatzes ist die Anzahl der Distanzebenen frei skalierbar, denn ein fixer Distanzwert zum Target gibt an, auf welcher Distanzebene man sich befindet und diese können frei gewählt werden. Für die Erstellung eines Prototyps wurden drei Distanzebenen definiert.

Far-Mode

Diese Ebene stellt die höchste Distanzebene dar. Sie gilt zur Darstellung allgemeiner Informationen Hamburgs. Darunter fallen unter anderem Informationen wie die Einwohnerzahl oder der Fläche und weitere Punkte.



Abbildung 4.9: Der Far-Mode mit den allgemeinen Informationen zu Hamburg

Between-Mode

Diese stellt die nächst nähere Distanzebene dar. In dieser werden auf der Stadtkarte selbst die Namen der Bezirke eingeblendet. Somit stellt dies die erste Verfeinerung der Stadtkarte selbst dar.



Abbildung 4.10: Der Between-Mode mit den Namen der Bezirke von Hamburg

Close-Mode

Diese stellt die niedrigste Distanzebene dar. Auf dieser sieht der Anwender die 3D Modelle der Sehenswürdigkeiten und die dazugehörigen Marker.



Abbildung 4.11: Der Close-Mode mit den initialen Markern der verfügbaren Sehenswürdigkeiten

Sobald das Target mit der HoloLens getracked wurde, ist dessen Position fix in der Szene und es kann auf die Koordinaten zugegriffen werden. Das heißt, dass das Target als Anker zur Distanzmessung genutzt werden kann. Relativ zum Target ist es möglich zur Laufzeit auf die Position der HoloLens zuzugreifen. Eine Position wird in Unity als Vector3 Objekt initialisiert und beinhaltet den entsprechenden x, y und z Wert. Da bei dieser Implementation ein hinzugefügtes Element, der QR-Code, als Target fungiert, ist die Realisierung der Distanzebenen nicht durch die gesamte Entfernung vom Target bis zur HoloLens verwendbar, sprich die Länge des Vektors, der sich durch die Subtraktion beider Positionsvektoren ergibt. Würde man das so berechnen, dann würde der Anwender verschiedene Distanzebenen dargestellt bekommen, je nachdem an welcher Seite der Stadtkarte sich dieser mit der HoloLens befindet. Laut Anforderung soll die Entfernung zur gesamten Stadtkarte für das Einblenden der jeweiligen Distanzebene zuständig sein, da das auch die Immersion und die Intuition verbessert. Ein Workaround ist die Verwendung der Differenz der y Werte, somit also der Höhenunterschied zwischen HoloLens und Stadtkarte. Dadurch ist es egal wo der Anwender mit der HoloLens positioniert ist. Würde der Anwender sich weiter weg von der Stadtkarte entfernt befinden und dort die Höhe verändern, so würde dies zwar auch zu ein Wechsel der Distanzebenen führen, jedoch passiert das im Falle der Verwendung dieses Anwendungsfalls

nicht.

In dieser Implementation sind die Ebenen jeweils 15 cm voneinander entfernt. Wird das Target das erste Mal mit der HoloLens getracked, so befindet sich die Applikation im Far-Mode. Da hier allgemeine Informationen zu Hamburg eingeblendet werden, stellt dies einen guten Einsatzpunkt dar. Nähert sich der Anwender der Stadtkarte mit der HoloLens um 15 cm, so wechselt der Modus in den Between-Mode und noch weitere 15 cm und es wird der Close-Mode.

4.4 Scripts und Animationen

Das Herz der Applikation befindet sich in den Skripten. Damit die Logik der Distanzebenen, das Tracking des Targets, die Erkennung der *AirTaps* und das Anschauen der Marker, die Drehung der Texte zum Anwender und der Detail-Mode möglich ist, müssen Skripte programmiert und an Objekte der Unity Szene gebunden werden. Im Anwendungsfall werden an mehreren Stellen Text dargestellt und dem Anwender präsentiert. Die Marker haben einen Text, wenn man sie anschaut und zeigen den Namen der Sehenswürdigkeit; im Detail-Mode werden Informationen zum Modell angezeigt; im Far-Mode werden allgemeine Informationen zu Hamburg dargestellt. Diese Texte werden mit Skripten so transformiert, dass sie zur Laufzeit immer zum Anwender ausgerichtet werden. Somit kann sich der Anwender frei um die Stadtkarte bewegen und weiterhin den jeweiligen Text lesen. Die Namen der Bezirke im Between-Mode bleiben davon unberührt und liegen planar auf der Stadtkarte selbst.

Das Vuforia Asset Paket bietet wie bereits beschrieben einige Prefabs an, die vorkonfiguriert genutzt werden können. Darunter fällt das standardmäßige TrackingPrefab, welches genutzt wird um das gewünschte Bild als Tracking-Element auszuwählen und alle Kinder dieses Objekts in der Unity Szene bei der Erkennung darzustellen. Für diesen Anwendungsfall wurde das dazugehörige Skript als Grundlage verwendet und daraufhin erweitert, dass zusätzlich zum Erkennen des Targets noch die Entfernung hinzugefügt wurde, sodass die Distanzebenen realisiert werden konnten. Alle darzustellenden Objekte dieses Anwendungsfalls sind abhängig von der Distanz des Anwenders zum Target. Daher ist das TrackingPrefab der Parent all dieser Objekte und reguliert über das Skript welche anhand der Distanz angezeigt werden sollen. Im vordefinierten Szenario befindet sich der Anwender bei der Höhe, auf dem das Target getracked wird, im Far-Mode und ändert bei je 15 cm zum Between- und Close-Mode. Das Skript wurde dahingehend angepasst, sodass über öffentliche Variablen in der Unity

Szene die Höhe der jeweiligen Distanzebene eingestellt werden kann.

Der Detail-Mode ist besonders, da dieser die Distanzebenen deaktivieren muss, damit nicht die Inhalte der Distanzebenen dargestellt werden, wenn der Anwender die Höhe verändert, während dieser das Modell betrachtet. Daher ist ein weiteres Kriterium für die Distanzebenen, dass zusätzlich zur Höhe auch geprüft werden muss, ob sich ein Modell im Detail-Mode befindet. Damit sich die Szene nach dem Detail-Mode wieder im Ursprungszustand befindet, muss gespeichert werden, welche Marker und welche Miniaturmodelle angezeigt werden. Beendet der Anwender den Detail-Mode eines Modells, so verschwindet das Modell und der dazugehörige Marker wird wieder dargestellt. Befindet der Anwender sich dann im Close-Mode, so wird dieser Marker angezeigt und wieder die Miniaturmodelle und Marker, wie sie vor dem Detail-Mode angezeigt wurden. Die Darstellung der Informationen einer Sehenswürdigkeit im Detail-Mode geschieht ebenfalls über ein Skript. Damit dies dynamisch für eine beliebige Anzahl an Informationen funktioniert, wurden die Prefabs der 3D Modelle so realisiert, dass diese einen direkten Child namens Informations haben. Das GameObject Informations wiederum hat eine flexible Anzahl an Kinder, die die Informationen darstellen. Das Skript eines Modells wird beim Detail-Mode alle Child-Elemente von Informations einblenden und beim Beenden auch alle wieder ausblenden. Somit kann ohne weitere Konfiguration weitere Elemente hinzugefügt werden, die zur Darstellung von Informationen genutzt werden sollen.

Zur Veranschaulichung wurden kurze Animationen implementiert. Führt der Anwender ein *AirTap* auf einen Marker aus, so wird der Alpha Wert des Marker Modells linear innerhalb einer Sekunde von eins auf null gesetzt, wodurch ein *Fade Out* realisiert wird. Zeitgleich startet die Animation des dazugehörigen Modells der Sehenswürdigkeit, wobei dieser zunächst aktiviert wird und so klein ist, dass das Modell zunächst nicht zu erkennen ist. Innerhalb einer Sekunde wird das Modell von dieser Größe auf die Miniaturgröße gebracht. Bei der Ausführung des *AirTaps* auf ein Miniaturmodell, wechselt das Modell in den Detail-Mode und dazu wird ebenfalls eine Animation abgespielt. Bei dieser Animation wird das Modell innerhalb von drei Sekunden auf ungefähr das Zehn fache vergrößert und im dazugehörigen Skript wird eine Rotation aktiviert, die langsam das Modell zur Betrachtung drehen lässt. Außerdem wird der dazugehörige Text, der weitere Informationen zur Sehenswürdigkeit beinhaltet, zwei Sekunden nach Start der Animation des Modells aktiviert und dessen Alpha Wert steigt linear von null auf eins binnen einer Sekunde und ist fertig, sobald auch das Modell die vollständige Größe erreicht hat. Der Text realisiert somit ein *Fade In*. Die Rotation des Modells hat keinen Effekt auf den Text, sodass der Text weiterhin fortlaufend zum Anwender

ausgerichtet wird, egal wo sich dieser befindet. Zuletzt gibt es noch die Animation zum Ausblenden des Modells beim Verlassen des Detail-Modes. Sobald der Anwender ein *AirTap* auf das Modell im Detail-Mode ausführt, so startet die Animation und verkleinert das Modell innerhalb von drei Sekunden auf eine nicht mehr sichtbare Größe und deaktiviert das Modell am Ende. Gleichzeitig wird der Alpha Wert des Textes binnen einer Sekunde linear von eins auf null gesetzt, wodurch ein weiterer *Fade Out* realisiert wird.

5 Evaluation der Umsetzung

5.1 Auswertung

Im Rahmen der Implementation des Anwendungsfalls wurden die funktionalen Anforderungen explizit umgesetzt, um eine Funktionalität gemäß der Konzeption zu gewährleisten. Neben den funktionalen Anforderungen, die primär für die Lauffähigkeit zuständig sind, wurden noch nichtfunktionale Anforderungen definiert, die zusätzliche Bedingungen für ein besseres Erlebnis des Anwendungsfalls sicherstellen. Zunächst werden die nichtfunktionalen Anforderungen der Reihenfolge nach auf Gegebenheit überprüft.

Intuitiv: Die Applikation soll so realisiert werden, dass die Benutzung dieser mit dem Wissen der Bedienungsweise der HoloLens ohne Einweisung möglich ist

- Dem Anwender der HoloLens müssen die Gestiken, insbesondere der *AirTap*, bekannt sein, um die Anwendung bedienen zu können. Da diese jedoch auch für den generellen Umgang der Brille notwendig sind, sind diese als intuitiv anzusehen. Des Weiteren wurde der Stadtkarte ein QR-Code hinzugefügt, damit alle computergenerierten Objekte sich anhand diesem an der analogen Stadtkarte ausrichten können. Da ein QR-Code im allgemeinen Kontext dafür verwendet wird, dass bei der Aufnahme mit einer Kamera weiterführende Informationen dargestellt werden, ist es für den Anwender auch hier intuitiv, dass ein Blick auf diesen QR-Code mit aufgesetzter Brille etwas auslöst. In diesem Fall werden bei der Erkennung die Objekte des Far-Modes angezeigt. Beginnt der Anwender die Stadtkarte zu untersuchen, so ist es sehr wahrscheinlich, dass dieser die weiteren Modi, den Between- und Close-Mode, ohne Erklärung entdeckt, da die jeweiligen Distanzebenen nur 15 cm voneinander entfernt sind und diese bei der Bewegung um die Karte schnell erreicht werden. Fängt der Anwender beispielsweise an die Stadtkarte näher zu untersuchen, um die Schrift auf dieser lesen zu können, muss bereits eine nähere Betrachtung stattfinden. Dort würden bereits die anderen Modi entdeckt werden. In den Testdurchläufen wurden die 15 cm oft zufällig überschritten während man sich bei der Benutzung bewegt hat, da der benötigte Höhenunterschied

zu den Modi bei der natürlichen Bewegung des Kopfes während der Beobachtung der Karte automatisch erreicht wird. Die möglichen *AirTaps* sind im Close- und Detail-Mode möglich. Im Anwendungsfall ergeben sich initial prinzipiell drei Gelegenheiten, zu denen es denkbar wäre eine Interaktion auszuführen. Im Far-Mode könnte eine Interaktion mit dem allgemeinen Text vorgenommen werden, im Between-Mode mit den Namen der Bezirke und im Close-Mode mit den Markern. Da es neben dem *AirTap* von Haus aus nur noch die *Bloom* Gestik gibt, die jedoch nicht primär zur Interaktion genutzt wird und initial nur die drei möglichen Elemente zur Verfügung stehen, sollte bei einer explorativen Herangehensweise die Interaktion mit den Markern im Close-Mode ohne Erklärung stattfinden. Zusätzlich wird bei der Betrachtung der Marker der Name der jeweiligen Sehenswürdigkeit eingeblendet und ein Hinweiston abgespielt, was explizit darauf hinweist, dass dort eine Interaktion möglich ist. Die *AirTaps*, die auf die Miniaturmodelle der Sehenswürdigkeiten im Close-Mode und auf das jeweilige Modell im Detail-Mode möglich sind, sind vergleichsweise zu den anderen Funktionen des Anwendungsfalls weniger intuitiv. Ohne Erklärung kann der Anwender nicht explizit wissen, dass auch dort *AirTaps* möglich sind. Dennoch würde der Anwender, wenn dieser den *AirTap* auf einen Marker verwendet hat, versuchen weitere *AirTaps* zu verwenden. So sollte auch das ohne Einweisung passieren. Eine kurze Erläuterung im Vorfeld für den Anwender, dass dieser während der Benutzung des Anwendungsfalls versuchen soll *AirTaps* zu verwenden, würde nahezu garantieren, dass dieser im Endeffekt alle Funktionalitäten findet und verwendet.

Reaktionszeit: Die Darstellung der Objekte verschiedener Distanzebenen soll sofort geschehen, sobald eine Distanzebene erreicht wurde

- Damit das verifiziert werden konnte, wurde ein UI Canvas Element in Unity hinzugefügt. Dabei handelt es sich um ein Objekt, welches sich planar im Sichtfeld des Anwenders durch die Brille befindet. Dadurch wurde am Displayrand permanent ein Distanz-Tracker eingebunden, der darstellt, wie weit sich der Anwender von dem realen Objekt befindet, welches angeschaut wird. Das ist möglich, da die HoloLens, wie in Kapitel 2.2.2 beschrieben, *Spatial Mapping* verwendet, um die reale Umgebung wahrzunehmen und darauf basierend ein Raum zu erstellen, damit die eingeblendeten computergenerierten Objekte auf dieser Basis skaliert und positioniert werden können. Der Cursor der HoloLens wird dabei immer auf die Oberflächen des wahrgenommenen Raumes gelegt. So ist es zur Laufzeit möglich, über Vektorrechnung die Distanz zwischen der HoloLens und dem Cursor zu berechnen, was somit die Darstellung der

Entfernung des Anwenders zur Stadtkarte ermöglicht. Das Ergebnis dieser Berechnung wird dem UI Canvas Element übermittelt und ermöglicht somit, dass während der Ausführung des Anwendungsfalls überprüft werden kann, wann ein Distanzebenenwechsel stattfindet und welche Entfernung die HoloLens dabei zur Stadtkarte aufweist. Unter diesen Bedingungen wurde festgestellt, dass der Wechsel einer Distanzebene auch genau dann passiert, wenn der entsprechende Distanzwert erreicht wurde. Es konnte keine Verzögerung festgestellt werden.

Performanz: Die Applikation soll bei der Darstellung der Szene mit den dazugehörigen Objekten mindestens 24 FPS aufweisen

- Um diese Anforderung zu verifizieren, gibt es innerhalb des Mixed Reality Toolkit Asset Pakets bereits ein Prefab, welches zur Laufzeit der Anwendung ein Objekt im Sichtbereich des Anwenders mit der HoloLens platziert, welches die momentanen FPS anzeigt. Dieses Objekt bewegt sich mit der Änderung der Blickrichtung des Anwenders, sodass sich diese zur jeder Zeit im Sichtbereich des Anwenders befindet. Auf diesem Weg wurden alle möglichen Darstellungen hervorgerufen und überprüft, wieviele FPS erreicht wurden. Interessant sind zur Überprüfung dieser Anforderung die Szenarios, in der potentiell eine hohe Leistungsbeanspruchung der Hardware vorliegt. Im Vorfeld lässt sich bereits vermuten, dass während des Detail-Modus eines Modells wahrscheinlich der größte Leistungsanspruch liegt, da dort die größten Modelle mit zusätzlichen Informationen als Textobjekte angezeigt werden. Auch in der Praxis hat sich herausgestellt, dass dort die FPS am niedrigsten waren. Dabei ist ausschlaggebend, aus wievielen Knoten und Kanten ein 3D Modell besteht, da beobachtet werden konnte, dass dies bei verschiedenen Modellen der Faktor war, der antiproportional zur FPS steht. Der in diesem Anwendungsfall in diesem Kontext größte Leistungseinbruch konnte somit dann erschaffen werden, wenn der Detail-Mode der Elbphilharmonie vorliegt. Dieses Modell besteht aus 95.630 Knoten und 191.472 Polygonen und ist somit ein sehr detailliertes Modell. Selbst im Detail-Mode der Elbphilharmonie wird die 24 FPS Grenze nicht unterschritten und befindet sich bei rund 30 FPS.

Optische Anforderung: Die verwendeten 3D Modelle sollen den realen Sehenswürdigkeiten nachempfunden werden, sodass eine Vorstellung des realen Erscheinungsbildes der Sehenswürdigkeit vermittelt wird

- Für diesen Anwendungsfall wurden keine eigenen 3D Modelle der Sehenswürdigkeiten erstellt. Hier wurde primär auf Modelle der Modell-Datenbank von SketchUp¹

¹<https://3dwarehouse.sketchup.com/index.html> (zuletzt aufgerufen am 18.12.2017)

zurückgegriffen, die hauptsächlich frei verfügbare Exemplare der Sehenswürdigkeiten anbietet, die von Google Maps extrahierbar sind. Speziell für die Elbphilharmonie wurde mir von RICHTIG DRUCK² ein sehr genaues und besonders originalgetreues Modell für diese Ausarbeitung zur Verfügung gestellt. Somit ist eine reelle erweiterte Visualisierung der Sehenswürdigkeiten im Rahmen dieses Anwendungsfalls möglich. Die verwendeten Modelle weisen somit genügend einzigartige Merkmale auf, sodass das reale Erscheinungsbild der jeweiligen Sehenswürdigkeit vermittelt wird.

Umgebungsanforderung: Die von der HoloLens zur Verfügung stehenden Gestiken sollen zur Bedienung verwendbar sein

- Prinzipiell wäre es auf vielen Weisen möglich eine Verwendung einer HoloLens Applikation zu ermöglichen, ohne explizit die Gestiken zu verwenden. Das Betrachten der Szene durch die HoloLens, indem der Cursor auf computergenerierte Objekte platziert wird, nennt sich *Gaze*. Das Mixed Reality Toolkit Asset Paket bietet speziell dafür ein *GazeManager* Interface, was die Erkennung eines *Gazes* ermöglicht. Daraufhin können Aktionen implementiert werden, die die Steuerung einer HoloLens Applikation darüber ermöglicht. Dies wäre auch für diesen Anwendungsfall denkbar gewesen, wodurch jedoch einige Designentscheidungen geändert werden müssten, da beispielsweise aktuell der *Gaze* dafür verwendet wird zu erkennen, wann ein Marker angeschaut wird, damit der dazugehörige Name der Sehenswürdigkeit dargestellt werden kann. Dies würde bei der Steuerung durch den *Gaze* in diesem Fall dazu führen, dass bei der reinen Betrachtung um den Namen zu sehen bereits das Modell angezeigt werden würde. Daher müsste man hier beispielsweise bereits alle Namen im Vorfeld anzeigen lassen, sodass der *Gaze* wieder frei wäre, um die Darstellung des Modells zu realisieren. Für diese Implementierung wurde die Verwendung des *AirTaps* durchgesetzt, um diese Anforderung zu erfüllen. Dies stellt auch die bessere Lösung dar, da so der *Gaze* als weiteres Steuerungselement verwendet werden kann. Die weitere native Gestik, der *Bloom*, ist von der HoloLens bereits standardmäßig mit dem Beenden einer Applikation und dem Öffnen des Hauptmenüs verknüpft, wodurch eine eigene Re-Implementierung dieser Gestik für andere Zwecke der generellen Steuerung der HoloLens widersprechen würde und somit eher zur Verwirrung führen würde. Eigene Gestiken wurden für diesen Anwendungsfall nicht implementiert.

Offline-Fähigkeit: Die Applikation soll ohne Internetverbindung verwendbar sein, um sie unabhängig nutzen zu können

²<https://richtig-druck.de/> (zuletzt aufgerufen am 18.12.2017)

- Es ist nicht notwendig, dass die HoloLens für die Ausführung der Applikation eine bestehende Internetverbindung hat. Alle Inhalte, die verwendet werden, befinden sich lokal auf der HoloLens und werden von dort aus in die Anwendung geladen. Theoretisch wäre es möglich, dass beispielsweise die verwendeten 3D Modelle der Sehenswürdigkeiten zur Einsparung des Festplattenspeichers nicht lokal abgelegt werden, sondern zur Laufzeit über das Internet runtergeladen werden. Jedoch würde dies die Benutzung der Applikation einschränken, sodass diese ohne Internetverbindung keine Modelle anzeigen könnte oder nur Platzhalter, was den Nutzen drastisch senken würde.

Neben den funktionalen und nichtfunktionalen Anforderungen zeigt sich, dass der entwickelte Anwendungsfall mit der HoloLens bereits gut umsetzbar war. Die Positionierung der Objekte auf der analogen Stadtkarte mit dem QR-Code als Anker funktioniert sehr gut und hat in der Praxis nur selten dazu geführt, dass die Position nicht korrekt eingehalten wurde. Da der Wechsel und die generelle Darstellung der Elemente der Distanzebenen zuverlässig funktioniert, erweitert die Applikation die analoge Stadtkarte gut und bietet dem Anwender passende und vor allem vorher nicht vorhandene, relevante Informationen. Die HoloLens stellt die computergenerierten Objekte klar und deutlich dar, wodurch ein reeller Mehrwert generiert werden kann, indem der Anwender die originalgetreuen Modelle näher betrachten kann, ohne vor Ort sein zu müssen. Die Betrachtung der 3D Modelle durch die Brille stellt eine gute Erweiterung zur traditionellen Beobachtung auf einem Display dar, da sich das Bild direkt vor den Augen des Anwenders befindet und dazu die Möglichkeit besteht, sich physisch um das Modell zu bewegen und somit Beobachtungen aus verschiedenen Perspektiven zu realisieren. Die Immersion ist in der Summe hoch und stellt eine positive Erweiterung der analogen Stadtkarte dar. Die Erstellung eines funktionierenden Prototypen mit definiertem Funktionsumfang für diesen Anwendungsfall ist somit gelungen.

5.2 Schwierigkeiten und Grenzen

Bereits während der Konzeption und später bei der Implementierung sind einige Probleme aufgetreten. Bei der Wahl des Trackingsverfahrens hat sich gezeigt, dass das Erkennen von Targets zum heutigen Zeitpunkt eine Schwierigkeit darstellt. Für das Tracking von Bildern wurde Vuforia verwendet, was eine führende native Lösung darstellt, was die Implementierung von Vuforia in Unity im neusten Update zeigt³. Damit ein Target zur Laufzeit auch erkannt wird, ist es notwendig, dass dieses genügend Features aufweist, sodass ein hohes

³<https://library.vuforia.com/articles/Training/getting-started-with-vuforia-in-unity-2017-2-beta.html> (zuletzt aufgerufen am 18.12.2017)

Rating erreicht werden kann. Dies setzt die Grundlage für diesen Anwendungsfall, damit alle Objekte korrekt positioniert und skaliert werden können. Optimal wäre die Verwendung der Stadtkarte selbst als Multi-Level Target gewesen, was aufgrund des schlechten Ratings der resultierenden Bilder der Stadtkarte zu diesem Zeitpunkt nicht möglich ist.

Des Weiteren ist die Erkennung des realen Umfelds mit Hilfe des *Spatial Mappings* stark von der Umgebungsbeleuchtung anhängig. Bei schlechten Lichtverhältnissen wird ein ungenaues und teils verkehrtes *Mesh* aufgezeichnet, wodurch die Positionierung und Skalierung der computergenerierten Objekte nicht zuverlässig funktioniert. Außerdem kann dies dazu führen, dass der QR-Code nicht erkannt wird, wodurch die Anwendung nicht gestartet wird. Dafür müssen jedoch sehr schlechte Verhältnisse vorliegen, die selbst eine generelle Betrachtung der analogen Stadtkarte beeinträchtigen würde.

In der Praxis hat sich gezeigt, dass das kleine FOV der HoloLens ein Problem bei der Darstellung der Modelle im Detail-Mode darstellt. Außerhalb des Detail-Modus sind die computergenerierten Objekte gut innerhalb des FOV erfassbar und kann somit vom Anwender vollständig wahrgenommen werden. Sobald größere Modelle dargestellt werden, wie es im Detail-Mode der Fall ist, ist es nahezu unmöglich eine nähere Betrachtung durchzuführen und das gesamte Modell im Sichtbereich darzustellen. Außerdem werden zu dem Modell der Sehenswürdigkeit noch weitere Informationen eingeblendet, die dadurch nur betrachtet werden können, wenn man einen großen Teil des Modells außerhalb des Sichtbereichs setzt oder eine größere Entfernung einhält.

Während der Bearbeitung des Anwendungsfalls wurde des Öfteren das Google Pixel Smartphone verwendet, um die Anwendung darauf laufen zu lassen. Dabei wurde stark bemerkbar, dass die Orientierung und damit verbundene Fixierung des Smartphones in der Szene der Anwendung Schwierigkeiten aufwies. Während die HoloLens zuverlässig die Positionierung und Skalierung der Objekte nach dem Tracking des QR-Codes umsetzt, verliert das Pixel Smartphone schnell den Bezug zur Szene und positioniert die Objekte falsch oder verliert gar das Tracking, womit die Objekte nicht mehr angezeigt werden. Dies zeigt jedoch, dass die HoloLens da einen großen Vorteil aufweist und diese Aufgabe zuverlässig ausführt. Dies gilt nur für den Fall, dass die Implementation so wie auf der HoloLens auf dem Smartphone realisiert wird. Wird ein Smartphone verwendet, welches das ARKit oder ARCore Framework unterstützt und die Implementation so verändert, dass diese auch genutzt werden, funktioniert die Positionierung und Skalierung ähnlich wie bei der HoloLens.

6 Zusammenfassung

Diese Ausarbeitung hat das Themenfeld Augmented Reality behandelt und hat dabei verschiedene Aspekte erläutert. Zunächst wurde der Begriff erklärt und der Kontext beschrieben, damit deutlich wird, wie sich dieses Feld positioniert und von anderen abgrenzt. Dabei wurde der Unterschied zur Virtual Reality verdeutlicht und anhand des Realitäts-Virtualitäts-Kontinuum der Zusammenhang verdeutlicht. Darauf aufbauend wurden die verschiedenen Darstellungs- beziehungsweise Wahrnehmungsarten vorgestellt; diese erstrecken sich von head-attached Varianten über hand-attached zu spatial Ansätzen. Nach einem detaillierten Einblick in die Historie zu Augmented Reality wurde speziell die Microsoft HoloLens vorgestellt, die für die praktische Umsetzung des Anwendungsfalls verwendet wurde. Aufbauend auf dieser Grundlage wurde in Idee und Konzeption anhand einer Anforderungsanalyse definiert, was der Prototyp nach der Implementation leisten muss. Die beiden Trackingverfahren Multi-Level Target und Single Target Reference wurden ins Detail ausgearbeitet um zu definieren, welches das am besten geeignete Verfahren für diesen Anwendungsfall darstellt. Zusammen mit dem ausgewählten Trackingverfahren des Single Target Reference Ansatzes und der Anforderungsanalyse, wurde die Durchführung der Implementation geschildert. Über die Erstellung einer geeigneten Stadtkarte Hamburgs, die Erläuterung der Vorgehensweise zur Erstellung der Applikation auf der HoloLens und der Programmierung der Funktionalität durch Skripte und Animationen der Unity Elemente, wurde das Ergebnis des erstellten Prototyps dargestellt. Dieses stellt die Grundlage der daraufhin durchgeführten Evaluation samt Auswertung des Anwendungsfalls und der Auflistung von Schwierigkeiten und Grenzen der Implementation dar.

Die Implementation des Prototyps lief in Zusammenarbeit von Unity mit der HoloLens nach persönlichem Empfinden sehr gut. Mit den richtigen Asset Paketen Mixed Reality Toolkit und Vuforia ließ sich alles nativ in der Unity Szene modellieren. Um auf die HoloLens spezifischen Möglichkeiten zuzugreifen, wie die Erkennung eines *AirTaps* oder dem *Gaze*, die Beobachtung eines virtuellen Objekts, bietet das Mixed Reality Toolkit Interfaces, die in allen Skripten in C# verwendet werden können. Somit unterscheidet sich die Entwicklung des Prototyps für

die HoloLens kaum von der Entwicklung für andere Plattformen. Der direkte Vergleich vom Smartphone, in diesem Fall dem Google Pixel, und der HoloLens hat verdeutlicht, wie viel besser das Tracking und die Positionierung mit Vuforia auf der HoloLens funktionieren. Dank der Tatsache, dass die HoloLens ein *Mesh* der Umgebung erstellt und die Objekte der Szene sich daran orientieren können, ermöglicht die im Vergleich zum Smartphone präzise Positionierung der Objekte, die nach dem Tracking eines Markers dargestellt werden. Während die Objekte schnell den Anker in der Szene verlieren und somit falsch oder gar nicht angezeigt werden nachdem das Smartphone bewegt wird und der Marker nicht mehr im Sichtbereich ist, bleiben die Objekte bei der HoloLens an der richtigen Stelle, selbst wenn man sich im Raum bewegt oder auch den Raum verlässt und wiederkehrt. Wird ein Smartphone verwendet, welches in der Lage ist das ARKit oder das ARCore Framework zu nutzen und wird die Implementation daraufhin für diese Frameworks angepasst, so funktioniert die Darstellung auf dem Smartphone ähnlich wie auf der HoloLens und die Objekte verlieren nicht den Anker in der Szene. Auch wenn die HoloLens technisch noch ausbaufähig ist, was besonders den FOV und die Darstellungsqualität angeht, war die Umsetzung der bestehenden Möglichkeiten und das generelle Erstellen einer Applikation für die Brille sehr angenehm und bietet viel Spielraum zur Verwirklichung vieler Szenarien.

7 Ausblick

7.1 Weiterentwicklung GeoMeta

Der ausgearbeitete Anwendungsfall lässt sich in vielen Aspekten erweitern. Basierend auf der bestehenden Funktionalität, kann die Anzahl der verfügbaren Sehenswürdigkeiten erhöht werden. Dafür müssen lediglich passende 3D Modelle zur Verfügung stehen, jeweils ein Marker verwendet werden, die nötigen Skripte angehängt werden und die Animationen hinzugefügt werden, damit der Wechsel von Marker über Modell zum Detail-Mode stimmig verläuft. Bisher wurden rein textliche Informationen einer Sehenswürdigkeit im Detail-Mode implementiert. Möglich sind aber auch akkustische Elemente, wie Konzertausschnitte bei Betrachtung der Elbphilharmonie oder bildliche Elemente wie reale Fotos zum Vergleich des Modells mit der Realität. Des Weiteren bietet das Mixed Reality Toolkit von Microsoft Interfaces zum Hand-Tracking an, wodurch weitere Gestiken und Aktionen mit der Hand implementierbar sind. So ist es möglich, dass bei der Ausführung eines *AirTaps*, wobei die Finger bei der Ausführung nicht wieder auseinander gehen sondern zusammengedrückt bleiben, die darzustellenden Objekte in der Szene verschoben oder rotiert werden. Dies könnte dazu genutzt werden die Modelle im Detail-Mode selber zu rotieren oder auch an einer Position zu verschieben, um eine bessere und individuellere Betrachtung zu ermöglichen. Auch denkbar wäre eine Modell-Datenbank der im Anwendungsfall verfügbaren Modelle zu implementieren. Als Anlehnung der von Haus aus bestehenden Anwendung *Hologram* der HoloLens, kann im Anwendungsfall eine Liste der verfügbaren Modelle angezeigt werden, wodurch der Anwender einen Überblick über alle Modelle erhält. Dies kann als Grundlage dazu verwendet werden, dass dem Anwender basierend auf dem Modell, welches ausgewählt wurde, nur noch der Marker angezeigt wird, der auf diese Sehenswürdigkeit verweist. Gerade wenn GeoMeta soweit ausgebaut wird, dass zu viele Marker gleichzeitig dargestellt werden, bietet diese Implementation die explizite Suche nach einer Sehenswürdigkeit, basierend auf der Vorschau des 3D Modells in der Liste. Eine weitere Implementationsmöglichkeit wäre die Funktion einer Augmented Reality Tour durch verfügbare 3D Modelle der Sehenswürdigkeiten. Beispielsweise könnte zusätzlich zum Detail-Mode ein weiterer Modus eingebaut

werden, der, sofern das Modell auch mit den inneren Komponenten modelliert wurde, eine Tour durch das Innenleben gewährt.

Was zu diesem Zeitpunkt aufgrund dem Stand von Vuforia nicht möglich ist, ist die Realisierung des Multi-Level Target Ansatzes. Besonders intuitiv und vorteilhaft für weitere Szenarien wäre die Realisierung der Targets explizit als Ausschnitt der Stadtkarte, in der die Sehenswürdigkeiten sich befinden. Der grundlegende Vorteil der daraus resultiert ist, dass die analoge Stadtkarte nicht mehr zwangsweise genutzt werden muss. Da die Targets auf der Karte von Google Maps basiert, wäre es somit möglich, dass der Teil der Stadtkarte auf Google Maps auf dem Smartphone angezeigt wird, auf die richtige Größe skaliert wird und von der HoloLens erkannt wird, sodass das dazugehörige Modell auf dem Smartphone dargestellt wird. Dabei gibt es viele Devices, die Google Maps Karten anzeigen können, wodurch neben Smartphones noch weitere Geräte als Grundlage zum Tracking verwendet werden können. Diese Form der Implementation würde die Anwendung noch eigenständiger machen, da neben der HoloLens kein weiteres spezielles Element benötigt wird, da man heutzutage davon ausgehen kann, dass der Großteil der Menschen ein Smartphone besitzen, besonders die, die in die Situation kämen die HoloLens zu verwenden. Dabei ist die Verwendung von Google Maps nur beispielhaft, sodass alle Kartenarten als Trackinggrundlage definiert werden können, je nachdem wie die Anforderung definiert wäre.

Grundsätzlich sei an dieser Stelle erwähnt, dass die Möglichkeiten der Weiterentwicklung von GeoMeta oder generell der Art des Anwendungsfalls nahezu endlos ist. Es gibt eine unbestimmte Anzahl an Features, die alle im Zusammenhang mit Stadtkarten eine Form der Erweiterung darstellen können. Allein bestehende Konzepte wie Routenplanung, Google Street View, Darstellung der Satelliten Bilder, Verkehrsdichte, Blitzer und vieles mehr ist implementierbar. Vieles würde sich mit dem heutigen Stand der Technik, vor allem der HoloLens und Vuforia, bereits gut lösen, andere Funktionen würden ein besseres Tracking voraussetzen, damit die Umsetzung realisierbar wäre.

7.2 Die Zukunft von Augmented Reality

Wie dem Abschnitt Historie der technischen Grundlagen zu entnehmen, reicht das Thema Augmented Reality bereits knapp 50 Jahre zurück. Dabei wurden viele Darstellungs- und Wahrnehmungsarten realisiert und die Möglichkeiten und vor allem die Immersion ist mit der Entwicklung der Technik stark gestiegen. Stellt man die beiden relativen Extreme, das

erste Head-Mounted Display für Augmented Reality von Ivan E. Sutherland und der HoloLens von Microsoft, gegenüber, so ist eine starke Entwicklung zu beobachten. Während das Head-Mounted Display von Ivan E. Sutherland noch an der Decke fixiert wurde, damit eine Positionsermittlung stattfinden konnte, um die dargestellten Bilder perspektivisch umsetzen zu können, ist die HoloLens in der Lage das reale Umfeld wahrzunehmen, ein *Mesh* dessen zu erstellen und die computergenerierten Objekte anhand dieser so zu positionieren und zu skalieren, dass es für den Anwender so erscheint, dass sich dieses realitätsgetreu verhält und sich dem Umfeld und der Perspektive des Anwenders anpasst.

Aus heutiger Sicht und dem Stand der Technik ergibt sich jedoch das Ergebnis, dass selbst heute noch die Umsetzung von Augmented Reality Geräten und Inhalten sehr ausbaufähig sind. Wie der Name des Themas bereits schließen lässt, ist die Wahrnehmung des realen Umfeld maßgeblich für die immersive Erweiterung der Realität. Umso präziser und korrekter diese wahrgenommen wird, kann aufbauend darauf eine Verarbeitung dessen durchgeführt werden, damit passendere Informationen eingeblendet werden können. Denkbar wäre das Szenario, dass in Zukunft die Erkennung des Umfelds so gut funktioniert, dass reale Objekte identifiziert werden können und basierend darauf kognitive Aktionen durchgeführt werden können. Ein Beispiel dessen wäre, dass das Gerät im realen Umfeld eine Parkbank erkennt, analysiert wurde, dass der Anwender bereits über eine längere Zeit in Bewegung ist und eine passende Augmented Reality Anwendung die Parkbank hervorhebt und den Anwender darauf hinweist, dass dort eine Pause eingelegt werden kann. Die HoloLens nimmt den Umriss des Umfelds wahr und über Vuforia können statische Trackingelemente eingebaut werden, damit für spezielle Lösungen eine Erkennung stattfinden kann. Für die Zukunft ist daher relevant, dass zusätzlich zum Umriss des Umfelds auch die Objekte erkannt werden. Außerdem ist die Darstellungsqualität der computergenerierten Objekte auf den bestehenden Augmented Reality Head-Mounted Displays vergleichsweise schlecht. Das liegt primär daran, dass die gesamte Rechenleistung im Head-Mounted Gerät selber verbaut ist, damit die Mobilität gewährleistet werden kann. Da die bisherigen Darstellungsarten auch vergleichsweise rechenintensiv sind, resultiert in Summe ein schlechteres Bild. Die bisherige Entwicklung der Technik zeigt, dass leistungsstärkere Hardware in immer kleinerer Form realisierbar ist. Daraus lässt sich erschließen, dass zur Steigerung der Qualität der darzustellenden computergenerierten Objekte auf Head-Mounted Displays Zeit vergehen muss, bis es Hardwarekomponenten gibt, die leistungsstark genug sind, dass sie visuell realitätsnahe Objekte generieren kann und dabei klein genug sind, dass sie in einem mobilen Head-Mounted Gerät verbaut werden können.

Literaturverzeichnis

- [Amin und Govilkar] AMIN, Dhiraj ; GOVILKAR, Sharvari: Comparative study of augmented reality SDK's. 5, Nr. 1, S. 11–26
- [Azuma a] AZUMA, Ronald: Tracking requirements for augmented reality. 36, Nr. 7, S. 50–51
- [Azuma u. a.] AZUMA, Ronald ; BAILLOT, Yohan ; BEHRINGER, Reinhold ; FEINER, Steven ; JULIER, Simon ; MACINTYRE, Blair: Recent advances in augmented reality. 21, Nr. 6, S. 34–47
- [Azuma b] AZUMA, Ronald T.: A survey of augmented reality. 6, Nr. 4, S. 355–385
- [Barfield] BARFIELD, Woodrow: *Fundamentals of wearable computers and augmented reality*. CRC Press
- [Billinghurst u. a.] BILLINGHURST, Mark ; CLARK, Adrian ; LEE, Gun ; OTHERS: A survey of augmented reality. 8, Nr. 2, S. 73–272
- [Bimber und Frohlich] BIMBER, O. ; FROHLICH, B.: Occlusion shadows: using projected light to generate realistic occlusion effects for view-dependent optical see-through displays, IEEE Comput. Soc, S. 186–319. – URL <http://ieeexplore.ieee.org/document/1115088/>. – Zugriffsdatum: 2017-11-29. – ISBN 978-0-7695-1781-0
- [Bimber und Raskar] BIMBER, O. ; RASKAR, R.: *Spatial Augmented Reality: Merging Real and Virtual Worlds*. CRC Press. – URL <https://books.google.de/books?id=JEu3BgAAQBAJ>. – ISBN 978-1-4398-6494-4
- [Bowman und McMahan] BOWMAN, Doug A. ; MCMAHAN, Ryan P.: Virtual Reality: How Much Immersion Is Enough? 40, Nr. 7, S. 36–43. – URL <http://ieeexplore.ieee.org/document/4287241/>. – Zugriffsdatum: 2017-11-20. – ISSN 0018-9162
- [Brown und Hua] BROWN, L.D. ; HUA, H.: Magic Lenses for augmented virtual environments. 26, Nr. 4, S. 64–73. – URL <http://ieeexplore.ieee.org/document/1652928/>. – Zugriffsdatum: 2017-11-28. – ISSN 0272-1716

- [Cakmakci und Rolland] ÇAKMAKCI, Ozan ; ROLLAND, Jannick: Head-worn displays: a review. 2, Nr. 3, S. 199–216
- [Clarke] CLARKE, Keith C.: Mobile mapping and geographic information systems. 31, Nr. 3, S. 131–136
- [Doerner u. a.] DOERNER, Ralf ; BROLL, Wolfgang ; GRIMM, Paul ; JUNG, Bernhard: *Virtual und Augmented Reality (VR / AR)*. Springer Berlin Heidelberg (eXamen.press). – URL <http://link.springer.com/10.1007/978-3-642-28903-3>. – Zugriffsdatum: 2017-11-28. – DOI: 10.1007/978-3-642-28903-3. – ISBN 978-3-642-28902-6 978-3-642-28903-3
- [Feng Zhou u. a.] FENG ZHOU ; DUH, Henry Been-Lirn ; BILLINGHURST, Mark: Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR, IEEE, S. 193–202. – URL <http://ieeexplore.ieee.org/document/4637362/>. – Zugriffsdatum: 2017-11-28. – ISBN 978-1-4244-2840-3
- [Graham u. a.] GRAHAM, Mark ; ZOOK, Matthew ; BOULTON, Andrew: Augmented reality in urban places: contested content and the duplicity of code. 38, Nr. 3, S. 464–479
- [Hecht u. a.] HECHT, Brent ; ROHS, Michael ; SCHOENING, Johannes ; KRUEGER, Antonio: Wikeye - using magic lenses to explore georeferenced Wikipedia content. In: *Proceedings of the 3rd International Workshop on Pervasive Mobile Interaction Devices (PERMID)*
- [Hedley u. a.] HEDLEY, Nicholas R. ; BILLINGHURST, Mark ; POSTNER, Lori ; MAY, Richard ; KATO, Hirokazu: Explorations in the Use of Augmented Reality for Geographic Visualization. 11, Nr. 2, S. 119–133. – URL <http://www.mitpressjournals.org/doi/10.1162/1054746021470577>. – Zugriffsdatum: 2017-12-18. – ISSN 1054-7460, 1531-3263
- [Kalawsky] KALAWSKY, Roy S.: *The science of virtual reality and virtual environments: a technical, scientific and engineering reference on virtual environments*. Addison-Wesley. – ISBN 978-0-201-63171-5
- [Leifer u. a.] LEIFER, Larry ; TOYE, George ; LOOS, Machiel Van der: BioRobotics'95 1.5 Final. . – URL <https://web.stanford.edu/group/rrd/People/vdl/publications/Biorobotics/BioRobotics.final.html>. – Zugriffsdatum: 2017-11-26

- [Liarokapis u. a.] LIAROKAPIS, F.; GREATBATCH, I.; MOUNTAIN, D.; GUNESH, A.; BRUJIC-OKRETIC, V.; RAPER, J.: Mobile Augmented Reality Techniques for GeoVisualisation, IEEE, S. 745–751. – URL <http://ieeexplore.ieee.org/document/1509156/>. – Zugriffsdatum: 2017-12-18. – ISBN 978-0-7695-2397-2
- [Milgram und Kishino] MILGRAM, Paul; KISHINO, Fumio: A taxonomy of mixed reality visual displays. 77, Nr. 12, S. 1321–1329
- [Milgram u. a.] MILGRAM, Paul; TAKEMURA, Haruo; UTSUMI, Akira; KISHINO, Fumio: Augmented reality: A class of displays on the reality-virtuality continuum. In: *Telemanipulator and telepresence technologies* Bd. 2351, International Society for Optics and Photonics, S. 282–293
- [Murase u. a.] MURASE, Kaori; OGI, Tetsuro; SAITO, Kota; KOYAMA, Takahide; SQUARE BIRU, Kudan: Correct Occlusion Effect in the Optical See-through Immersive Augmented Reality Display System.
- [Narzt u. a.] NARZT, Wolfgang; POMBERGER, Gustav; FERSCHA, Alois; KOLB, Dieter; MÄLLER, Reiner; WIEGHARDT, Jan; HÄRTNER, Horst; LINDINGER, Christopher: Augmented reality navigation systems. 4, Nr. 3, S. 177–187. – URL <http://link.springer.com/10.1007/s10209-005-0017-5>. – Zugriffsdatum: 2018-01-07. – ISSN 1615-5289, 1615-5297
- [Parks] PARKS, Theodore E.: Post-Retinal Visual Storage. 78, Nr. 1, S. 145. – URL <http://www.jstor.org/stable/1421101?origin=crossref>. – Zugriffsdatum: 2017-11-30. – ISSN 00029556
- [Pohl] POHL, Klaus: *Requirements Engineering: Grundlagen, Prinzipien, Techniken*. dpunkt.verlag
- [Reitmayr u. a.] REITMAYR, G.; EADE, E.; DRUMMOND, T.: Localisation and interaction for augmented maps, IEEE, S. 120–129. – URL <http://ieeexplore.ieee.org/document/1544673/>. – Zugriffsdatum: 2017-12-18. – ISBN 978-0-7695-2459-7
- [Rosenberg] ROSENBERG, L.B.: Virtual fixtures: Perceptual tools for telerobotic manipulation, IEEE, S. 76–82. – URL <http://ieeexplore.ieee.org/document/380795/>. – Zugriffsdatum: 2017-11-26. – ISBN 978-0-7803-1363-7
- [Schoening u. a.] SCHOENING, Johannes; ROHS, Michael; KRATZ, Sven; LOECHTEFELD, Markus; KRUEGER, Antonio: Map torchlight: a mobile augmented reality camera projector

- unit, ACM Press, S. 3841. – URL <http://portal.acm.org/citation.cfm?doid=1520340.1520581>. – Zugriffsdatum: 2017-12-18. – ISBN 978-1-60558-247-4
- [SchÄ¶ning u. a.] SCHÄ¶NING, Johannes ; KRÄ¶GER, Antonio ; MÄ¶LLER, Hans J.: Interaction of mobile camera devices with physical maps.
- [Starter u. a.] STARNER, Thad ; MANN, Steve ; RHODES, Bradley ; LEVINE, Jeffrey ; HEALEY, Jennifer ; KIRSCH, Dana ; PICARD, Rosalind W. ; PENTLAND, Alex: Augmented reality through wearable computing. 6, Nr. 4, S. 386–398
- [Steptoe u. a.] STEPTOE, William ; JULIER, Simon ; STEED, Anthony: Presence and discernability in conventional and non-photorealistic immersive augmented reality, IEEE, S. 213–218. – URL <http://ieeexplore.ieee.org/document/6948430/>. – Zugriffsdatum: 2017-11-21. – ISBN 978-1-4799-6184-9
- [Sutherland] SUTHERLAND, Ivan E.: A Head-Mounted Three-Dimensional Display. . – URL http://90.146.8.18/en/archiv_files/19902/E1990b_123.pdf
- [Van Krevelen und Poelman] VAN KREVELEN, DWF ; POELMAN, Ronald: A survey of augmented reality technologies, applications and limitations. 9, Nr. 2, S. 1
- [Wagner und Schmalstieg] WAGNER, Daniel ; SCHMALSTIEG, Dieter: History and Future of Tracking for Mobile Phone Augmented Reality, IEEE, S. 7–10. – URL <http://ieeexplore.ieee.org/document/5232244/>. – Zugriffsdatum: 2018-01-07. – ISBN 978-1-4244-4437-3
- [Zhou u. a.] ZHOU, Feng ; DUH, Henry Been-Lirn ; BILLINGHURST, Mark: Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR. In: *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, IEEE Computer Society, S. 193–202

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

Hamburg, 12. Januar 2018

René Schramowski